Analog Integr Circ Sig Process
DOI 10.1007/s10470-016-0914-y

Modeling, simulation and implementation of circuit elements in an open-source tool set on the FPAA

Aishwarya Natarajan[1] · **Jennifer Hasler**[1]

Received: 11 August 2016 / Revised: 13 December 2016 / Accepted: 18 December 2016

Abstract An open-source simulator to design and implement circuits and systems, replicating the results from the Field Programmable Analog Array (FPAA) is presented here. The fundamental components like the transistors, amplifiers and floating gate devices have been modeled based on the EKV model with minimal parameters. Systems including continuous-time filters and the analog front-end of a speech processing system have been built from these basic components and the simulation results and the data from the FPAA are shown. The simulated results are in close agreement to the experimental measurements obtained from the same circuits compiled on the FPAA fabricated in a 350 nm process.

Keywords FPAA · Level=2 models · Analog circuit design · Modelica · Floating gates

1 Modeling circuit elements for the FPAA

Field Programmable Analog Arrays (FPAA) [1], with their reconfigurability and programmability, have enabled to design and implement large-scale mixed-signal systems, with a diverse set of applications, including, signal processing and neuromorphic computing. The need for a single platform facilitating a hardware-software codesign is critical in implementing such applications [2], and provides flexibility and helps the user to utilize the hardware to its potential.

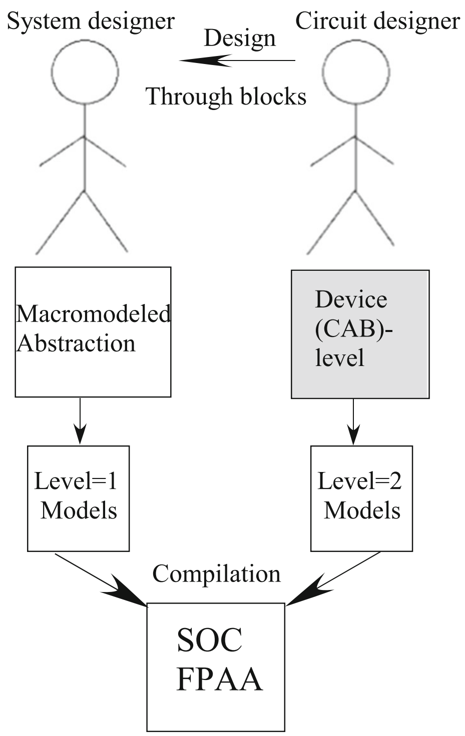


Fig. 1 The concept of implementing the circuit/system idea from the designer's mind to simulating and obtaining results from the FPAA. It is essential to have a support for circuit design as much as a system design since the number of parameters are much higher in system design than circuit design and is significant to realize the dynamics of the system. How this low-level circuit design is done is the question in this paper and is highlighted

Figure 1 shows a simulator based on real data from silicon with a simple platform having minimal parameters for a comprehensive analysis, built in Xcos/Scilab, and is shown in this paper. We will work with Hasler et al. who built the open-source Xcos/Scilab infrastructure [3] to

✉ Aishwarya Natarajan
anatarajan35@gatech.edu

Jennifer Hasler
jennifer.hasler@ece.gatech.edu

[1] Georgia Institute of Technology, Atlanta, GA, USA

Published online: 06 January 2017

integrate this into their distribution. The infrastructure [3] enables one to compile the desired circuits and systems on the FPAA to obtain silicon data and simulate them at the same time to analyze and go back and forth between the simulation and data to obtain different dynamics, done either in a SPICE-similar setting with the fundamental components like transistors, capacitors, amplifiers etc., which is the primary focus of this work or build it as a system implementing the Ordinary Differential Equations (ODEs) shown in [4], depending on the user's choice, thereby incorporating the perspectives of both the circuit and system designers.

The conventional circuit simulators like SPICE, Cadence etc. employ models which have a large number of parameters and switch between different levels of models like BSIM (e.g. level=44) [5, 6], etc., to obtain different levels of accuracy and different times of simulation for convergence. However, the basic components built in this toolset are based on the EKV models, with a minimal number of parameters, and are easy to model.

An accurate and a relatively fast simulator is essential [7] for a circuit-designer and the user has the flexibility to intuitively tune the parameters, in this work. The W/L ratios being fixed for the devices on the FPAA is abstracted out through parameters like threshold current for the transistor which is a more relevant term in this discussion and one need not hypothesize the required W/L, which is generally a significant factor to be decided on [8], in SPICE simulators.

Due to the availability of the remote system and open-source toolset encapsulated in a virtual machine [3], any user can compile their required circuits and systems, send it to a remote system which programs the design, measures the output from the given input and sends them the results from the FPAA back in an email [9]. A design cycle of waiting for a period of time for the fabrication as in the case of a custom chip isn't required here and we can readily take data due to the access to the FPAA.

The reconfigurability is obtained by programming the charges on the floating gates on the FPAA. Hence, it is important to judge and have an idea of the bias currents and reference DC levels to be used in the systems to get the desired results. One can design the system beforehand using the simulation models in the infrastructure and set the different parameters in the system and further use the remote system to replicate the results and also, obtain a diverse set of dynamics by tuning the properties of the circuit.

This paper elucidates the models used for the design and implementation of the circuits and its close agreement to the experimental measurements obtained from the FPAA. Section 2 introduces the concept of level=1 and level=2 models while Sect. 3 describes the FPAA and the tool infrastructure used for obtaining the results. We discuss the characterization of the CAB components as level=2 blocks in Sect. 4. Section 5 introduces us to the floating gate components and their accuracy in simulating such circuits and systems, with the experimental measurements from such devices. Validation of the level=2 blocks and building upon these to reach a system are further discussed in Sects. 6 and 7 concludes the discussion, comparing this work and conventional simulator's compilation time.

2 Level=1 and Level=2 models

The tool infrastructure built in Scilab and Xcos [3] enables us to perform simulations and take the experimental measurements for compiling various circuits and systems. The different blocks in the tools are classified as primarily level=1 and level=2 blocks, as shown in Fig. 2. Level=1 blocks aim to model systems, wherein the focus is on a macromodeled simulation [10], defining the ODEs and algebraic equations for the system operating in a voltage mode with a voltage signal in and voltage out, with vectorized inputs and outputs. While the level=2 aims to model the circuits and individual CAB components, through a modelica based scilab framework, implementing implicit ODEs and the input and output signals can be voltage or current mode. Hence, the level=1 models are similar to a typical data flow graph which defines the equations of the system, as introduced in [3] and [4]. Level=2 models aim to model the individual CAB elements, in a SPICE-like setting shown in this paper.

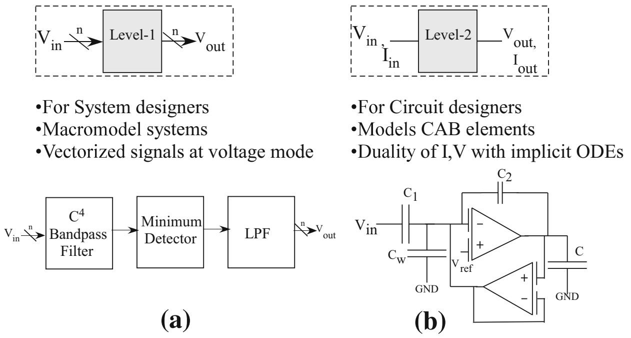


Fig. 2 Level=1 and Level=2 blocks. **a** Level=1 blocks is used for emulating the system behavior. It operates in voltage mode and serves to design the system at a high level. The analog front-end of a speech processing system is shown as an example of a level=1 system. **b** Level=2 blocks is used for emulating the circuit behavior. A C^4 bandpass filter is shown as an example of a level=2 circuit

3 FPAA and tool infrastructure overview

The Field Programmable Analog Array (FPAA) SOC [1] is a reconfigurable and programmable SOC consisting of Computational Analog blocks (CABs) and Computational Logic Blocks (CLBs), which are connected through Manhattan style routing, integrated with a processor, ADCs, DACs and peripherals, as shown in Fig. 3. The CAB components are shown in Fig. 3(b) which are primarily modeled here. The results shown here are taken from the SOC FPAA fabricated on a 350 nm process.

The level=2 models are built through a modelica based scilab framework to implement the EKV models. Modelica, an object oriented language to write and solve the ODEs and model the dynamics of a system, enables directly implementing electrical systems as physical level components in Xcos [11]. This modeling is similar to SPICE level simulations, thereby giving a better intuition too, rather than breaking down the ODEs of the system and modeling each operation individually and integrating to a system. A standard Runge–Kutta-4(5) (RK-45) [12] is used as the solver in the simulations and can be chosen from a number of solvers. The electrical blocks have voltage and current values in their input and output ports, which is specified in the computational function associated with the Xcos block.

A component-based approach [13] is followed here wherein the behavior for all the components we use has been described in the computational function associated with that block and a library of components is created. During compilation, the Xcos/Scilab converts the Xcos schematic to a set of files. All the instances are called in a netlist-like modelica file , which describes the connections between the nodes of all the components present in the diagram through the “connect” command, analogous to an electrical netlist which describes the circuit. The other modelica file describes the equations of the system, obtained by following KCL and KVL at the nodes. These modelica .mo files are compiled to a . C file which describes the flow of the simulation through the event-driven flags, which further define the flow of operations from the initialization of the ODEs to the running for the time period of the simulation defined by the period of the clock pulse generated and applied at regular periodic intervals of time.

4 Characterization of CAB components

The CAB components comprise nFETs, pFETs, OTAs, FGOTAs, T-gates and current mirrors. The EKV model [14] is used to model the characteristics, based on the fundamental device physics properties and covers subthreshold to above threshold regions accurately [15]. FG characteristics showing the capacitive coupling effects can be modeled too which help in accurately simulating FG circuits and systems [16].

4.1 Transistors: nFET, pFET

The measurements are taken from a golden nFET and pFET, whose position is fixed on a particular CAB on the FPAA IC, to get uniform characteristics and to use

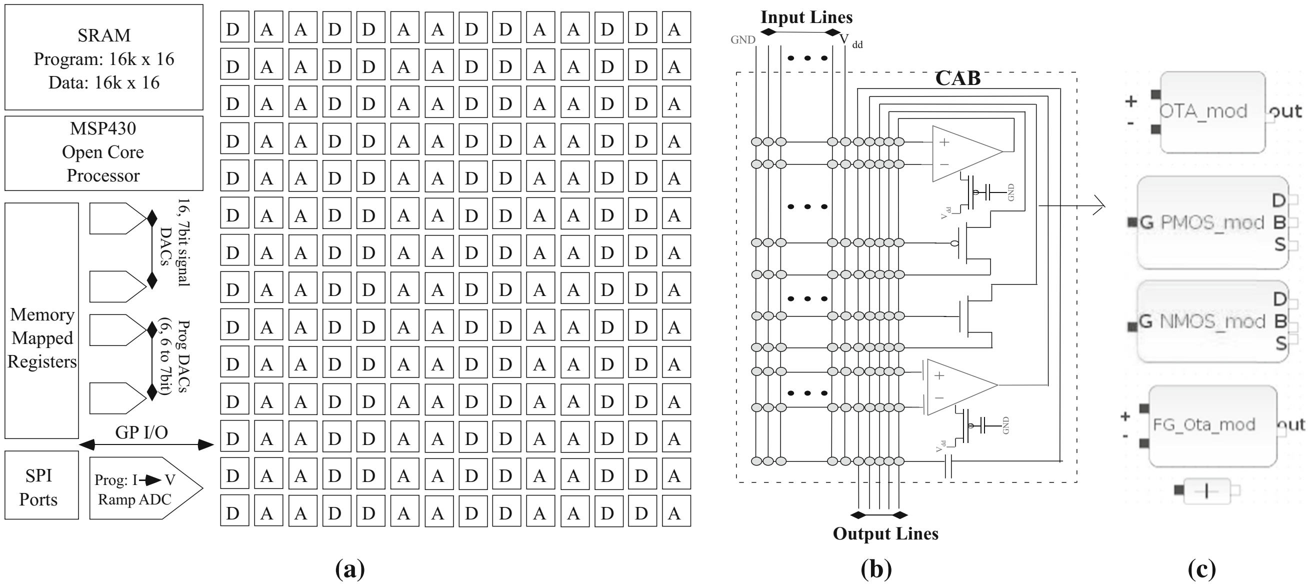


Fig. 3 FPAA and tool infrastructure overview. **a** FPAA fabric consists of CABs, CLBs and other peripherals, fabricated on a 350 nm process. **b** The CAB components consisting of nFET, pFET, OTA, FGOTA and capacitor, with the FGpFETs is shown in one CAB. **c** The Xcos palette with the blocks simulated with modelica corresponding to each CAB component is shown

the same parameters while building other circuits and systems from these transistors. To identify what terminal voltages' need to be swept to determine the required parameters for the FETs, we look at the simplified EKV model in subthreshold and above threshold regions [17].

The version of the EKV model used is shown in the table in Fig. 4(a), where I_{th} is the current at threshold, V_{T0} is the threshold voltage, U_t is the thermal voltage, κ is the fractional change in the surface potential, V_d is the drain voltage, V_b the bulk voltage, V_g the gate voltage and V_s is the source voltage. The source voltage is fixed at a V_{dd} of 2.5 V while V_g and V_d are swept to determine the parameters for the EKV model [29]. These parameters are substituted in the simulation model to get a close overlap with the data.

Looking at the equations in Fig. 4, one can identify the slopes to be determined to extract the necessary parameters. The drain current measurements against the gate voltage sweep are performed and fit to the EKV model to determine I_{th} , V_{T0} and κ while the drain current measurements against the drain voltage sweep

	nFET	pFET
EKV	$I_{th}\ln^2(1+e^{(\kappa(V_g-V_{T0})-V_s+\sigma V_d)/2U_T})$	$I_{th}\ln^2(1+e^{(\kappa(V_b-V_g-V_{T0})-(V_b-V_s)+\sigma(V_b-V_d))/2U_T})$
	$-\ln^2(1+e^{(\kappa(V_g-V_{T0})-V_d+\sigma V_s)/2U_T})$	$-\ln^2(1+e^{(\kappa(V_b-V_g-V_{T0})-(V_b-V_d)+\sigma(V_b-V_s))/2U_T})$
For sub V_{T0}: Ohmic	$I_{th}e^{\kappa(V_g-V_{T0})/U_T}\,(e^{-V_s/U_T}-e^{-V_d/U_T})$	$I_{th}e^{\kappa(V_b-V_g-V_{T0})/U_T}\,(e^{-(V_b-V_s)/U_T}-e^{-(V_b-V_d)/U_T})$
Saturation	$I_{th}e^{(\kappa(V_g-V_{T0})-V_s+\sigma V_d)/U_T}$	$I_{th}e^{(\kappa(V_b-V_g-V_{T0})-(V_b-V_s)+\sigma(V_b-V_d))/U_T}$
For Above V_{T0}: Ohmic	$\frac{I_{th}}{4U_T^2}\,((\kappa(V_g-V_{T0})-V_s)^2-(\kappa(V_g-V_{T0})-V_d)^2)$	$\frac{I_{th}}{4U_T^2}\,((\kappa(V_b-V_g-V_{T0})-(V_b-V_s))^2-(\kappa(V_b-V_g-V_{T0})-(V_b-V_d))^2)$
Saturation	$\frac{I_{th}}{4U_T^2}(\kappa(V_g-V_{T0})-V_s+\sigma V_d)^2$	$\frac{I_{th}}{4U_T^2}(\kappa(V_b-V_g-V_{T0})-(V_b-V_s)+\sigma(V_b-V_d))^2$

(a)

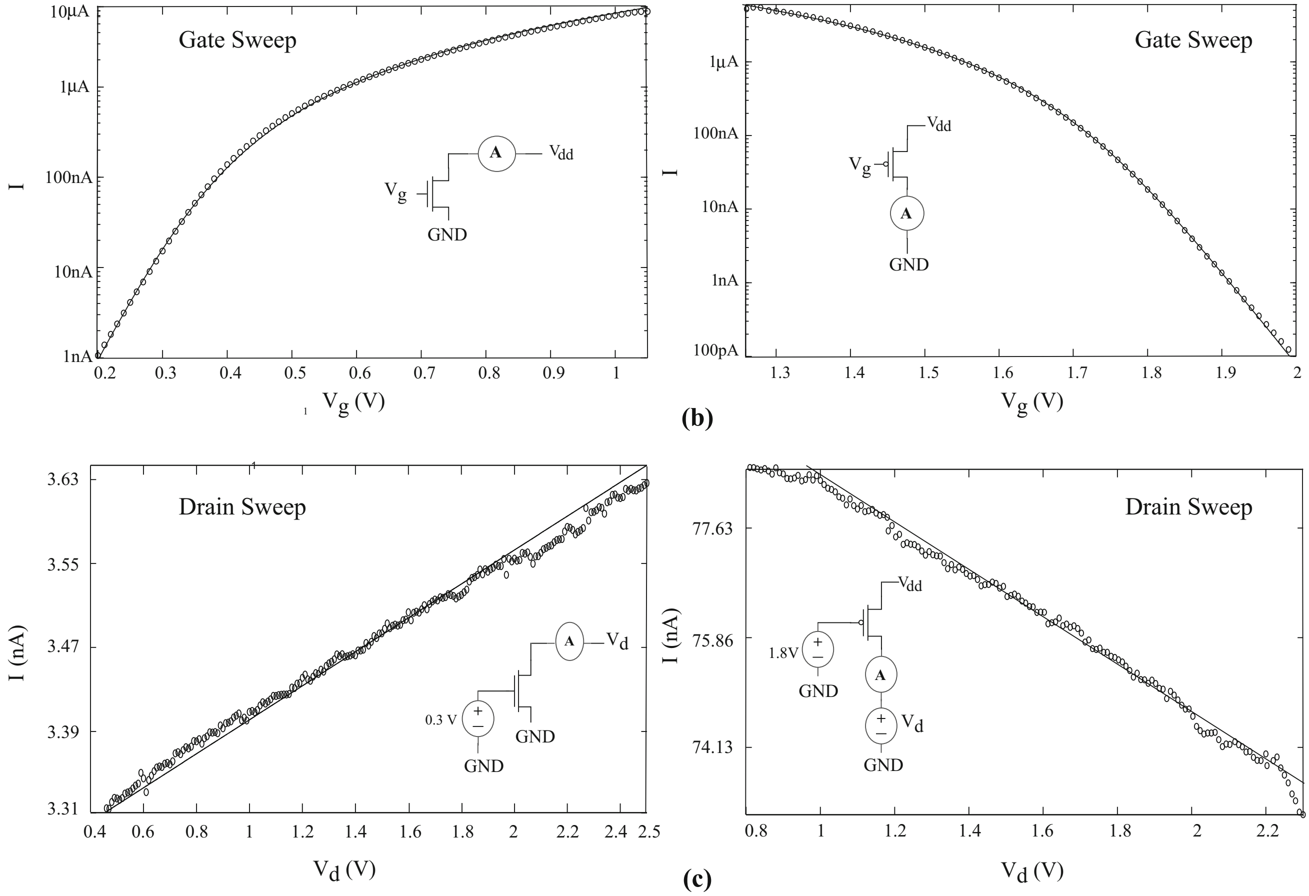


Fig. 4 Simulated and measured current as a function of gate voltage and drain voltage in log scale, at a V_{dd} of 2.5 V. The measured *data* is represented as the fine discrete points while the simulation is the *thick continuous line*. **a** The version of the EKV model for the level=2 models of the transistors is shown. **b** The subthreshold region with a slope of $\frac{\kappa}{U_T}$ and the above threshold regions can be observed. The parameters obtained from the EKV curve-fit on the gate sweep data, used for the simulation are I_{th} of 53.58nA, V_{T0} of 0.32 V and κ of 0.84 for the nFET and I_{th} of 111.20nA, V_{T0} of 0.75 V and κ of 0.76 for the pFET. **c** The drain voltage is swept and the gate voltage is at 0.3 V for the nFET and source voltage is fixed at a V_{dd} of 2.5 V for pFET, to measure the slope in the sub-threshold region. σ extracted from the data, used for the simulation is 0.00039 and 0.0049 for nFET and pFET respectively

determine the σ value. These values are then substituted in the modelica Scicos simulations for the transistors to get the desired behavior. As shown in Fig. 4, it can be seen that there is a close overlap between the data and the simulations.

4.2 Building amplifiers from the transistors

Using the pFET and nFET models introduced in Sect. 4.1, experiments are performed for a nFET source follower and a common source amplifier, as shown in Fig. 5. In the common drain amplifier, the current flows through the nFETs , determined by the bias voltage, V_{ref}. The source voltage follows the gate terminal, to which the input voltage is swept, at a gain of κ. The variation of κ with the input voltage seen here is less and the depletion capacitance is almost a constant as the FET is in the subthreshold region. Only the gain of κ is highlighted in the transfer curve and the offset of κV_{ref} isn't shown due to the voltage offset from the output buffer used at the output of the experimental setup. Consider that the transistors are in Sub V_{T0} and doing a large-signal analysis for intuition [29],

$$\begin{aligned} I_{th} e^{(\kappa(V_{in}-V_{T0})-V_{out})/U_T} &= I_{th} e^{(\kappa(V_{ref}-V_{T0}))/U_T} \\ V_{out} &= \kappa(V_{in}-V_{ref}) \end{aligned} \tag{1}$$

The common source amplifier has the pFET biased to 1 V, which sets the bias current through the FETs. Again doing a large-signal analysis,

$$I_{thp} e^{\kappa(V_{dd}-V_{ref}-V_{T0})+\sigma_p(V_{dd}-V_{out})/U_T} =$$
$$I_{thn} e^{\kappa(V_{in}-V_{T0})+\sigma_n V_{out}/U_T}$$

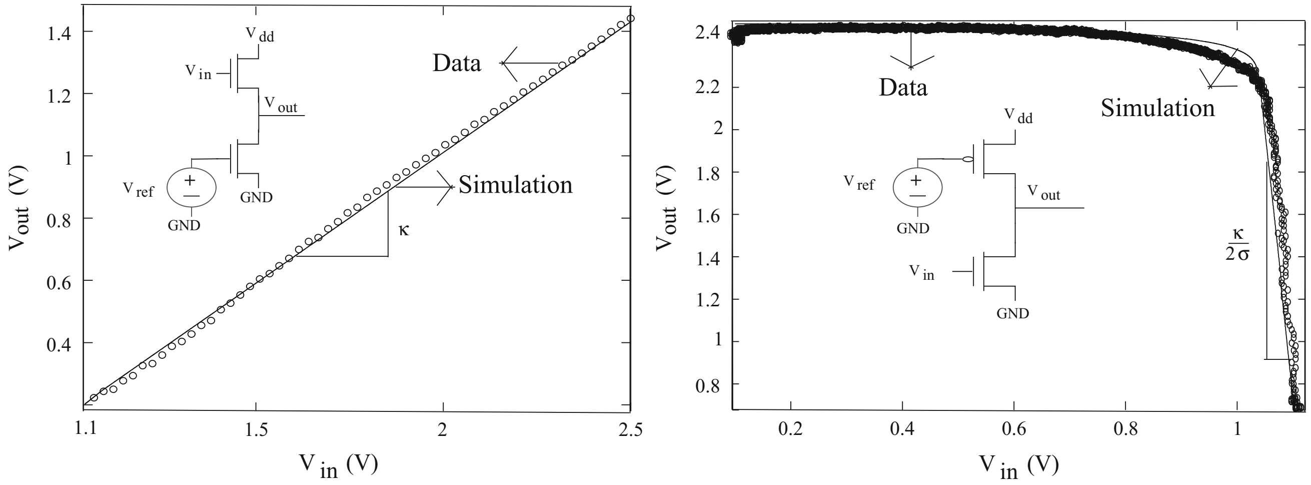
Assuming identical FETs,

$$\begin{aligned} \kappa(V_{dd}-V_{ref})+\sigma_p V_{dd} &= (\sigma_p+\sigma_n)V_{out}+\kappa V_{in} \\ V_{out} &= \frac{-\kappa}{\sigma_p+\sigma_n} V_{in} + V_{constant} \end{aligned} \tag{2}$$

Hence a high gain of $\kappa/2\sigma$ is observed in the transfer curve of V_{out} vs V_{in}, as shown in Fig. 5.

4.3 OTA

Another CAB component, the OTA has been built as a level=2 model. The schematic of the 9 transistor differential transconductance amplifier is shown in Fig. 6. It consists of a pFET differential pair and a set of current mirrors. The bias current, I_{bias} is set by the FG-pFET at the top, which acts as a current source for the amplifier [18]. Considering that the input pFET differential pair is in saturation, the I_{bias} is split into currents I_1 and I_2 in the



(a) **(b)**

	κ	I_{th}	V_{TO}	σ
pFET	0.679	111.84nA	0.866V	0.0049
nFET	0.808	53.58nA	0.313V	0.00039

(c)

Fig. 5 Verifying and building circuits using level=2 models. **a** Simulated and measured output voltage as a function of input voltage for a nFET source follower. A gain of κ can be observed due to the capacitive coupling. The nFET is biased at a reference voltage of 500 mV. **b** Simulated and measured Output voltage as a function of input voltage for a common source amplifier. A finite gain of $\frac{\kappa}{2\sigma}$ is observed. **c** The parameters used for the simulation of the transistors in the amplifier are shown

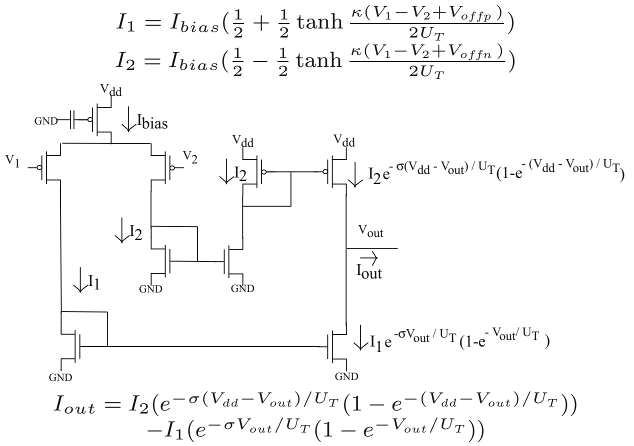

$I_1 = I_{bias}(\frac{1}{2} + \frac{1}{2}\tanh\frac{\kappa(V_1 - V_2 + V_{offp})}{2U_T})$
$I_2 = I_{bias}(\frac{1}{2} - \frac{1}{2}\tanh\frac{\kappa(V_1 - V_2 + V_{offn})}{2U_T})$
Vdd
GND
Ibias
V1
V2
Vdd
Vdd
I2
I2 e^- σ(Vdd - Vout) / UT(1-e^- (Vdd - Vout) / UT)
I2
Vout
Iout
I1
GND
GND
I1 e^-σVout / UT (1-e^- Vout/ UT)
GND
GND
$I_{out} = I_2(e^{-\sigma(V_{dd}-V_{out})/U_T}(1 - e^{-(V_{dd}-V_{out})/U_T})) - I_1(e^{-\sigma V_{out}/U_T}(1 - e^{-V_{out}/U_T}))$

Fig. 6 Schematic of the 9 transistor OTA in the CAB, with the I_{bias} current split into currents I_1 and I_2 in the input differential pair branches and reflected as I_2'' and I_1'', due to the output transistors

input differential pair branches. The derivation from the exponential relationship to the tanh relation can be followed from[1] and through intuition from the graph and its solution. Since the drain current is dependent on the drain voltage in saturation due to the early effect, the effect of σ is significant especially in the output transistors and included in the output relation from I_{out} to V_{out}. The current is not reflected out equally from the current mirrors due to mismatches. The V_{T0} and I_{th} differences between the individual transistors are responsible for the voltage and current offsets which too are included in the equations.

Different experiments are performed to study the behavior of the OTA as shown in Fig. 7. It shows the OTA as an element in the CAB, which is built from the level=2 modelica simulation models. Each parameter of the nFET and pFET like V_{T0}, I_{th}, σ etc. can be manipulated to vary the effect of mismatches and offsets, especially from the current mirrors to get different dynamics in the OTA.

Figure 7(b)–(d) shows the characteristics of an OTA built as a level=2 model. A close agreement can be seen between the experimental measurements and the simulations.

While the input voltage is swept, the output voltage is kept at midrail and the ammeter is connected in series with the output node of the OTA. As expected, we can see a tanh behavior in the I_{out} curve as a function of V_{in}. The transconductance, g_m, as a function of I_{bias} is measured from the slope of the curve. The output current to which the OTA settles at the upper and lower end is not identical due to the finite current gain from the current mirror and is reflected as an exponential variation in the threshold voltage differences.

V_1 and V_2 are fixed to observe the effect of V_{out} on I_{out}. The finite output conductance is shown in the zoomed in plot and can be measured from the slope of the curve, over a range of the output voltage. This output resistance is due to the early effect imposed by the finite σ of the output transistors. We can see that the current drops exponentially near the V_{dd} of 2.5 V, which is due to the point when the transistors transit from the saturation region.

The Voltage transfer curve of the OTA in the open loop configuration, wherein V_{out} is obtained as a function of V_{in} is also measured. The slope of the curve gives the gain of the OTA, which is similar to the $A_v = g_m\ R_{out}$, where g_m and R_{out} have already been obtained from the I_{out} vs V_{in} and I_{out} vs V_{out} curves respectively.

5 Floating gate components

A FG circuit is one in which the gate of the FET is electrically isolated and is capacitively connected to the other nodes [16]. Since there is no DC path from the floating node to any fixed voltage, it is difficult to model FG in conventional SPICE simulators wherein it is modeled with a DC voltage at the node with a resistor. Charge is stored on the FG and it can be modulated through electron tunneling and hot-electron injection, for computation. The FG programming is encapsulated in the infrastructure [20], abstracted away from the user.

5.1 FGpFET

The input, routing, overlap and oxide capacitances represented by C_1, C_w C_{ov}, C_{ox} respectively form a capacitive divider at the gate terminal with C_T being the total capacitance at the gate terminal.

$$C_T = C_1 + C_w + C_{ov} + C_{ox}(1 - \kappa)$$

$$V_{fg} = \frac{C_1}{C_T} V_g + \frac{C_{ov}}{C_T} V_d + \frac{C_w}{C_T} V_{dd}$$

Experimental measurements and simulations are performed for the FGpFET, from the Multiple-Input Translinear Element (MITE) block [21, 22]. The effective value of κ seen is lesser compared to a pFET due to the capacitive coupling as shown in the above equations. V_{fg} is the effective FG voltage at the node due to the capacitive coupling from the gate node, with a voltage offset which is dependent on the initial charge stored on the node. Similar to the pFET, gate sweep is performed to extract I_{th}, V_{T0} and κ and drain sweep gives the σ value. The capacitor node to be swept can be chosen , thereby giving different capacitive coupling ratios from the gate terminal at the input to V_{fg}, as shown in Fig. 8.

[1] [19], Chap. 5, pp. 68.

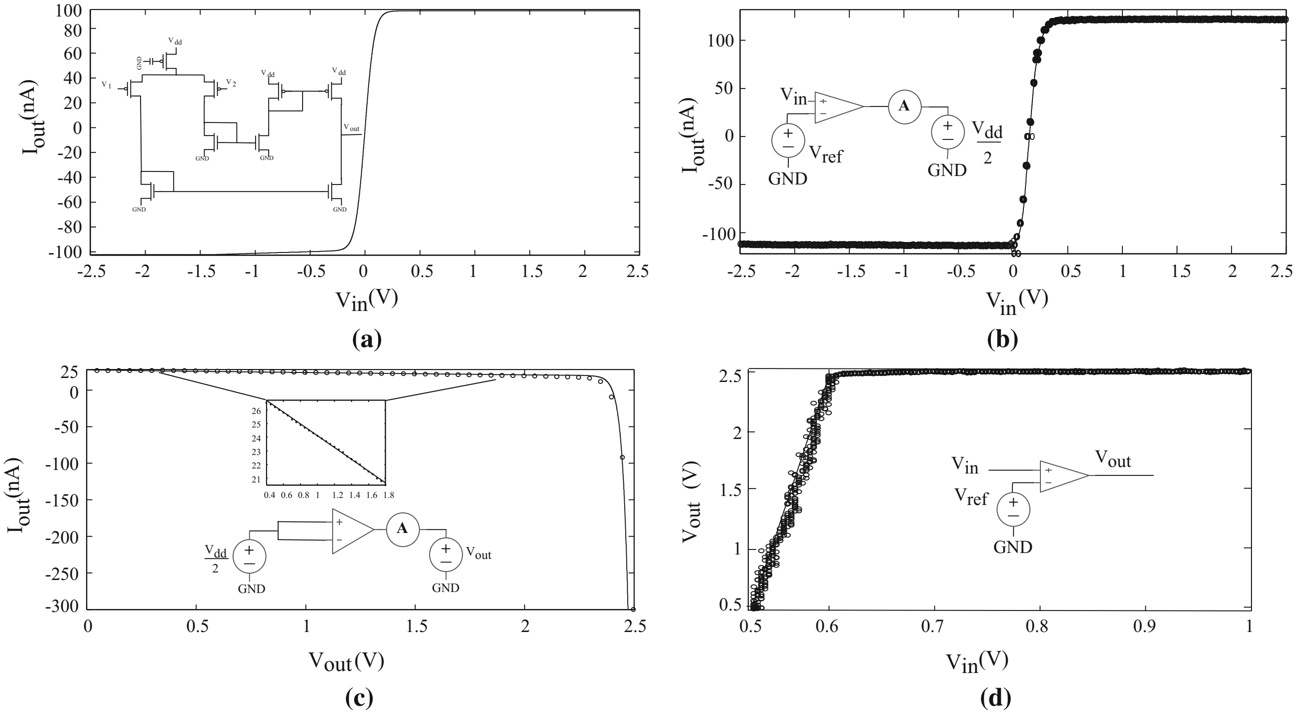
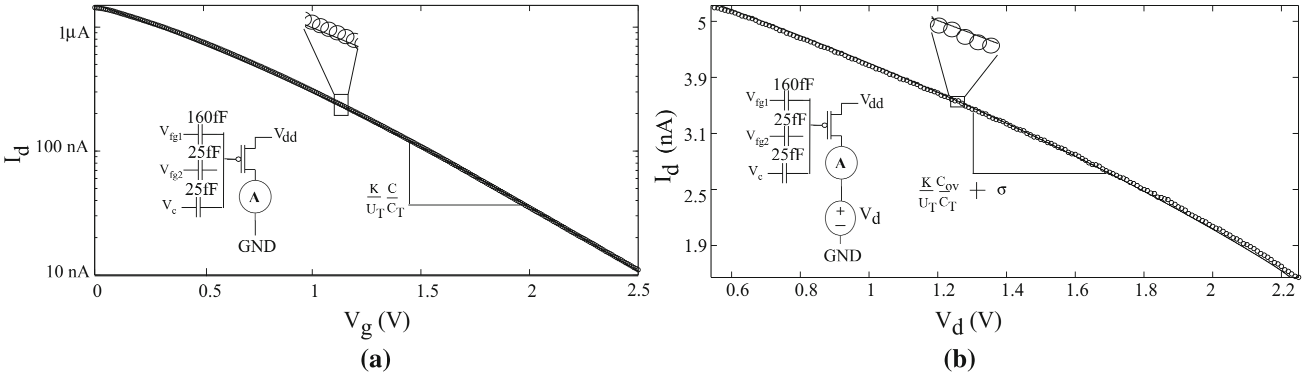


Fig. 7 Simulated and measured characteristics of the OTA biased at 100 nA. **a** Output current as a function of differential input voltage. The nine transistor OTA is built using the level=2 modelica transistors. The tanh behavior is observed and the offset in the currents between the positive and negative inputs can be seen due to the pFET current mirror. **b** Output current as a function of differential input voltage for an OTA. The transconductance can be measured from the slope which shows a tanh behavior. The current offsets from the biased current can be seen due to the threshold mismatch in the current mirrors in the 9T OTA structure. **c** Output current as a function of output voltage. The exponential dependency of the current is seen through an upper limit near V_{dd} above which the current decreases. The finite output conductance of the OTA is observed in the finite slope in the midrange, shown in the *inset* zooming into the midrange. **d** Voltage transfer curve, output voltage as a function of input voltage. The positive terminal is fixed at 500 mV



	κ	I_{th}	V_{TO}	σ
FG pFET	0.712	512.36nA	0.854V	0.0071

(c)

Fig. 8 Simulated and measured current as a function of gate voltage and drain voltage of a FGpFET, from the MITE block in log scale. **a** The FG input depending on the input capacitances to be swept can be chosen. The capacitive coupling can be seen in the slope and the range of currents obtained as the input voltage is swept. The source voltage is fixed at a V_{dd} of 2.5 V. A slope of $\frac{\kappa}{U_T}\frac{C}{C_T}$ can be observed. **b** The Source voltage is fixed at a V_{dd} of 2.5V. A slope of $\frac{\kappa}{U_T}\frac{C_{ov}}{C_T}+\sigma_p$ can be observed. **c** The parameters obtained through curve-fit from the data, which are then used for the simulations are shown

5.2 FGOTA

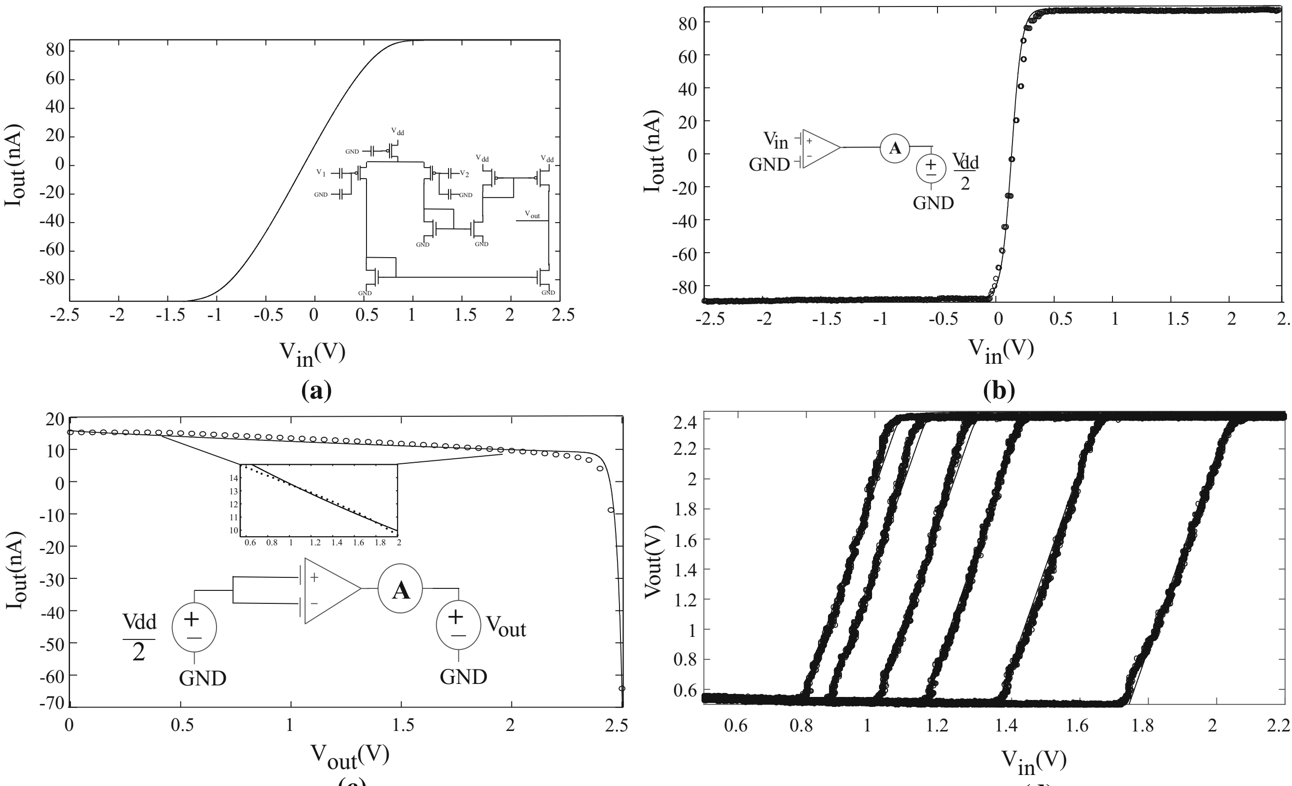
The effective FG voltages are set by the capacitive divider at the inputs of the FGpFETS, at the differential amplifier input terminals.

$$V_{fgp} = \frac{C_1}{C_T} V_1 + V_{fgp0}$$
$$V_{fgn} = \frac{C_1}{C_T} V_1 + V_{fgn0}$$

The 9T FGOTA structure in Fig. 9(a). built from level=2 models of FGpFETs and pFETs shows characteristics similar to that obtained from the FGOTA built as a single level=2 model. In addition to the current source which can be programmed, the FGOTA has two FGpFETs at the two inputs which can be programmed [23]. Each parameter of the individual transistors in the 9T structure can be tweaked to get different linearities and transconductances ranges.

As shown in Fig. 9(b), the tanh behavior and the improved linearity can be observed in the I_{out} vs V_{in} plot, whose slope gives the transconductance. The linear range of the FGOTA is higher than the OTA due to the capacitive divider at the two inputs. The drain induced barrier lowering effect (DIBL) is again responsible for the finite output conductance, measured from the slope of the I_{out} vs V_{out} curve, as shown in the zoomed in inset in Fig. 9(c) and the sudden exponential drop in current can be seen as the output voltage approaches V_{dd}.

Another advantage of the FGOTA is that it helps to set a desired voltage offset which is programmable. The charges



(a)

(b)

(c)

(d)

	g_m	R_{Out}
OTA	1486.5nA/V	233.4MΩ
FG OTA	285.5nA/V	277.81MΩ

(e)

Fig. 9 Simulated and measured characteristics for a FGOTA. **a** The voltage offset is obtained for FGOTA built using the level=2 modelica FG transistors by setting the offsets at the input FGpFETs. **b** Output current as a function of differential input voltage for a level=2 FG OTA whose slope gives the transconductance and shows its tanh behavior and the linearity. **c** The output current as a function of output voltage shows the exponential dependency of the current, seen through an upper limit near V_{dd} above which the current decreases. The finite slope, shown in the inset zooming into the midrange shows the output conductance of the FGOTA. **d** Voltage transfer curve, output voltage as a function of input voltage. The positive terminal is fixed at 1.5 V. The voltage offsets can be seen due to the offsets programmed at the input FGpFETs. The positive input was biased at 10 nA current and the negative input bias was increased in steps of 10 from 10 to 60 nA. **e** The slopes of the curves show the g_m and R_{out} of the amplifiers

on the two FGpFETs at the inputs can be modulated thereby producing the required voltage offset. The FGpFET which acts as the current source is at a fixed bias while the input FGpFET is tuned from 10nA to 60nA. This tuning is responsible for the voltage shifts as observed in the V_{out} vs V_{in} voltage transfer curve in Fig. 9(d).

6 Verification and building level=1 systems using level=2 models

To further verify and validate the models, in addition to the circuits built in Sect. 4.2, other circuits and systems are built from the level=2 models and the experimental measurements are compared with the simulations, seen in Fig. 10.

The OTA is connected in a follower configuration, to act as a first order low pass filter, which is driving a capacitive load, considered to be the sum of the parasitic and routing capacitances at the output node of the OTA. FGOTA could also be used here to get a better linearity. The response of the filter integrating the step input is close to the simulations, with the time constant, τ, being $2U_T\mathrm{C}/\kappa I_{bias}$.

$$\tau\frac{\mathrm{d}V_{out}}{\mathrm{d}t}=\frac{2U_T}{\kappa}\tanh\frac{\kappa(V_{in}-V_{out})}{2U_T}$$

Which when linearized,

$$\tau\frac{\mathrm{d}V_{out}}{\mathrm{d}t}+V_{out}=V_{in} \tag{3}$$

Consider the Capacitively Coupled Current Conveyer, C^4 based second-order band-pass filter [24]. The transconductances of the FGOTAs and the feedback and load capacitance set the time constants of the filter. A step input of 400mV around $V_{dd}/2$ is applied and again there is a close overlap between the data and the simulation results. The equations for C^4 and its level=1 modeling is shown in [3].

Building on these components and systems, the analog front-end of a speech processing system is shown here in Fig. 11 [1, 25]. It consists of the C^4 which splits the incoming band within a set of frequencies as determined by the bias currents set by the FGOTAs in the bandpass filters. This is fed to a minimum detector to track the amplitude at the minimum points on the envelope of the input waveform. The time constants are set by the load capacitances and the currents to which the FG-pFET and OTA are biased.

The bias current of the OTA in the minimum detector is tuned such that it stays in high gain, assuming the pFETs are in Sub V_{T0} saturation. The LPF at the output filters the ripples and smoothens the minimum detector output to give a clean result. There is a close overlap in the C^4 output, while the dynamics of the minimum detector is very close in both the data and simulations where it closely follows the minimas. An exact fit is not obtained though, in the midrange, since there is a slight DC offset observed in the output. Through this system, we have further emphasized that we get an accurate correspondence between the simulations and the experimental data for larger systems too.

7 Conclusion

An open-source simulator based on EKV models has been implemented and shown here, which is used to obtain accurate results close to the experimental results from the

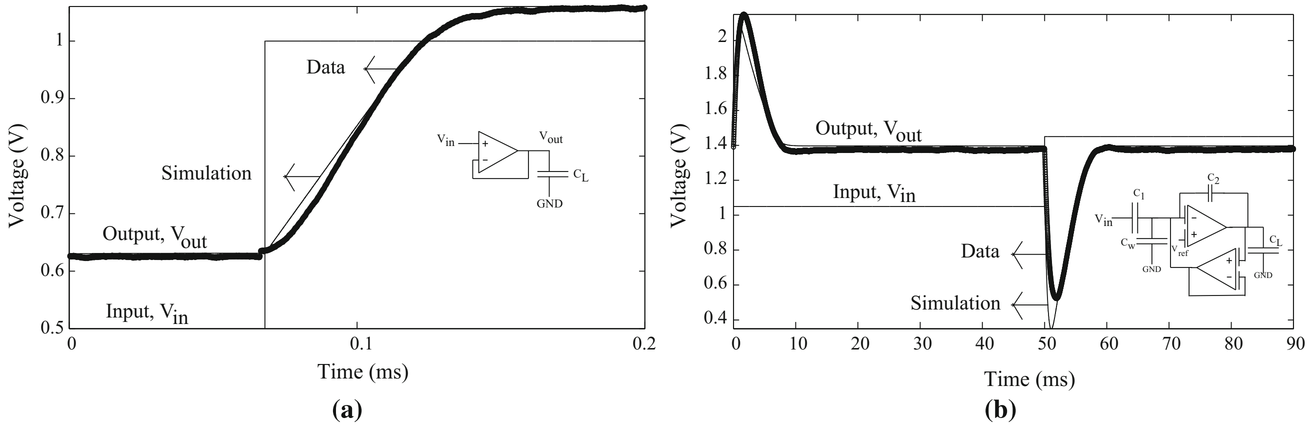


Fig. 10 Simulated and measured output voltage as a function of input voltage for Continuous-time Filters. **a** Step response for a first-order low-pass filter. A step input is given to an OTA biased at 5 nA, connected in a follower configuration. The extracted capacitance value, which comprises the parasitic and routing capacitance at the output of OTA for the required time constant is 460 fF. **b** Step response for a capacitively coupled current conveyer, C^4 based bandpass filter. A step input from 1.05 to 1.45 V is given to a FGOTA biased at 50 nA, with a DC input bias, V_{ref} at 1.4 V, while the feedback FGOTA is biased at 500 pA, to get the required dynamics

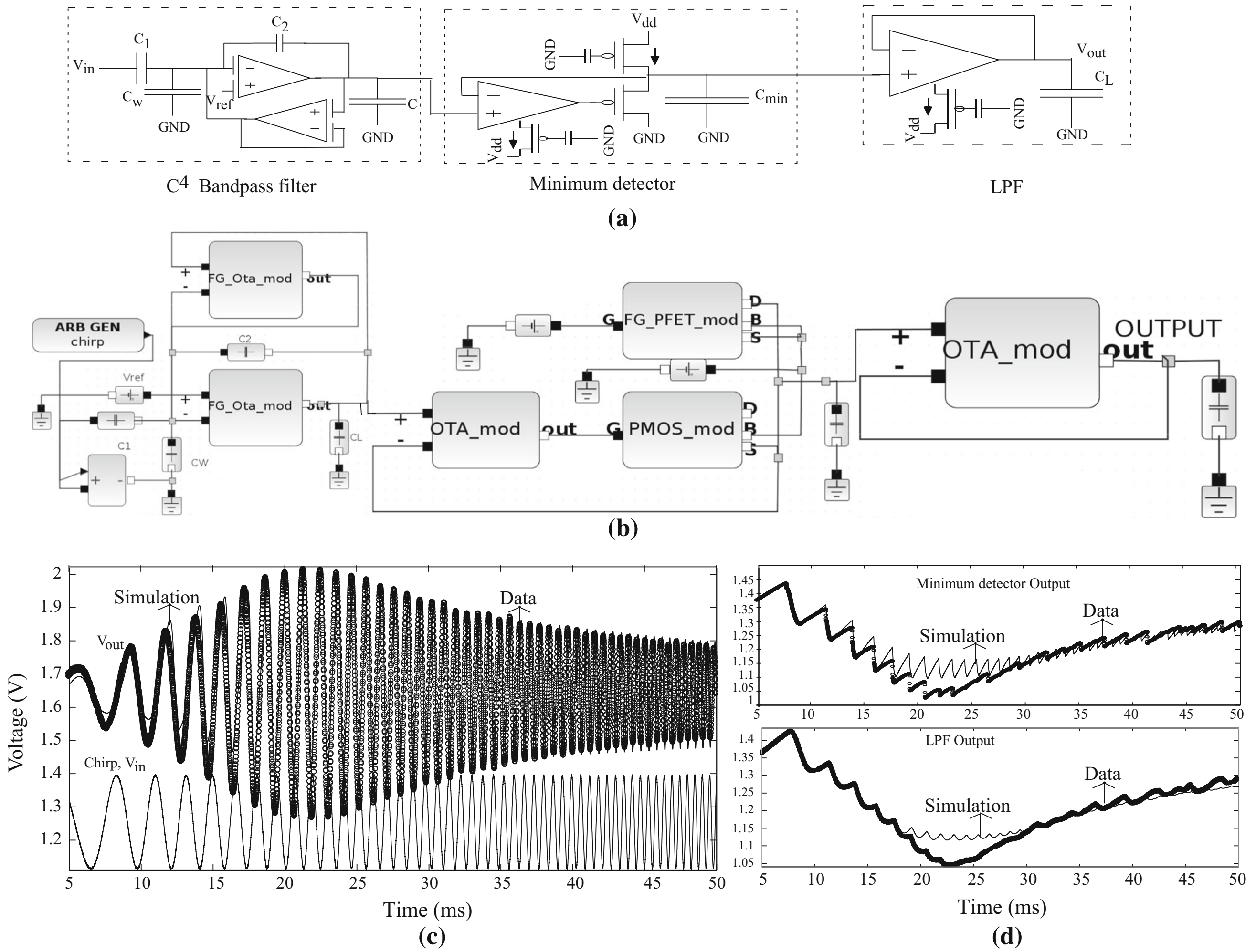


Fig. 11 Simulated and measured dynamics for the analog front-end of a speech processing system. **a** The schematic of the system consisting of a C^4 bandpass filter, minimum detector and LPF are shown. **b** The Xcos/Scilab diagram with the level=2 models corresponding to each block, used for simulation is shown. **c** A chirp input between 1 Hz and 20 Khz with an offset of 200 mV around 1.25 V is applied at the input of the system. The C^4 limits the spectral band to a set of frequencies, set by the FGOTA in the gain stage biased at 300 nA, while the feedback stage FGOTA is biased at 6 nA. **d** The minimum detector tracks the minimum points while the LPF gives out a smooth result

Table 1 Comparison of simulation times with a conventional Ngspice simulator

	With level=2 OTA (s)	With all level=2 36T (s)	Ngspice with EKV (s)	Ngspice with BSIM (s)
Sin input (20 Hz) over 50 ms	0.721	1.632	3.456	5.384
Chirp input over 50 ms	0.945	1.893	4.296	6.596
Sin input (1Khz) over 5 ms	0.548	1.526	2.985	3.241
Sin input (1Khz) over 5 s	105.732	206.132	457.938	508.143

FPAA. To further highlight the performance of our simulator, experiments have been performed to compare the behavior against the conventional Ngspice simulator, installed in the same virtual machine in the Ubuntu environment where the tool infrastructure is also located. Our simulator is similar to a Matlab/Simulink type environment based on Modelica being different from one with all the ODEs for the system written in C language, and shows a faster behavior. The Ngspice simulator [26, 27] uses the EKV model [28] in a 500 nm process while our simulator in this work uses a 350 nm process. The experiments were performed for the front-end of a speech processing system as shown in Fig. 11 for different set of inputs to identify the initial settling time, convergence and compilation times.

Level=2 models for the OTA were used for abstracting out the 9T-OTA structure, thereby giving an accurate and faster simulation while the netlist for the system consisting of 36T was simulated in Ngspice with the same time resolution. Even using a 36T structure for the system gives a faster behavior than Ngspice. The shorter input time gives us an idea about the initial setup time and solving for the DC operating point times while the longer input gives an estimate of the longer computation time, in general for larger systems.

A RK(45) solver is used, which determines the step size depending on the error in the solutions, thereby giving an accurate result, which can be observed from the overlap with the data from the FPAA in the results. We can see from the equations of the EKV model used in this paper that it's an analytic function with derivatives existing and being continuous over all values. This is advantageous as compared to a BSIM model with a large number of parameters which takes a longer time to converge since the error maybe higher due to the existence of discontinuities. Table 1 shows this aspect with the BSIM simulation time being greater than the simulation with the EKV model.

Moreover, the routing capacitances can also be estimated through transient responses by observing the time constants while measuring the output relevant to the particular system. The capacitances can be modeled in the simulations, customizing depending on the CAB location where the circuit elements have been placed, hence aiding in their characterization too.

References

1. George, S., Kim, S., Shah, S., Hasler, J., Collins, M., Adil, F., et al. (2016). A programmable and configurable mixed-mode FPAA soc. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *24*(6), 2253–2261.
2. Michell, G. D., & Gupta, R. K. (1997). Hardware/software co-design. *Proceedings of the IEEE*, *85*(3), 349–365.
3. Collins, M., Hasler, J., & George, S. (2016). An open-source tool set enabling analog-digital-software co-design. *Journal of Low Power Electronics and Applications*, *6*(1), 3.
4. Schlottmann, C. R., & Hasler, J. (2014). High-level modeling of analog computational elements for signal processing applications. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *22*(9), 1945–1953.
5. Foty, D. P. (1997). *MOSFET modeling with SPICE: Principles and practice*. Upper Saddle River, NJ: Prentice-Hall.
6. Cheng, Y., & Hu, C. (2002). *MOSFET modeling & BSIM3 user guide*. Berlin: Springer.
7. Tsividis, Y. P., & Suyama, K. (1994). MOSFET modeling for analog circuit cad: problems and prospects. *IEEE Journal of Solid-State Circuits*, *29*(3), 210–216.
8. Guerra-Gómez, I., Mcconaghy, T., & Tlelo-Cuautle, E. (2013). Operating-point driven formulation for analog computer-aided design. *Analog Integrated Circuits and Signal Processing*, *74*(2), 345–353.
9. Hasler, J., Shah, S., Kim, S., Lal, I. K., & Collins, M. (2016). Remote system setup using large-scale field programmable analog arrays (FPAA) to enabling wide accessibility of configurable devices. *Journal of Low Power Electronics and Applications*, *6*(3), 14.
10. Schlottmann, C., Petre, C., & Hasler, P. E. (2012). A high-level simulink-based tool for FPAA configuration. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *20*(1), 10–18.
11. Campbell, S. L., Chancelier, J.-P., & Nikoukhah, R. (2006). *Modeling and simulation in Scilab/Scicos*. New York, NY: Springer.
12. Lam, K.C.A., & Zwolinski, M. (June 2013). Circuit simulation using state space equations. In *2013 9th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)* (pp. 177–180).
13. Tiller, M. (2001). *Introduction to physical modeling with modelica*. Berlin: Springer.
14. Enz, C. C., Krummenacher, F., & Vittoz, E. A. (1995). 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*, *8*(1), 83–114.
15. Low, A., & Hasler, P. (1999). Cadence-based simulation of floating-gate circuits using the EKV model. In *42nd Midwest Symposium on Circuits and Systems* (vol. 1, pp. 141–144).
16. Hasler, P. E. (2005). Floating-gate devices, circuits, and systems, invited. In *Proceedings of the 5th IEEE International Workshop on System-on-Chip for Real-Time Applications (IWSOC 2005), 20-24 July 2004, Banff, Alberta, Canada* (pp. 482–487).
17. Tsividis, Y. P. (1987). *Operation and modelling of the MOS transistor*. New York, NY: McGraw-Hill.
18. Odame, K. M., & Hasler, P. E. (2009). Theory and design of OTA-C oscillators with native amplitude limiting. *IEEE Transactions on Circuits and Systems*, *56–I*(1), 40–50.
19. Mead, C. (1989). *Analog VLSI and neural systems*. Bostan, MA: Addison-Wesley Longman Publishing Co., Inc.
20. Kim, S., Hasler, J., & George, S. (2016). Integrated floating-gate programming environment for system-level ics. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *99*, 1–9.
21. Schlottmann, C., Abramson, D. N., & Hasler, P. E. (2012). A MITE-based translinear FPAA. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *20*(1), 1–9.
22. Graham, D. W., Farquhar, E., Degnan, B., Gordon, C., & Hasler, P. (2007). Indirect programming of floating-gate transistors. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *54*(5), 951–963.
23. Chawla, R., Adil, F., Serrano, G. J., & Hasler, P. E. (2007). Programmable g_m- C filters using floating-gate operational transconductance amplifiers. *IEEE Transactions on Circuits and Systems*, *54–I*(3), 481–491.
24. Graham, D. W., Hasler, P. E., Chawla, R., & Smith, P. D. (2007). A low-power programmable bandpass filter section for higher order filter applications. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *54*(6), 1165–1176.
25. Ramakrishnan, S., Basu, A., Chiu, L. K., Hasler, J., Anderson, D., & Brink, S. (2014). Speech processing on a reconfigurable analog platform. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *22*(2), 430–433.
26. Ngspice Web Site. http://ngspice.sourceforge.net/.
27. Nenzi, P., & Vogt, H. Ngspice users manual. http://ngspice.sourceforge.net/docs/ngspice26-manual.pdf.
28. *EKV Website*, http://ekv.epfl.ch/.
29. Degnan, B. (2013). *Temperature robust programmable subthreshold circuits through a balanced force approach.* Ph.D. dissertation, Georgia Institute of Technology, Atlanta, GA, May 2013.

Aishwarya Natarajan received her B.S. from University of Mumbai, India, and M.S. from Georgia Institute of Technology, Atlanta. She is currently pursuing the Ph.D. degree in electrical and computer engineering at Georgia Institute of Technology, Atlanta. Her research interest involves analog and mixed signal integrated circuits and systems design, and low-power bio-inspired circuits and systems.

Jennifer Hasler is a Professor in the School of Electrical and Computer Engineering at Georgia Institute of Technology. Dr. Hasler received her M.S. and B.S.E. in Electrical Engineering from Arizona State University in 1991, and received her Ph.D. From California Institute of Technology in Computation and Neural Systems in 1997. Her current research interests include low power electronics, mixed-signal system ICs, floating-gate MOS transistors, adaptive information processing systems, "smart" interfaces for sensors, cooperative analog-digital signal processing, device physics related to submicron devices or floating-gate devices, and analog VLSI models of on-chip learning and sensory processing in neurobiology. Dr. Hasler received the NSF CAREER Award in 2001, and the ONR YIP award in 2002. Dr. Hasler received the Paul Raphorst Best Paper Award, IEEE Electron Devices Society, 1997, IEEE CICC best paper award, 2005, Best student paper award, IEEE Ultrasound Symposium, 2006, IEEE ISCAS Sensors best paper award, 2005, and best demonstration paper, ISCAS 2010.