A Historical Perspective on Hardware AI Inference, Charge-Based Computational Circuits and an 8 bit Charge-Based Multiply-Add Core in 16 nm FinFET CMOS

Kayode A. Sanni, *Member, IEEE*, and Andreas G. Andreou, *Fellow, IEEE*

***Abstract*— The second wave of AI is about statistical learning of low dimensional structures from high dimensional data. Inference is done using multilayer, data transforming networks using fixed point arithmetic with parameters that have limited precision (4-16 bits). In this paper we give a historical perspective on hardware AI inference deep Artificial Neural Networks (ANNs) or in short Deep Neural Networks (DNN) and deep learning. We review custom chip implementations of ANNs from thirty years ago and from the more recent publications in the last five years. With only few exceptions, hardware AI architectures are digital but we argue that if done right, i.e. in the charge domain, analog computation will have a role in future hardware AI systems. We make our discussion concrete by presenting the architecture, implementation and measurements from a mixed-signal, charge-based 8-bit analog multiplier for limited precision linear algebra in AI systems. Using a capacitor array, and charge redistribution, the architecture performs the multiplication operation in the charge domain at the thermal noise limit with near minimum energy dissipation. The charge redistribution multiplier core was fabricated in a 16 nm FinFET CMOS process, with measured energy 1.4 fJ for the analog multiplication operation. Compared to a conventional digital implementation synthesized and simulated in the same technology, the proposed design achieves the same performance at 37% less energy.**

***Index Terms*— Energy aware, analog multiplier, deep neural networks, hardware AI.**

I. Introduction

OVER the last half century computer scientists, architects and engineers have envisioned building computers that match the parallel processing capabilities of biological brains. Seventy years ago, the fathers of computer science Alan Turing [1] and John von-Neumann [2] looked at the brain for inspiration in order to advance the science of computing. At the same time, the psychologist Frank Rosenblatt [3] employed probabilistic graphical models to abstract the organization and computational capabilities of brains. Thirty years ago, the connectionist movement emerged as an alternative approach to Artificial Intelligence (AI) for solving hard problems in perception and cognition. Today the abundance of empirical successes on deep learning and deep convolutional networks (see excellent review in [4]) has been supported by theoretical advances and partial understanding as to the "why" deep neural networks avoid the curse of dimensionality of their "shallow" cousins from thirty years ago [5]. Deep neural networks and deep learning is the second wave of AI that we witness today!

The central doctrine in the connectionist movement is that the cognitive abilities of the brain are a result of a highly interconnected network of simple non-linear units. These simple computational units abstract the function of neurons while synapses abstract the connections between neurons. The strength of the synaptic connections in networks of such units is determined through a learning algorithm. The two volume edited book-set by the "Parallel Distributed Research Group" [6], [7] defined the research agenda in the field of connectionist architectures and neural networks in the decades that followed. But for over thirty years the then so promising brain inspired "artificial neural networks" technology has remained dormant. The only application that had been developed at the scale was alphanumeric character recognition using convolutional networks [8], [9].

Today, a huge portion of human communication, activity, and experience—in the form of speech, natural-language text and images —is mediated by computers, thanks to the popularity of the Internet, digital telephony, email, and other electronic media. "The web has become both our public square and our Library of Alexandria, mediating and archiving huge swaths of human culture and knowledge, ranging from the magisterial to the quotidian".[1]

Manuscript received May 9, 2019; revised July 26, 2019; accepted July 31, 2019. Date of publication August 7, 2019; date of current version September 17, 2019. This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) Defense Sciences Office (DSO) under Contract HR00111720056 and in part by the Northrop Grumman Faculty Award to support graduate student research. This article was recommended by Guest Editor K. Kailas. *(Corresponding author: Andreas G. Andreou.)*

K. A. Sanni was with the Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD 21218 USA. He is now with Northrop Grumman Corporation, Baltimore, MD 21240 USA.

A. G. Andreou is with the Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: andreou@jhu.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JETCAS.2019.2933795

[1]Professor Jason Eisner, direct quote; AGA personal communication, circa 2012

Our individual projects and relationships also evolve through electronic text, speech and images: private documents, emails with picture attachments, tweeting or short messages, phone conversations, and human-computer interactions such as web searches on text and in the future sound and images. Hence the ability to distill useful knowledge and insight from these massive data sources is key to such diverse fields as universal information access, medical records management and diagnosis, financial trading, disease outbreak detection, social networking, and the automatic construction of structured knowledge bases from the myriad sources of scientific and other useful information written by humans for humans. All these, thanks to advances and exponential improvements made in the scaling of size and the density of devices, coupled with the advances in algorithms and computer architectures, CAD and workstation technologies that have enabled the design of complete truly complex Systems On a Chip (SOC) in information and communication technologies that support our daily lives and modern economies. Alex Szalay, a physicist and computer scientist at Johns Hopkins University and the late Jim Gray of Microsoft Research argue that the Moore's law [10] i.e. the successive generation of inexpensive CMOS sensors is creating an exponential growth of data [11].

With the exponential growth of data, computing in data centers, the engines behind our insatiable desire for global communication, instant connectedness and interaction has grown dramatically at an economic and environmental cost. From 2000 to 2005, the total electricity consumption associated with data center servers doubled, reaching 45 billion kWh/year in the US and 120 billion kWh/year worldwide. The corresponding electricity costs were $2B/year, respectively (at $0.04/kWh). Realizing the importance of energy cost, large data centers like those of Google and Microsoft are built in locations with easy access to cheap energy and water cooling. Microsoft's data center in Quincy, Washington is presently one of the world's largest, taking up 47,000 square meter and running on 47 MW, and a new data center is now being built next to the existing one. With todays technology, a Petaflops (10^{15} floating operations per second) machine requires approximately 5 MW of power, which costs approximately 2 Million dollars a year. Historically the number of computations per kWh has increased by a factor of 1.5 every year (Koomey's Law) [18]. According to this figure, a projected Exaflops (10^{18} floating operations per second) computer operating in 2020 would require 50 MW and would generate a yearly electricity bill of $50M. In comparison, the subway in New York runs on a mere 20 MW. We have already passed the point at which the cost of running and maintaining a high performance computer over its lifetime surpasses its capital costs. The main bulk of these additional expenses go towards building space and cooling.

Datacenter computing has traditionally relied on high-end chip-multiprocessor CPUs with large cache memories such as the Intel Xeon-Phi [19] that are commensurate with High Performance Computing (HPC) workloads. Most commodity chip-multiprocessors as well consumer grade GPUs do not have adequate on-chip memory for scientific computing [20], which was the goal of Nvidia's CUDA and AMD's CAL frameworks. Consumer grade GPUs necessitate a factor of 4 to 8 increase in memory per core to be effective in non graphics computation (see Figure 20 in [20]). In recent years the latter problem has been partially resolved thanks to advances in 2.5D and 3D packaging technology [21], [22]. Today high GPU performance at moderate power is possible through such advanced packaging. The use of 2.5D silicon interposer [23] technology and 3D integration [24], [25] using 3D DRAM such as the Micron-IBM Hybrid-Memory-Cube (HMC) [26] or the AMD-Hynix High Bandwidth Memory (HBM) [27] effectively allow for off-the-die high bandwidth memory access at over 2 Gbit/s and at less than 2 pJ/bit energy cost. Hence, data centers have begun to employ the latest high end GPUs [28] from Nvidia [29] and AMD [30] for processing the "BIGDATA" exponentially growing in the cloud i.e., text, images, speech and sound, thus enabling deep neural network (AI) applications to be developed at the scale.

II. The Great AI Awakening

Fueled by the latter advances in GPU technologies and the abundance of data for performing statistical learning and distilling the knowledge into multilayer networks, the last five years we have been witnessing the "Great AI Awakening" [31]. During this period language translation technology using recurrent neural networks and deep neural network models has advanced from an academic innovation [32] to full fledged systems at the scale with impressive results [33]. During the same period there was the realization by Jeff Dean[2] that full fledged systems for real-time language translation where orders magnitude short of what state-of-the-art GPU technology in Google data-centers could support. He spearheaded the effort for accelerating these deep neural network/AI algorithms which resulted to the genesis of the Google Tensor Processing Unit (TPU). TPU details were revealed to the public after 36 months of internal use [34] followed by a system with improved memory interfaces the Tensor Processing Unit 2 (TPU-2) [35]. The hardware architectures were complemented with an extensive software environment known as TensorFlow [36] to facilitate the training and deployment of applications.

It is worth noting that Stu Feldman, computer scientist and VP of engineering at Google, in a 2009 seminar at Johns Hopkins University argued that: "Extreme computing is by definition specialized. If you were to use off-the-shelf components for the system, it is by definition not extreme", suggesting that Google would at some point or another consider specialized processors. Yet at the same time, Bill Dally a prominent computer architect, when asked in a 2010 interview on the future of hardware accelerator and the development of specialized architectures to support data center computing, he did not see any need for such architectures [37].

During the same period deep neural networks have been applied to image feature extraction [38] and classification [39] and hence there was also an increased interest in accelerating deep neural networks for computer vision using

[2]AGA personal communication during lunch with Jeff Dean, Lead of Google A.I. (https://ai.google), April 26, 2018

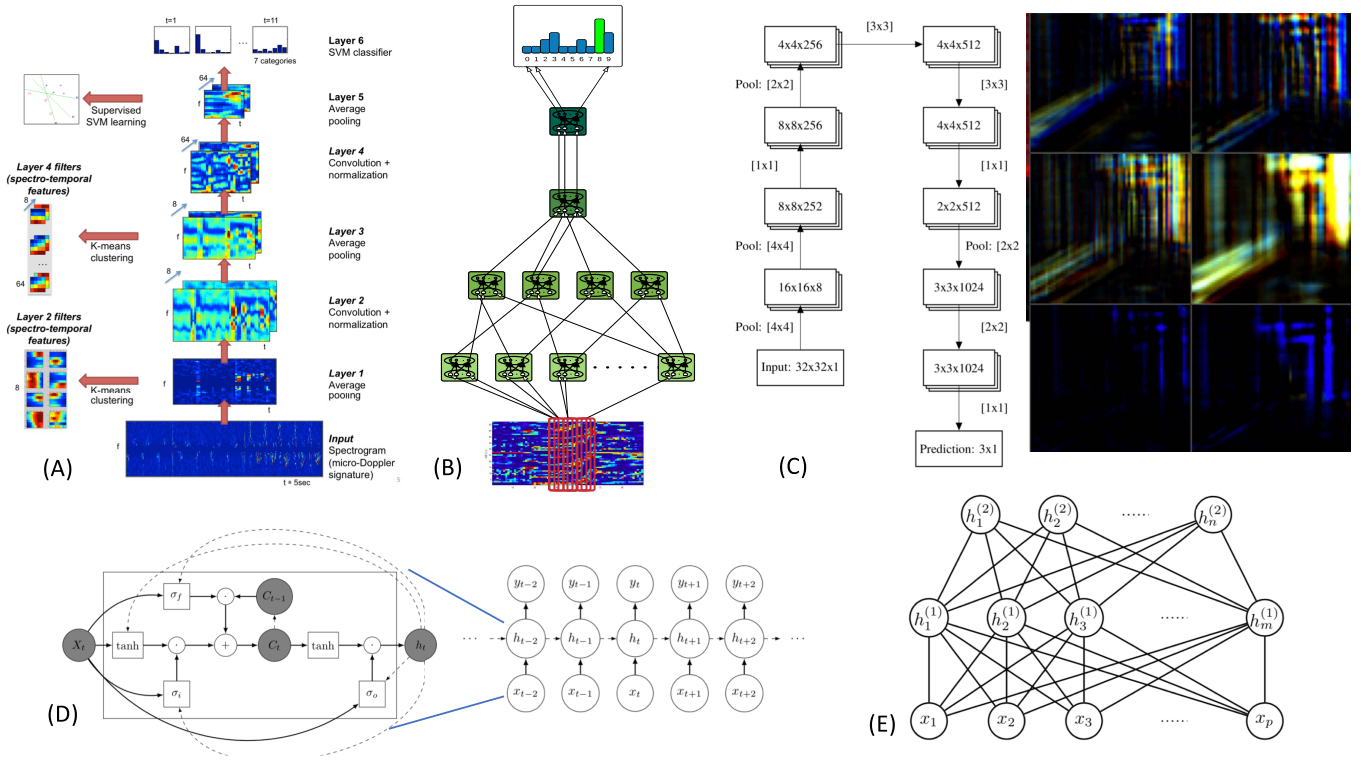


Fig. 1. The 2nd-Wave of AI is exemplified by systems that involve statistical learning of low dimensional structures from high dimensional data. Inference is carried out using multilayer data transforming architectures, with models that have parameters between 4 and 12 bit precision. We list here examples from our own work: (A) Multimodal feature integration from micro-Doppler sonar and sound for human behavior recognition with convolutional neural networks [12]. (B) Real-time sensory information processing, speech recognition and language processing using convolutional networks running on the IBM TrueNorth neuro-synaptic system [13], [14]. (C) Self-driving toy car with neuromoprhic vision sensor and convolutional neural networks for end-to-end navigation [15]. (D) Action recognition from sonar signatures using the Long-Short Term Memory (LSTM) recurrent neural network [16]. (E) Character recognition using a Recurrent Boltzmann Machine (RBM) deep-belief network [17].

FPGAs [40] and hardware accelerators [41]. The prospect of AI assisted semi-autonomous car driving systems [42] has lead to increased interest in energy aware deep neural network accelerators. Artificial neural network accelerators (today called AI processors) have seen a renaissance with a plethora of specialized processors for deep neural networks reported in the literature [43]–[49]. Besides the server oriented solution [34] Google has also developed their own specialized energy aware Tensor Processor Unit for IoT and Edge computing [50], [51].

Much like the hardware neural accelerators a quarter century ago, most architectures for accelerating neural networks have a broadcast data bus. The Adaptive Solutions N64000 chip and the CNAPS server [52] had a linear array of processing units. The T0 vector processor and SPERTII system [53], [54] also had a linear array of fixed point units and a broadcast bus architecture. The Ring Array Processor (RAP) system was developed by the same group using commercial off the shelf TMS32040 DSP units arranged on a ring [55]. Perhaps the most elaborated system developed at that time was the SYNAPSE-1 server that was based on the Siemens MA16 chip [56]. The latter was designed upon careful examination of the workloads in artificial neural network requirements and much like the TPU [34] and the recent work from IBM [46] employed a parallel systolic engine for computations [56]. An excellent overview and description of the systems developed at that time can be found in the edited volume by Ramacher and Ruckert [57]. The interest in custom neural network chips declined as advances in Digital Signal Processing (DSP) chips such as the TI TMS32080 [58] rendered custom neural network hardware obsolete.

III. Analog AI Hardware: Why and How

In the early 90s, the heydays of the "shallow" neural network revolution, Carver Mead [59], advocated using analog transistor physics to perform neural computation, directly mimicking the currents in neuron ion channels. At the same time the pioneer of sub-threshold analog integrated circuits, Eric Vittoz had published a highly influential paper titled "Analog VLSI signal processing: Why, where, and how?" [60] arguing that there is plenty of room for analog signal processing in perceptual systems.

In this section we revisit the topic of analog signal processing in the context of brain like systems and AI hardware. Before addressing the question of "why and how" we first do a brief detour to put the discussion in a historical context. In 1986, Mead's group at Caltech was employing bulk CMOS technology with λ between 2.5μm and 0.7μm (page 59 of [61]). A quick review of our own publications and laboratory notebooks from that period, reveals that we were fabricating chips in 4μm Silicon On Sapphire (SOS)–CMOS technology and in 3μm p-well bulk CMOS. Alas! Thirty

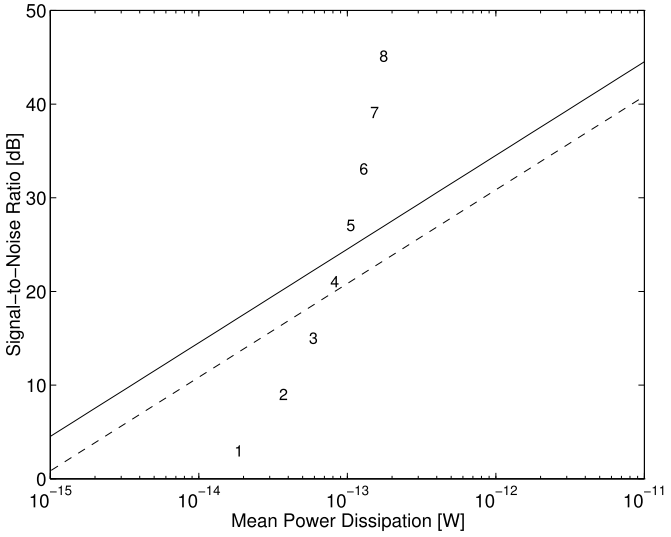


Fig. 2. Signal-to-noise ratio as a function of mean power ($f_p = 0.5$ $f_s =$ 100 MHz, $\epsilon = 1E-12$). CVCT (analog) solid, CVDT (analog sampled data) dashed, DVDT (digital) number of bits.

years later, with foundry FinFET CMOS technologies at the 16nm, 10nm and 7nm nodes, analog neural networks have not capitalized the ($\times$200000) improvements in digital MOS transistor area density to support the engineering of cognitive machines that match the effectiveness and energetic efficiency of the human brain. Our lack of knowledge about the inner workings of brain function and behavior has contributed to this chasm. Today, matching the information processing capabilities of biological neural structures in state-of-the-art silicon technology remains an elusive dream despite the stunning advances in microelectronics.

In the early days of neural networks [62]–[64] analog transistors where employed to emulate the biophysics of neurons and perform analog multiplication and other non-linear computations. Five years ago we have suggested [65] that in the nano-CMOS era, the engineering of large-scale systems aimed at intelligent brain like machines must be carried out at a different level of abstraction. Instead of using analog transistors to emulate the biophysics of neurons, digital transistors must be employed to perform the arithmetic equivalent to the behavior of a neuron. Combined with the advances in high-density 3D CMOS [21], 3D DRAM technology [27], [26], 2.5D integration [25] as well as energy efficient high-speed communication interconnects [66], we see today a renaissance in the engineering of digital deep neural networks hardware (a.k.a. hardware AI) systems at the scale. Nonetheless, our theoretical work [67] as well as calculations by others [68], and more recently by [69] suggest that from an energetic efficiency perspective, analog computation and mixed signal circuits, including charge based sampled data circuits, are a viable alternative to digital circuits for low precision computations.

Figure 2 shows a plot of SNR vs power for an analog (Continuous Value Continuous Time-CVCT-), sampled data (Continuous Value Discrete Time - CVDT-) and a digital (Discrete Value Discrete Time-DVDT-) systems. Using the equations from [67] the analog and sampled data system power consumption as a function of desired signal to noise ratio (S/N) is given by: $P_m = 4kTf_pSNR_{CVCT}[1-\frac{f_s}{f_p}arctan(\frac{f_p}{f_s})]$ and $P_m = 2kTf_pSNR_{CVDT}[1 - sinc(2\pi\frac{f_p}{f_s})]$ respectively. For the digital system the corresponding equation is: $P_m = \frac{kTf_s}{2}\log_2(1 + SNR_{DVDT})[\text{erfc}^{-1}(\frac{\epsilon}{4SNR_{DVDT}})]^2$. These equations are derived for idealized systems operating at the thermal noise limit. The three plots in Figure 2 suggest that at low S/N, analog and analog sampled data circuits are more efficient than digital with the breakpoint at about 4 to 5 bits.

A. Charge Based Processing: Analog Computation Done Right!

In the nano-CMOS era with products at the 10nm and 7nm nodes is there any possibility that analog and or mixed-signal computation can be done right so that it can have an impact on the future hardware AI systems [67].

Since the early days of MOS transistors, charge based circuits have been employed in signal processing to perform vector arithmetic at a time that there were no other options [70], [71]. Charge based circuits were also used for vector quantization in the early 1990s [72]. In the latter work charge based analog circuits performed Euclidean distance computation and search, something that is also done today in some AI accelerators. Subsequently it was recognized that charge domain processing can be employed in CMOS for computations ranging from simple weighted multiplication and addition [73]–[75] and more recently high frequency analog FFT [76]. Charge based computation has also been employed for neural network accelerators in the past [77], [78]. Impressive energy efficiency per operation has been attained when performing 1-bit bitwise vector-vector multiplication [79], [80] and in recent work multi-bit vector-vector multiplication [81]. While not targeted specifically for AI, charge based neuromorphic circuits of [82] and more recently of [83] and [84] essentially perform mixed-signal low precision dot product operations in arrays of "neuronal" elements. A charge based, passive vector dot-product multiplier circuit followed by a SAR converter was also reported in the literature [85]. A 6T+1C SRAM based binary compute in memory accelerator also computes in charge domain [86]. A similar approach was used in earlier work where a pseudo-DRAM is employed to store the binary state vectors and computations are done using binary [80] or multi-bit inputs [81] in the charge domain.

A number of charge based architectures for low precision Vector-Vector Multiply ACcumulate (VV MAC) compute in memory units have also been investigated by Sanni [87], [88]. In the latter architectures computation is carried out by counting at the thermal noise limit, using packets of about 1000 electrons. These systems are neither analog nor digital in the traditional sense but employ mixed signal circuits to count the packets of charge and hence we call them *Quasi-Digital*. By amortizing the energy costs of the mixed signal encoding and decoding circuits over a large number of elements in the vectors high energy efficiencies can be achieved. Figure 3 summarizes the results from five generations of chips fabricated in 65/55nm CMOS technology.

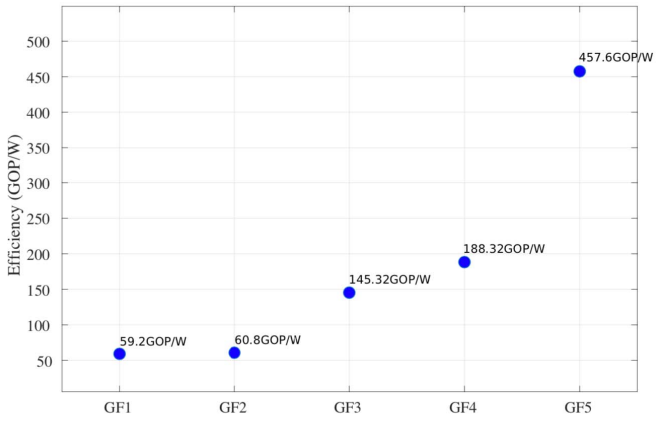


Fig. 3. Low precision MAC operations per watt measured in the work of [87], [88].

IV. Charge Based Analog Multiplier

In the this section, we take an alternative approach to charge based multiplier taking inspiration from the architecture of the Successive Approximation Register (SAR) analog to digital converters. The successive approximation principle and charge re-distribution employed in the multiply-accumulate core is based on the widely used Successive Approximation Analog to Digital Converter (SA ADC). The emergence of MOS technology in the early 70s has created opportunities for new data converter architectures that were charge based. The successive approximation (SA) concept is described in two seminal papers by the Berkeley analog integrated circuits group [89], [90] where the successive approximation register architecture was proposed as the basis for all MOS charge based analog-to-digital converters.

As one of the most popular architectures for high speed low-power data conversion, SA ADC has been widely adapted for general purpose data conversion [91]–[93], in CMOS imager [94], and in sensor networks [95], More specifically the charge redistribution SA ADC, which comprises of an analog switched capacitor array, a comparator, and some digital decoding logic, can be designed for both high resolution and high speed while being implemented in relatively small area. Using a successive approximation register (SAR), the SA ADC achieves efficient data conversion by performing a binary search through all possible quantization levels to converge to the correct digital output representing the analog input. SA ADCs have been designed in silicon-on-sapphire (SOS) CMOS [96], and in 16nm FinFET CMOS [97], [98].

A. Operation

The proposed mixed-signal Multiply-ADD (MADD) architecture, computes a fixed-point multiply-add operation as

$$y = wx + c, \tag{1}$$

for a signed weight w, input x, and offset c encoded as signed magnitude. Conventionally, the unsigned product of the magnitude of w and x can be computed as the sum of the partial products deduced from the individual bits of the input and weight as illustrated in Table I. The sign of this product is computed separately from the logically AND operation of the most significant bit (MSB) of the weight and input. Subsequently, the offset c is added to the resulting product to derive the output of this operation.

TABLE I
Binary Multiplication

$$\begin{array}{cccccccc} & & & & \mathbf{w_{n-1}} & \cdots & \mathbf{w_1} & \mathbf{w_0} \\ & & & \times & \mathbf{x_{n-1}} & \cdots & \mathbf{x_1} & \mathbf{x_0} \\ \hline & & & & \mathbf{p_{(n-1,0)}} & \cdots & \mathbf{p_{(1,0)}} & \mathbf{p_{(0,0)}} \\ & & & \mathbf{p_{(n-1,1)}} & \cdots & \mathbf{p_{(1,1)}} & \mathbf{p_{(0,1)}} & \\ & & & \vdots & & & & \\ + & \mathbf{p_{(n-1,n-1)}} & \cdots & \mathbf{p_{(1,n-1)}} & \mathbf{p_{(0,n-1)}} & & & \\ \hline \mathbf{y_{2n-1}} & & & \cdots & & & \mathbf{y_1} & \mathbf{y_0} \end{array}$$

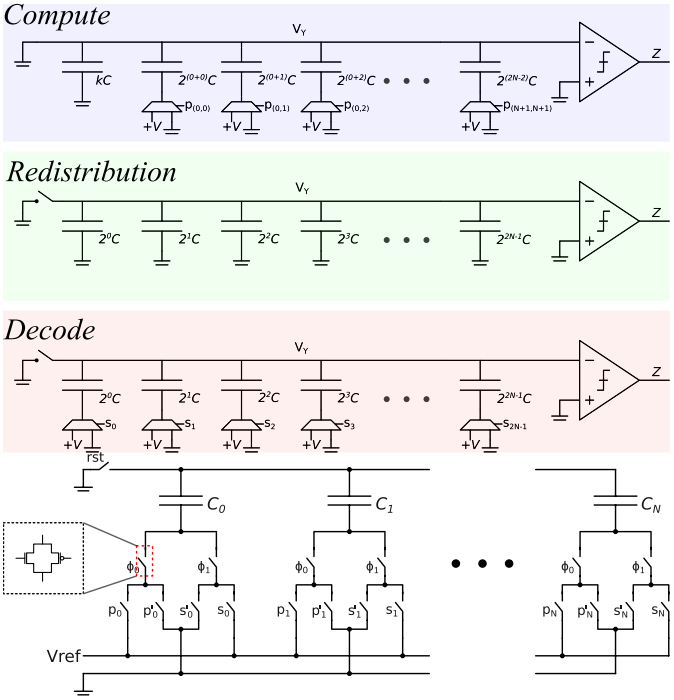


Fig. 4. (Top) Reconfiguration of the switches in the core of the SA based MULTiply-ADD core. Multiplication-Addition comprises of three different phases/steps – the *Compute* phase, *Redistribution* phase, and finally the *Decode* phase – for integrating the weighted partial product bits. (Bottom) Detail circuit of the programmable capacitor array.

B. Computational Phases

As opposed to using a cascade of full adders for summing the partial products, a programmable capacitor array is used to sum the scaled partial product bits as charge. Computing in the charge domain allows the operations to be performed with the ones and zeroes being represented as charge packets of about 1000 electrons. Moreover, this charge can then be converted efficiently using a Successive Approximation (SA) based binary search. Integrating together this computational and decoding structure, allows for re-use of components; the mixed-signal multiplier steps are shown in Figure 4.

First, in the *Compute* phase, a charge equivalent to the sum of the weighted partial product bits is injected into the array of capacitors. The top plates of all capacitors are connected to a common ground node, while the bottom plates are isolated

and connected to either a power supply with $+V$ volts or the common ground node depending on the partial product bits. This total charge Q_Y can be expressed as

$$Q_Y = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} 2^{i+j} p_{(i,j)}(CV), \tag{2}$$

where i and j are the indexes of the partial product bits generated from the logical AND operation between the weight and input as shown in Table I, C is the unit capacitance, and V is the voltage potential induced across the capacitors. Thus, by connecting a set of capacitor in parallel using switches, the partial product can be integrated as the total charge Q_Y injected onto this capacitor array.

After this first phase, the charge Q_Y is translated into a voltage V_Y in the *Redistribution* phase. In this phase, charge is redistributed evenly across all the capacitors in the array by disconnecting the top plates of the capacitors to ground, and connecting all the bottom plates of the capacitors to ground. Since the charge is conserved between the Compute and Redistribution phase,

$$Q_Y = C_T V_Y \tag{3}$$

$$\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} 2^{i+k} p_{(i,j)}(CV) = C_T V_Y \tag{4}$$

$$V_Y = \left(\frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} 2^{i+k} p_{(i,j)} C}{C_T} \right) V, \tag{5}$$

where C_T is the total capacitance of the array when all the capacitor are connected. C_T is sized for the precision of the computation, which is,

$$C_T = (2^{2N} - 1)C \tag{6}$$

from the computation ($2N$ bit precision). Thus, the voltage V_Y represents the sum of the weighted partial product bits scaled by the voltage of the power supply V and the capacitance of the array C_T.

Finally, this voltage representing the output of computation is converted back into a digital value in the *Decode* phase. Using a digital control word s_i, the capacitor array is reconfigured from a parallel configuration to a series-parallel configuration in order to perform the binary search to decode the voltage V_Y into the digital word. In the series-parallel configuration, the V_Y is voltage divided according to each bit position in the digital word, and compared to the common ground node indicating the magnitude of that bit for the decoded result. Although the charge from the computation is preserved on the capacitor array, an additional charge is injected into the array while decoding, which in turns creates a delta voltage; this voltage V_D is given as

$$V_D = V_Y + \left(\frac{C_D}{C_T} \right) V, \tag{7}$$

where C_T is given earlier as $(2^N - 1)C$, C_D is total capacitance of the capacitors with the bottom plate connected to the power supply with voltage V in the capacitor array, and V_Y is initial voltage for this phase that represents the output of the computation. During decoding, C_D changes exponentially at a rate of 2^i according to the control word s_i from most-significant to least-significant bit. Through the binary-search the result of the computation can be decoded in N time steps, where $N = 8$ is the bit-precision of the decoded output.

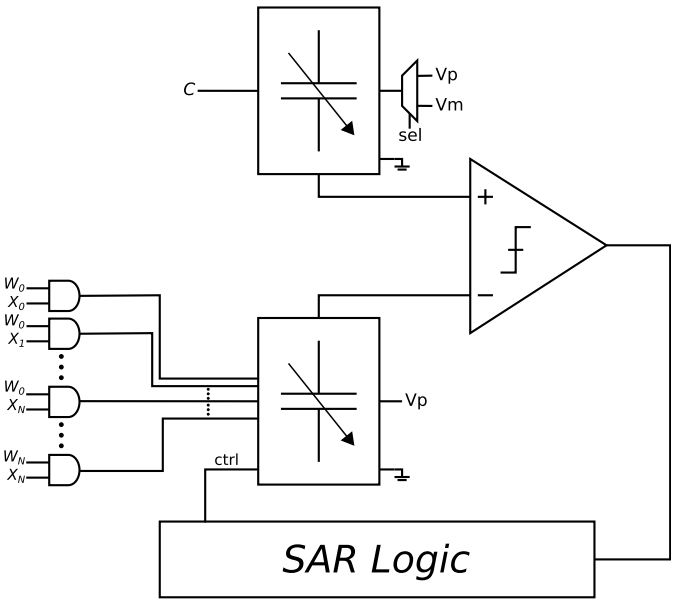


Fig. 5. SA Architecture for Multiply-Add Operations. This architecture operates in three phases. As shown in the full architecture diagram (Figure 4), the partial product bits are computed from the weight w and the input x. These bits are then used to program a capacitor array where y is the output encoded as the bit-stream on the output of the comparator. The connected top plates of this capacitor array connects to the comparator which is used in decoding the output of the computation from the analog domain. An additional programmable capacitor array is used for generating a voltage offset used during decoding for implementing the addition operation and also calibration if necessary.

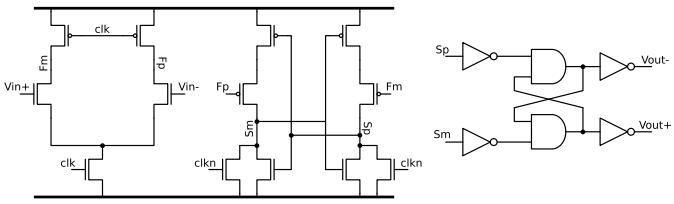


Fig. 6. Dynamic comparator and latch for the SA architecture.

Furthermore, by adapting this architecture to a differential topology, the addition operation can be implicitly performed with a voltage offset on the other input of the comparator. Hence allowing the creation of efficient pipeline computational structures. Also, this additional capacitor array can be used to correct the comparator offset.

C. Circuits for Multiplier Core Architecture

The full architecture implementing the fixed-point multiply-add operation can be seen in Figure 5.

By further optimizing the circuit, the 3 phase multiplication step can be simplified into two phases, where the first phase ϕ_0 does the computation, and the second phase ϕ_1 does the redistribution and the decode step. Analog multiplexers com-

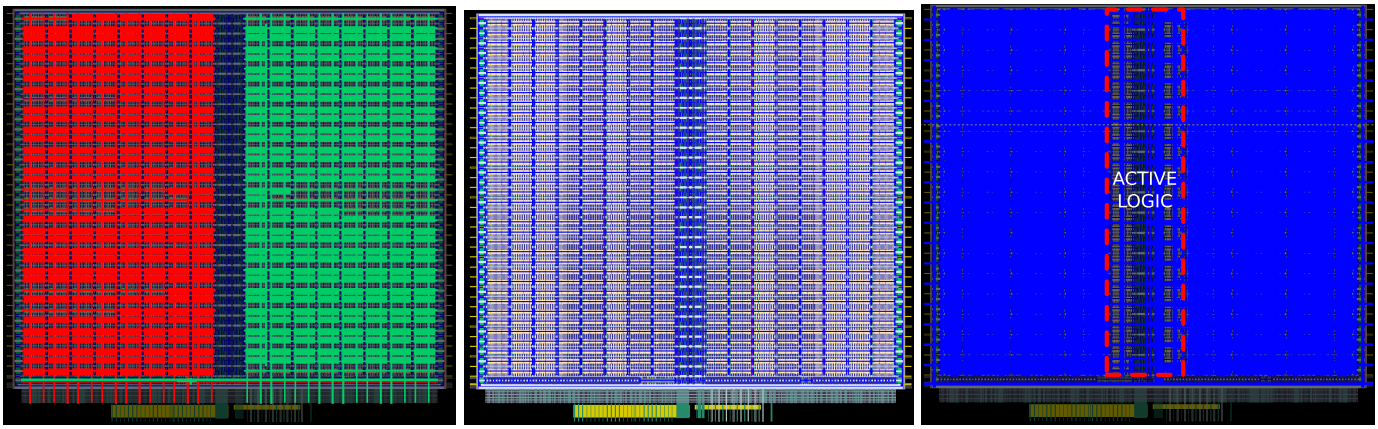


Fig. 7. (Left) Layout view of the SA Multiply-Add core with the active capacitors highlighted. The main capacitor array for summing the partial product bits is highlighted in red, and the capacitor array for calibration and the additional add operation is in green. (Middle) Layout of the SA multiply-add core without SAR decoding logic. (Right) Layout view of the SA multiply-add core with the inactive logic highlighted (the layout highlighted in blue is the inactive logic that is situated under the capacitors).

prised of transmission gates are used to charge and discharge the capacitor to either the reference voltage or the common ground. The rst switch is used to reset the top node plate for each processing cycle.

A two-stage dynamic comparator (shown in Figure 6), along with a SR latch is used for decoding the output computation from the analog domain. When the clk is low, the comparator sets Sp and Sm low, which causes the latch to hold its value. Then when clk is high, the difference between the inputs $Vin+$ and $Vin-$ is amplified and reflected across Sp and Sm, which pull to the respective rail through positive feedback. This values is then stored on the latch. Moreover, this comparator has minimal static power dissipation as none of the stages are biased with a constant current.

V. 16nm CMOS FinFET Test Chip

A test chip, that includes the custom layout for the charge based multiplier-add circuit was fabricated in 16nm FinFET CMOS. The core layout, which was matched to the circuitry described in Section IV, was optimized for minimal area and energy, and it was validated with parasitic extraction simulations. In order to maximize the efficiency of area utilization, the capacitor array for the DAC was constructed as a 2D grid of cells comprised of a unit capacitor and the pass gate logic for driving the cell (as seen in Figure 7). A metal-oxide-metal (MOM) capacitor design is used for the unit capacitors. The pass gate logic and the capacitor are laid out jointly in a 3D layout topology, where the MOM capacitors, which were designed across the higher level metals, are underlaid with the periphery circuitry and routing. The layout for the unit cell for the DAC can be seen in Figure 8.

The MOM capacitor is designed across two metal layers (M4 and M5), and is shielded below and laterally to minimize parasitic coupling from the digital signals and adjacent cells. This cell measures 1.248μm by 2.976μm (3.714μm^2) in area with an approximate unit capacitance of 4fF, which was deduced from the parasitic extraction analysis.

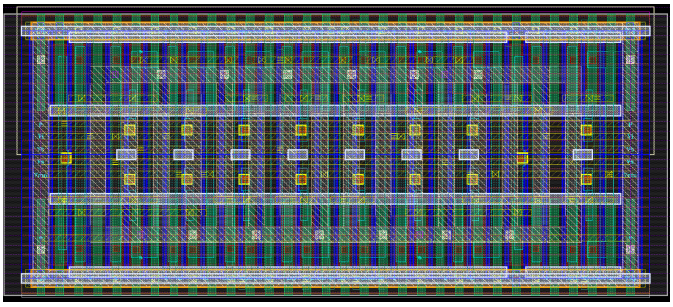


Fig. 8. Layout of the unit cell of the DAC for the SA multiply-add core.

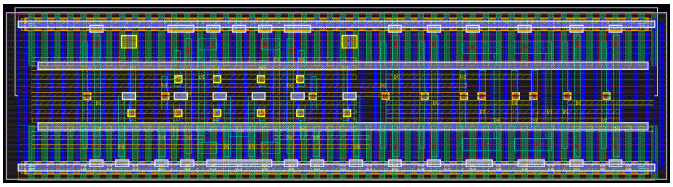


Fig. 9. Layout of the dnamic comparator and latch for the SA multiply-add core.

The dynamic comparator and latch (circuit shown in Figure 6) were laid out in 1.056μm by 4.604μm (4.862μm^2) silicon area, and a netlist, including parasitic nodes, was extracted and simulated to verify functionality and measure performance. The layout for this comparator and latch can be seen in Figure 9. The complete layout of the capacitor array, comparator, latch, and periphery circuits for the SAR multiply-add architecture can be seen in Figure 7.

The comparator was simulated to run at 1GHz, and a Monte Carlo simulation was also performed to characterize the comparator offset. A histogram plot of the result of this simulation can be seen in Figure 10.

The design measures 51μm by 53μm in area (2703μm^2) without the digital periphery circuit (which can be shared across cores when designing a vector unit), and computes 8-bit products and additions. Comparing to a digital implementation, which would require roughly 96μm^2 of silicon

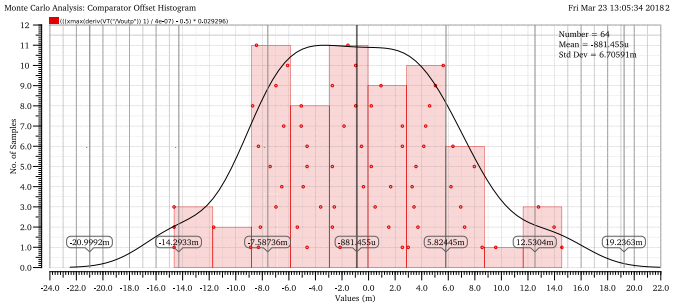


Fig. 10. Histogram of the comparator offset with fabrication variation. The maximum comparator offset with this design with process and mismatch variations did not exceed 21mV.

TABLE II
SIMULATED AND MEASURED PERFORMANCE OF THE SA MULTIPLY-ADD CORE.

Technology	16nm FinFET Simulated	16nm FinFET Measured
Supply Voltage	0.4V	0.8V
Clock Frequency	50MHz	20MHz *
Throughput	3.57MOPs	2MOPs *
Power	24.5nW	–
Total Energy/Op	6.85fJ	–
Performance	146TOPs/W	–
Analog Energy/Op	0.66fJ	1.4fJ

* Maximum clock frequency and minimum operating voltage was limited by the test setup. The total power for the core could not be measured directly as the power supply pad is shared with other test circuits and the I/O pads hence the results are not included in the third column.

area, this SA-based multiply-add core has higher area cost. Nonetheless, majority of this area is primarily used for the layout of the unit capacitors, as shown in Figure 7. For an 8-bit product and addition computation, a total of 510 capacitors are used for both capacitor arrays. In order to facilitate the routing and sharing of the periphery logic circuitry for driving the capacitors, the capacitor arrays are arranged in a 2D grid of 36 by 8 unit capacitors Thus, 33 dummy capacitors are incorporated in each capacitor array. It is worth noting that since the periphery logic that drives the cells are shared across capacitors, most of the underlaid circuitry is inactive and unused; Figure 7(left) highlights the inactive logic in the core layout. The total active logic area is $271.8\mu m^2$, which accounts for approximately 10% of the total area of the core. Furthermore, majority of this active logic used in this core is pass gate logic, which has minimal dynamic power, as cells are not constantly switching between the power rails.

A. Energy Efficiency

Measurements should ideally achieve the simulated performance shown in Table II. However this is not possible during testing due to bandwidth constraints from limited number of pins and the serial I/O used to move data from the test-board to the chip. Furthermore, due to pin limitations, power and ground are shared among multiple test structures, multiplexing and pads hence accurate measurements of power dissipation

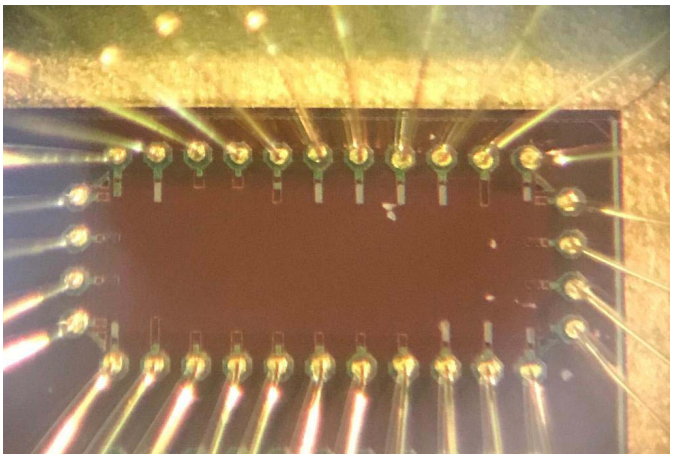


Fig. 11. Photomicrograph of the packaged test-chip that includes the SA MADD core. The pads are shared with other test structures on the same chip.

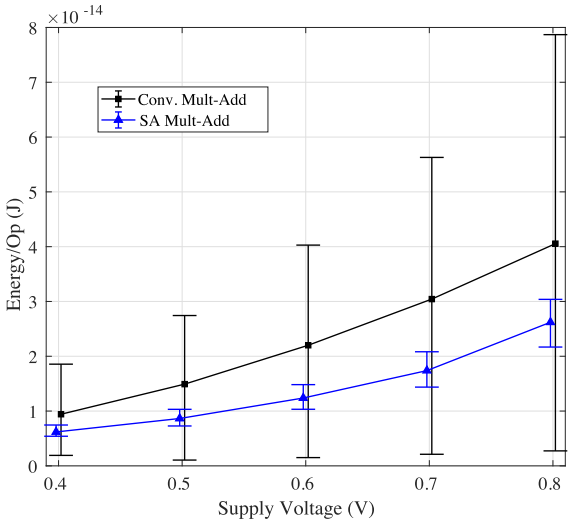


Fig. 12. Simulation results showing energy plotted across different supply voltages for the 8bit SA MADD core and for a synthesized 8 bit multiplier using standard library cells.

was not possible. The photomicrograph of the fabricated test chip can be seen in Figure 11.

In order to compare the computational efficiency of this design to that of conventional architectures, circuit simulations along with an energy analysis is done. The mixed-signal SA architecture was constructed to compute 8-bit multiply-add operations with 5-bit signed weights and inputs, and an 8-bit offset. The capacitor array was designed with unit capacitors in the order of 4fF, with an input voltage bias of 50mV (and an LSB of $200\mu V$).

The SPICE models of both implemented designs (charge based and traditional) were simulated with different supply voltages and clock rates in order to measure performance and efficiency. With a nominal process supply voltage of 0.8V, the power supply is swept from 0.4-0.8V for both designs, and the total energy from these simulations were measured. The simulation results are summarized Figure 12.

As the energy fluctuates with both leakage current and dynamic current from gates switching, the measurements are

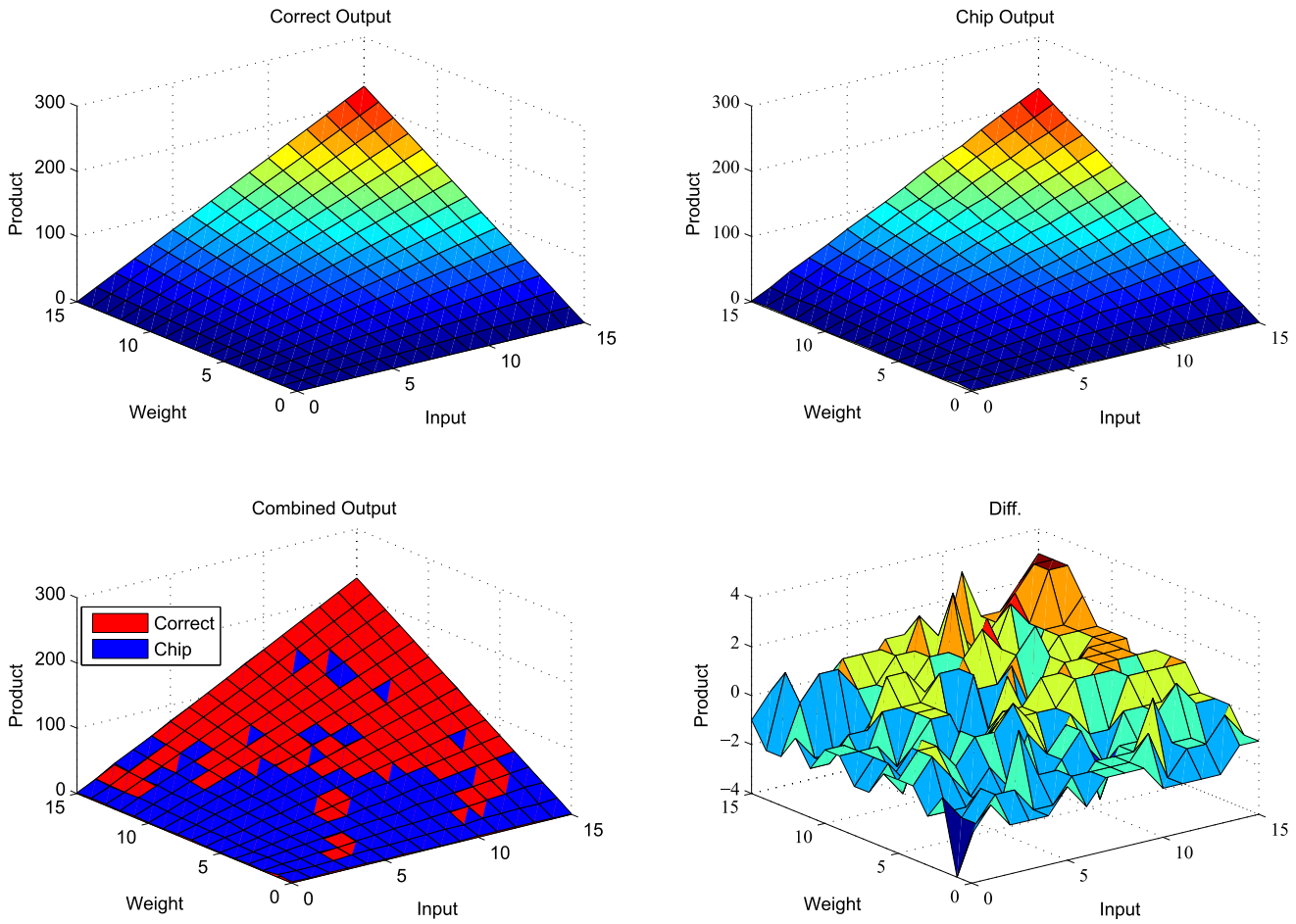


Fig. 13. Measured results of computation through a multiplier sweep of input and weights, before and after calibration. The figure on the bottom right corner plots the absolute error in the computation.

recorded as the inputs are randomly sampled while measuring the distribution of the energy cost for different supply voltages. The SA-based multiply-add design on average is more energy-efficient, and has lower maximum energy-cost compared to a conventional implementation.

VI. Experimental Results

The chip was packaged and interfaced to Spartan-6 Opal-Kelly FPGA using a fabricated PCB board for testing and measurements. A custom DAC/ADC board provides biases for the analog circuits on the test-chip. Results are shown in Figures (13) and (14).

The measured characteristic from the test chip are shown in Table II (right). The clock frequency of the SA Multiply-Add core is limited by the pad I/O cells. As mentioned earlier, the power and energy estimates factors in the cost of the periphery circuits, which include the I/O cells, serial interface, and input and output registers. Nonetheless, as the energy of the analog block (capacitor array) is measured separately, we see that the Energy/Op of the charge computational circuit is comparable between the simulation and measured results. This test chip is capable of computing products with an average output precision of 8bits and a minimum of 6bits. As the energy scales quadratically with the supply voltage, both designs achieve their minimum bound of energy cost per operation at 0.4V as evident from the simulation results.

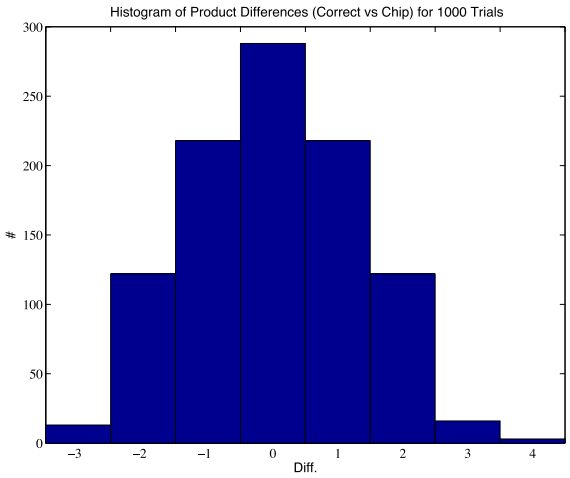


Fig. 14. Histogram of product differences between measured results and theoretical predictions created from one thousand random trials.

VII. Discussion

While the SA Multiply-Add core is superficially similar to the earlier work by Bankman and Murmann [85] the details

of operation are different. In the latter work they employ a passive charge re-distribution technique followed by a SAR data converter. In the work presented in this paper there is only one multiplier which is broken up into the digital logic for producing the partial product bits, and the SAR ADC for the implicit summing of these partial products. Both designs employ a signed magnitude encoding; in the architecture presented here we simply set the MSB (a simple XOR gate), while in the their architecture they worked it into their multiplier switched-capacitor circuit.

The conventional multiply-add design uses an average of 9.4fJ per operation, while the SA multiply-add design uses 6.85fJ on average per operation, which equates to an energy savings of roughly 37%. Furthermore, the majority of the energy cost is attributed to the dynamic comparator, while the capacitor array, which does the compute operation, uses roughly 0.66fJ. An improved dynamic comparator design will improve the power dissipation of the overall multiplier. The summary of the simulated characteristics of the SA multiply-add core can be seen in TableII.

Measurements presented in this table are solely based on SPICE model simulations, and are not fully indicative of the performance and efficiency of a fabricated core. Furthermore, the energy dissipated from the SAR decoding logic is not included in this simulation as this logic can be shared across multiple cores with the exception of the state-holding Successive Approximation Register (SAR). Nonetheless, these results show that the proposed SA multiply-add core can compute 8-bit multiply-add operations with an average efficiency 146TOPs/W.

It is clear from Figure 7 that in a FinFET CMOS process, much area is wasted to implement the capacitor array for the charge based multiplier. We believe that this simply a technological limitation that can be resolved by using an eDRAM (embedded DRAM) technology or fine grain 3D integration of the analog circuit core with a layer optimized for designing capacitors.

VIII. Conclusion

In this paper, a charge based mixed-signal architecture is presented as an energy efficient alternative to conventional digital signal processors for fixed-point multiply-add in hardware AI. Based on the widely-popular successive approximation SA ADC, this architecture utilizes a programmable capacitor array to compute products and sums, that are subsequently decoded into a digital value using a binary-search scheme. Designed in a 16nm FinFET process, the architecture was simulated to compute an 8-bit multiply-add operations at 6.85fJ at 0.4V supply voltage, and an average energy efficiency of 146TOPs/W. Experimental results from measurements are in agreement with the simulation results. Compared to a conventional digital architecture implementing the same operation at comparable bit-depth, the proposed design used 37% less energy.

Acknowledgment

The authors are grateful to Dr. Isidoros Doxas and Dr. Louise Sengupta for their support and encouragement. Chip fabrication was supported by the DARPA CRAFT program.

References

[1] A. M. Turing, "The chemical basis of morphogenesis," *Philos. Trans. Roy. Soc. London, B, Biol. Sci.*, vol. 237, pp. 37–72, Aug. 1952.
[2] J. V. Neumann, *The Computer and the Brain*. New Haven, CT, USA: Yale Univ. Press, 1958.
[3] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.
[4] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
[5] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao, "Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review," *Int. J. Autom. Comput.*, vol. 13, no. 7553, p. 178, Mar. 2017.
[6] D. E. Rumelhart, J. L. McClelland, and PDP Research Group, *Parallel Distributed Processing: Foundations*, vol. 1. Cambridge, MA, USA: MIT Press, 1987.
[7] J. L. McClelland, D. R. Rumelhardt, and PDP Research Group, *Parallel Distributed Processing: Psychological and Biological Models*, vol. 2. Cambridge, MA, USA: MIT Press, 1987.
[8] Y. Le Cunn, L. Bottou, and Y. Bengio, "Reading checks with multilayer graph transformer networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 1997, pp. 151–154.
[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[10] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, pp. 114–117, Apr. 1965.
[11] A. Szalay and J. Gray, "2020 computing: Science in an exponential world," *Nature*, vol. 440, no. 7083, pp. 413–414, Mar. 2006.
[12] S. Dura-Bernal, G. Garreau, J. Georgiou, A. G. Andreou, S. L. Denham, and T. Wennekers, "Multimodal integration of micro-Doppler sonar and auditory signals for behavior classification with convolutional networks," *Int. J. Neural Syst.*, vol. 23, no. 5, pp. 1350021-1–1350021-15, 2013.
[13] A. G. Andreou *et al.*, "Real-time sensory information processing using the truenorth neurosynaptic system," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Mar. 2016, pp. 1–3.
[14] D. R. Mendat, A. S. Cassidy, G. Zarrella, and A. G. Andreou, "Word2vec word similarities on IBM's truenorth neurosynaptic system," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, Oct. 2018, pp. 595–598.
[15] K. D. Fischl *et al.*, "Neuromorphic self-driving robot with retinomorphic vision and spike-based processing/closed-loop control," in *Proc. 51st Annu. Conf. Inf. Sci. Syst.*, Mar. 2017, pp. 1–6.
[16] J. Craley, T. S. Murray, D. R. Mendat, and A. G. Andreou, "Action recognition using micro-Doppler signatures and a recurrent neural network," in *Proc. 51st Annu. Conf. Inf. Sci. Syst.*, Mar. 2017, pp. 1–5.
[17] K. Sanni *et al.*, "FPGA implementation of a deep belief network architecture for character recognition using stochastic computation," in *Proc. 49th Conf. Inf. Sci. Syst. (CISS)*, Mar. 2015, pp. 1–5.
[18] J. G. Koomey, S. Berard, M. Sanchez, and H. Wong, "Assessing trends in the electrical efficiency of computation over time," Intel Microsoft, Santa Clara, CA, USA, Tech. Rep., Aug. 2009.
[19] A. Sodani *et al.*, "Knights landing: Second generation xeon phi product," *IEEE Micro*, vol. 36, no. 2, pp. 34–36, Mar./Apr. 2016.
[20] A. S. Cassidy and A. G. Andreou, "Beyond Amdahl's law: An objective function that links multiprocessor performance gains to delay and energy," *IEEE Trans. Comput.*, vol. 61, no. 8, pp. 1110–1126, Aug. 2012.
[21] R. S. Patti, "Three-dimensional integrated circuits and the future of system-on-chip designs," *Proc. IEEE*, vol. 94, no. 6, pp. 1214–1224, Jul. 2006.
[22] J. Y. Chen, "GPU technology trends and future requirements," in *IEDM Tech. Dig.*, Dec. 2009, pp. 1–6.
[23] A. Arunkumar *et al.*, "MCM-GPU: Multi-chip-module GPUs for continued performance scalability," in *Proc. 44th Annu. Int. Symp.*, 2017, pp. 320–332.
[24] S. Borkar, "3D integration for energy efficient system design," in *Proc. 48th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2011, pp. 214–219.
[25] G. H. Loh, "Computer architecture for die stacking," in *Proc. Tech. Program VLSI Technol., Syst. Appl.*, Apr. 2012, pp. 1–2.
[26] *Micron Hybrid Memory Cube*, Micron-Technol., Boise, ID, USA, 2013, pp. 1–2.

[27] H. Jun *et al.*, "HBM (high bandwidth memory) DRAM technology and architecture," in *Proc. IEEE Int. Memory Workshop (IMW)*, May 2017, pp. 1–4.

[28] S. W. Keckler, W. J. Dally, B. Khailany, M. Garland, and D. Glasco, "GPUs and the future of parallel computing," *IEEE Micro*, vol. 31, no. 5, pp. 7–17, Sep. 2011.

[29] "NVIDIA Tesla V100 GPU architecture," NVIDIA, Santa Clara, CA, USA, Tech. Rep., Jun. 2017.

[30] AMD. (Aug. 2017). *Radeon's Next-Generation Vega Architecture*. [Online]. Available: https://radeon.com/_downloads/vega-whitepaper-11.6.17.pdf

[31] G. Lewis-Kraus, "The great A.I. awakening," *New York Times Mag.*, pp. 1–38, Dec. 2016.

[32] T. Mikolov, M. Karafiát, L. Burget, J. Cernocky, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. Int. Conf. Speech Commun. Technol. (INTERSPEECH)*, 2010, pp. 1045–1048.

[33] Y. Wu *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," Sep. 2016, *arXiv:1609.08144*. [Online]. Available: https://arxiv.org/abs/1609.08144

[34] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proc. 44th Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2017, pp. 1–17.

[35] J. Dean, "Machine learning for systems and systems for machine learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Apr. 2019, pp. 1–54.

[36] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.

[37] A. G. Andreou, T. Abel, W. J. Dally, and T. J. Sejnowski. *The Future of Computing, From Extreme to Green*. [Online]. Available: http://www.kavlifoundation.org/science-spotlights/future-computing-extreme-green

[38] Q. V. Le *et al.*, "Building high-level features using large scale unsupervised learning," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 1–11.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. 25 (NIPS)*, 2012, pp. 1097–1105.

[40] C. Farabet, B. Martini, P. Akselrod, S. Talay, Y. LeCun, and E. Culurciello, "Hardware accelerated convolutional neural networks for synthetic vision systems," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2010, pp. 257–260.

[41] A. Dundar, J. Jin, B. Martini, and E. Culurciello, "Embedded streaming deep neural networks accelerator with applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 7, pp. 1572–1583, Jul. 2017.

[42] M. Bojarski *et al.*, "End to end learning for self-driving cars," Apr. 2016, *arXiv:1604.07316*. [Online]. Available: https://arxiv.org/abs/1604.07316

[43] L. Cavigelli and L. Benini, "Origami: A 803 GOp/s/W convolutional network accelerator," Dec. 2015, *arXiv:1512.04295*. [Online]. Available: https://arxiv.org/abs/1512.04295

[44] B. Moons, R. Uytterhoeven, W. Dehaene, and M. Verhelst, "ENVISION: A 0.26-to-10 TOPS/W subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28 nm FDSOI," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 246–247.

[45] Y. H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.

[46] B. Fleischer *et al.*, "A scalable multi-TeraOPS deep learning processor core for AI trainina and inference," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2018, pp. 35–36.

[47] Z. Yuan *et al.*, "Sticker: A 0.41-62.1 TOPS/W 8 bit neural network processor with multi-sparsity compatible convolution arrays and online tuning acceleration for fully connected layers," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2018, pp. 33–34.

[48] S. Yin *et al.*, "An ultra-high energy-efficient reconfigurable processor for deep neural networks with binary/ternary weights in 28 nm CMOS," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2018, pp. 37–38.

[49] K. Ueyoshi *et al.*, "QUEST: A 7.49 TOPS multi-purpose log-quantized DNN inference engine stacked on 96 MB 3D SRAM using inductive-coupling technology in 40 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018, pp. 216–218.

[50] *Coral Dev Board Datasheet*, Google, Menlo Park, CA, USA, Mar. 2019.

[51] O. Tenam, H. Khaitan, R. Narayanawami, and D. H. Woo, "Neural network accelerator with parameters resident on chip," U.S. Patent 2019 0050717 A1, Feb. 14, 2019.

[52] D. W. Hammerstrom, "A VLSI architecture for high-performance, low-cost, on-chip learning," in *Proc. Int. Joint Conf. Neural Netw.*, Feb. 1990, pp. 537–544.

[53] K. Asanovic *et al.*, "CNS-1 architecture specification," Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. TR-93, Apr. 1993.

[54] K. Asanovic, N. Morgan, and J. Wawrzynek, "Using simulations of reduced precision arithmetic to design a neuro-microprocessor," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 6, no. 1, pp. 33–44, 1993.

[55] N. Morgan, J. Beck, P. Kohn, J. Bilmes, E. Allman, and J. Beer, "The ring array processor: A multiprocessing peripheral for connectionist applications," *J. Parallel Distrib. Process.*, vol. 14, no. 3, pp. 248–259, Mar. 1992.

[56] U. Ramacher, "SYNAPSE—A neurocomputer that synthesizes neural algorithms on a parallel systolic engine," *J. Parallel Distrib. Process.*, vol. 14, no. 3, pp. 306–318, Mar. 1992.

[57] U. Ramacher and U. Ruckert, *VLSI Design of Neural Networks*. Norwell, MA, USA: Kluwer, 1991.

[58] *TMS320C80 Digital Signal Processor*, Texas Instrum., Dallas, TX, USA, 1997.

[59] C. Mead, "Neuromorphic electronic systems," *Proc. IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990.

[60] E. Vittoz, "Analog VLSI signal processing: Why, where, and how?" *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 8, no. 1, pp. 27–44, 1994.

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

[62] M. Holler, S. Tam, H. Castro, and R. Benson, "An electrically trainable artificial neural network (ETANN) with 10240 floating gate synapses," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 1989, pp. 191–196.

[63] B. E. Boser, E. Sackinger, J. Bromley, Y. Le Cun, and L. D. Jackel, "An analog neural network processor with programmable topology," *IEEE J. Solid-State Circuits*, vol. 26, no. 12, pp. 2017–2025, Dec. 1991.

[64] E. Sackinger, B. E. Boser, J. Bromley, Y. LeCun, and L. D. Jackel, "Application of the ANNA neural network chip to high-speed character recognition," *IEEE Trans. Neural Netw.*, vol. 3, no. 3, pp. 498–505, May 1992.

[65] A. S. Cassidy, J. Georgiou, and A. G. Andreou, "Design of silicon brains in the nano-CMOS era: Spiking neurons, learning synapses and neural architecture optimization," *Neural Netw.*, vol. 45, pp. 4–26, Sep. 2013.

[66] W. J. Dally, C. T. Gray, J. Poulton, B. Khailany, J. Wilson, and L. Dennison, "Hardware-enabled artificial intelligence," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2018, pp. 3–6.

[67] P. M. Furth and A. G. Andreou, "Comparing the bit-energy of continuous and discrete signal representations," in *Proc. 4th Workshop Phys. Comput. (PhysComp)*, 1996, pp. 127–133.

[68] R. Sarpeshkar, "Analog versus digital: Extrapolating from electronics to neurobiology.," *Neural Comput.*, vol. 10, no. 7, pp. 1601–1638, Oct. 1998.

[69] B. Murmann, D. Bankman, E. Chai, D. Miyashita, and L. Yang, "Mixed-signal circuits for embedded machine-learning applications," in *Proc. 45th Asilomar Conf. Signals, Syst. Comput.*, Feb. 2016, pp. 1341–1345.

[70] T. L. Vogelsong, J. J. Tiemann, and A. J. Steckl, "Charge-domain integrated circuits for signal processing," *IEEE J. Solid-State Circuits*, vol. SSC-20, no. 2, pp. 562–570, Apr. 1985.

[71] F. L. J. Sangster and K. Teer, "Bucket-brigade electronics: New possibilities for delay, time-axis conversion, and scanning," *IEEE J. Solid-State Circuits*, vol. SSC-4, no. 3, pp. 131–136, Jun. 1969.

[72] G. T. Tuttle, S. Fallahi, and A. A. Abidi, "A low-power analog CMOS vector quantizer," in *Proc. Data Compress. Conf. (DCC)*, Mar. 1993, pp. 410–419.

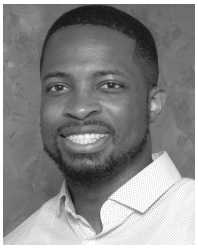
[73] K. Yang and A. G. Andreou, "Multiple input floating-gate MOS differential amplifiers and applications for analog computation," in *Proc. 36th Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 1993, pp. 1212–1216.

[74] K. Yang and A. G. Andreou, "A multiple input differential amplifier based on charge sharing on a floating-gate MOSFET," *Analog Integr. Circuits Signal Process.*, vol. 6, pp. 197–208, Nov. 1994.

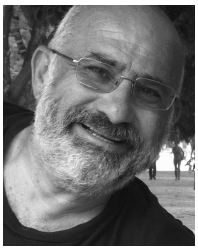
[75] S. Nakamura and Y. Nagazumi, "A matched filter design by charge-domain operations," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 5, pp. 867–874, May 2005.

[76] B. Sadhu, M. Sturm, B. M. Sadler, and R. Harjani, "Analysis and design of a 5 GS/s analog charge-domain FFT for an SDR front-end in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 48, no. 5, pp. 1199–1211, May 2013.

[77] A. J. Agranat, C. F. Neugebauer, R. D. Nelson, and A. Yariv, "The CCD neural processor: A neural network integrated circuit with 65536 programmable analog synapses," *IEEE Trans. Circuits Syst.*, vol. 37, no. 8, pp. 1073–1075, Aug. 1990.
[78] Z. Tang, O. Ishizuka, and H. Matsumoto, "Programmable MOS charge-mode neural circuits," *Electron. Lett.*, vol. 28, no. 22, pp. 2059–2060, Oct. 1992.
[79] R. Genov and G. Cauwenberghs, "Charge-mode parallel architecture for vector-matrix multiplication," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 48, no. 10, pp. 930–936, Oct. 2001.
[80] R. Karakiewicz, R. Genov, and G. Cauwenberghs, "1.1 TMACS/mW fine-grained stochastic resonant charge-recycling array processor," *IEEE Sensors J.*, vol. 12, no. 4, pp. 785–792, Apr. 2012.
[81] G. Tognetti, J. Sengupta, P. O. Pouliquen, and A. G. Andreou, "Characterization of a pseudo-DRAM crossbar computational memory array in 55 nm CMOS," in *Proc. 53rd Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2019, pp. 1–5.
[82] F. O. Folowosele *et al.*, "A switched capacitor implementation of the generalized linear integrate-and-fire neuron," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2009, pp. 2149–2152.
[83] C. Mayr *et al.*, "A biological-realtime neuromorphic system in 28 nm CMOS using low-leakage switched capacitor circuits," Dec. 2014, *arXiv:1412.3233*. [Online]. Available: https://arxiv.org/pdf/1412.3233.pdf
[84] M. Noack, M. Krause, C. Mayr, J. Partzsch, and R. Schueffny, "VLSI implementation of a conductance-based multi-synapse using switched-capacitor circuits," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Feb. 2014, pp. 1–4.
[85] D. Bankman and B. Murmann, "An 8-bit, 16 input, 3.2 pJ/op switched-capacitor dot product circuit in 28-nm FDSOI CMOS," in *Proc. IEEE Asian Conf. Solid-State Circuits*, Nov. 2016, pp. 21–24.
[86] H. Valavi, P. J. Ramadge, E. Nestler, and N. Verma, "A 64-tile 2.4-Mb in-memory-computing CNN accelerator employing charge-domain compute," *IEEE J. Solid-State Circuits*, vol. 54, no. 6, pp. 1789–1799, Jun. 2019.
[87] K. A. Sanni, "Heterogeneous chip multiprocessor: Data representation, mixed-signal processing tiles, and system design," Ph.D. dissertation, Johns Hopkins Univ., Baltimore, MD, USA, 2019.
[88] K. Sanni, T. Figliolia, G. Tognetti, P. O. Pouliquen, and A. G. Andreou, "A charge-based architecture for energy-efficient vector-vector multiplication in 65 nm CMOS," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5.
[89] J. L. McCreary and P. R. Gray, "All-MOS charge redistribution analog-to-digital conversion techniques. I," *IEEE J. Solid-State Circuits*, vol. SSC-10, no. 6, pp. 371–379, Dec. 1975.
[90] R. E. Suarez, P. R. Gray, and D. A. Hodges, "All-MOS charge-redistribution analog-to-digital conversion techniques. II," *IEEE J. Solid-State Circuits*, vol. SSC-10, no. 6, pp. 379–385, Dec. 1975.
[91] B. P. Ginsburg and A. P. Chandrakasan, "500-MS/s 5-bit ADC in 65-nm CMOS with split capacitor array DAC," *IEEE J. Solid-State Circuits*, vol. 42, no. 4, pp. 739–747, Apr. 2007.
[92] K.-P. Pun, L. Sun, and B. Li, "Unit capacitor array based SAR ADC," *Microelectron. Rel.*, vol. 53, no. 3, pp. 505–508, 2013.
[93] P. Harpe, E. Cantatore, and A. van Roermund, "A 10 b/12 b 40 kS/s SAR ADC with data-driven noise reduction achieving up to 10.1 b ENOB at 2.2 fJ/conversion-step," *IEEE J. Solid-State Circuits*, vol. 48, no. 12, pp. 3011–3018, Dec. 2013.
[94] R. Özgün, J. H. Lin, F. Tejada, P. O. Pouliquen, and A. G. Andreou, "A low-power 8-bit SAR ADC for a QCIF image sensor," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2011, pp. 841–844.
[95] N. Verma and A. P. Chandrakasan, "An ultra low energy 12-bit rate-resolution scalable SAR ADC for wireless sensor nodes," *IEEE J. Solid-State Circuits*, vol. 42, no. 6, pp. 1196–1205, Jun. 2007.
[96] E. Culurciello and A. G. Andreou, "An 8-bit 800-μW, 1.23-MS/s successive approximation ADC in SOI CMOS," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 9, pp. 858–861, Sep. 2006.
[97] E. Martens, B. Hershberg, and J. Craninckx, "A 69-dB SNDR 300-MS/s two-time interleaved pipelined SAR ADC in 16-nm CMOS FinFET with capacitive reference stabilization," *IEEE J. Solid-State Circuits*, vol. 53, no. 4, pp. 1161–1171, Apr. 2018.
[98] L. Kull and M. Braendli, "A 24 to 72 GS/s 8 b time-interleaved SAR ADC with 2.0 to 3.3 pJ/conversion and 30 dB SNDR at Nyquist in 14 nm CMOS FinFET," in *IEEE Int Solid-State Circuits Conf. Dig. (ISSCC) Tech. Papers*, Nov. 2018, pp. 1–3.



Kayode A. Sanni (M'15) received the B.S. degree in computer engineering from the University of Maryland, Baltimore (UMBC), in 2012, and the M.S. and Ph.D. degrees from the Johns Hopkins University in 2014 and 2019, respectively. He enrolled in Ph.D. Program in electrical and computer engineering at Johns Hopkins University in 2012. His dissertation research was focused on the design mixed-signal VLSI and system design for heterogeneous chip multiprocessors in 55 nm CMOS and 16 nm FinFET CMOS. He is currently a Principal ASIC Engineer with Northrop Grumman Corporation.



Andreas G. Andreou (F'96) is currently a Professor of electrical and computer engineering, computer science and the Whitaker Biomedical Engineering Institute, Johns Hopkins University. He is the Co-Founder of the Johns Hopkins University Center for Language and Speech Processing. His research focuses on the science and technology of sensory communication and computing machinery; that is abstract and physical computational structures for sensory information processing, pattern analysis, and machine intelligence.

Notable micro-systems achievements over the last 25 years, include a contrast sensitive silicon retina, the first CMOS polarization sensitive imager, silicon rods in standard foundry CMOS for single photon detection, and a large scale mixed analog-digital associative processor for character recognition. Significant algorithmic research contributions include the vocal tract normalization and heteroscedastic linear discriminant analysis for speech recognition, and algorithms for multimodal bio-inspired action recognition.

In 1996, he was elected as an IEEE Fellow, for his contribution in energy efficient sensory microsystems.