A 65nm CMOS Analog Programmable Standard Cell Library for Mixed-Signal Computing

Pranav O. Mathews, *Graduate Student Member, IEEE,* Praveen Raj Ayyappan, *Graduate Student Member, IEEE,* Afolabi Ige, *Graduate Student Member, IEEE,* Swagat Bhattacharyya, *Graduate Student Member, IEEE,* Linhao Yang, *Graduate Student Member, IEEE,* and Jennifer O. Hasler, *Senior Member, IEEE*

***Abstract*—Integrated Circuit (IC) design for analog computing requires similar toolflow and synthesis as large-scale digital systems, in-turn necessitating a library of general-purpose analog cells. To this end, we present a programmable, Floating-Gate (FG)-based analog standard cell library in a commercially available 65nm process that allows analog IC designers to use synthesis tools with an abstracted design mindset similar to large-scale digital design. We fabricate the test cells, which include filters with programmable corners, an analog classifier, and an arbitrary waveform generator; experimentally characterize FG programming; and experimentally demonstrate the performance of the standard cells. Overall, the standard cells achieve a similar or smaller footprint than previous approaches while leveraging the benefits of FG programming at smaller technology nodes.**

***Index Terms*—Floating-Gate, FPAA, Analog Standard Cell, Reconfigurable Analog, 65nm**

I. The Need for Analog Standard Cells

A library of analog standard cells has been created that allows designers to flexibly shift parameters to accommodate specific design needs. This mindset is similar to digital design, where standard cells allow designers to abstract architectures and reuse cells through high level synthesis tools, a process so widespread that open-source tools have emerged ([1]–[3]). An analog standard cell library enables designers to sidestep a pervasive issue with traditional analog integrated circuit (IC) design: needing to significantly modify a large number of parameters to meet new specifications even for an existing architecture, a problem that is magnified when considering large-scale analog computing ([4]–[6]).

Recent tools that are designed to alleviate the high cost of analog redesign have moved towards using basic analog primitives to build small analog components, focusing primarily on layout automation from an existing schematic ([7], [8]). Current analog design tools focus mostly on macromodeling for mixed-signal simulation ([9]–[15]). Both these tools help expedite new designs in the verification and layout stages but do not fix the core requirement that a large number of parameters need to be tweaked for new specifications.

Floating-Gate (FG) analog standard cells allow for a digital mindset through programmability. The programmability of FGs is a solution to the requirement that any standard analog structure needs to be able to meet multiple design specifications, which has limited previous attempts towards analog tools for system design ([16]–[20]). An entire library of standard cells has been built in the open-source Skywater 130 nm CMOS process with a test structure being taped out for characterization ([21], [22]).

Pranav O. Mathews, Praveen Raj Ayyappan, Afolabi Ige, Swagat Bhattacharyya, Linhao Yang, and Jennifer O. 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

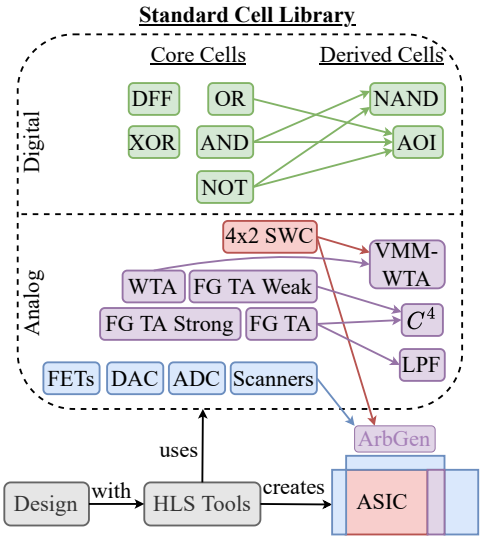
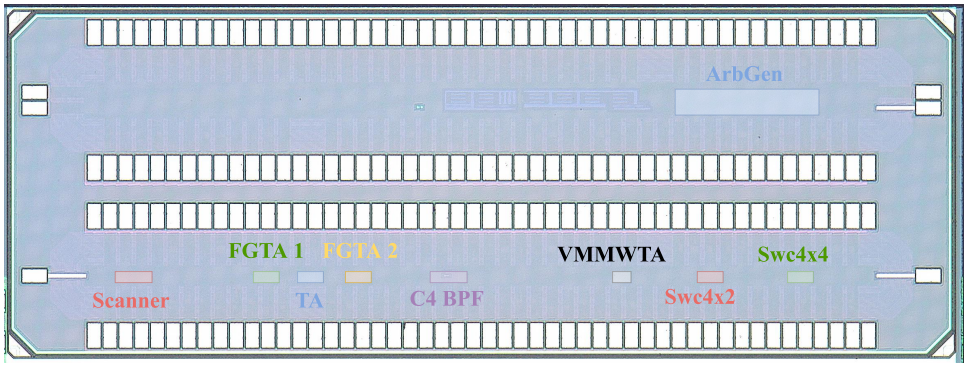


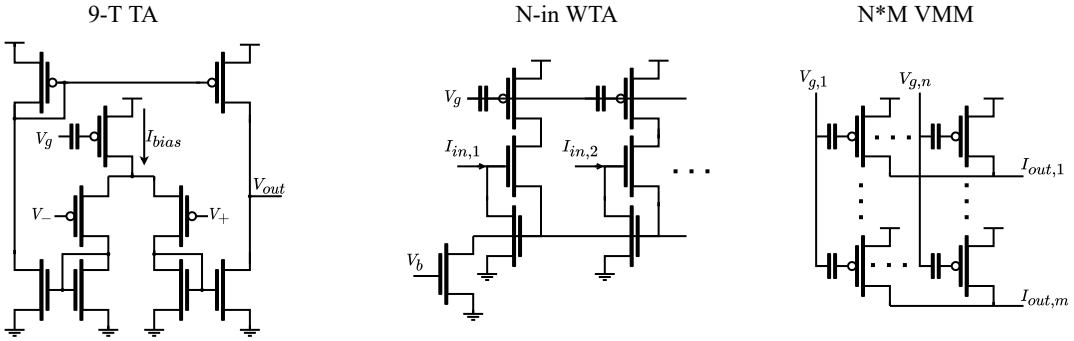
Fig. 1. Analog high-level synthesis (HLS) tools require a flexible, programmable analog standard cell library. [23]. Cells can be FG based (red), non-FG based (blue), or a combination (purple); systems and more complex standard cells can be created with combinations of core cells, similar to digital design. HLS tools can take these cells and synthesize analog systems.

In this work FG analog standard cells and their underlying design mindset have been used to generate a standard cell library in a commercially available 65 nm process; cells were fabricated and thoroughly tested experimentally both by themselves and as part of small systems such as an Arbitrary Waveform Generator (AWG) (Fig. 2). The standard cells share the same 6.5 µm pitch and are arranged so that the FG cells share drainlines, following history that collapses FG systems into one large crossbar for programming to isolate a FET with gateline and drainline control. The systems built out of multiple standard cells (AWG) use on-chip FG control circuitry for FG isolation and programming selection similar to Field Programmable Analog Array (FPAA) devices [24].

The standard cells tested in the 65 nm process extend the initial 130 nm discussion ([21], [22]), following the same design and pitch with small optimizations for the different process requirements. Because both sets of standard cells



(a)



(b)

Programmable Analog Standard Cell Library

FG Cells	BPF	2-TA	4x2 VMM	4-WTA	AWG	FG Char	FG Bootstrap
65nm Area (μm^2)	420.2	116.2 175.4 (FG)	119.11 (Direct) 13.8 (Indirect)	89.4	21,036	198.1	119.7
130nm Area (μm^2)	266.6	116.5 182.6 (FG)	130.88 (Direct) 130.78 (Indirect)	91.5	–	238.6	194.7

Non-FG Cells	4-bit Cap Bank	4-SPST T-gate	6-bit DAC	Scanner (4x)	2 nFET + 2 pFET (W/L = 10)
65nm Area (μm^2)	–	31.4	6,533	177.1	18.1
130nm Area (μm^2)	238.6	30.9	6,632	304.6	22.9

(c)

Fig. 2. The programmable analog standard cell library. (a) Die photo of the 65nm test chip. (b) Transistor level schematics of some circuits in the standard cell library. (c) List of standard cells and their area, both in 65nm and 130nm [21].

(130 nm and 65 nm) are designed to be the same, they can both uniquely be used in existing analog high-level synthesis tools ([23]); the tools in [23] use an island concept, a method of grouping FG and non-FG cells together into programmable blocks, to effectively generate application-specific IC (ASIC) designs from the proposed standard cells to meet analog computing benchmarks ([25]). Also of interest are the synthesis and targeting of FPAAs, devices that further enable a standard framework for flexible analog computing ([23], [26], [27]).

This effort focuses on the first experimental measurements and resulting characterization of fabricated programmable analog standard cells in a 65 nm process (Fig. 2). Section II explains the characterization of FG programming parameters in the 65 nm process. FG devices are then programmed to test the core standard cells (Section III): nFET and pFET blocks, programmable Transconductance Amplifiers (TAs), programmable filters, analog scanners, and voltage DACs as a subset of the overall library. Section IV tests single island systems built out of the 65 nm standard cells, demonstrating how the individual blocks can come together to form classifiers and waveform generators. Finally the capabilities of the standard cells and how they can be further used is summarized in the closing discussion (Section V).

II. Characterizing Floating-Gate Programming

Programmability is essential to maximize the use of any standard cell. FGs are highly programmable analog devices, allowing circuits to adapt to a wide variety of specifications. Characterization of FG programming and subsequent programming algorithm development is imperative for effectively using FG standard cells.

FG devices consist of a standard CMOS transistor with a capacitively coupled input. The FG node, or gate voltage, can be shown as

$$V_{fg} = \frac{C_1}{C_T} V_g + \frac{C_{tun}}{C_T} V_{tun} + V_Q, \tag{1}$$

where C_T is the total capacitance on the FG node, C_1 is the control gate capacitance, C_{tun} is the tunneling capacitance,

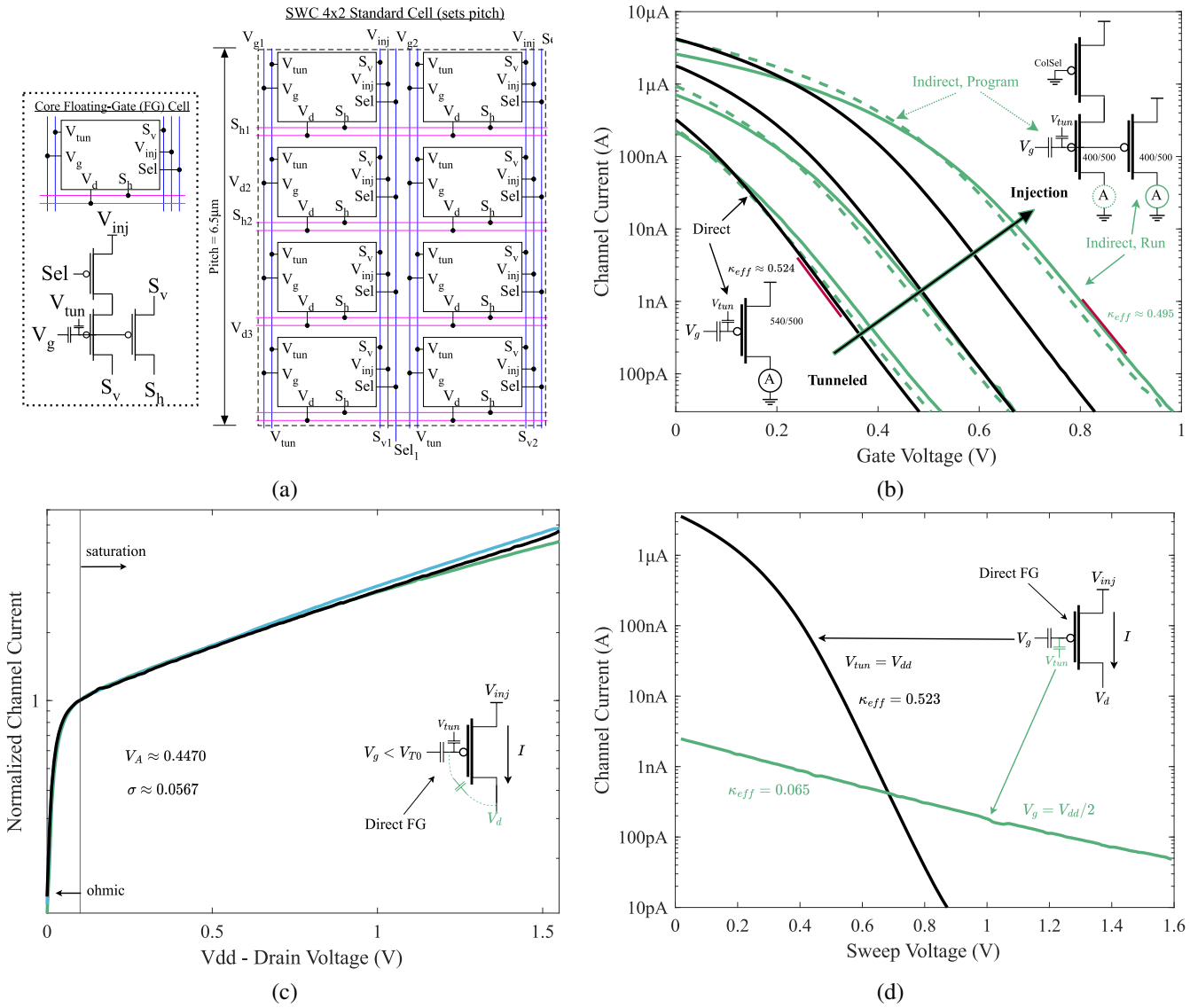


Fig. 3. Transistor characteristics of the Floating-Gate (FG) devices used in a Vector-Matrix-Multiplier (VMM) structure. (a) High-level schematic of the indirect 4x2 VMM standard cell. As the core FG cell in the library, it sets the pitch of the every cell, 6.5 μm for 65nm. (b) Gate sweeps on a single FG transistor in direct and indirect 4x2SWC cells over different injection levels. Different programmed charges allow control over the response of a FG transistor. An indirect FG VMM uses an auxiliary transistor that shares a FG with the transistor used in operation, allowing for programming without additional switches; however, threshold mismatch between the run and program FETs can lead to variation. (c) Drain sweeps of a FG transistor, showing the increased capacitance coupling from the drain side. This can be modeled as an effective σ. (d) Sweep of the larger gate input capacitor and the much smaller tunneling capacitor. Both capacitors couple their input to the FG Node according to their capacitance.

and V_Q is the voltage resulting from any charge on the FG node. A MOS varactor is used for both the tunneling junction and control gate, allowing FG devices to be built in a standard $65\,\mathrm{nm}$ CMOS process rather than requiring a special double-poly option or process.

The following subsections discuss the framework around which FG devices are integrated into analog standard cells. Section II-A gives an overview of the core FG standard cell, the 4x2SWC cell, and explains how it forms the basis for every other standard cell in a given process. Section II-B discusses methods to find programming parameters for FG devices and resulting measurements from the $65\,\mathrm{nm}$ test chip. Section II-C adds on by discussing the Autozeroing Floating Gate Amplifier (AFGA) and the resulting programming parameters that can be extracted from the circuit.

A. The Floating-Gate VMM

The basic building block of the FG standard cells is the 4x2 Vector-Matrix-Multiply (VMM) block, which brings FG devices into a crossbar structure. VMMs are ideal for programming because of their selectivity and useful as a computational element (Fig. 3a). A VMM is also an area-efficient way to package FG devices, as any biases needed for a FG cell can be pulled from the VMM. Because most FG-based cells use the 4x2 VMM at their core, the overall pitch for a standard cell is set by the 4x2 VMM, as cell height is constrained by the minimum necessary height of an FG crossbar.

FG programming allows each transistor in the VMM to be set to a desired current for use in a design. Tunneling acts as a global erase, setting all currents in the VMM to

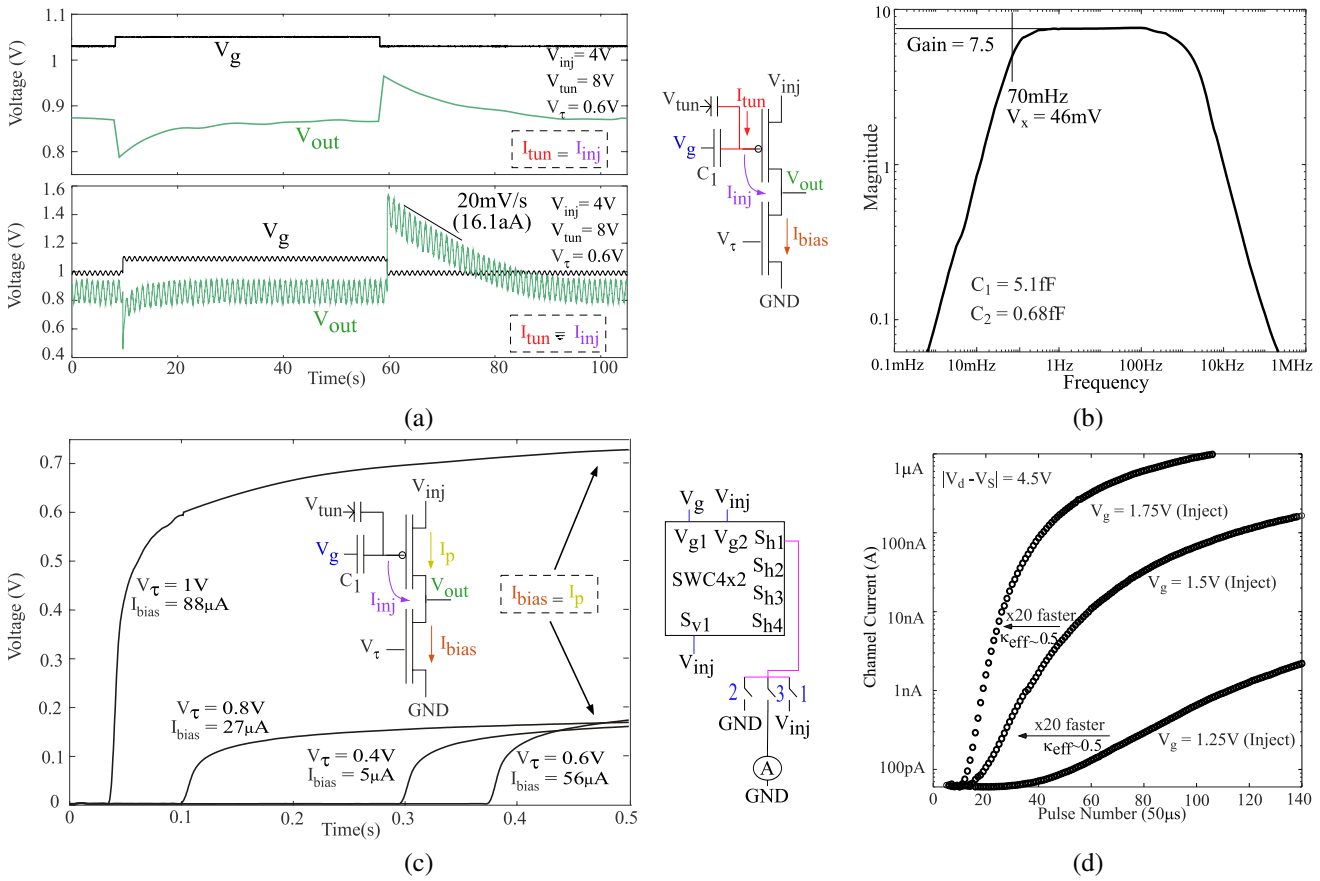


Fig. 4. Characterization of FG devices, where (a-b) show initial tunneling and injection measurements and (c-d) show precision programming. (a) An Autozeroing Floating-Gate Amplifier (AFGA) showing a bandpass response to a small step input in. The adaptation is due to injection and tunneling currents in the top FG device, the circuit returns to equilibrium ($I_{tun} = I_{bias}$) after an external perturbation. (b) Frequency response of the AFGA for a small 10 mV sinusoidal input. The corners are set by the injection and tunneling currents. (c) Programming with the AFGA. The bottom nFET is held to the desired current and the circuit is ramped to injection voltages at t=0; over time the FG pFET injects and V_{out} rises indicating that the desired current has been reached. (d) Change in current output for a fixed gate voltage over many injection steps. The typical S-curve response is shown for different V_{ds} during injection.

an extremely low value (on the order of 10 pA). Injection increases drain current by adding electrons to the gate of the FG FET, decreasing the gate voltage and allowing the same device to operate over several decades of current biases.

There are two 4x2 VMMs that emerge from the two standard FG architectures: direct and indirect (Fig. 3b). Direct VMMs are programmed by asserting voltages on and measuring the same FG pFET that will be used during operation, using switches to disconnect the FG pFET from its circuit during programming and reconnect it during run mode. Indirect VMMs assert voltages on and measure a separate program pFET that shares its gate with the run mode bias pFET. Although sharing the gate reduces switch count, it introduces mismatch between the run and program mode pFETs.

External signals couple into the gate of a FG pFET through the input capacitor, designed to be larger than parasitics to increase the coupling. A normal pFET has a coupling coefficient from the gate as κ, but the FG capacitor decreases the effective coupling to κ_{eff} (Fig. 3a,b). The tunneling voltage also couples onto the gate, an effect that is minimized by decreasing the size of the tunneling capacitor. This leads to a $\kappa_{eff,tun}$ that is much smaller compared to κ_{eff} (Fig. 3d).

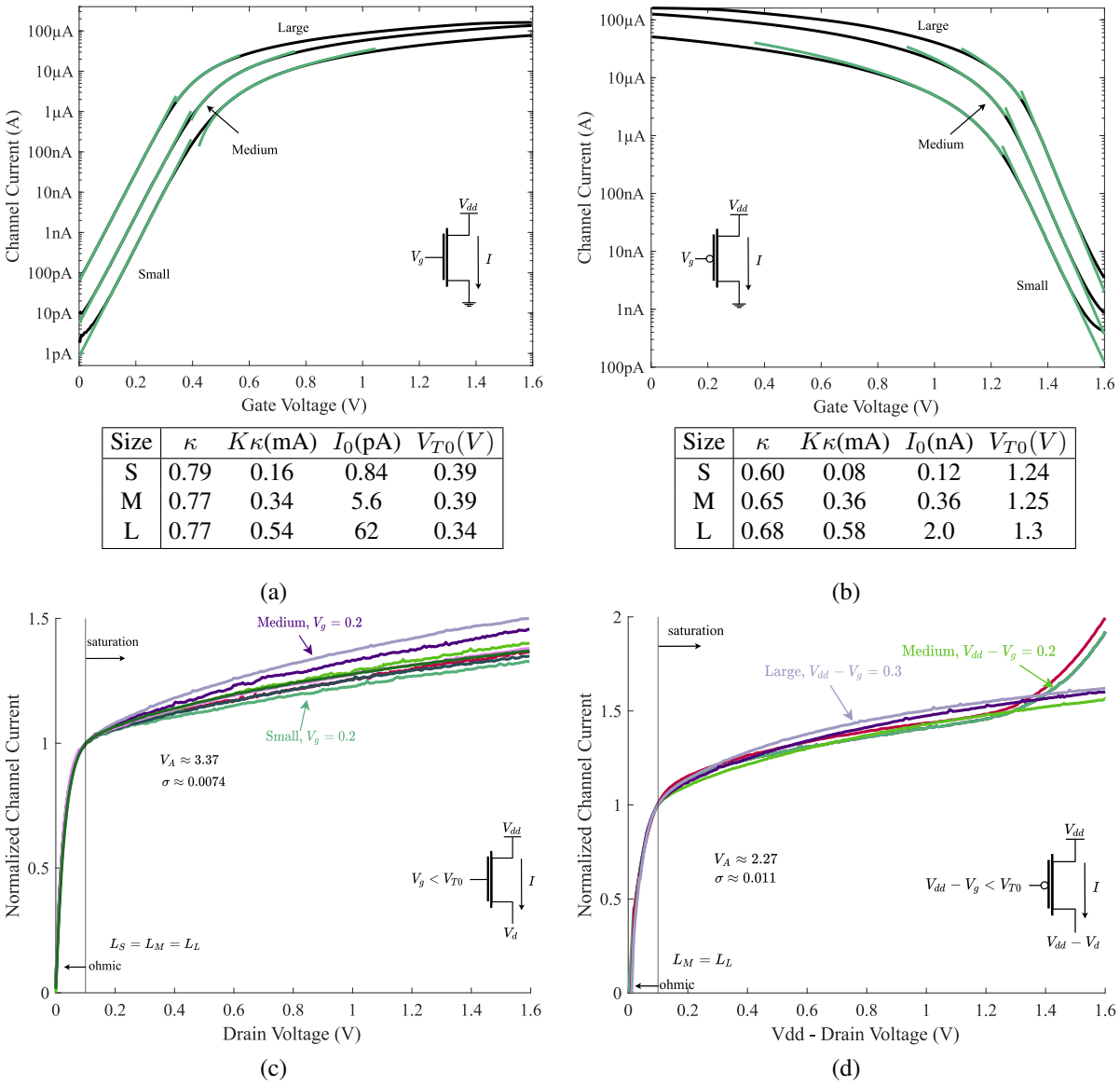
Drain to gate capacitance is typical in any FET. In normal devices the gate voltage is fixed by some driver and therefore does not change with variations on the drain. In a FG FET this is not true and the coupling does affect the floating node, modeled alongside the DIBL effect as an effective σ or Early voltage (Fig. 3c).

B. FG Programming Characterization

Varactors or double-poly capacitors are the best options for coupling to an FG because of potential leakage currents from gate contacts on floating node from external coupling capacitors. The process node in this work required varactors, allowing for FG coupling at the cost of a nonlinear capacitance to voltage relationship and an area penalty due to well spacing process requirements.

The charge on the FG node, V_Q, is changed through two processes: hot-electron injection and electron tunneling. To inject a FG pFET the source is raised to a higher voltage (4.5 V in 65nm), the gate is fixed, and the drain is pulsed from the source voltage to a low voltage. Injection rate depends on the V_{sd} and source current into the transistor I_s, which also depends on V_g; injection current can be modeled as

$$I_{inj} = I_{inj,0} \left(\frac{I_s}{I_{th}} e^{-\kappa \Delta V_{fg}/V_{inj}} \right) e^{-\Delta V_{ds}/V_{inj}}. \qquad (2)$$



Size	κ	$K\kappa$(mA)	I_0(pA)	$V_{T0}(V)$
S	0.79	0.16	0.84	0.39
M	0.77	0.34	5.6	0.39
L	0.77	0.54	62	0.34

Size	κ	$K\kappa$(mA)	I_0(nA)	$V_{T0}(V)$
S	0.60	0.08	0.12	1.24
M	0.65	0.36	0.36	1.25
L	0.68	0.58	2.0	1.3

Fig. 5. EKV parameters extracted from the low-voltage transistor standard cells. These standard cells come in three different sizes (small, medium, and large) and in 'p' and 'n' type variants. (a) Gate sweeps of nFET devices with different W/L (1.5,9.5,95) and their extracted EKV parameters. (b) Gate sweeps of pFET devices with different W/L (1,9.6,96) and their extracted EKV parameters. (c) Subthreshold drain sweeps of nFET devices with different W/L. Each transistor varies in W but has the same L leading to similar drain coupling for each. (d) Subthreshold drain sweeps of nFET devices with different W/L. The medium and large transistors have the same L, leading to similar drain coupling for both. The medium transistor experiences some hot impact ionization for a higher V_{ds}, leading to the double exponential curve.

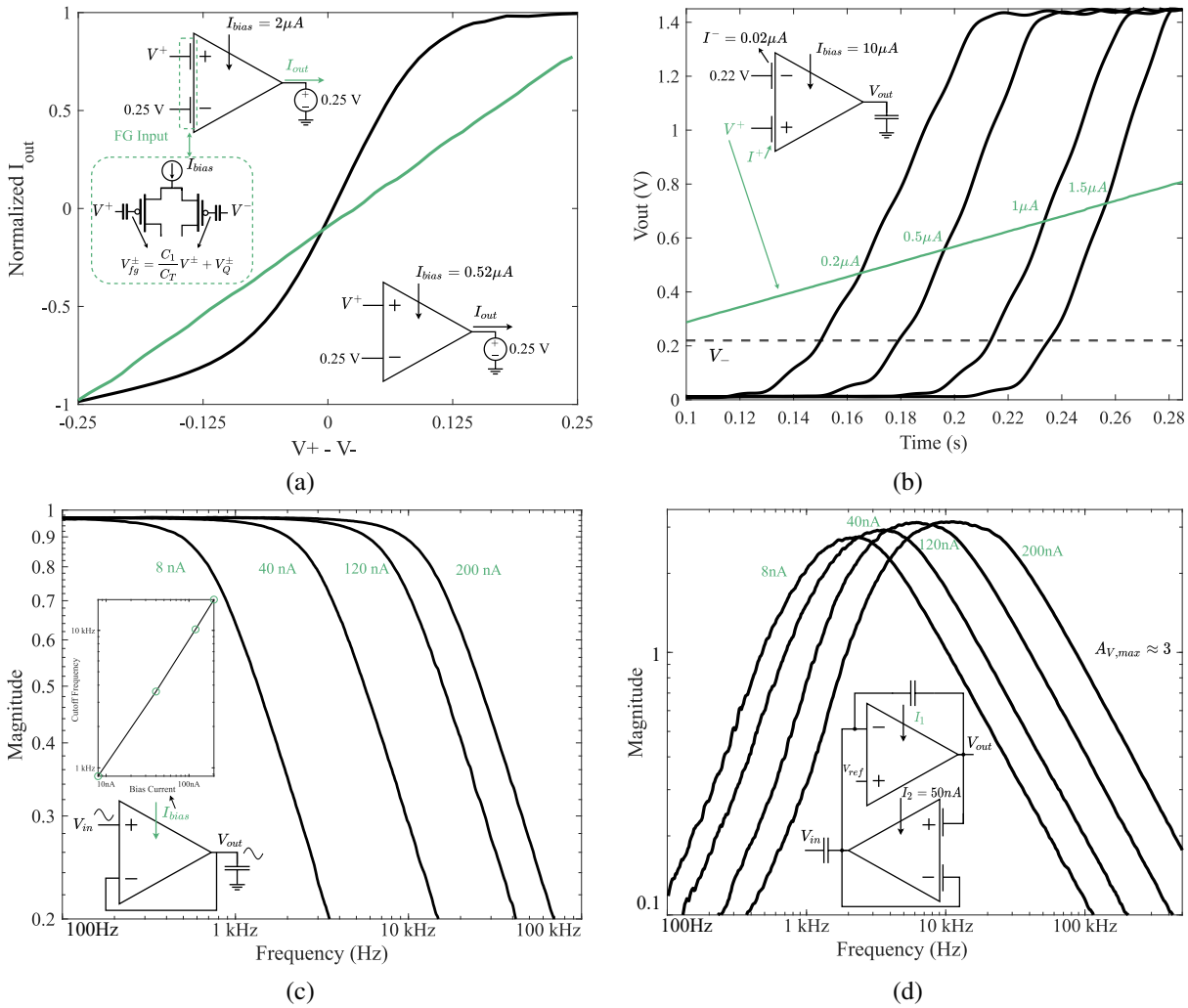
Starting from a tunneled, subthreshold state, each pulse injects more electrons onto the floating node, decreasing V_{fg}, which increases the FG pFET source current. In the subthreshold region higher source current increases injection rate with each subsequent pulse. Once the current reaches above-threshold levels the injection rate drops with each subsequent pulse and the source current saturates, resulting in the characteristic s-curve (Fig. 4c).

In contrast, tunneling reduces source current by removing electrons from the FG node via a high electric field from the tunneling junction to the FG node, increasing V_{fg}. As V_{fg} rises, the FG pFET source current and the voltage difference across the tunneling capacitor decrease, which reduces the tunneling rate asymptotically to a halt.

C. The Autozeroing FG Amplifier

The Autozeroing Floating Gate Amplifier (AFGA) uses continuous tunneling and injection currents to dynamically respond to a changing input signal [29]. A FG pFET is set above an nFET, with both the tunneling voltage and the source voltage high to facilitate tunneling and injection. As the input voltage changes, the current from the FG pFET changes to be above or below the bias current set by the nFET. Injection and tunneling processes then work to restore equilibrium, changing the V_Q and moving the pFET current to equal the nFET (Fig. 4a). This process creates a bandpass filter, where the corners come from the injection and tunneling rates (Fig. 4b).

AFGA dynamics can also be used to target a specific current. The equilibrium state of the circuit is when the current



Iout vs ΔV	
Parameter	Result
I_{bias} (no FG)	520 nA
I_{bias} (FG)	2 μA
I^+	1.7 μA
I^-	2 μA

Vtrip vs ΔI^+	
I^+ (μA)	V_{trip} (V)
0.2	0.47
0.5	0.6
1.0	0.67
1.5	0.72

Lowpass Filter	
I_{bias} (nA)	f_c (kHz)
8	0.87
40	3.59
120	10.18
200	16.74

C^4 Bandpass Filter	
I_1 (nA)	f_{cen} (kHz)
8	2.25
40	3.57
120	6.17
200	11.39

(e)

Fig. 6. TA-based analog standard cells exhibiting different behavior depending on the programmed bias current. The core TA used is a 9-transistor variant. (a) Output current vs differential voltage of a TA programmed at 520 nA and a TA with weak FG inputs programmed at 2 µA. The capacitive divider on the input of the weak FG TA results in a linear range that is nearly rail-rail. (b) Output voltage of a TA with strong FG inputs over time as the negative terminal is fixed and the positive terminal it swept. The offset depends on the programmed charge on the FG inputs, and can be adjusted to create larger offsets. Offset charge can be directly programmed with the right programming structure ([28]). (c) An LPF created with a TA and capacitor. The cutoff frequency depends on the bias current and can be shifted through FG programming. (d) A C^4 bandpass filter over different center frequencies. By adjusting the bias currents in the TAs both the Q-factor and corner frequencies can be shifted. e) Programmed bias currents for each curve in (a-d).

through the top FG pFET equals the current through the bottom nFET. If the FG pFET starts tunneled and the circuit is ramped up, injection will occur at an accelerating rate until V_{out} rises, lowering the injection rate until the AFGA is at equilibrium and both currents are equal (Fig. 4c). A target current can be reached by choosing the starting nFET bias current to equal the target before injection starts. This method is less precise than a closed-loop pulsed method because overshoot is possible; injection takes a finite time to stop.

III. Measuring Analog Standard Cells

Once the specifics of the core 4x2 standard cell are put in place, the rest can be designed to fit in pitch. One would like to characterize each cell both to learn behavior and to extract physical parameters that can then be used in simulation models. Section III-A shows the characterization of nFET and pFET standard cells and the extraction of the relevant parameters from each sweep for fitting to the EKV transistor model. Several programmable TA circuits, both by themselves and in specific filter topologies are then characterized in Section

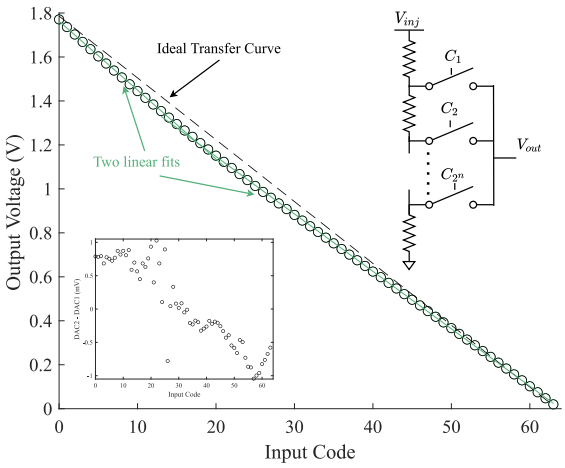


Fig. 7. Measurements of a voltage DAC (resistor string) standard cell from two different chips for all input codes. The outputs from both chips match well, and two straight line fits can be plotted against the DAC output curve.

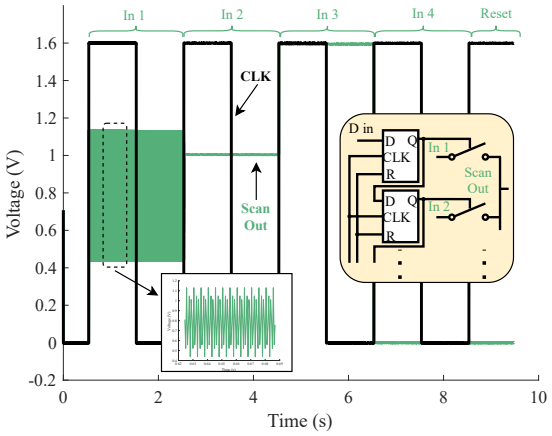


Fig. 8. Output of a four in one out analog scanner. Four different inputs are applied and the output only shows one at a time; the output shown is controlled by the digital shift register and changes as a "1" is clocked through.

III-B. Finally, other standard cells that are necessary on the periphery of systems, both for programming and measurement of fabricated circuits, are characterized in Section III-C.

A. Transistor Parameter Extraction

The fundamental nature of transistors makes them integral for the synthesis of larger cells and the development of accurate models for analog standard cell simulation. To accommodate different design requirements, three standard FET cells, both nFEt and pFET, are fabricated with W/L ratios of roughly 1, 10, and 100. We fit the source current versus gate and drain voltages through each transistor using a reduced EKV model to extract relevant parameters for analog modeling in both the subthreshold and above-threshold regions.

We perform gate sweeps on all devices followed by linear fits to extract EKV model parameters (Fig. 5). Assuming a saturated subthreshold transistor,

$$I = I_{th} e^{(\kappa(V_g - V_{T0}) - V_s + \sigma V_d)/U_T} = I_0 e^{(\kappa V_g - V_s)/U_T}, \quad (3)$$

where a linear fit to $\log(I)$ vs V_g extracts the gate coupling parameter κ, the zero current I_0, an approximation of the threshold current I_{th}, and the approximate threshold voltage V_{th}. A similar process is repeated for the above threshold case, where assuming saturation,

$$I = \frac{K}{2\kappa} (\kappa(V_g - V_t) - V_s)^2 (1 + \lambda V_d), \quad (4)$$

where a linear fit to $\sqrt{I}$ vs V_g extracts K/κ and another estimate of I_{th} and V_{th}.

An important parameter not extracted from the gate sweeps is channel length modulation, modeled in the EKV equations as $\sigma = 1/\lambda$, where λ is the Early voltage. A drain sweep and a linear fit to the saturated region gives us λ and therefore σ, for all transistors. Since the length of each standard cell transistor matched, λ is similar across the devices.

B. Programmable Transconductance Amplifiers

TAs have many uses in analog circuits ranging from amplifiers to G_mC filters where operation is contingent on TA bias current, I_{bias}. The TA standard cell offers programmable bias currents over several orders of magnitude by replacing the bias transistor with an FG pFET, enabling block reuse across designs (Fig. 6a). Two TAs are fit into one standard cell in order to the area efficiency of the FG biases.

FGs can also be used in the input differential pair, allowing designers to program or remove input offsets (Fig. 6b) ([30]). Depending on the strength (i.e. effective coupling) of the input capacitor, FG input differential pairs can also improve the linear range of a TA, which can be useful in various circuit topologies. One such example is the C^4 bandpass filter (Fig. 6c), which takes advantage of a wide linear-range TA in the feedback path to improve filter linearity [31].

A programmable bias currents allows G_mC filters with different frequency responses to be built out of a single block (Fig. 6c,d). The gain bandwidth product can be directly shifted, trading bias current power for a higher bandwidth in a lowpass filter (LPF) or adjusting the offset of the two TA bias currents to shift the center frequency of the C^4. As expected, the current and cutoff frequency scale linearly in the LPF.

C. Supporting Circuitry

The mechanisms for programming an FG FET have been discussed in section II; however, a practical array of FG FETs requires some external circuitry for selecting, isolating, and measuring a transistor during the programming process. A global run/program mode signal switches high-voltage FG FETs from any low-voltage transistors in the rest of a circuit during programming. Specific standard cells are designed for mode switching and for control in program mode. A ramp ADC can be used for measurement, and charge pumps can be used to control the high voltage levels in program mode [24].

Once the circuit is in program mode, analog multiplexers switch both drain and gate voltages to isolate the FG FET at the specified address and deselect all others. Unselected

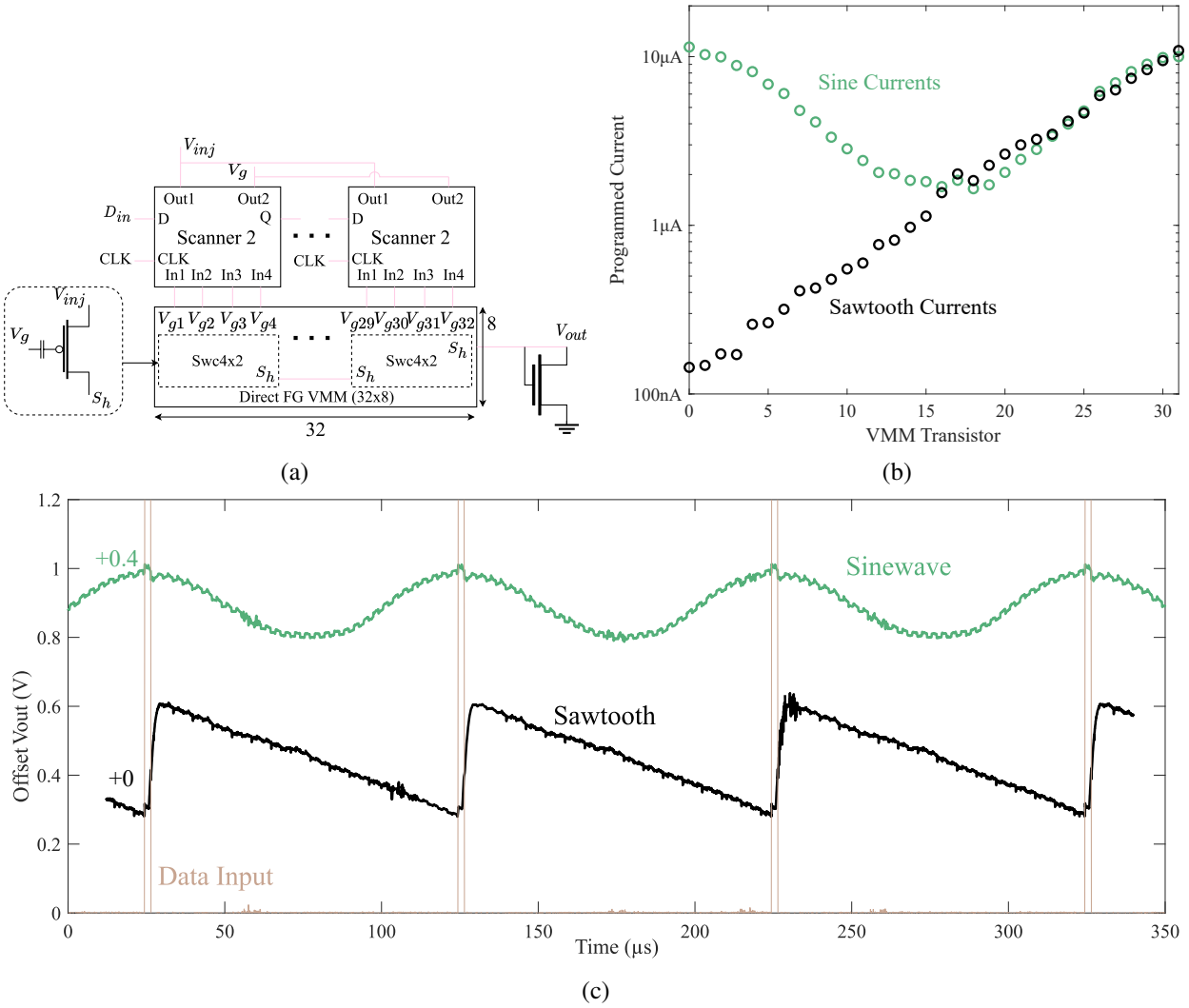


Fig. 9. An AWG made out of the analog standard cells. (a) A high-level schematic of the AWG, showing its composition of cells from the 65nm library. (b) Currents programmed into the VMM to output the corresponding waveform. (c) Output of the AWG programmed to a sine and sawtooth waveform, where the sine is offset for clarity (+0.4 DC); both waveforms would normally overlap.

gatelines (columns) are connected to the injection voltage to cut off all current, and selected gatelines are held at a voltage that is suitable for fast injection. Unselected drainines (rows) are connected to the injection voltage, and selected rows are connected to pulsing circuitry for controlled injection [28].

DACs set the gate and drain voltages for the selected FET. A monotonic 6-bit resistor string DAC was fabricated and experimentally found to roughly follow the ideal transfer curve for the circuit (Fig 7). The DAC transfer curve from two different chips have a maximal variation of $\pm 1\,\mathrm{mV}$, demonstrating the reproducibility of our approach.

An analog scanner is helpful for multiplexing analog outputs onto a single pin, reducing pin count for systems with a large amount of inputs/outputs (IO) or simplifying debugging at multiple test points. The analog scanner comprises a chain of D flip-flops which feed T-gates; clocking a 'one' through the system changes which output is allowed onto the shared output line. Fig. 8 shows an example of four different inputs being clocked out in order.

IV. Building Systems with Analog Standard Cells

Individual standard cells are combined to form systems, following architectural approaches from the island mindset. FG cells and their supporting circuitry occupy one side of the island, while lower voltage cells occupy the other side. The standard cells support a wide variety of computation, specifically aiming to perform well on standard benchmarks [25]. Section IV-A discusses an analog classifier efficiently solving a nonlinear problem, and Section IV-B shows a programmable arbitrary waveform generator (AWG) that can be used by itself or within larger analog computational systems.

A. Programmable Arbitrary Waveform Generator

An Arbitrary Waveform Generator (AWG) was constructed from the analog standard cells by combining direct VMMs into a larger 8x32 VMM, using the scanner block for the output and a scanner variant to select columns. The AWG uses FG FETs to store coefficients and a scanner to select which coefficients should be outputted. Outputs can be used

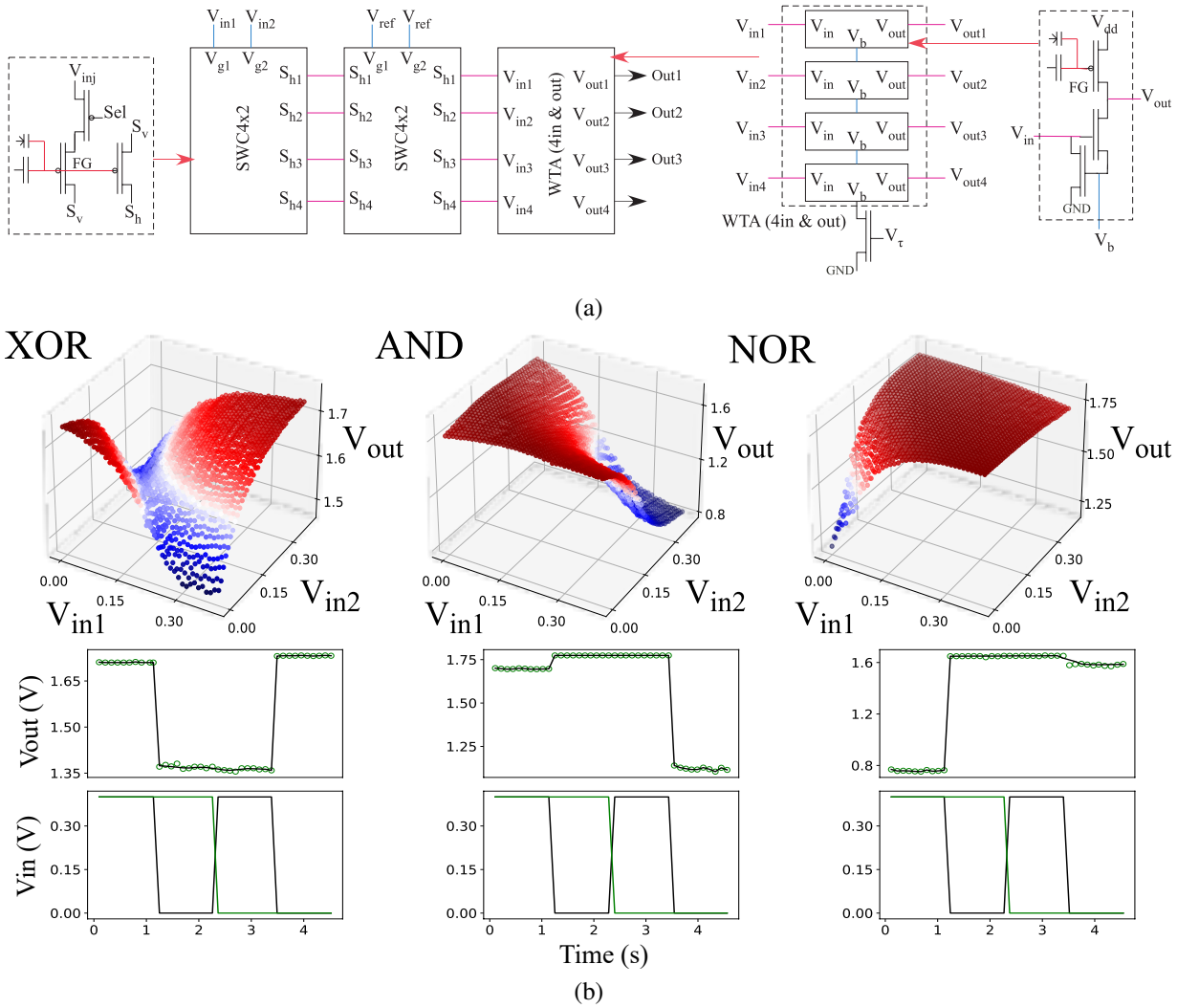


Fig. 10. A classifier constructed out of analog standard cells. (a) High level circuit schematic of the VMM+WTA, demonstrating how it is constructed out of other standard cells. (b) Outputs of the VMM-WTA when programmed to solve the XOR problem. Only one output represents the XOR solution, the other outputs work to set the non-linear decision boundaries but also solve the AND and NOR logic functions.

for waveform generation or matrix-matrix multiplication [32]. In particular, waveforms are the summation of any selected coefficients for each time step, which can be used to create functions such as a sine and sawtooth (Fig. 9). The frequency and precision of the output waveform are set by the input clock signal, and the FG programming, respectively; thus, the high precision of FG programming and high clock rate of digital circuits make the AWG an appealing architectural choice.

B. Computing with the VMM+WTA

Analog circuits can effectively perform machine learning classification; one classification structure is the single-layer, universal classifier that can be built out of an array of VMM standard cells and a Winner-Take-All (WTA) structure [33]. We program a 3x3 VMM+WTA such that the first column acts as a constant reference, and the last two accept binary inputs. Weights are programmed into the VMM so that the WTA output row wins when the inputs do not match, drawing two decision boundaries to solve the non-linear XOR problem within a single layer of computation (Fig. 10). Much larger structures can be built using the VMM+WTA concept, and the standard cells support extending this classifier to a multitude of applications.

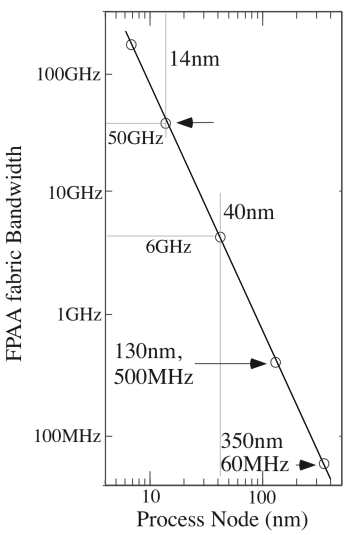
V. Discussion and Summary

This work presents the first experimental measurements from a fabricated library of programmable analog standard cells in 65 nm CMOS. Results show that FG FETs allow one cell to operate across a wide range of specifications via flexible biasing. Furthermore, the standardized layout and interfacing of each cell allows them to be tiled, leading to a system design process more akin to digital design, where a high level structure can be defined and then compiled to layout through the use of HLS synthesis tools.

Parameter extraction from experimental measurements, as shown in this work, are important for building accurate models for each standard cell, allowing for accurate macromodeling in a simulation tool. Once a standard cell is characterized, designers can simulate their circuits with greater confidence that the simulated results will match reality.

		Proposed	[21]	[24]	[34]	[35]
Tech Node (nm)		65	130	350	350	90
Programmable Biasing?		Yes	Yes	Yes	Yes	No
Voltage (V)	Supply	1.8	1.6	2.5	2.5	1.2
	Injection	3.25	4.0	6.0	6.0	–
	Tunneling	6.25	8.0	12.0	12.5	–
BPF Metrics	TA Input Lin. (mV)	±(85–850)	–	±(60–670)	±80	–
	Bandwidth (kHz)	1.8-15	–	0.1-4	0.02-20	0.13-0.47
	Power (µW)	1.1-7.6	–	0.02-0.6 (est)	0.001-1.1	0.005*
Cell Pitch (μm)		6.5	6.5	–	–	–
Area (μm^2)	BPF	420.2	266.6	222,071	133,000	–
	2-TA	116.2	116.5	3,733	–	–
	2-FG TA	175.4	182.6	5,599	–	–
	4x2 VMM (Direct)	119.11	130.8	1,452	–	–
	4x2 VMM (InDirect)	136.8	–	–	–	–
	4-bit Cap Array	–	238.6	1,577 (3-bit)	–	–
	4-SPST T-gate	31.4	30.9	1,428	–	–

(a)



(b)

Fig. 11. Comparisons of 65nm standard cells with other similar circuits. (a) Table comparing parameters of the measured $65\,\mathrm{nm}$ standard cells to other fabricated chips with similar systems. The $65\,\mathrm{nm}$ standard cells have a lower injection and tunneling voltage than prior implementations, and the area occupied by each standard cell is also less than counterparts in other designs. (b) Expected scaling of FPAA fabric bandwidth with process node, showing potential speed gains with a more advanced process [5].

Comparisons to other fabricated work, both ASICs and FPAAs, show favorable results from the $65\,\mathrm{nm}$ cells (Fig. 11a). Due to the smaller process node, both injection and tunneling voltages are lower than previous approaches. TA metrics also compare positively; the FG TA has a large linear range, and the C^4 bandpass filter has a wide tuning range while occupying less area than previous implementations ([24], [34], [35]). Each standard cell generally takes less area than comparable cells in other processes, and further optimization can further decrease the area.

The $65\,\mathrm{nm}$ cells are one part of an overarching standard cell library spanning multiple processes. Original designs for these cells were done in $130\,\mathrm{nm}$ and then ported over, [21], keeping the same pitch. The same process can be repeated for other technology nodes, allowing for area benefits as pitch can be optimized to scale down with technology and speed benefits even up to RF levels [36] (Fig. 11b). Having the same set of standard cells in multiple processes allows a designer to create a system without having to work in a specific technology, meaning that work done in one process is not wasted. In addition the process node becomes a design parameter, allowing designers to optimize between cost and scaling benefits.

Standard cells and HLS synthesis tools combine to form a design flow that can successfully and effectively target meaningful benchmark systems [25]. The goal of this paper, and ones that will follow, is to build the foundation of the standard cell design mindset with experimental data on individual cells and systems built with them. Other work will build upon these tools, expanding the capabilities of simulators, routing tools, and FPAA synthesis. Many open-source tools can be used ([26], [37], [38]), enabling designers across all spaces to use their skills in advancing higher level analog synthesis.

VI. Acknowledgement

The authors would like to thank Kevin Nielson for help with packaging chips and populating test boards.

References

[1] B. Khailany, E. Krimer, R. Venkatesan, J. Clemons, J. S. Emer, M. Fojtik, A. Klinefelter, M. Pellauer, N. Pinckney, Y. S. Shao, S. Srinath, C. Torng, S. L. Xi, Y. Zhang, and B. Zimmer, "INVITED: A Modular Digital VLSI Flow for High-Productivity SoC Design," in *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*. IEEE, Jun. 2018.

[2] M. Shalan and T. Edwards, "Building OpenLANE: A 130nm OpenROAD-based Tapeout- Proven Flow : Invited Paper," in *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, Nov. 2020.

[3] S. Reda, "Overview of the OpenROAD Digital Design Flow from RTL to GDS," in *2020 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Aug. 2020.

[4] M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. J. Yang, and R. S. Williams, "Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication," in *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, Jun. 2016.

[5] J. Hasler, "Large-Scale Field-Programmable Analog Arrays," *Proceedings of the IEEE*, vol. 108, no. 8, Aug. 2020.

[6] T. P. Xiao, C. H. Bennett, B. Feinberg, S. Agarwal, and M. J. Marinella, "Analog architectures for neural network acceleration based on non-volatile memory," *Applied Physics Reviews*, vol. 7, no. 3, Jul. 2020.

[7] H. Chen, M. Liu, B. Xu, K. Zhu, X. Tang, S. Li, Y. Lin, N. Sun, and D. Z. Pan, "MAGICAL: An Open- Source Fully Automated Analog IC Layout System from Netlist to GDSII," *IEEE Design & Test*, vol. 38, no. 2, Apr. 2021.

[8] T. Dhar, K. Kunal, Y. Li, M. Madhusudan, J. Poojary, A. K. Sharma, W. Xu, S. M. Burns, R. Harjani, J. Hu, D. A. Kirkpatrick, P. Mukherjee, S. Yaldiz, and S. S. Sapatnekar, "ALIGN: A System for Automating Analog Layout," *IEEE Design & Test*, vol. 38, no. 2, Apr. 2021.

[9] G. Boyle, D. Pederson, B. Cohn, and J. Solomon, "Macromodeling of integrated circuit operational amplifiers," *IEEE Journal of Solid-State Circuits*, vol. 9, no. 6, Dec. 1974.

[10] G. Casinovi and A. Sangiovanni-Vincentelli, "A macromodeling algorithm for analog circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, no. 2, Feb. 1991.

[11] R. Batra, P. Li, L. Pileggi, and Yu-Tsun Chien, "A methodology for analog circuit macromodeling," in *2004 IEEE International Conference on Cluster Computing (IEEE Cat. No.04EX935)*. IEEE, 2004.

[12] J. Wang, X. Li, and L. T. Pileggi, "Parameterized Macromodeling for Analog System-Level Design Exploration," in *2007 44th ACM/IEEE Design Automation Conference*, Jun. 2007.

[13] Y. Wei and A. Doboli, "Structural Macromodeling of Analog Circuits Through Model Decoupling and Transformation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 4, Apr. 2008.

[14] J. Kim, M. Jeeradit, B. Lim, and M. A. Horowitz, "Leveraging designer's intent: A path toward simpler analog CAD tools," in *2009 IEEE Custom Integrated Circuits Conference*, Sep. 2009.

[15] S. Liao and M. Horowitz, "A Verilog piecewise-linear analog behavior model for mixed-signal validation," in *Proceedings of the IEEE 2013 Custom Integrated Circuits Conference*, Sep. 2013.

[16] T. Pletersek, J. Trontelj, L. Trontelj, I. Jones, and G. Shenton, "High-performance designs with CMOS analog standard cells," *IEEE Journal of Solid-State Circuits*, vol. 21, no. 2, Apr. 1986.

[17] R. Bowman, "Introduction to CMOS analog standard cell ASIC design," in *Proceedings., Second Annual IEEE ASIC Seminar and Exhibit*. Rochester, NY, USA: IEEE, 1989.

[18] C. Toumazou, G. Moschytz, and B. Gilbert, "Trade-offs in analog circuit design : the designer's companion," 2002.

[19] D. R. D'Mello and P. G. Gulak, "Design Approaches to Field-Programmable Analog Integrated Circuits," *Analog Integrated Circuits and Signal Processing*, vol. 17, no. 1, Sep. 1998.

[20] B. C. Lim and M. Horowitz, "An Analog Model Template Library: Simplifying Chip-Level, Mixed-Signal Design Verification," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 1, Jan. 2019, conference Name: IEEE Transactions on Very Large Scale Integration (VLSI) Systems.

[21] J. Hasler, P. R. Ayyappan, A. Ige, and P. O. Mathews, "A 130nm CMOS Programmable Analog Standard Cell Library," *submitted (first revision) to IEEE Transactions on Circuits and Systems I*, 2023.

[22] J. Hasler, B. Muldrey, and P. Hardy, "A CMOS Programmable Analog Standard Cell Library in Skywater 130nm Open-Source Process," *WOSET*, 2023.

[23] A. Ige, L. Yang, H. Yang, J. Hasler, and C. Hao, "Analog System High-Level Synthesis for Energy-Efficient Reconfigurable Computing," *Journal of Low Power Electronics and Applications*, vol. 13, no. 4, Dec. 2023.

[24] 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, Jun. 2016.

[25] J. Hasler and C. Hao, "Programmable Analog System Benchmarks leading to Efficient Analog Computation Synthesis," *ACM Transactions on Reconfigurable Technology and Systems*, Sep. 2023.

[26] S. Kim, S. Shah, R. Wunderlich, and J. Hasler, "CAD synthesis tools for floating-gate SoC FPAAs," *Design Automation for Embedded Systems*, vol. 25, no. 3, Sep. 2021.

[27] J. Hasler, "The Rise of SoC FPAA Devices," in *2022 IEEE Custom Integrated Circuits Conference (CICC)*, Apr. 2022.

[28] 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*, 2015.

[29] Hasler, B. A. Minch, and C. Diorio, "An autozeroing floating-gate amplifier," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 1, Jan. 2001.

[30] V. Srinivasan, G. J. Serrano, J. Gray, and P. Hasler, "A Precision CMOS Amplifier Using Floating-Gate Transistors for Offset Cancellation," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 2, Feb. 2007.

[31] 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, Apr. 2012.

[32] A. Bandyopadhyay, J. Lee, R. Robucci, and Hasler, "MATIA: A Programmable 80μW/frame CMOS Block Matrix Transform Imager Architecture," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 3, Mar. 2006.

[33] S. Ramakrishnan and J. Hasler, "Vector-Matrix Multiply and Winner-Take-All as an Analog Classifier," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 2, Feb. 2014.

[34] B. Rumberg, D. W. Graham, S. Clites, B. M. Kelly, M. M. Navidi, A. Dilello, and V. Kulathumani, "RAMP: accelerating wireless sensor hardware design with a reconfigurable analog/mixed-signal platform," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*. ACM, Apr. 2015.

[35] M. S. Diab and S. A. Mahmoud, "Field programmable analog arrays for implementation of generalized nth-order operational transconductance amplifier-C elliptic filters," *ETRI Journal*, vol. 42, no. 4, 2020.

[36] J. Hasler and H. Wang, "A Fine-Grain FPAA fabric for RF+Baseband," *GOMAC*, 2015.

[37] A. B. Kahng, L. Wang, and B. Xu, "TritonRoute: The Open-Source Detailed Router," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 3, Mar. 2021, conference Name: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.

[38] K. E. Murray, O. Petelin, S. Zhong, J. M. Wang, M. Eldafrawy, J.-P. Legault, E. Sha, A. G. Graham, J. Wu, M. J. P. Walker, H. Zeng, P. Patros, J. Luu, K. B. Kent, and V. Betz, "VTR 8: High-performance CAD and Customizable FPGA Architecture Modelling," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 13, no. 2, Jun. 2020.