Received 2 February 2020; revised 19 April 2020; accepted 28 April 2020.
Date of publication 4 May 2020; date of current version 6 July 2020.

Digital Object Identifier 10.1109/JXCDC.2020.2992228

A Fully Integrated Reprogrammable CMOS-RRAM Compute-in-Memory Coprocessor for Neuromorphic Applications

JUSTIN M. CORRELL [1] (Member, IEEE), VISHISHTHA BOTHRA [2], FUXI CAI [3] (Member, IEEE), YONG LIM [4], SEUNG HWAN LEE [5], SEUNGJONG LEE [1] (Member, IEEE), WEI D. LU [1] (Fellow, IEEE), ZHENGYA ZHANG [1] (Senior Member, IEEE), and MICHAEL P. FLYNN [1] (Fellow, IEEE)

[1]Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA
[2]Apple, Cupertino, CA 95014 USA
[3]Applied Materials, Santa Clara, CA 95050 USA
[4]Samsung Electronics, Yongin 17113, South Korea
[5]Intel, Albuquerque, NM 87124 USA

CORRESPONDING AUTHOR: J. M. CORRELL (correllj@umich.edu)

This work was supported by DARPA Unconventional Processing of Signal for Intelligent Data Exploitation (UPSIDE).

ABSTRACT Analog compute-in-memory with resistive random access memory (RRAM) devices promises to overcome the data movement bottleneck in data-intensive artificial intelligence (AI) and machine learning. RRAM crossbar arrays improve the efficiency of vector-matrix multiplications (VMMs), which is a vital operation in these applications. The prototype IC is the first complete, fully integrated analog-RRAM CMOS coprocessor. This article focuses on the digital and analog circuitry that supports efficient and flexible RRAM-based computation. A passive 54 $\times$ 108 RRAM crossbar array performs VMM in the analog domain. Specialized mixed-signal circuits stimulate and read the outputs of the RRAM crossbar. The single-chip CMOS prototype includes a reduced instruction set computer (RISC) processor interfaced to a memory-mapped mixed-signal core. In the mixed-signal core, ADCs and DACs interface with the passive RRAM crossbar. The RISC processor controls the mixed-signal circuits and the algorithm data path. The system is fully programmable and supports forward and backward propagation. As proof of concept, a fully integrated 0.18- μm CMOS prototype with a postprocessed RRAM array demonstrates several key functions of machine learning, including online learning. The mixed-signal core consumes 64 mW at an operating frequency of 148 MHz. The total system power consumption considering the mixed-signal circuitry, the digital processor, and the passive RRAM array is 307 mW. The maximum theoretical throughput is 2.6 GOPS at an efficiency of 8.5 GOPS/W.

INDEX TERMS ADC, analog, compute-in-memory, DAC, resistive random access memory (RRAM), vector-matrix multiplication (VMM).

I. INTRODUCTION

THE energy consumption of data movement tasks behind artificial intelligence (AI) and machine learning presents a significant bottleneck in system performance. Recent work on near- and in-memory computing promises to improve the efficiency of vector-matrix multiplication (VMM) by reducing or removing the data movement barrier. Full digital compute-in-memory solutions using SRAM have been demonstrated (see [1], [2]) but suffer from large leakage currents. Mixed-signal solutions using passive and active resistive random access memory (RRAM) crossbar arrays have been demonstrated (see [3]–[5]) but require discrete or external peripheral devices for operation. In [5], 3-bit multiplication is realized by combining multiple 1-b RRAM cells, but the compute-in-memory macro still requires an external field-programmable gate array (FPGA) interface for the data path and control. This article describes the first complete single-chip RRAM coprocessor with single-cell multibit RRAM. This article focuses on the CMOS circuitry that supports RRAM computation and complements [6], which introduces the entire system but focuses on the devices and algorithms.

An RRAM crossbar, built with an RRAM device at the intersection of every row and column, is well suited to perform VMM. The device conductance values of the crossbar array store the weight values of the matrix. Voltage inputs are

applied to the crossbar rows to perform analog VMM, and the resulting column currents are measured. The column currents flowing into virtual grounds are the vector product of the row voltages and RRAM conductances. The RRAM bitcell conductances are preprogrammed to represent the weights of a neural network or a part of a neural network. An advantage is that this analog approach allows us to exploit physics to execute direct computing of data-intensive processing without continuously accessing the weights. Compute is in-memory and in parallel and, in a single step [6].

Challenges to practical analog crossbar operation include the nonidealities of the RRAM as well as the extensive mixed-signal circuitry required to support the crossbar operation. The highly nonlinear voltage-to-current relationship of RRAM means that the voltages cannot directly serve as the input vector. Our solution is to represent the input vector as a pulse modulated signal. We integrate the column output currents and perform VMM in the charge domain. Parallel, individual pulse-domain DACs apply the "read" input to the rows. At each column, a two-stage regulator presents a high-quality virtual ground. High-resolution (13-bit) charge-domain ADCs integrate the column currents. The RRAM bitcells are programmed using dedicated write DACs. For maximum throughput, the array should operate in parallel; therefore, each of the rows and columns of the crossbar has dedicated hardware—ADC, read DAC, and write DAC. Furthermore, to enable a wide range of algorithms, our architecture supports transverse operation, where the input is applied to the columns, and the current is read from the rows, terminated to virtual grounds.

Our system incorporates a reduced instruction set computer (RISC) processor for maximum flexibility and programmability of crossbar operations. Compiled C code directs the operation of the crossbar. It is vital to include the RRAM crossbar array and supporting peripheral circuitry on the same silicon die to optimize system performance and minimize latency. The prototype IC incorporates a 54 × 108 RRAM crossbar, mixed-signal interface, and RISC processor. Section I introduces the RRAM coprocessor with the integrated crossbar, discusses design considerations, and reviews system operating modes. Section II details the mixed-signal circuits used to perform analog VMM. Section III focuses on the processor used to control the mixed-signal core and for algorithm implementation. Section IV shows the fabricated prototype. Section V discusses the measurement setup and algorithm implementation on the prototype IC. Finally, Section VI is the conclusion.

II. SYSTEM ARCHITECTURE

A. RRAM COPROCESSOR

Our self-contained fully programmable coprocessor flexibly maps neuromorphic algorithms onto an integrated 54 × 108 RRAM crossbar. The RRAM crossbar performs the VMM operations and is postprocessed directly onto the CMOS die. A mixed-signal core with specialized DACs and ADCs applies the input vector to excite the crossbar and digitizes the output vector. The prototype coprocessor, shown in Fig. 1, includes a RISC CPU for control and configuration as well as complete mixed-signal circuitry for writing to the RRAM bitcells and VMM operation. Compute is in place and in parallel, thus maximizing VMM throughput. Parallel

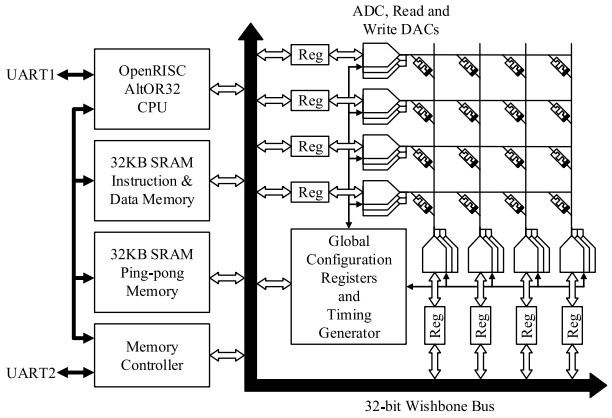

FIGURE 1. RRAM coprocessor.

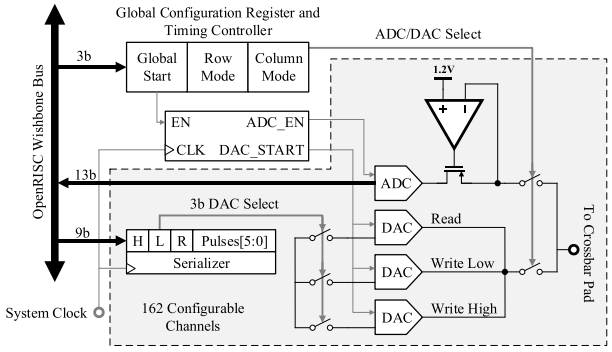

FIGURE 2. Mixed-signal interface.

operation is supported by dedicated mixed-signal peripheral hardware for each row and column of the crossbar.

Dedicated hardware for each row and column supports read and write operations. Write hardware programs the RRAM bitcells to the appropriate weight value. The read hardware performs read–verify during write operations and performs the VMM operations. During programming, separate row and column DACs apply timed high-voltage pulses to the selected bitcell. After each write step, the updated RRAM bitcell conductance is read-verified using a row read DAC and a column ADC. Once fully programmed, to perform a VMM, read DACs in each row (column) apply input pulses to excite the RRAM crossbar and column (row) ADCs integrate and digitize the charge collected at each column. With a full set of mixed-signal circuitry at each row and column, the system supports the forward pass (inner product) and backward pass (transpose inner product) operations.

For easy operation and flexibility, the entire mixed-signal interface (see Fig. 2), comprised of DACs, ADCs, global timing controller, and configuration registers, is memory-mapped to a custom on-chip RISC CPU. Processor instructions written in standard C code configure the mixed-signal core for different modes and implement RRAM-crossbar-mapped algorithms. The complete coprocessor facilitates convolutional or fully connected network layers. It handles different activation functions and learning rules and supports both spiking and nonspiking networks.

B. RRAM CROSSBAR

The 54 × 108 passive crossbar array is comprised of WO_x-based RRAM bitcells without selector devices.

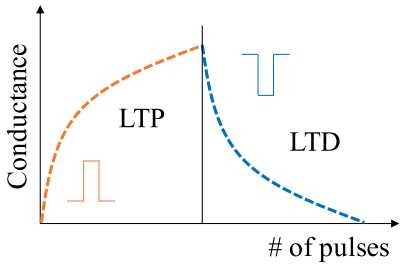

FIGURE 3. RRAM long-term potentiation (LTP) and long-term depression (LTD) curves.

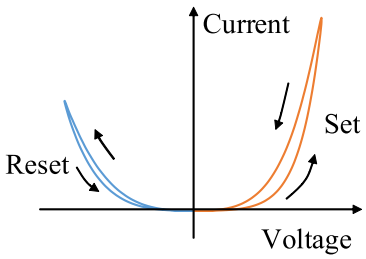

FIGURE 4. RRAM *I–V* characteristics.

An RRAM bitcell is a two-terminal device whose resistance is modulated by the history of external stimulation. RRAM resistance is both nonvolatile and reversible. The bitcell is programmed by applying high-voltage positive and negative pulse trains to the opposite sides of the device. Fig. 3 shows conceptually how the device conductance changes with the number of positive or negative pulses. Once the bitcell is programmed, a low-voltage pulse train reads the state of the device without perturbing the conductance.

The crossbar structure is formed by fabricating an RRAM bitcell at the intersection of every row and column of the crossbar array. A passive array, without selector devices, increases the compute density and reduces circuit design complexity but suffers from sneak-path currents and channel crosstalk [7]. Although selector devices, such as those found in 1T1R crossbar arrays [8]–[10], help mitigate these issues, they increase the bitcell area and require considerable extra circuitry and address decoding.

We address the sneak-path and crosstalk at the bitcell and system levels. In the bitcell, we use devices that exhibit highly nonlinear I–V characteristics to minimize parasitic leakage currents in neighboring cells. Fig. 4 shows the typical I–V characteristics of a WO_x-based RRAM bitcell. At the system level, during each operating mode, we use protective (write) or common-mode (read) voltage schemes, as described in Section III-C.

C. DESIGN CONSIDERATIONS

Practical WO_x-based RRAM devices are not suitable for voltage-to-current multiplication because they are highly nonlinear. Doubling the voltage across an RRAM bitcell does not lead to a doubling of the current. In some cases, increasing the device voltage could reprogram the bitcell. Time-based operation with PWM DACs is an alternative, but a conventional PWM DAC introduces significant nonlinearity due to the nonzero rise times and fall times. Finally, the column ADC must present a robust virtual ground to the column and

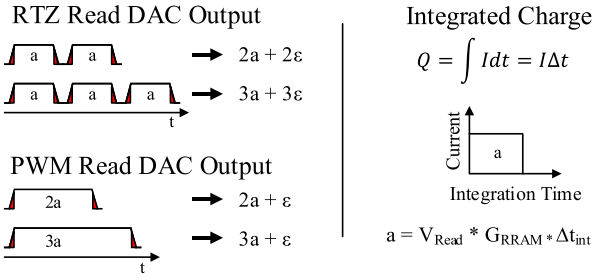

FIGURE 5. Charge-domain operation with pulse-mode RTZ DACs.

accurately integrate and digitize the column charge.

To address the accuracy and linearity challenges, we operate the crossbar in the time and charge domains instead of in the voltage and current domains. The pulse-mode read DACs deliver fixed-amplitude, return-to-zero (RTZ) pulse trains where the number of pulses is equal to the digital input code. An advantage of this approach is that the nonlinearity associated with the rise time and fall time is proportional to the input and appears as a gain error that can be canceled in software. Pulsed charge-domain operation also relaxes the bandwidth requirement of the virtual ground regulator. Fig. 5 shows the example RTZ pulse trains with proportional gain errors. The voltage amplitude is selected based on the RRAM dynamics. The fixed duration pulses modulate with the crossbar array producing "charge packets" that flow into the virtual ground of a new hybrid 13-b charge-integrating ADC.

We choose a relatively high ADC resolution (13 bits) to accommodate a range of RRAM parameters and to facilitate a wide variety of VMM applications. High accuracy is also essential for verifying the write of a single bitcell. We decide on the nominal resolution by considering the extremes of operation. The highest integrated charge is for the largest input vector and the highest set of bitcell conductances. With a 6-bit input, 54 rows, and 4-bit RRAM resolution, the theoretical dynamic range is around 16 bits. In practical applications, the output does not exceed 1/8 of the theoretical maximum so that 13 bits of resolution is sufficient. A 13-bit resolution is also sufficient to measure a single bitcell conductance to more than 5-bit accuracy during write verify. In summary, 13-bit ADC resolution is a good tradeoff for both online learning and inference tasks, given the size of the crossbar.

D. OPERATING MODES

The coprocessor supports five different RRAM crossbar operating modes—these modes cover write and read operations. The write operations include positive write, negative write, and read–verify and are used to program and verify RRAM bitcells. Forward read and backward read operations compute the inner product and the transpose inner product. A combination of settings in a global configuration register and discrete settings in the row and column hardware determine each mode (see Table 1). A global timing generator provides the mode timing.

For the positive and negative write operations, two sets of 6-bit row and column write DACs are used in conjunction to increase or decrease the conductance state of a selected RRAM bitcell. After each write step, the bitcell conductance is read using a 6-bit row read DAC and a 13-bit column ADC. Only the row and column addresses unique to the selected

TABLE 1. Global configuration register settings and crossbar voltage levels.

Mode	Row Mode	Column Mode	Row Voltage	Column Voltage	RRAM Voltage
Positive Write	1 = DAC	1 = DAC	1.9V	0.1V	1.8V
Negative Write	1 = DAC	1 = DAC	0.1V	1.9V	-1.8V
Read-Verify	1 = DAC	0 = ADC	1.8V	1.2V virt. gnd	0.6V
Forward Read	1 = DAC	0 = ADC	1.8V	1.2V virt. gnd	0.6V
Backward Read	0 = ADC	1 = DAC	1.2V virt. gnd	1.8V	0.6V

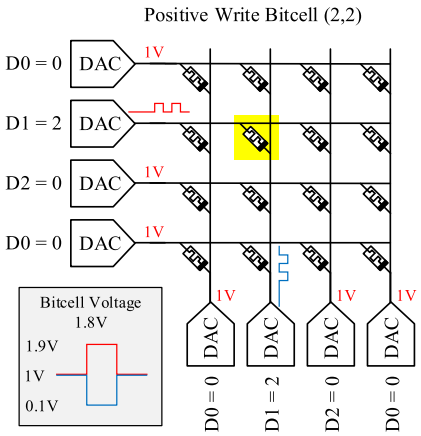

FIGURE 6. Positive write mode bitcell write for row 2, column 2.

device are loaded with the input pulse-train data. All other rows and columns are set to a zero-input code so that DACs of the unselected rows and columns output a protective half voltage to mitigate sneak-path currents.

We now consider the write process in more detail, beginning with a write to increase the conductance of a bitcell. The RISC processor loads the registers for one high-voltage row write DAC and one low-voltage column write DAC with the pulse-train input code equal to the number of desired pulses. The selected row high-voltage write DAC pulses the crossbar row between the 1-V protective half voltage and the 1.9-V write high voltage. At the same time, the selected column low-voltage write DAC pulses the crossbar column between the 1-V protective voltage and the 0.1-V write low voltage. The unselected row and column DACs provide the 1-V protective voltage for the duration of the pulse train. In this way, the selected device is pulsed with a 1.8-V pulse train. The same operation is performed but in the reverse direction to decrease the conductance of a bitcell, resulting in a −1.8-V voltage pulse train. The resultant 1.8-V voltage difference, either positive or negative, across the selected device is sufficient to potentiate the conductance of the bitcell, as shown in Fig. 3. As an example, Fig. 6 shows the crossbar configuration for forward write of the bitcell in row 2, column 2.

For the read–verify step, the processor reconfigures the peripheral hardware to read out the conductance of the selected device. The row and column write DACs are tristated from the crossbar, and the row read DACs and column ADCs connected. The processor clears all of the read DAC input

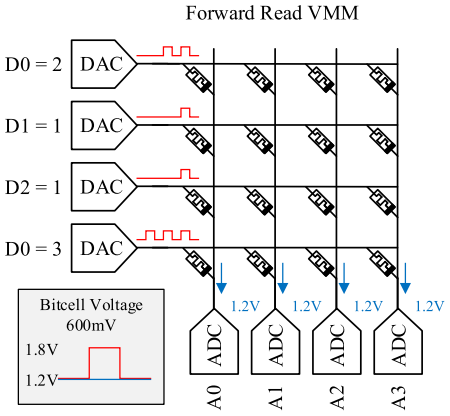

FIGURE 7. Forward read mode for VMM.

registers and loads only the address unique to the selected device with the input pulse train. The columns are connected to ADCs which present a strong 1.2-V virtual ground. With a zero-input read DAC code, all of the unselected read DACs also output the 1.2-V common-mode voltage. The addressed read DAC pulses the crossbar row between the 1.2-V common-mode voltage and the 1.8-V read voltage. This operation pulses the device with a 600-mV pulse train, and the resulting column current flows into the 1.2-V virtual ground of the ADC and is digitized. The processor converts the ADC code into a conductance value so that the write step described earlier can be verified. The write and verify operations repeat until the RRAM bitcell is programmed to the desired weight value.

During read operations, the entire array operates in parallel for matrix multiplications. In the forward-read direction, the row read DACs are loaded with the input vector data and connected to the crossbar. The column ADCs provide a 1.2-V common-mode virtual ground to the columns of the crossbar. The read DACs generate 1.2–1.8-V pulse trains in number equal to the DAC input codes from the input vector. The rows of the crossbar are excited with 600-mV pulses, and the ADCs digitize each of the resulting integrated column currents forming the VMM output vector. For backward read, the process reverses, and row ADCs provide a 1.2-V virtual ground to the rows, whereas column DACs apply pulse trains to the columns. Fig. 7 shows the forward read crossbar configuration for VMM.

III. MIXED-SIGNAL ARCHITECTURE

A. MIXED-SIGNAL CONTROLLER

The mixed-signal controller includes the global configuration register and a global timing controller that configure the array and drive the timing of the peripheral hardware. During operation, the processor sequentially loads all of the memory-mapped mixed-signal registers and then writes a global-start bit to trigger the mixed-signal timing. During the mixed-signal operation, the compute occurs in parallel in the mixed-signal time domain, while the processor waits (NOP) a deterministic amount of clock cycles until finished. The control is handed back to the processor and sequentially proceeds to the next step.

The global configuration register (see Fig. 2) is a one-hot, 3-bit register that controls the write/read mode of all the rows

TABLE 2. Write and read DAC output voltages.

Mode		RTZ DAC Pulse Low and High Voltage	
Programming	Write High	1V	1.9V
	Write Low	1V	0.1V
	Read	1.2V	1.8V
VMM	Forward Read	1.2V	1.8V
	Backward Read	1.2V	1.8V

and columns and the start command for the global timing controller. The two lower bits set the row mode (bit 1) and column mode (bit 0). These are global control signals that select whether the DACs (1) or ADCs (0) are connected to the rows and columns of the crossbar, as shown in Fig. 2. For example, in the forward-read mode, all 54 rows connect to read DACs, and all 108 columns connect to ADCs. In this case, the row-mode bit is set to 1, and the column-mode bit is set to 0. In the forward-write mode, the processor sets both the column- and row-mode bits to 1, connecting DACs to all the rows and columns. As explained in the following section, the individual DAC type (i.e., read, write high, and write low) is set locally as part of the DAC input code. The DAC output voltages are summarized in Table 2.

The global timing generator is a state machine triggered by the MSB of the global configuration register. The period of the mixed-signal timing is equal to the period of the system clock multiplied by the number of clock cycles needed for the 6-bit RTZ DAC operation. The timing generator outputs global control signals, DAC_START and ADC_EN, for the DACs and ADCs and a global stop signal to signal the end of the mixed-signal timing period. The global control signals are active during both write and read operations, but the ADC_EN signal is ignored during write operations. The ADC_EN signal starts and ends one clock cycle before and after the DAC_START signal to power cycle for the ADC.

During write operations, an asserted DAC_START signal enables the selected row and column DACs to output their respective pulse trains. During read operations, ADC_EN wakes up and turns off the power-hungry integrator of the ADC. In a single VMM operation, first, the ADC_EN signal wakes up the ADC, and then, the DAC_START signal both starts the read DAC pulse train and signals the beginning of the ADC integration period. The ADC integrates until DAC_START is deasserted, and then, the collected charge is integrated. The ADC is powered down when ADC_EN is deasserted.

B. PULSE-DOMAIN WRITE AND READ DACs

We address the need for a linear time-domain DAC with a two-level RTZ pulse-mode scheme. A 9-bit input register controls the DACs for each mixed-signal channel. The 9-bit code comprises three configuration bits and the 6-bit DAC value (see Fig. 2). The three one-hot configuration bits enables either the read (R), write high (H), or write low (L) DAC while tristating the unused DACs. The 6-bit DAC input code is serialized into a 1-bit pulse train where the number of pulses is equal to the value of the input code. The serialized pulse train drives logic that controls two switches for each of the DACs, connected to three off-chip voltage references. The output of the DAC is connected to the crossbar row (or column) and drives the voltage between two off-chip references (determined by the control bits) under the control of the DAC logic. The six CMOS switches are sized at 20 μm width to minimize the voltage drop from the DAC references.

C. ACTIVE VIRTUAL GROUND AND INTEGRATING CHARGE DOMAIN ADC

An active virtual ground terminates each crossbar output column (or row) and provides a current signal to a dedicated charge-integrating ADC. A high-quality virtual ground is essential so that the row (column) input voltage and the RRAM transconductance determine the current through each RRAM device. As we operate the crossbar in the charge domain to avoid nonlinearity in the DAC operation, the ADC passively integrates the current delivered through the virtual ground. The ADC digitizes the total charge integrated during the conversion period.

Our approach tackles the challenges of high ADC accuracy and small die area. We implement a programmable current attenuation and a high ADC resolution of 13 bits to support a wide range of algorithms and RRAM structures. The active virtual ground incorporates programmable current attenuation so that we can accommodate different RRAM configurations. A novel hybrid ADC architecture is efficient and dramatically reduces the size of the integration capacitor. The 13-bit ADC is a cascade of a 5-bit first-stage first-order incremental ADC and a 9-bit second-stage successive approximation register (SAR) ADC for high resolution. Since directly integrating the current from crossbar requires large capacitance, instead, the first-stage ADC integrates the charge multiple times with a smaller capacitor and, at the same time, performs coarse digitization. When the integration period ends, the second-stage fine SAR sub-ADC digitizes the residual charge and produces the 13-bit ADC output code.

D. ACTIVE VIRTUAL-GROUND WITH VARIABLE GAIN CONTROL

Fig. 8 shows a simplified schematic of the active virtual ground circuit. Since the current from the RRAM crossbar is too large to integrate directly, we divide the current with a programmable attenuation ranging from 1/64 to 8/64. The current divider splits the current from crossbar into two paths, a dump path (63/64–56/64), and signal path (1/64–8/64). Only the signal path current is integrated and digitized by the ADC.

We introduce a two-stage regulation structure for accurate current division and high output-current linearity. The first stage is a regulated common-gate amplifier that presents a 1.2-V virtual ground voltage to the crossbar. The second-stage maintains the drain voltage of the first-stage regulator transistor at 1.0 V. Only a subset of the second-stage regulated common-gate transistors connect to the ADC—the remainder connects to the dump output. Variable current gain is easily achieved by changing the current ratio using switches. The two-stage regulator provides much higher output impedance compared with a single-stage regulator. This higher impedance improves the linearity of the current

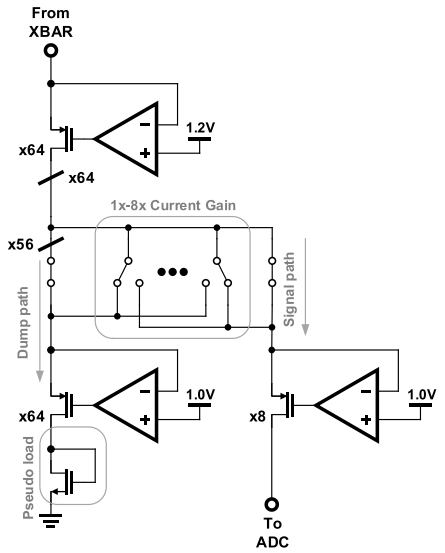

FIGURE 8. **Two-stage, active virtual ground with gain control. Adjusting the signal path and dump path ratio sets the ADC current. Current gain 1×–8× corresponds to the signal path (1/64–8/64) and dump path (63/64–56/64).**

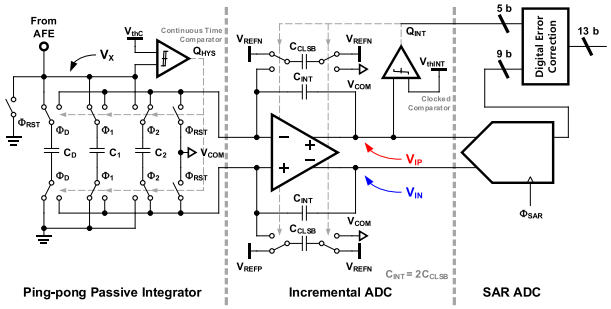

FIGURE 9. **Schematic of the hybrid charge integrating ADC.**

supplied to the ADC input. To ensure accurate current division, we load the output of the second-stage dump regulator with a diode-connected pseudo load. Due to cascaded regulation and the pseudo load, the simulated current division error is less than 0.4%, and the full-range current gain variation is less than 0.1%. Each regulation amplifier is an nMOS-input folded-cascode amplifier with an open-loop dc gain of 58 dB, a gain bandwidth product (GBW) of 16 MHz, and a current consumption of 45 μA.

E. CHARGE DOMAIN 13-B HYBRID ADC

The new 13-bit hybrid integrating ADC combines a 5-bit first-stage incremental ADC with a 9-bit second-stage SAR ADC (see Fig. 9). The first stage is a first-order incremental ADC that performs coarse digitization while alternatively passively integrating the input current on two capacitors, C_1 and C_2. For high resolution, it is critical that the input charge is collected and transferred to incremental ADC without leakage. The passive ping-pong integrator consists of a continuous-time hysteresis comparator and three capacitors, C_D, C_1, and C_2. When the integration-enable signal goes high, the reset switches in Fig. 9 open, and capacitors C_1 and C_D begin to accumulate charge. The continuous-time hysteresis comparator senses when V_X exceeds a threshold, and when this occurs, C_1 disconnects from the input and

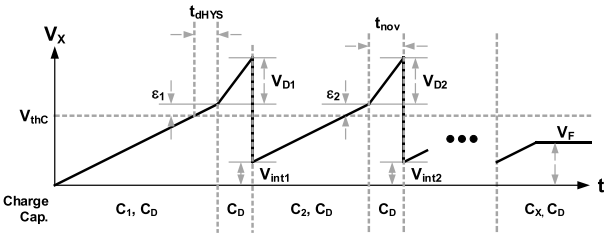

FIGURE 10. **Detailed capacitor waveform for passive integrator.**

connects to the active integrator of the incremental ADC. To prevent glitching and other nonideal switching effects, we incorporate hysteresis in the continuous comparator decision. After C_1 disconnects, C_2 connects to the current input and begins accumulating charge. Nonoverlapped clocking for the C_1 and C_2 switches is essential to prevent charge leakage. The extra capacitor C_D continues accumulating charge, whereas C_1 and C_2 are disconnected during the nonoverlap period, preventing V_X from going too high. When the clocked comparator determines that the output of the active integrator exceeds the threshold voltage V_{thINT}, then the precharged capacitor C_{CLSB} connects and subtracts charge in the next integration phase. Since C_{CLSB} is half the size of C_{INT} and precharge voltage is V_{REFP}–V_{REFN}, the subtracted voltage V_{CLSB} is (V_{REFP}–V_{REFN})/2. The coarse digital code is the number of subtraction occurrences.

When the period integration ends, the fine charging capacitor and C_D connect to the active integrator and integrate the charge. Then, the integrator connects to the second-stage SAR ADC, which starts fine conversion. The conversion range of the SAR ADC is twice the nominal output range of the active integrator, $2V_{\text{CLSB}}$, to provide redundancy. This redundancy accommodates comparator offset as well as any excess output voltage caused by the final residue integration. Considering this one bit of redundancy, the overall resolution of the two-stage ADC is 13 bits.

Lossless charge transfer from the passive integrator is vital for high-resolution ADC operation. The continuous-time comparator is relatively slow, and therefore, during the comparator decision, excessive charge integrates on C_1 (or C_2) and C_D. Furthermore, the input voltage strongly influences the comparator speed. Our approach accurately transfers charge, despite inaccuracy in the comparator decision. Fig. 10 shows how V_X progresses during the integration phase. The total charge Q_{tot} after N cycles of passive integration is

$$\begin{aligned} Q_{\text{tot}} = {} & (C_1 + C_D)(V_{\text{thC}} + \varepsilon_1) + C_D V_{D1} + (C_2 + C_D) \\ & \times (V_{\text{thC}} + \varepsilon_2 - V_{\text{int.1}}) + C_D V_{D2} \\ & + \ldots + (C_X + C_D)(V_F - V_{\text{int.}N-1}) \end{aligned}$$

where ε is the voltage increase during comparator delay, V_D is voltage increase during the nonoverlap phase, C_X is the connected capacitor at the final phase, V_F is the final voltage when the integration signal goes low, and C_{int} is initial voltage once a capacitor is connected. During the nonoverlap phase, only C_D is connected to V_X. After C_2 is connected, charge redistribution occurs and the initial voltage for the next phase is

$$V_{\text{int.1}} = \frac{C_D\,(V_{\text{ref1}} + \varepsilon_1 + V_{D1})}{C_2 + C_D}.$$

Substituting for V_{int}, the total charge is

$$Q_{\text{tot}} = C_1(V_{thC} + \varepsilon_1) + C_2(V_{thC} + \varepsilon_2) + \ldots + (C_X + C_D)V_F.$$

The equation shows that the total charge is transferred without any loss.

The active integrator uses a three-stage fully differential ring amplifier [11] with an auxiliary first-stage autozero (similar to [12]) to improve energy efficiency. The clocked comparator is a double-tail latched comparator [13]. The SAR ADC employs bottom-plate sampling for accuracy and asynchronous logic for speed. When the integration signal goes low, the SAR ADC samples the output of the active integrator output and begins the SAR algorithm. The overall ADC code is the sum of the first- and second-stage codes. The ADC consumes 1.3 mW at 19 MS/s and occupies 0.066 mm^2 (74 μm $\times$ 890 μm).

IV. DIGITAL ARCHITECTURE

A custom OpenRISC processor with 64 kB of dual-port SRAM supports and controls the mixed-signal core. The registers for the DACs, ADCs, and global configuration are memory mapped to the OpenRISC address space via the shared 32-bit Wishbone bus. The memory-mapped registers are accessible in a standard C code application. Since the core mainly performs register-level manipulations, a stripped-down version, the Alternative OpenRISC 1000 or AltOR32, was chosen to save area and power. The AltOR32 removes the floating-point unit, hardware multiplier, and pipeline delay slot instructions and hardware from the implementation. We developed a custom Verilog module that implements the mixed-signal core as a peripheral slave to the 32-bit Wishbone bus. This module places all of the mixed-signal registers in the address space of the processor. The module is treated just like any other OpenRISC peripheral.

The on-chip SRAM is divided into two blocks—32 kB for both processor instruction and data memory and 32 kB of "ping-pong" memory. The ping-pong memory can be used for loading input data from off-chip during runtime for large-scale neural network applications. All 64 kB of SRAM is dual-port and is mapped to both the OpenRISC 32-bit Wishbone bus and a custom memory controller external to the processor. The memory controller implements a backdoor UART port that provides access to all of the SRAM.

The processor instructions, written in the standard C code, are compiled and run on-chip to control the mixed-signal core. A typical C program implements the sequential instructions to set up the global configuration registers and peripheral hardware and then starts the global timing controller to compute a VMM. The processor waits (NOP) the necessary time until the state machine is finished. The program then reads out the ADC results into memory. In [6], we demonstrated the mapping of complete algorithms on-chip.

The following pseudocode writes the bitcell in row 2, column 2 with two write pulses

```
// Write bitcell(2,2) with pulse = 2 row = 2;
col = 2;
pulses = 2;
row_mode = 1;                  // Select row DACs
col_mode = 1;                  // Select col DACs
for(int i=0;i<162;i++){
   DAC[i] = 0;                 // Clear all DACs
}
DAC[row-1] = pulses;
DAC[col-1+54] = pulses;

//Start global timing generator
start();
//Read-verify bitcell
value = read_verify(row,col).
```

V. PROTOTYPE

A. RRAM CROSSBAR

The W/WO_x/Pd RRAM crossbar array was patterned using e-beam lithography, etching [for the W bottom electrode (BE)] and liftoff [for the Pd top electrode (TE)] processes. The WO_x switching layer was created through rapid thermal annealing of the exposed W BE surface with oxygen gas at 425 °C for 60 s. A SiO_2 spacer structure was formed by plasma-enhanced chemical vapor deposition (PECVD) followed by reactive-ion etching (RIE) back before WO_x growth and TE deposition to allow for better step coverage of the TE at the crosspoints. Details on CMOS integration can be found in [6].

The WO_x devices do not require a forming step—a key requirement for passive crossbar arrays. Without selector devices, the high-voltage forming process can damage other already formed cells in the array. Switching behavior in WO_x devices is categorized as bulk-type. Conductance modulation is proportional to the effective conductive area change [14] instead of the opening or closing of the depletion gap found in filamentary (TaO_x, HfO_2) RRAM devices. Bulk-type switching leads to forming-free behavior, but device retention is limited. For the prototype devices, the device retention is limited to minutes but is sufficient to demonstrate the algorithms in [6]. More details on the WO_x devices and their switching behavior can be found in [14] and [15].

B. INTEGRATED COPROCESSOR

The prototype IC is fabricated in 180-nm CMOS and occupies 62 mm^2. This includes 162 ADCs, 486 DACs, the OpenRISC processor, 64 kB of SRAM, and the integrated RRAM crossbar fabricated on the surface of the die. A single channel is comprised of three DACs and one ADC and occupies 0.13 mm^2. Fig. 11 shows the die photograph with expanded channel layout. The system performance is determined from the power consumption and throughput of the entire system at the 148 MHz maximum operating frequency (Table 3).

VI. MEASUREMENT RESULTS

A. MEASUREMENT SETUP

The prototype RRAM coprocessor is wirebonded to a 391-pin PGA package for testing and measurement. A custom printed circuit board provides analog and digital supplies as well as the off-chip biasing for the mixed-signal peripheral hardware. The instructions for the coprocessor are written in standard C and compiled into binary machine code on a host PC using the or1knd toolchain [16]. The C program contains the high-level instructions for algorithm implementation as well as the low-level instructions supporting the on-chip hardware configuration settings.

Access to the coprocessor dual-port SRAM is provided by an on-chip "back-door" memory controller and the UART port of the OpenRISC. A custom C bootloader is compiled and loaded into the SRAM using the memory controller and

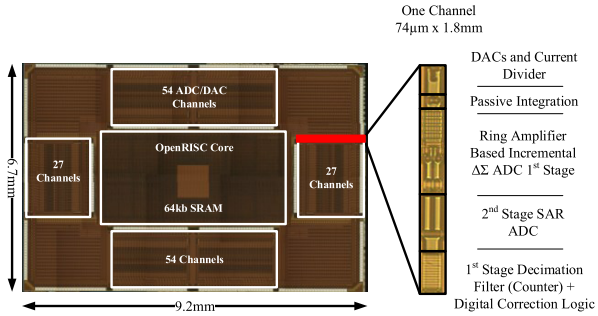

FIGURE 11. **Die photograph of the 180-nm prototype.**

TABLE 3. **System summary.**

Crossbar	RRAM Crossbar	Passive 1R
	RRAM Bitcell	WO_x
	Array Dimension	54x108
	Minimum Resistance (kOhm)	300
	Maximum Resistance (kOhm)	600
	Levels	16
	Variation	4.2%
	Power (mW)	7
Mixed-Signal Interface	# of ADCs	162
	ADC Resolution (bits)	13
	# of DACs	486
	DAC Resolution (bits)	6
	Power (mW)	64
CPU	Power (mW)	235
System Performance	System Clock (MHz)	148
	Input (bits)	6
	VMM/S	448 K
	OP/S	2.6 G
	OPS/W	8.5 G
Mixed-Signal Efficiency (6 bit inputs)	Energy/VMM (nJ)	144
	Energy/Op (pJ)	25

Opal Kelly XEM7001 FPGA. The bootloader loads neuromorphic applications directly into the SRAM instruction memory address space from a PC through a USB-to-UART translator to the UART port peripheral of the OpenRISC processor.

B. CHARGE-DOMAIN MULTIPLICATION

The pulse-mode DAC and charge-domain ADC linearity are tested by replacing the RRAM connection between DAC and ADC with a selection of discrete resistors and applying 6-b input pulse trains. The integrated charge per step is the multiplication of the input value and the conductance—or one "op" in a VMM. Fig. 12 shows the integrated charge versus DAC input code for five different input currents. The integrated charge error is the difference between the expected charge and the charge measured by the ADC. Since the input is applied as RTZ pulses, any nonidealities or errors show up as a gain error, as shown in Fig. 12.

C. SYSTEM PERFORMANCE

The system performance is determined from the power consumption and throughput of the entire system at the 148-MHz maximum operating frequency. The mixed-signal

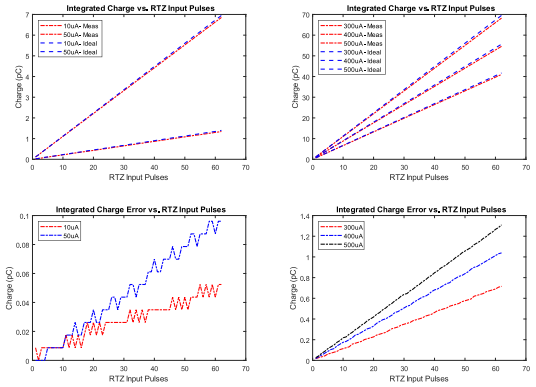

FIGURE 12. **Integrated charge (top row) and error (bottom row) for 10, 50, 300, 400, and 500 μA—6-b input pulse trains measured at the 48-MHz system clock frequency.**

core, including the ADCs and DACs supporting the 54 × 108 crossbar, consumes 64.4 mW. The mixed-signal energy efficiency is 144 nJ/VMM or 25 pJ/Op. The OpenRISC core and passive crossbar array consume 235.3 and 7 mW, respectively. The total system power is 307 mW, and the maximum theoretical throughput is 2.6 GOPS at 8.5 GOPS/W.

VII. CONCLUSION

Analog RRAM promises very efficient VMM but requires extensive circuit support. This article presents the mixed-signal circuitry for efficient and flexible analog VMM and describes the first fully integrated single-chip analog RRAM system. We introduce specialized mixed-signal circuits to stimulate and read the RRAM crossbar efficiently. Charge-domain operation with pulse-mode DACs ensures high linearity. Two-stage regulation provides a robust virtual ground and programmable attenuation. Hybrid two-stage ADCs integrate and digitize crossbar output current. A RISC processor interfaced with a memory-mapped mixed-signal core controls the crossbar operation. The integrated, reconfigurable coprocessor implements unsupervised and supervised online learning, feature extraction, and classification in a multilayer network.

ACKNOWLEDGMENT

The authors thank Daniel Hammerstein, Kerry Bernstein, and Andreas Olofsson.

REFERENCES

[1] J. Zhang, Z. Wang, and N. Verma, "A machine-learning classifier implemented in a standard 6T SRAM array," in *Proc. IEEE Symp. VLSI Circuits (VLSI-Circuits)*, Honolulu, HI, USA, Jun. 2016, pp. 1–2.

[2] M. Kang, S. K. Gonugondla, A. Patil, and N. R. Shanbhag, "A multi-functional in-memory inference processor using a standard 6T SRAM array," *IEEE J. Solid-State Circuits*, vol. 53, no. 2, pp. 642–655, Feb. 2018.

[3] P. M. Sheridan, F. Cai, C. Du, W. Ma, Z. Zhang, and W. D. Lu, "Sparse coding with memristor networks," *Nature Nanotechnol.*, vol. 12, no. 8, pp. 784–789, Aug. 2017.

[4] C. Li *et al.*, "Analogue signal and image processing with large memristor crossbars," *Nature Electron.*, vol. 1, no. 1, pp. 52–59, Jan. 2018.

[5] C.-X. Xue *et al.*, "A 1Mb multibit ReRAM computing-in-memory macro with 14.6ns parallel MAC computing time for CNN based AI edge processors," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2019, pp. 388–390.

[6] F. Cai *et al.*, "A fully integrated reprogrammable memristor–CMOS system for efficient multiply–accumulate operations," *Nature Electron.*, vol. 2, no. 7, pp. 290–299, Jul. 2019.

[7] M. A. Zidan, H. A. H. Fahmy, M. M. Hussain, and K. N. Salama, "Memristor-based memory: The sneak paths problem and solutions," *Microelectron. J.*, vol. 44, no. 2, pp. 176–183, Feb. 2013.
[8] S. Kim, J. Zhou, and W. D. Lu, "Crossbar RRAM arrays: Selector device requirements during write operation," *IEEE Trans. Electron Devices*, vol. 61, no. 8, pp. 2820–2826, Aug. 2014.
[9] E. J. Merced-Grafals, N. Dávila, N. Ge, R. S. Williams, and J. P. Strachan, "Repeatable, accurate, and high speed multi-level programming of memristor 1T1R arrays for power efficient analog computing applications," *Nanotechnology*, vol. 27, no. 36, Sep. 2016, Art. no. 365202.
[10] W.-H. Chen *et al.*, "A 65 nm 1 Mb nonvolatile computing-in-memory ReRAM macro with sub-16ns multiply-and-accumulate for binary DNN AI edge processors," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2018, pp. 494–496.
[11] Y. Lim and M. P. Flynn, "A 1 mW 71.5 dB SNDR 50 MS/s 13 bit fully differential ring amplifier base SAR-assisted pipeline ADC," *IEEE J. Solid-State Circuits*, vol. 50, no. 12, pp. 2901–2911, Sep. 2015.
[12] Y. Lim and M. P. Flynn, "A calibration-free 2.3 mW 73.2 dB SNDR 15b 100 MS/s four-stage fully differential ring amplifier based SAR-assisted pipeline ADC," in *Proc. Symp. VLSI Circuits*, Kyoto, Japan, Jun. 2017, pp. 98–99.
[13] M. Miyahara, Y. Asada, D. Paik, and A. Matsuzawa, "A low-noise self-calibrating dynamic comparator for high-speed ADCs," in *Proc. IEEE Asian Solid-State Circuits Conf.*, Fukuoka, Japan, Nov. 2008, pp. 269–272.
[14] T. Chang, S.-H. Jo, K.-H. Kim, P. Sheridan, S. Gaba, and W. Lu, "Synaptic behaviors and modeling of a metal oxide memristive device," *Appl. Phys. A, Solids Surf.*, vol. 102, no. 4, pp. 857–863, Mar. 2011.
[15] T. Chang, S.-H. Jo, and W. Lu, "Short-term memory to long-term memory transition in a nanoscale memristor," *ACS Nano*, vol. 5, no. 9, pp. 7669–7676, Sep. 2011.
[16] *Overview: AltOR32—Alternative Lightweight openRISC CPU: openCores*. Accessed: Jan. 2016. [Online]. Available: https://opencores.org/projects/altor32

JUSTIN M. CORRELL (Member, IEEE) received the B.S. degree in electrical engineering from the University of Florida, Gainesville, FL, USA, in 2015, and the M.S. degree in electrical and computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2018, where he is currently pursuing the Ph.D. degree.

He interned in the High-Speed Data Converter Group, Analog Devices, Greensboro, NC, USA, in 2015. His current research interests include mixed-signal neuromorphic systems and high-speed ADCs.

Mr. Correll was awarded the Rackham Graduate School Summer Research Opportunity Program (SROP) from the University of Michigan in 2014.

VISHISHTHA BOTHRA , photograph and biography not available at the time of publication.

FUXI CAI (Member, IEEE) received the B.S. degree in electronic and communications engineering from the University of Hong Kong, Hong Kong, in 2013, and the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2015 and 2019, respectively.

From 2013 to 2019, he was a Research Assistant with the Nanoelectronics Group, University of Michigan, under Prof. Lu. He is currently a Physicist/Scientist with Applied Materials Inc., Santa Clara, CA, USA. He has published several highly cited articles in multiple influential journals. His research interests include resistive switching memory (ReRAM), memristor crossbar arrays and neuromorphic applications, and in-memory computing.

Dr. Cai was the Vice Chair of Technical Activities (EDS) of the IEEE Southeastern Michigan Chapter IV from 2017 to 2018.

YONG LIM received the B.S. and M.S. degrees in electrical engineering from Yonsei University, Seoul, South Korea, in 2004 and 2006, respectively, and the Ph.D. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2017.

He has been with Samsung Electronics since 2006, where he developed CMOS image sensor readout circuits until 2012. Since 2017, he has been developing energy-efficient data converters for communication systems.

Seung Hwan Lee was born in Seoul, South Korea, in 1982. He received the B.S. and M.S. degrees in physics and applied physics from Osaka University, Osaka, Japan, in 2006 and 2008, respectively, and the Ph.D. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2020.

From 2008 to 2015, he was with the Non-Volatile Memory Technology Development Group, SK Hynix, Icheon, South Korea. He is currently a Failure Analysis Staff Engineer with Intel, Albequerque, NM, USA. His research interests include emerging devices (ReRAM, PCRAM, and MRAM) and their applications in crossbar arrays for next generation computing architectures.

SEUNGJONG LEE (Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 2011 and 2013, respectively. He is currently pursuing the Ph.D. degree with the University of Michigan, Ann Arbor, MI, USA.

From 2013 to 2016, he was an Assistant Research Engineer with Silicon Works, Daejeon, South Korea, working on mixed-signal circuits. His current research interests include data converters and power management ICs.

WEI D. LU (Fellow, IEEE) received the B.S. degree in physics from Tsinghua University, Beijing, China, in 1996, and the Ph.D. in physics from Rice University, Houston, TX, USA, in 2003.

He is currently a Professor with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA. From 2003 to 2005, he was a Post-Doctoral Research Fellow with Harvard University, Cambridge, MA, USA. He joined the UM Faculty in 2005. His research interest includes resistive-random access memory (RRAM)/memristor devices, neuromorphic systems, aggressively scaled transistor devices, and low-dimensional systems.

ZHENGYA ZHANG (Senior Member, IEEE) received the B.A.Sc. degree in computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2003, and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley (UC Berkeley), Berkeley, CA, USA, in 2005 and 2009, respectively.

He has been a Faculty Member with the University of Michigan, Ann Arbor, MI, USA, since 2009, where he is currently an Associate Professor with the Department of Electrical Engineering and Computer Science. His current research interests include low-power and high-performance VLSI circuits and systems for computing, communications, and signal processing.

Dr. Zhang was a recipient of the David J. Sakrison Memroial Prize from UC Berkeley in 2009, the National Science Foundation CAREER Award in 2011, the Intel Early Career Faculty Award in 2013, and the Neil Van Eenam Memorial Award from the University of Michigan, College of Engineering in 2019. He has been on the Technical Program Committees of Symposium on VLSI Circuits and the IEEE Custom Integrated Circuits Conference (CICC) since 2018. He was an Associate Editor of the IEEE Transactions on Circuits and Systems—I: REGULAR PAPERS from 2013 to 2015, and the IEEE Transactions on Circuits and Systems—II: Express Briefs from 2014 to 2015. He has been an Associate Editor of the IEEE Transactions on Very Large Scale Integration Systems since 2015.

MICHAEL P. FLYNN (Fellow, IEEE) received the Ph.D. degree from Carnegie Mellon University, Pittsburgh, PA, USA, in 1995.

From 1995 to 1997, he was a Member of Technical Staff with Texas Instruments, Dallas, TX, USA. During the four-year period from 1997 to 2001, he was with Parthus Technologies, Cork, Ireland. He joined the University of Michigan, Ann Arbor, MI, USA, in 2001, where he is currently a Professor. His technical interests are in RF circuits, data conversion, serial transceivers, and biomedical systems.

Dr. Flynn is a 2008 Guggenheim Fellow. He received the NSF Early Career Award in 2004, the 2005–2006 Outstanding Achievement Award from the Department of Electrical Engineering and Computer Science, University of Michigan, the 2010 College of Engineering Ted Kennedy Family Team Excellence Award from the College from Engineering, University of Michigan, the 2011 Education Excellence Award, and the 2016 University of Michigan Faculty Achievement Award. He was the Editor-in-Chief of the IEEE Journal of Solid-State Circuits (JSSC) from 2013 to 2016. He served as an Associate Editor for the IEEE JSSC and the IEEE Transactions on Circuits and Systems. He is the Chair of the Data Conversion Committee of the International Solid-State Circuits Conference. He formerly served on the Technical Program Committees for ESSCIRC, A-SSCC, and the Symposium on VLSI Circuits. He is also a former Distinguished Lecturer of the IEEE Solid-State Circuits Society.