A Digitally Enhanced Dynamically Reconfigurable Analog Platform for Low-Power Signal Processing

Craig R. Schlottmann, *Student Member, IEEE*, Samuel Shapero, *Student Member, IEEE*, Stephen Nease, and Paul Hasler, *Senior Member, IEEE*

***Abstract*—We present a field-programmable analog array designed for accurate low-power mixed-signal computation. This 25-mm^2 350 nm-CMOS reconfigurable analog IC incorporates digital enhancements to increase compatibility in embedded mixed-signal systems. The chip contains 78 computational analog blocks (CABs) which house a variety of processing elements. There are 36 general CABs with hundreds of common analog primitives for computation, 18 digital-to-analog converter (DAC) CABs, each with 8-b compilable DAC capabilities, and 24 vector-matrix multiplier CABs, for low-power parallel processing. A floating-gate routing matrix connects these analog elements to one another, both within individual CABs and between CABs. To facilitate digital interfacing and dynamic reconfigurability, we included a novel network of volatile switches based on digital shift and select registers that control analog switches. These dynamically controlled switches span all of the rows and columns of the internal routing, allowing for run-time system modification and scanning I/O. The digital registers can also double as on-chip memory. We introduce a new hybrid floating-gate switch matrix, which includes switches that eliminate previously observed mismatch issues to provide highly precise computation. To highlight the potential of this digitally enhanced analog processor, we demonstrate a dynamically reconfigurable image transformer, an arbitrary waveform generator, and a mixed-signal FIR filter.**

***Index Terms*—Analog signal processing, field-programmable analog array (FPAA), rapid analog prototyping, vector-matrix multiplier (VMM).**

I. Analog Reconfigurability

As monolithic integration of analog and digital circuitry pervades the market, integrated circuit designers are faced with the increasingly difficult task of verifying complex mixed-signal systems. The most common approach for this task is to simulate the analog subsystem, fabricate and test the mixed-signal system, and then repeat [1]. We propose that a faster and more efficient approach is to prototype mixed-signal systems using reconfigurable analog hardware, similar to the common strategy of using FPGAs to prototype digital systems.

In addition to simple prototyping, reconfigurable analog systems are extremely powerful for embedded computing applications, acting as coprocessors. Analog signal processing (ASP) systems can easily utilize subthreshold transistor operation to perform ultra low-power computation, especially in parallel computing systems [2].

We present the RASP 2.9v (Fig. 1), which is the next generation of field-programmable analog array (FPAA). FPAAs are an ideal platform for analog computing, signal processing, and prototyping. The RASP 2.9v includes over 76 000 programmable analog parameters and a varied toolbox of components (e.g., OTAs, FETs, caps, multipliers, and T-gates) to synthesize almost any analog system (complete parameters are in Table I). The RASP 2.9v, unlike previous FPAAs [3],

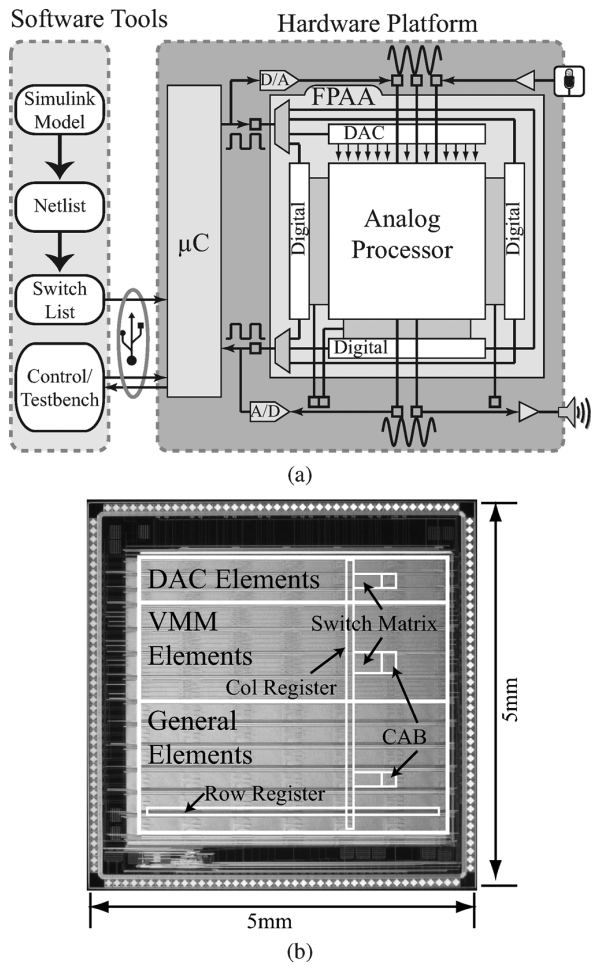


Fig. 1. Architecture and layout of the RASP 2.9v FPAA. (a) The system-level diagram shows the analog core and surrounding digital control and interfacing. The analog processor communicates directly with the microcontroller via an SPI interface. A complete software tool chain is available for analog synthesis in Simulink and connects to the hardware platform with USB. (b) The RASP 2.9v IC was fabricated in 350-nm CMOS and consumes 25 mm^2 of area.

Manuscript received April 14, 2011; revised January 21, 2012; accepted March 11, 2012. This paper was approved by Associate Editor Roland Thewes.

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: cschlott@gatech.edu; samshap@gatech.edu; stephen.nease@gatech.edu; phasler@ece.gatech.edu).

Digital Object Identifier 10.1109/JSSC.2012.2194847

TABLE I
RASP 2.9v SUMMARY OF PARAMETERS

Process	350nm CMOS
V_{DD}	2.4V
Die Size	5mm × 5mm
Number of CABs	18 DAC, 36 Regular, 24 VMM
Programmable parameters	> 76,000
Number of Volatile Switches	4728: 6 × 400-bit (vertical), 14 × 156-bit (horizontal), 6 × 24-bit (DAC)
Chip I/O	79 Analog, 20 Dynamic output lines 18 compilable DAC
Regular CAB Elements	132 OTA, 168 FgOTA, 36 T-gate, 72 nFET, 72 pFET, 36 OTA buffer, 144 500fF Cap
Programming Speed	Volatile Switch: 719ns FG Switch: 31 ±2ms Analog Indirect FG: 38 ±10ms Analog Direct FG: 36 ±8ms

includes a novel volatile switching architecture. This switching architecture allows digital control and dynamic reconfigurability that are important in embedded systems, especially if it is working in conjunction with a digital system. Debugging prototyped systems is quite easy with this architecture, because the registered switches can be used to multiplex internal circuit nodes out to measurement equipment.

While other FPAAs, such as the analog math co-processor in [4], have substantial digital interfacing capabilities, they tend to have far more limited application space. The co-processor is designed for ODE computation, while the hexagonal FPAA in [5] is designed to operate as a single high-dimension Gm-C filter. The higher density and greater variety of components on the RASP 2.9v permit it to reach a much wider application space.

The embedded digital control structures combine with the high-density analog arrays to produce the first dynamically reconfigurable FPAA. The digital enhancements were carefully designed to maximize their usefulness as control and storage devices, while minimizing their footprint on the overall chip. This architecture extends the high computational density of previous RASP, while the digital enhancements enable higher chip utilization, effectively increasing the size of realizable systems. The digital control also provides the ability to compile banks of on-chip data converters which increases the device's usefulness in embedded systems.

Another novel advancement of this chip is the hybrid switch matrix, which is comprised of both directly and indirectly programmed switches. The previous generation of FPAAs utilized an indirect programming scheme, which achieves very low switch resistances. However, the precision of these indirect switches has been plagued by mismatch issues inherent to the indirect scheme. To remedy this problem, we have added direct switch elements to the switch matrix, which do not have the same mismatch issues as indirect devices. Our introduction of the hybrid switch matrix eliminates the burden of characterizing and storing the offset of each switch for every chip. These new features—dynamic control, on-chip DACs, and high-precision switches—uniquely position this chip as the only platform reported that is capable of large-scale embedded digitally enhanced analog processing.

The FPAA hardware platform is also supported by a complete suite of software synthesis tools that allow the user to design systems in Mathworks Simulink and compile that design onto the FPAA [6].

The remainder of this paper proceeds as follows. Section II describes the processing elements of the system and Section III describes the routing architecture. Section IV elaborates on the software tools developed for working with the RASP 2.9 hardware as well as methods for using these tools to automate system testing and calibration. Section V provides results from several example systems that are newly realizable with this platform. These example systems include a dynamically reconfigurable image transformer, an arbitrary waveform generator, and a mixed-signal FIR filter. Section VI contains concluding remarks.

II. PROCESSING ELEMENTS

Fig. 1(a) illustrates the FPAA system-level architecture, with the analog processor at its core. This processor contains thousands of analog components that can be configured and routed to implement many analog signal processing systems. The RASP 2.9v features a novel volatile switching scheme that allows the user to scan thousands of outputs, assert a signal onto any internal node via 20 dedicated I/O pins, and store and retrieve digital values.

The analog processing core is composed of analog primitives which are arranged in computational analog blocks (CABs). The various CABs are shown in Fig. 2. The IC contains 78 CABs: 36 for general purpose analog computation, 18 designed for compiling current-mode digital-to-analog converters (DACs), and 24 optimized for performing vector-matrix multiplication (VMM) operations.

A. General Analog CAB

To accommodate the widest possible application space, the largest chip real estate was given to the general processing CAB. Each general CAB contains four operational transconductance amplifiers (OTAs), four FETs (50/50 split of n/p-type), one transmission-gate switch (T-gate), and four 500-fF capacitors.

Of the four OTAs, three are fully port accessible and one is connected in unity-gain feedback. Two of the port-accessible OTAs have floating-gate input stages with a capacitive divider attenuation of 1:9, which increases the linear input range by a factor of 9 (see Fig. 3). The floating-gate inputs can be programmed to compensate for—or introduce—a fixed offset. The floating gates are programmed by modifying the charge on the gate with hot electron injection and Fowler–Nordheim tunneling. On-chip circuitry can measure the floating gate's state and apply the necessary terminal voltages to modify the charge to the desired level.

The OTAs all utilize a wide-linear-range nine-transistor topology with pFET inputs. A floating-gate pFET transistor sets the tail current of the OTAs, programmable from 100 pA to 10 μA. We can accurately set the transconductance of each OTA, which is proportional to the bias current. Control over the transconductance of the device is useful for programming systems such as analog filters or voltage-controlled current sources. The power consumed by each OTA is calculated as

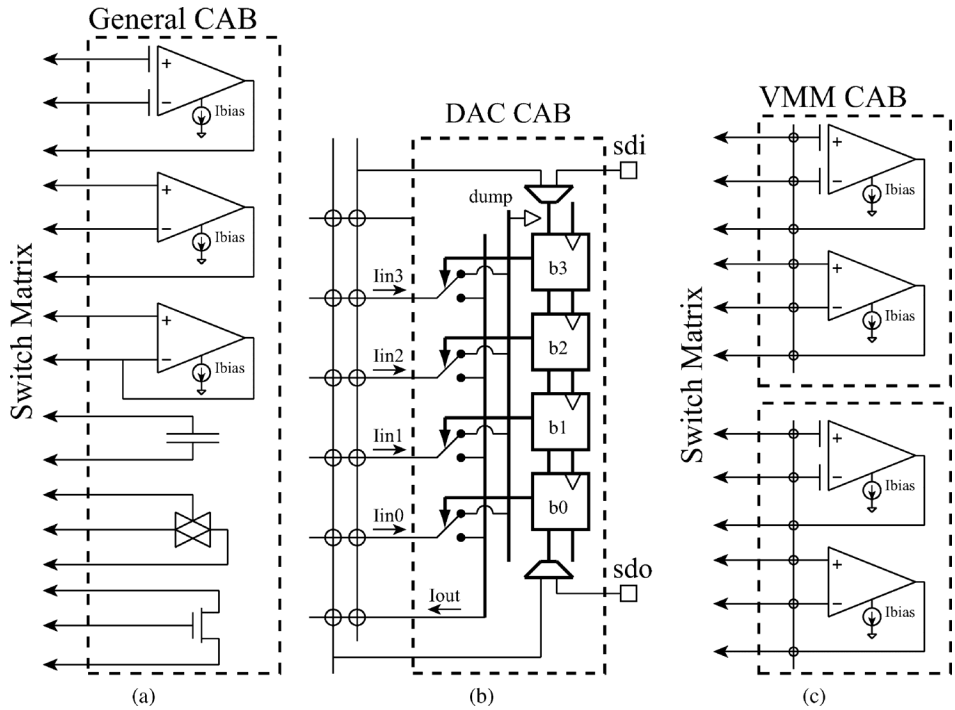


Fig. 2. Structure of the CABs. (a) The general CAB consists of common analog elements: OTAs, MOSFETs, capacitors, and transmission-gates. (b) The DAC CAB contains an 8-b register which toggles a bank of switches, allowing for multiple DAC topologies to be compiled. (c) The VMM CAB contains both regular and floating-gate-input OTAs, which are commonly used as the front-end to the VMM: the fgOTA for V/I conversion, and the OTA for the active feedback of the sense transistor.

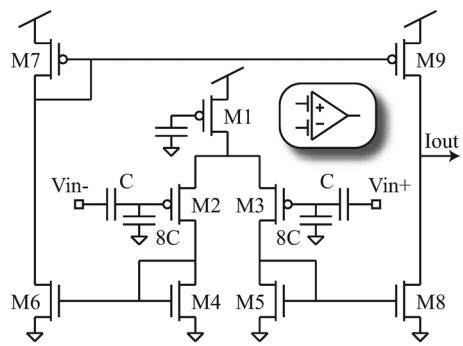


Fig. 3. Both the regular and FG-input OTAs use a nine-transistor structure. The bias current is set with a FG pFET and can be programmed from 100 nA to 30 μA. The FG-input has an attenuation factor of 1:9 on the input stage for wider input liner range (with lower gain). The FG input elements can also be programmed to remove any input offset.

$2I_{\rm bias}V_{\rm DD}$, where the factor of two is a result of the $I_{\rm bias}$ flowing in both branches.

The OTA can also be used as an amplifier with voltage gain. The voltage gain is independent of bias current, so we will choose current values only large enough to drive the amplifier's load. For power analysis of systems with voltage-gain OTAs and high impedance loads, we will choose $I_{\rm bias}$ of 100 nA, which results in 480 nW of static power.

B. DAC CAB

One improvement of the RASP 2.9v over previous FPAAs is the incorporation of dedicated current DAC sections. The DAC CAB is composed of digitally controlled switches connecting to the switch matrix, allowing users to compile binary-weighted current-mode DACs.

Each DAC CAB contains one 8-b register, with three CABs down a column connected in series. Thus, the DAC section could also be configured as six 24-b registers. These registers can be configured either to take the serial-data input from the off-chip source or from the switch matrix. The same configuration setting will determine if the output goes off chip or into the switch matrix. This capability allows the digital registers to double as storage for system data.

C. VMM CAB

VMM is an extremely efficient operation when performed in the analog domain [7], [8]. Because this computation is such a "killer app" for modern analog computation, we took special consideration to facilitate their large-scale design in the RASP 2.9v.

Each of the 24 VMM CABs contains four pairs of OTAs. In each pair, one OTA is for the current-scaling active current mirror, and the other is a floating-gate input OTA for V–I or I–V conversion. Between each pair of devices is a short vertical line to allow repeated connections to the same column address, drastically increasing routing efficiency. These particular floor plan choices increase the dimension of synthesizable matrix multipliers, while all of the OTAs can still be used for any purpose, resulting in no loss of flexibility.

III. Routing and Analog Switches

The RASP 2.9v FPAA is arranged in 13 rows and six columns of CABs. To interconnect the CAB elements to each other, we have incorporated a full crossbar switch matrix (SM).

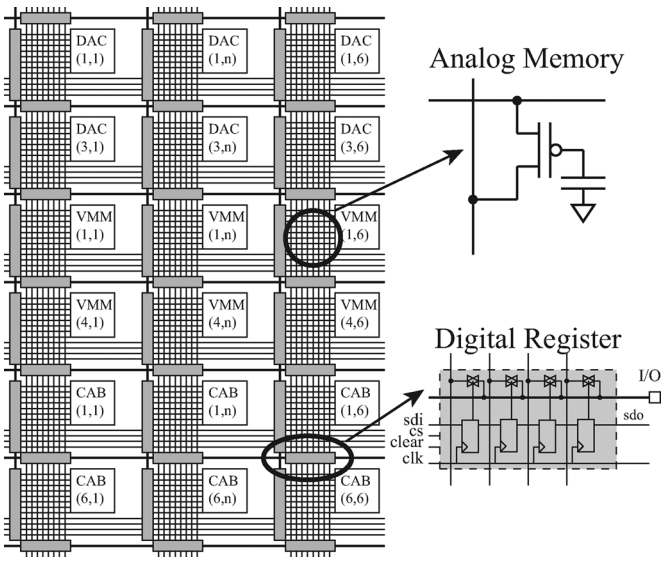


Fig. 4. CABs are arranged in six columns with 13 rows. There are 36 regular CABs, 24 VMM CABs, and 18 DAC CABs. The routing is a full crossbar switch matrix with floating-gate switches intersecting each row and column. This topology allows for great functional density, as each floating gate stores its own memory and acts as either a switch or an analog computation device. The volatile switches are controlled by digital shift registers that span all of the columns (156 b each) and rows (400 b each).

Nonvolatile switches are located at the intersection of each row and column line.

A. Routing

Fig. 4 shows the routing architecture. The crossbar matrix contains a mixture of global, local, and power routing. Each section of SM is composed of three global power lines, 11 vertical global lines, and 14 vertical local lines. The global lines span all of the CAB rows, where the local lines can be connected to each top or bottom neighbor with the bridge switch. The locals can be reconfigured into global lines, at the cost of higher parasitic resistance and capacitance. This combination of two line types allows for greater versatility.

Analysis from a similar (but smaller) FPAA structure extracted a parasitic capacitance of 1.6 pF for global vertical lines, 1.5 pF for global horizontal lines, and 220 fF for vertical local lines [3]. While the local lines have approximately the same length in the RASP 2.9v, the global vertical and global horizontal lines are 63% and 50% longer, respectively. We can use these to extrapolate respective capacitances of 2.6 and 2.3 pF. Good estimates of these capacitances allow designers to take routing into account when designing circuits.

The power lines support a chip-wide global V_{DD}, Gnd, and V_{ref}. The inclusion of a global V_{ref} is novel to this chip and was the direct result of previous FPAA design experience. Many analog signal processing systems use a common midrail voltage that feeds multiple elements. Including a global V_{ref} drastically reduces routing complexity. The V_{ref} line is pinned out where is can be driven off-chip to any voltage, or it can be left open at the pin and driven by an on-chip source.

B. Nonvolatile Switches

The crossbar SM is composed of programmable floating-gate (FG) transistor switched, of which there is a total of 76 000. Each element can be programmed using hot electron injection or Fowler–Nordheim tunneling. The FGs double as reconfigurable switches and nonvolatile memory that store their own conductance. Since the FGs are analog, they can be programmed to intermediate states, allowing their use for dense analog computation.

For general routing situations, we use the indirect switch programming scheme shown in Fig. 5(a). This structure allows us to measure the programmed current in the indirect device (M2)—which shares a gate with the in-circuit device (M1)—while removing selection circuitry (M3) from the signal path, minimizing parasitic resistances. However, the cost of this indirect system is the inherent mismatch between the device that is measured (M2) and the device that is used in the circuit (M1). This effect is not a problem for fully programmed switches, but can cause a loss in precision when the FGs are used for computation. This issue can be compensated by characterizing and storing the offset coefficients of each device.

In addition to the indirect switch, we have added direct switches—shown in Fig. 5(b)—to create a novel hybrid switch matrix. This programming method uses one FG as both the programmed device and the in-circuit device, so the mismatch between the program-time and run-time devices for a particular SM address has been eliminated [see Fig. 5(d)]. This method is an important improvement to the analog processor which relies heavily on the use of FG switch elements for precise computation. The direct device frees us from the cumbersome task of having to map all of the coefficients of the chip. To keep the same form factor of the switch cell, a single pFET (M5) is inserted above the fgFET (M4) for programming isolation. Because the pFET has low conductance at low voltages, the direct scheme makes a poor all-purpose switch. A comparison of the two switches' ON resistance is shown in Fig. 5(e).

An on-chip programmer, based on the design in [9], is used to program all of the nonvolatile switches and other programmable elements. For the direct-programmed switch, the routing column measures drain current in program mode. This approach means that all of the switches down a column must be of the same type: direct or indirect. The global verticals are therefore subdivided into three indirect lines and eight direct lines, and the local verticals are subdivided into six indirect and eight direct. The switches are skewed towards the direct configuration because it is very valuable for precise current sources and multiplier weights.

C. Volatile Switches

The incorporation of volatile switches on the RASP 2.9v marks a vast improvement in digital interfacing compared to earlier FPAAs. The volatile switches are composed of shift registers that control the selection of T-gate switches, referred to here as registered switches. The T-gates can connect routing lines to a common I/O bus. We have inserted registered switches across every CAB row and down every CAB column, for a total of 20 registers. This new tool allows us to probe any given circuit node in run mode.

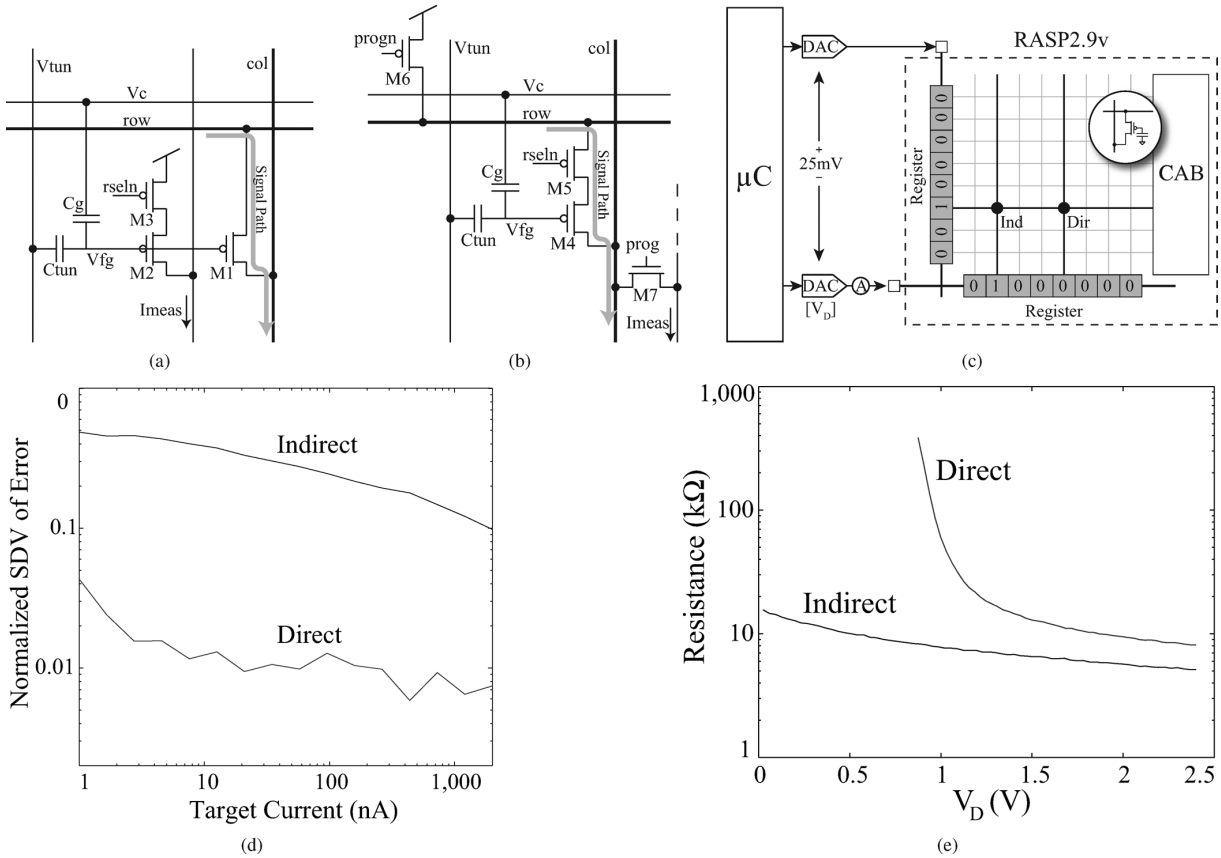


Fig. 5. Routing structure contains two variations of FG switches: indirect and direct. (a) Indirect-programmed FG switch provides a very good pass element since there are no MUXs in the signal path. (b) Direct-programmed FG switch was included for improved precision. However, it is not an optimal all-purpose switch because selection circuitry had to be added to the signal path for programming isolation. (c) Volatile switches can be leveraged to dynamically select which FG switch to read from in a measurement test. (d) Comparison of the two types of FG switches shows that the direct switch has a much lower first-pass programming error. (e) Each switch shows an on resistance of about 10 kΩ, however, the direct switch's resistance rises sharply at low voltages because of the pFET in the signal path.

The registers are loaded serially with the SDI line (serial data in), can be read on a common SDO line (serial data out), and clocked with a dedicated SCLK (serial clock). This SPI protocol lets the FPAA interface with most modern microcontrollers. The shift registers are buffered with a data latch that loads on a global chip select (CS). This data buffer allows us to shift configurations while maintaining the previous switch control. Communication with each register is multiplexed using a 5-b address. All of the registers are on a global clear.

Some of the registers (the ones in the DAC CAB) can be configured to take SDI signals from on-chip sources, as illustrated in Fig. 6. The timing diagram in the figure shows that when the select line is disabled, the register is filled from the default external source (likely the microcontroller). When the select line is enabled, the register is connected to a line in the SM so that it can be loaded with on-chip data. This option is useful when we want to store a digital pulse train that is generated on-chip, for instance when synthesizing a sigma-delta converter.

The registered switches come at the cost of pin count and nonvolatile switch density. The whole structure requires 29 dedicated pins (four SPI, five address, 20 I/O), reducing the general analog I/O to 79 pins (based on our 200 pin QFP). This cost is acceptable because the 20 register I/O greatly expand the effective I/O to serially reach every circuit net when operated as a scanner multiplexer. The other cost is in density; each bit of the register consumes the area of eight FG switches. This approach reduces the available analog routing lines, as well as eight local horizontal outputs per CAB. The great improvement in overall routing versatility by the run-mode volatile switches makes this an acceptable cost as well.

The unit capacitance of the register is simulated to be 50 fF. There are many gates in the flip-flop, so a lumped unit capacitance is extracted from the dynamic power of 3 μW simulated for one bit at 10 MHz. We use this lumped unit capacitance value to calculate the dynamic power of the registers in systems with the equation $P = N_{\text{bits}} C_{\text{unit}} V_{\text{DD}}^2 f$.

IV. Programming Methods

A. Programming Tools

The RASP 2.9 analog signal processor can be fully programmed and tested with the Mathworks MATLAB environment. Furthermore, the general CABs are supported by a Mathworks Simulink design framework. The user begins by creating a block level diagram of the desired circuit in Simulink, using a library of linear and nonlinear elements [see Fig. 7(a)]. Each block has both a signal processing function in Simulink and a corresponding SPICE subcircuit definition. After testing the

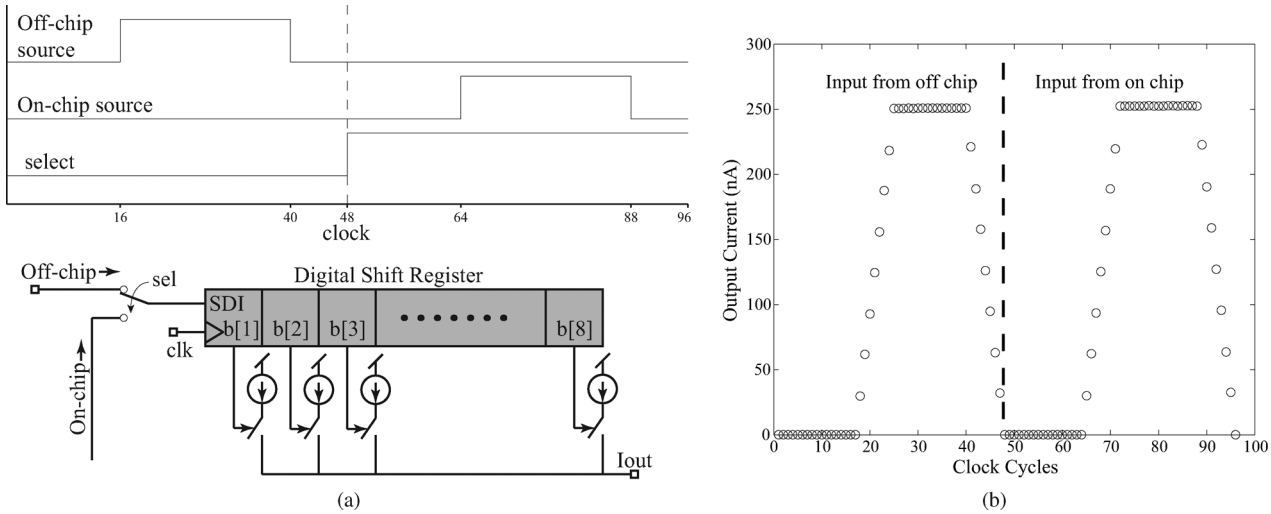


Fig. 6. Serial data for the registers can be loaded from either an off-chip source or the on-chip switch matrix. (a) Test setup for loading the register highlights the switch that selects between on- and off-chip sources. The top graph shows a timing diagram with trains of zeros and ones coming from each the input sources. The schematic diagram on the bottom shows each register bit controlling an equally-weighted current source for easy read out. (b) Output measurement shows identical current readings from both the on- and off-chip register data.

block diagram in Simulink, the MATLAB program "sim2spice" compiles the block diagram into a SPICE netlist of RASP 2.9 components [10]. The SPICE netlist can be viewed independently for debugging or immediately compiled into a switchlist via GRASPER, a C-based place-and-route tool [11]. The switchlist represents bias currents for included CAB elements, analog switch elements, and the digital routing between the programmed elements and the IOs. Compiling the netlist and the switchlist is typically accomplished with a single MATLAB operation.

To aid in debugging, there also exists the FPAA Routing & Analysis Tool (RAT) [6], a GUI that illustrates the topology of the programmed switches and CAB elements. Once a switchlist has been generated, MATLAB can program the circuit on the RASP via an AT91sam7s Microcontroller. Individual switches can be programmed to within 9.5 b of accuracy in less than 50 ms, using methods similar to [9].

Temperature and noise effects are important to any analog designer, however we are presenting a platform for countless system implementations. Each system will have its own non-idealities, of which we cannot provide a global specification. We have, however, provided an analysis on the line capacitance of the routing fabric, which will be useful when calculating the SNR of a given system. Our future work involves using the routing information to accurately model individual signal processing blocks. By incorporating these second-order effects into the Simulink-level blocks, the design engineer will have a better understanding of the overall system behavior.

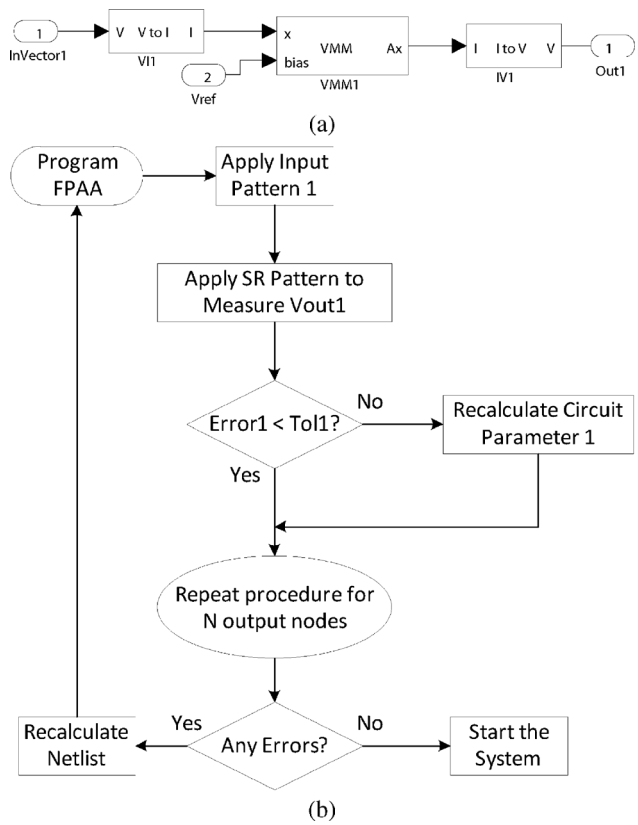


Fig. 7. Tools for creating and testing a circuit on the RASP 2.9. (a) A Simulink implementation of a VMM in the general CABs for linear transformation, with current and voltage converters. Each block represents an ideal function (used in Simulink simulations) and a circuit netlist. (b) Algorithm for testing a hardware circuit designed in Simulink. Each hardware block's output is compared to the ideal expected from the Simulink function and recalibrated accordingly. After every node is tested, the device is reprogrammed and retested until every device meets the desired tolerance.

B. Testing

The registered switches on the RASP 2.9v are designed to rapidly expedite testing. The current DACs, shift registers, and off-chip DACs allow the user to insert a desired current or voltage to any circuit node, and to probe every other node. Fig. 7(b) shows the algorithm for testing a circuit programmed on the FPAA. The shift registers allow us to test each block individually and to quickly debug and calibrate any programmed

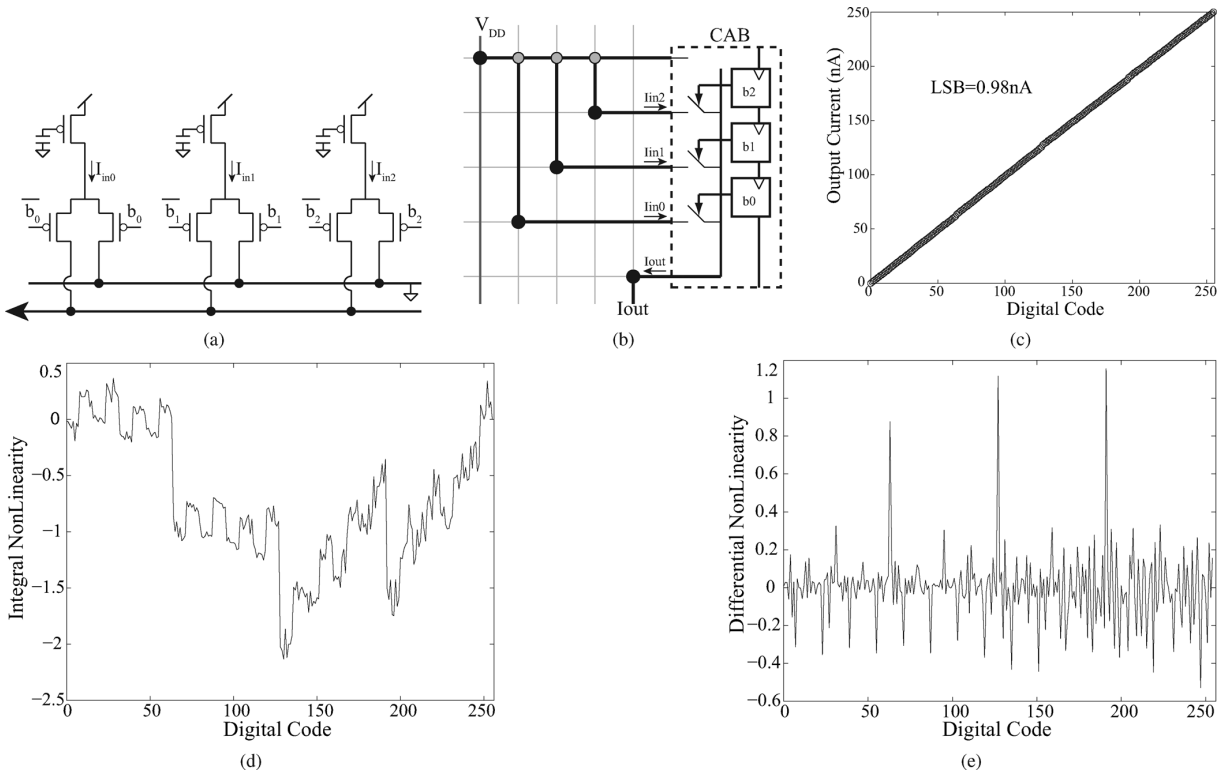


Fig. 8. On-chip reconfigurable DAC. (a) Schematic and (b) FPAA implementation of the FG current-source DAC. (c) Measured results from a compiled 8-b current DAC shows an LSB of 0.98 nA. (d) INL and (e) DNL plots from the 8-b current DAC.

device on the chip. MATLAB scripts can automatically perform calibration by comparing outputs with desired results, modifying the switchlist, and reprogramming the circuit for another test cycle.

V. Results and Applications

The RASP 2.9v adds capabilities for on-chip data conversion and digital enhancement, while maintaining the functionality of the earlier RASP 2.8 chips, which have been used to implement Gm-C filters, AM receiver, and speech processors [3].

Here, we report on the performance of four systems that highlight key improvements of the chip: the current-mode DACs, a large-scale image processor, an arbitrary waveform generator, and an analog architecture for bitwise arithmetic. Each of these example systems would have been difficult or impossible to compile on previous FPAA platforms. We validate the chip by demonstrating the versatility of systems that can be compiled, whereas we achieve comparable performance to systems that were fabricated in custom silicon.

A. Programmable DAC Core

One important use of the chip's digital infrastructure is to compile current-mode DACs onto the chip. This new capability allows users to easily apply inputs to current-mode circuits, using the chip's SPI protocol. In each column of CABs, SPI controls three 8-b DAC CABs connected serially. Taking advantage of the FPAA's reconfigurable nature, we provide the resources to compile DACs rather than include fixed DACs. This flexibility allows us to try various topologies, alter the least significant bit, or use that area for something else if DACs are not needed.

The RASP 2.9v architecture makes it easy to implement binary-weighted current DACs. Fig. 8 shows the schematic and FPAA implementation of a DAC based on individual current sources. The current source implementation has the benefit of ease and flexibility of design; even a nonstandard mapping can be programmed. Another potential topology is a FG-based diffuser tree. The diffuser tree implementation has a more constrained design, but the use of small conductance ratios dramatically reduces temperature dependence.

Fig. 8(c) shows the response of an 8-b FG current source DAC with LSB of 0.98 nA. Currently, the setting time of the DAC is limited by the SPI clock speed of the microcontroller. The system is clocked at 1.39 Mbit/s. This architecture is most efficient when all three DACs in a column are being utilized. Three DACs in a column can be clocked in 17.3 μs, yielding an effective SPI setting time of 5.77 μs/sample. For the 8-b DAC at 1.39 Mb/s, we calculate the dynamic power consumption to be 3.2 μW from $P_{\mathrm{dynamic}} = N_{\mathrm{bits}} C_{\mathrm{unit}} V_{\mathrm{DD}}^2 f$, where C_{unit} is the unit capacitance of the register. The static power is calculated as 614 nW, which is $P_{\mathrm{static}} = \left(2^{N_{\mathrm{bits}}}\mathrm{LSB}\right) V_{\mathrm{DD}}$. Our total power at 173 kS/s is thus 3.8 μW. The maximum INL and DNL are measured to be 2.13 and 1.16 LSB, respectively [shown in Fig. 8(d) and (e)].

The closest DAC architecture comparison from the literature is the FG current-mode DAC in [12]. This DAC is based on binary weighted FG current sources where the FGs are programmed to an LSB of 50 nA. This DAC reported 7-b accuracy with 0.5LSB linearity error, but no power or speed num-

TABLE II
SYSTEM PERFORMANCE SUMMARY

	This DAC	DAC in [13]
Resolution	8 bits	8 bits
INL	2.13 LSB	1.09 LSB
DNL	1.16 LSB	0.80 LSB
LSB	0.980nA	3.75μA
Conversion Rate	173kS/s (SPI speed)	5MS/s
Power	3.8μW	850μW
Number of channels	18	1
	This AWG	**AWG in [15]**
Wave	100nA DC 100nA$_{pp}$	300nA DC 100nA$_{pp}$
Clock	1.92MHz	250kHz
Elements	40	64
Power	2.7μW	not reported
THD	-25.5dB @ 310kHz -29.5dB @ 17.5kHz	not reported not reported
	This FIR Filter	**FIR Filter in [16]**
Size	24 bit	16 tap
Sample speed	40 kHz	50 kHz
Power	12μW	16mW
Filters	LPF	LPF, BPF, comb

bers. The next closest topology is the FG DAC presented in [13]. We cite this DAC in Table II for a more thorough comparison because it is very similar to our own, and the DAC in [12] did not include many specifications. This architecture also uses floating-gate current sources, but it is slightly different in that it uses multiple gates to couple onto the floating node rather than programming it with precise charge. The 8-b DAC in [13] was fabricated in 1.2-μm custom silicon and reports INL and DNL of 1.09 and 0.8 LSB, respectively. They used an LSB of 3.75 μA and achieved 5 MS/s at 850 μW.

B. VMM Applications

Fig. 9(a) illustrates the implementation of a current-mode VMM based on the design in [14]. The VMM is a modified current mirror, which uses the weights of directly programmed switch elements to multiply the input currents and sums the output currents via Kirchhoff's Current Law (KCL). Negative multiplications can be implemented with a differential configuration. Using a constant bias current for inputs allows for consistent speed and power. The VMM blocks in the RASP 2.9v were created to efficiently place and route this architecture, utilizing a large proportion of the routing fabric for computational purposes, as shown in Fig. 9(b).

Fig. 9(c) illustrates the performance of scalar multiplication using the directly programmable devices compared to using the indirect devices. The advantage of the direct FG is clearly shown with one programming pass. The direct FG VMM shows accurate four-quadrant multiplication of 4.5 b, whereas the indirect FG system shows significant gain error as well as large offset error. These errors are traditionally compensated for with multiple programming passes using an adaptive process. However, such an adaptive programming step is not always practical or even possible. By restricting the range around a bias current and calibrating each multiplication, the accuracy of the direct VMM can be increased to 6 b.

One application of the VMM circuit in the RASP 2.9v is as a low-power front-end image processor. Since CMOS imagers produce currents as outputs, current mode VMMs are a natural way of performing linear operations such as edge detection, smoothing, whitening, and discrete cosine transforms.

Fig. 9(d) illustrates an image transform system implemented in the RASP 2.9v. We scan in the image and perform a separable transform with two passes: the first is a convolution with the S1 vector, and the second is a convolution with S2. Data from the two transforms are shown on the right side of the figure: a 3×3 Sobel edge detector and a 9×9 smoothing filter. This system makes use of the on-chip DAC because the test image is being generated by a PC rather than a current-mode imager. This topology highlights a design feature of the compilable DACs in that they can be configured in serial; we only need to shift in one new sample and let the other samples shift to the next DAC.

C. Arbitrary Waveform Generator

The RASP 2.9v is particularly well suited for arbitrary waveform generation (AWG). Fig. 10 illustrates the architecture of an AWG programmed on the FPAA, as well as several waveforms that were generated. The AWG makes optimal use of the switch fabric, as every transistor acts as a memory element, holding the value of the current it will pass to the output channel. As the shift register scans the rows sequentially, the stored currents of the elements in that row flow down to the appropriate channels. We calibrated the scan bits to the number of devices in the signal, so that the first row of devices switched on just as the last row switched off. We were able to control the waveform frequency by changing the scan speed. This structure allows multiple columns to be selected by the row register at the bottom, to select among many stored waveforms. Additionally, since the waveforms are in current mode, the register can be set to select more then one column at a time, resulting in a waveform that is the sum of the two source waveforms.

We used a wide range amplifier in transimpedance configuration as a I-to-V converter in order to generate easily readable voltage outputs, shown in the lower portion of Fig. 10(a). We controlled the amplitude and offset of the output waveform with the amplifier bias and reference voltage respectively. This topology is also beneficial in that it fixes the output voltage of the current sources to the reference voltage. Although direct switches are used on the input-current side of the I-to-V, the output voltage signal must be routed on the indirect-switch lines, so that the output swing is not limited by the high resistance of the direct switches.

The time response [Fig. 10(b)] and frequency response [Fig. 10(c)] are shown for two waveform speeds. As with the on-chip DAC, the clock speed is limited by the SPI line from the microcontroller. Each wave is programmed with 40 elements, with one clocked at 17.5 kHz and the other at 310 kHz. These clock speeds result in waveforms that are 437 Hz (17.5 kHz/40) and 7.7 kHz (310 kHz/40). The number of elements in a waveform can be expanded up to the full length of the vertical register: 400 b. This AWG structure is similar to that reported in [15], with our system being compiled in the reconfigurable hardware and the other being fabricated in custom 0.5-μm silicon. Ours achieves similar speeds, with the previous design reporting maximum clock of 250 kHz, where we have run the SPI clock up to 1.92 MHz. At this maximum speed, we

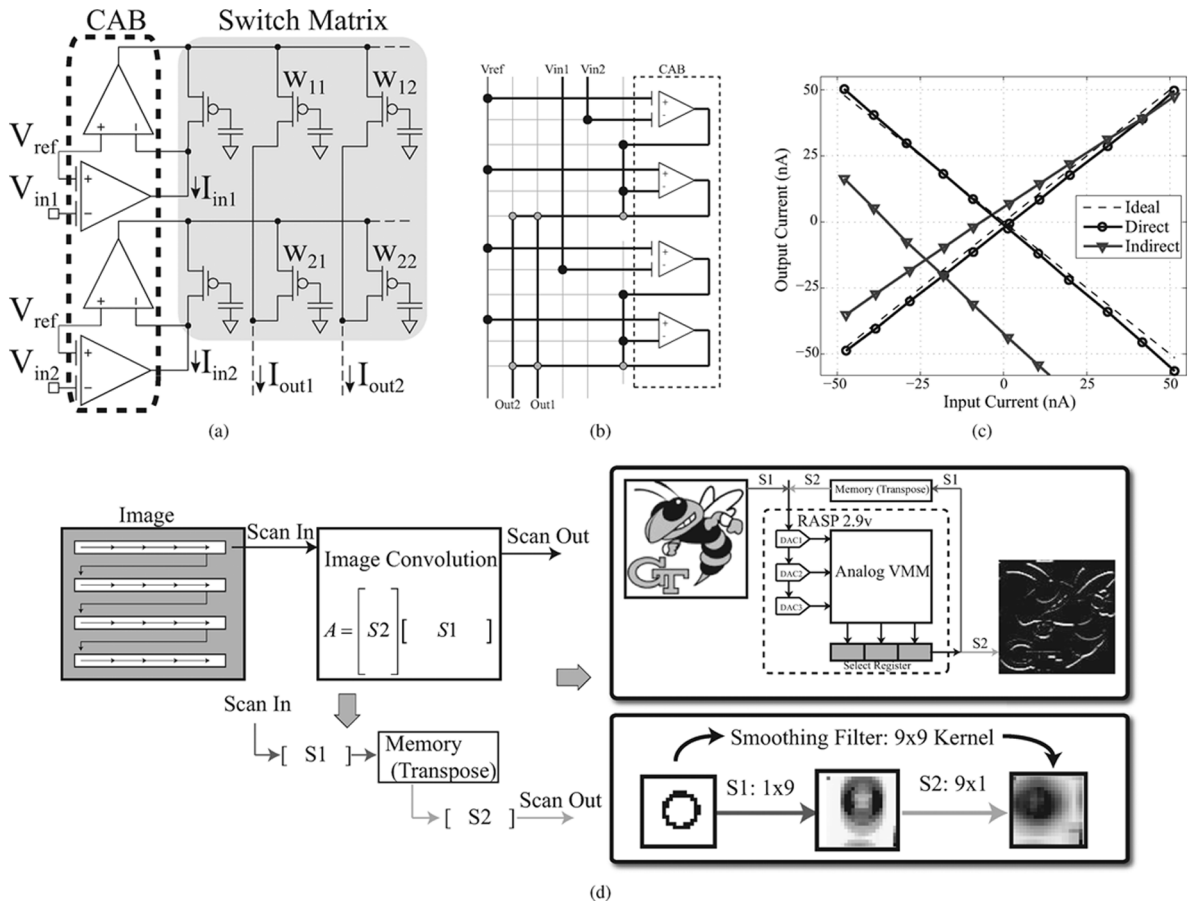


Fig. 9. (a) Schematic and (b) FPAA implementation of a 2 × 2 VMM. (c) Data from a differential 1 × 1 VMM. The directly programmable devices in the VMM cab allow accurate multiplication (to 4.5 b) on the first programming pass, eliminating the calibration needed in earlier RASP designs for linear processing. The one-pass programmed indirect-switch VMM shows signifigant gain and offset error. Calibration can be used, however, to increase the accuracy to 6 b. (d) Application of the VMM in an image processing system. The image processor performs separable transforms scanning in the image and convolving by S1 and S2 in two passes. The kernels chosen for this test are a 3 × 3 Sobel edge detector and a 9 × 9 smoothing filter. The system schematic of the image processor front end shows the on-chip DAC components providing the signals to the VMM.

calculate the dynamic power dissipation to be 1.11 μW. The system operates with a single bit shifting down the registers, so we only need to add up the two bits that are changing. The static power is from a combination of the signal DC current and the I/V stage. The dc current is 100 nA, and the I/V includes two OTAs: the voltage amplifier operating with 100 nA and the feedback Gm stage has a bias of 200 nA to sink the maximum signal current. The static power is 1.6 μW, which results in a total power of 2.7 μW.

D. Mixed-Signal FIR Filter

The ability of the RASP 2.9v to shift through control bits opens opportunities for bit-wise arithmetic. A distributed arithmetic FIR filter is a common and powerful bit-wise operation. An FIR filter has certain advantages over traditional analog filter—such as linear phase—which motivates its inclusion in our analog toolbox.

The RASP 2.9v is particularly well-suited for a current-mode implementation of this operation. The multiplication of bits by weights is implemented by current sources that can be left open or connected to the output, where the addition is achieved simply by KCL.

With our shift register controlling the bitwise activation of the current sources, the full system can be implemented in multiple ways. A classical architecture for a mixed-signal distributed arithmetic FIR filter is presented in [16] and involves a straightforward mapping of the digital blocks to the analog domain.

Fig. 11 shows a novel architecture for the mixed-signal FIR filter. The filter will maintain its linear phase, while having an analog input and output signal. The input stage is a integrate-and-fire sigma-delta converter [17]. The output of this stage is a digital pulse train, shown in Fig. 11(b). The spike rate is linear with input voltage, and can easily be modified by the size of the capacitor when compiled, as shown in Fig. 11(c).

The spike train is sampled by the register and filtered by the weighted current sources. The pulsewidth of the sigma-delta converter can be tuned with V_{bias} to ensure that it matches the sampling rate of the register. Any filter coefficients can be programmed into the current sources, where a differential convention can be used to implement four-quadrant coefficients. Initial results from the low-pass filter are shown in Fig. 11(d). The output current is accurately reconstructing a filtered version of the input signal. The dynamic power consumption for the 24-bit register is calculated to be 7 μW at 1 MHz. The static power is a combination of the current sources and the four OTAs. The

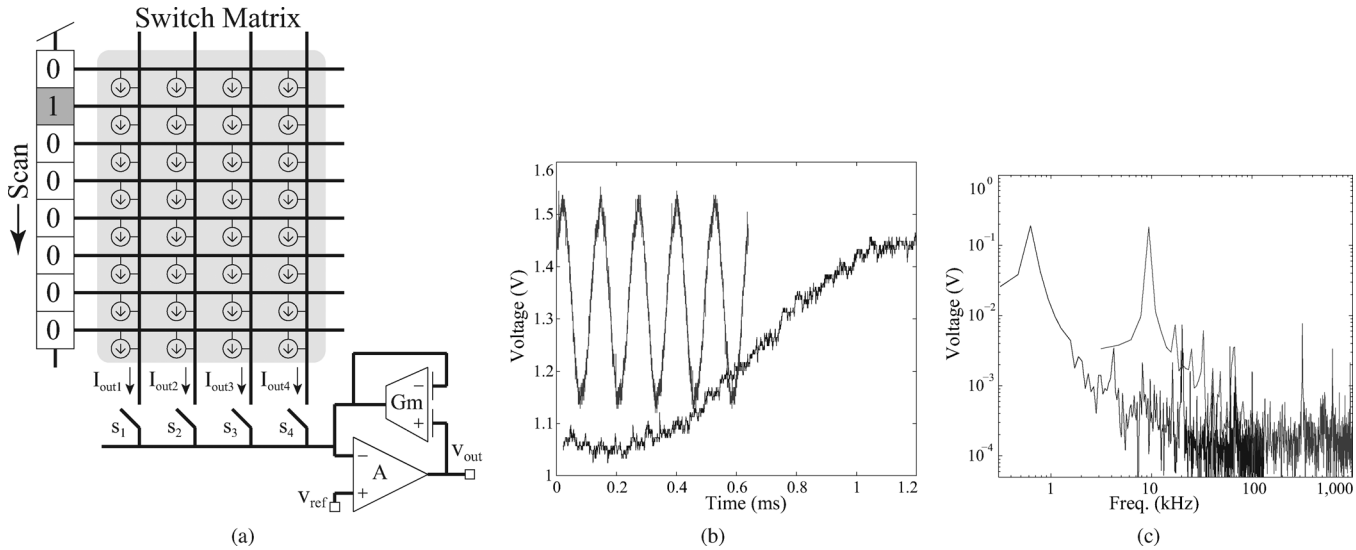


Fig. 10. Applications of the RASP 2.9v. (a) Architecture for a four-channel arbitrary waveform generator. The volatile switches short each row to $V_{\rm DD}$ serially, so each column passes the current supplied by the floating gate element at the intersection with the active row (shown here as empty circles). The volatile switch lines can be used to choose one channel—or combine multiple channels—to be converted to an output voltage. The I-to-V is an OTA with a FGOTA in feedback to provide a tunable transimpedance. Increasing the bias current of the FGOTA decreases its transimpedance. The input floating gates of the FGOTA can be used to program an arbitrary voltage offset for the output. (b) Two sine waves generated by the AWG, using 40 devices scanned at 17.5 kHz and 310 kHz. Note that the FGOTA has been programmed to allow different gain and offsets. (c) FFT of the two sine waves. Total harmonic distortion is -29.5 and -25.5 dB, respectively. Note the small spike at the scan frequencies.

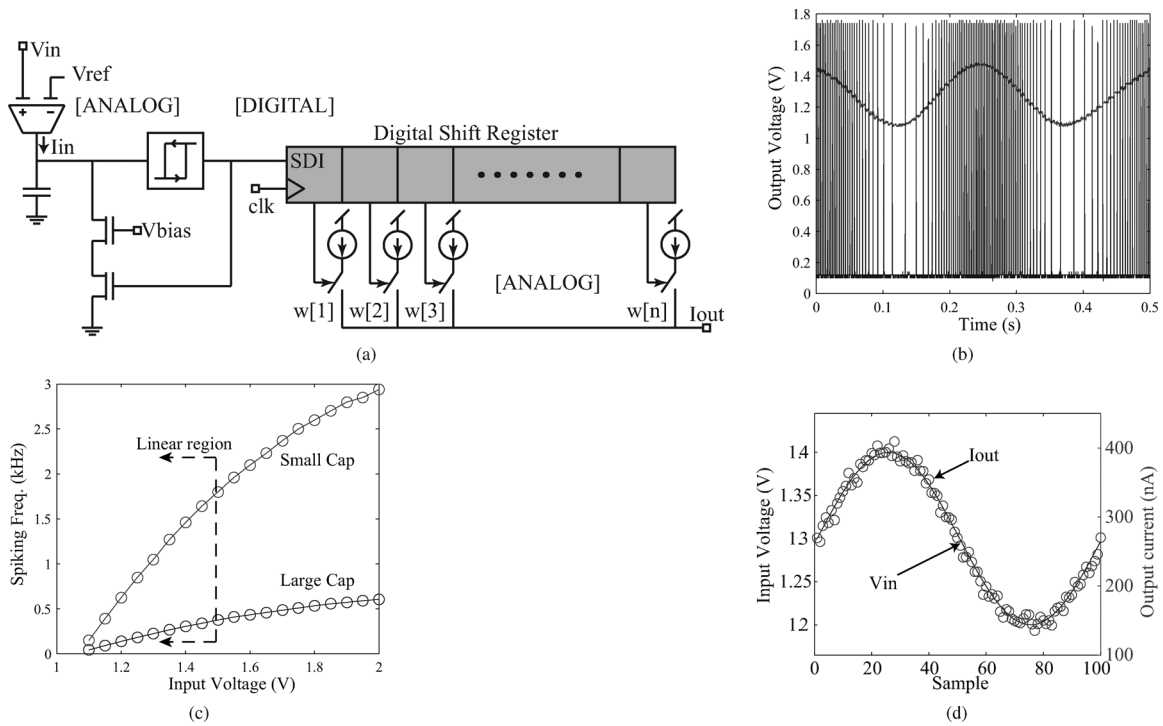


Fig. 11. (a) Proposed architecture for the mixed-signal FIR filter. (b) Integrate-and-fire spike generator produces digital pulses with a frequency based of the input current. (c) Integrating capacitor size can easily be reconfigured to tune the spiking frequency range. (d) Output of the mixed-signal system. The initial results show the output current correctly reconstructing a slow-moving input analog signal.

current sources are each biased at 10 nA, of which we average that half (12) are ON. The two OTAs in the I/V stage are biased at 100 nA for the voltage amplifier and 300 nA for the feedback Gm stage to handle the maximum signal current. The V/I OTA and comparator are biased at 500 and 100 nA, respectively. The static power is 5 μW, which results in a total power of 12 μW. This architecture was invented to take advantage of the RASP 2.9v elements; therefore, the comparisons to other systems must be made based on function rather than exact architecture. Our system falls under the category of floating-gate-based mixed-signal FIR filters. We use [16] in Table II as the closest functional comparison, whereas [18] would also fall into this category of filter.

VI. Conclusion

The RASP 2.9v is designed for mixed-signal computation, with compilable DACs for signal conversion, VMMs for efficient linear operations, generic analog CABs for many nonlinear operations, and digital registers for digital storage and dynamic reconfigurability. We have demonstrated a current-mode DAC, a VMM, an embedded image processor, an AWGs, and a bit-wise FIR filter. These are key building blocks allowing implementation of high impact systems like analog/software-defined radio [19] and low-power FFT processors [20], [21]. A summary of key parameters is provided in Table I, with a summary of system performance in Table II. We continue to refine our automation tools to improve our programming and testing speeds.

References

[1] G. Ying, A. Kuehlmann, K. Kundert, G. Gielen, E. Grimme, M. O'Leary, S. Tare, and W. Wong, "Guess, solder, measure, repeat – How do I get my mixed-signal chip right?," in *Proc. ACM/IEEE Design Autom. Conf.*, 2009, pp. 520–521.

[2] C. Mead, *Analog VLSI and Neural Systems*. Reading, MA: Addison-Wesley, 1989.

[3] A. Basu, S. Brink, C. Schlottmann, S. Ramakrishnan, C. Petre, S. Koziol, F. Baskaya, C. Twigg, and P. Hasler, "A floating-gate-based field-programmable analog array," *IEEE J. Solid-State Circuits*, vol. 45, no. 9, pp. 1781–1794, Sep. 2010.

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

[5] J. Becker, F. Henrici, S. Trendelenburg, M. Ortmanns, and Y. Manoli, "A field-programmable analog array of 55 digitally tunable OTAs in a hexagonal lattice," *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2759–2768, Dec. 2008.

[6] S. Koziol, C. Schlottmann, A. Basu, S. Brink, C. Petre, B. Degnan, S. Ramakrishnan, P. Hasler, and A. Balavoine, "Hardware and software infrastructure for a family of floating-gate based FPAAs," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 2794–2797.

[7] S. Chakrabartty and G. Cauwenberghs, "Sub-microwatt analog VLSI trainable pattern classifier," *IEEE J. Solid-State Circuits*, vol. 42, no. 5, pp. 1169–1179, May 2007.

[8] A. Bandyopadhyay, J. Lee, R. Robucci, and P. Hasler, "MATIA: A programmable 80 μW/frame CMOS block matrix transform imager architecture," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 663–672, Mar. 2006.

[9] A. Basu and P. Hasler, "A fully integrated architecture for fast and accurate programming of floating gates over six decades of current," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 45, no. 9, pp. 1781–1794, Sep. 2010.

[10] C. Schlottmann, C. Petre, and P. Hasler, "A high-level simulink-based tool for FPAA configuration," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 1, pp. 10–18, Jan. 2012.

[11] F. Baskaya, S. Reddy, S. Lim, and D. Anderson, "Placement for large-scale floating-gate field-programable analog arrays," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 8, pp. 906–910, Aug. 2006.

[12] G. Serrano and P. Hasler, "A floating-gate DAC array," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2004, pp. 357–360.

[13] L. Wong, C. Kwok, and G. Rigby, "A 1-V CMOS D/A converter with multi-input floating-gate MOSFET," *IEEE J. Solid-State Circuits*, vol. 34, no. 10, pp. 1386–1390, Oct. 1999.

[14] C. Schlottmann and P. Hasler, "A highly dense, low power, programmable analog vector-matrix multiplier: The FPAA implementation," *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 1, no. 3, pp. 403–411, Sep. 2011.

[15] R. Chawla, C. Twigg, and P. Hasler, "An analog modulator/demodulator using a programmable arbitrary waveform generator," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2005, vol. 6, pp. 6106–6109.

[16] E. Ozalevli, W. Huang, P. Hasler, and D. Anderson, "A reconfigurable mixed-signal VLSI implementation of distributed arithmetic used for finite-impulse response filtering," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 2, pp. 510–521, Feb. 2008.

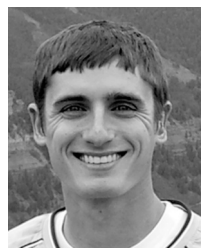
[17] D. Wei, V. Garg, and J. Harris, "An asynchronous delta-sigma converter implementation," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 4903–4906.

[18] W. Figueroa, D. Hsu, and C. Diorio, "A mixed-signal approach to high-performance low-power linear filters," *IEEE J. Solid-State Circuits*, vol. 36, no. 5, pp. 816–822, May 2001.

[19] M. Kitsunezuka, S. Hori, and T. Maeda, "A widely-tunable, reconfigurable CMOS analog baseband IC for software-defined radio," *IEEE J. Solid-State Circuits*, vol. 44, no. 9, pp. 2496–2502, Sep. 2009.

[20] A. Wang and A. Chandrakasan, "A 180-mV subthreshold FFT processor using a minimum energy design methodology," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, Jan. 2005.

[21] S. Suh, A. Basu, C. Schlottmann, P. Hasler, and J. Barry, "Low-power discrete Fourier transform for OFDM: A programmable analog approach," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 2, pp. 290–298, Feb. 2011.



Craig R. Schlottmann (S'07) received the B.S. degree from the University of Florida, Gainesville, in 2007, and the M.S. degree from the Georgia Institute of Technology, Atlanta, in 2009, where he is currently working toward the Ph.D. degree, all in electrical engineering.

During his graduate studies, he interned with MIT's Lincoln Laboratory in the field of embedded and high performance computing. His research interests include low-power analog signal processing, mixed-signal IC design, low-power embedded electronics, and multiple-input translinear element (MITE) systems.

Mr. Schlottmann was the recipient of the Best Live Demo Award at ISCAS 2010.

Samuel Shapero (S'08) received the B.S. and M.S. degrees in electrical engineering from Stanford University, Stanford, CA, in 2005 and 2006, respectively. He is currently working toward the Ph.D. degree in bioengineering at the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta.

He is a Georgia Tech Presidential Fellow and an NSF IGERT Fellow. As a Senior Design Engineer with Micron Technology, he received patents for his work in NAND flash memory design. His current research focuses on low-power analog circuit design, nonvolatile programmable circuits, neural modeling, and sparse coding in both software and VLSI systems.

Mr. Shapero was named a U.S. Presidential Scholar in 2001.

Stephen Nease received the B.S. degree in electrical engineering from Rose-Hulman Institute of Technology, Terre Haute, IN, in 2009, and received M.S. degree from the Georgia Institute of Technology, Atlanta, in 2011, where he is currently working toward the Ph.D. degree in electrical engineering.

His interests include neuromorphic circuits and systems, analog circuit design, and reconfigurable mixed-signal CMOS systems.

Paul Hasler (SM'04) received the B.S.E. and M.S. degrees in electrical engineering from Arizona State University in 1991, and the Ph.D. degree in computation and nueral systems from the California Institute of Technology, Pasadena, in 1997.

He is currently a Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta. His 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 National Science Foundation CAREER Award in 2001, and the Office of Naval Research YIP award in 2002. He received the

Paul Raphorst Best Paper Award, IEEE Electron Devices Society, 1997, CICC Best Student Paper Award, 2006, ISCAS Best Sensors Paper award, 2005, and the Best Paper Award at SCI 2001.