Mixed-Signal Spin/Charge Reconfigurable Array for Energy-Aware Compressive Signal Processing

Adrian Tatulian, Soheil Salehi, and Ronald F. DeMara
Department of Electrical and Computer Engineering
University of Central Florida, Orlando, FL 32816-2362
adrian.tatulian@ucf.edu, soheil.salehi@knights.ucf.edu, and ronald.demara@ucf.edu

***Abstract*— Recently, significant attention has been given to hardware realization of Orthogonal Matching Pursuit (OMP) algorithms for signal reconstruction. Some CMOS-only approaches have been proposed in the literature which minimize overheads impacting throughput by exploiting parallelism within OMP techniques. Herein, an approach using hybrid spin-CMOS hardware is presented as a reconfigurable logic fabric utilizing a palette of spintronic and MOS components. The resulting fabric utilizes slice-organized analog blocks providing amplifiers, transistors, capacitors, and low-/high-barrier Magnetic Tunnel Junctions (MTJs) which are configurable to realize OMP's required squaring and square-root operations in analog. Digital functional blocks include 6-input fracturable look up tables, and a spin-based analog-to-digital converter to realize matrix inversion needed by OMP. These functional blocks are connectable via programmable interconnect to a non-volatile crossbar to perform low energy vector-matrix multiplication with reduced area. Simulation results indicate a 5-fold reduction in energy consumption and a 26-fold decrease in area requirement compared to CMOS-only approaches.**

***Keywords*— Reconfigurable Mixed-Signal Processing, FPAA, Configurable Analog Block, Compressive Sensing, Magnetic Tunnel Junction (MTJ), Non-Volatile Memory Crossbar**

I. Introduction

Reconfigurable fabrics are effective solutions for signal processing applications. For instance, Huang *et al.* describes a digital FPGA-based scalable architecture for computing Discrete Cosine Transforms (DCT) in image/video coding applications [1]. Such applications can leverage dynamic partial reconfiguration for zonal coding of the target image, i.e., performing DCT for zones of any size from 1×1 to 8×8, as well as reconfigurability in the precision of DCT coefficients. It was shown that having the ability to reduce the precision of implementation, for example, for high compression ratio video encoding, can lead to significant savings in both power and area by leveraging reconfigurability to match a range of processing requirements using a single chip. Herein, we seek to extend such reconfiguration-based advantages to Compressive Sensing (CS) applications via leveraging mixed-signal processing.

Although digital-only FPGAs are commonly-used to realize general-purpose computation directly in hardware to avoid overheads of software bloat [2, 3], computations involving sensor interfacing and signal processing can generally be more efficiently-solved or more rapidly-approximated in the analog domain [4]. Thus, Field Programmable Analog Arrays (FPAAs) have gained attention as analog counterparts to FPGAs. It has been recently reported that using analog computation can lead to 1000-fold improvements in computational energy efficiency [5]. For example, various ultra-low power realizations of IoT sensing systems such as temperature sensors and heart-rate alarms have been developed using mixed-signal FPAAs [6].

CS is an emerging signal processing approach having good attributes for compatibility with analog computation. CS aims to reconstruct sparse signals, i.e., signals with a small number of non-zero values in some given basis, using sub-Nyquist sampling rates. The sampling and reconstruction computations allow tolerances for approximation and are often compressed when written to memory [7]. Compressive sampling is thereby an effective way to limit energy, storage and data transmission overheads in power-critical systems such as IoT devices [8].

Challenges facing CS are that its algorithms for signal encoding and reconstruction are computationally intensive and require run-time adaptation. Signal reconstruction requires determining the optimal solution to an undetermined system of equations, which is NP-hard [7]. Thus, most commonly-used algorithms seek to achieve an approximate solution to this problem. One such algorithm is Basis Pursuit (BP), which relies on convex optimization to reduce the complexity of the problem while still achieving an accurate solution [7]. An alternate approach is Orthogonal Matching Pursuit (OMP), which greedily determines the optimal solution to the system after a set number of iterations. OMP offers lower complexity than BP at the cost of reduced accuracy [9].

Recently, significant attention has been given to hardware realization of OMP for signal reconstruction, and several approaches have been proposed for minimizing overheads impacting throughput and area by exploiting parallelism and reusing hardware on Xilinx FPGAs [7, 9, 10]. While the optimizations used by these authors have been shown to be effective, further improvements may be obtained by researching hardware approaches to 1) accommodate analog computation, and 2) incorporate spin-based components now commercially-available in conjunction with CMOS devices.

Promising hardware approaches to attain aforementioned objectives are novel hybrid-device circuits leveraging the complementary strengths of CMOS and post-CMOS devices. Specifically, spin-based devices such as Magnetic Tunnel Junctions (MTJs), which are commercially-available as DRAM-replacement memory modules, offer significant benefits such as area and leakage energy reduction. They have been applied to reconfigurable fabrics via spin-based Look Up Tables (LUTs) to mitigate challenges of continued scaling of CMOS technology such as high static power due to leakage,

978-1-7281-1957-1/19/$31.00 ©2019 IEEE

Table 1: Comparison of mixed-signal field-programmable fabrics which are suitable for various signal processing tasks.

Work	Routing Architecture	CAB Elements	CDB Elements	Highlighted Contributions
Wunderlich *et al.* [4]	Manhattan	Operational transconductance amplifiers, transistors, capacitors, MITEs (multiple input translinear elements)	3-input Basic logic element (BLE)	Integrated analog/digital computation
George *et al.* [13]	Manhattan w/ µProc. Cores	Operational transconductance amplifiers, transistors, multipliers	4-input Basic logic element (BLE)	Integrated microprocessor with CABs/CLBs
Choi *et al.* [14]	Separate TCAB/ ALU/CLB arrays	Time configurable analog blocks (TCABs)	4-input programmable LUT	Programmability using TCABs
Schlottmann *et al.* [12]	Crossbar	Operational transconductance amplifiers, transistors	N/A	Dynamically reconfigurable FPAA
M-FPAA (proposed herein)	Crossbar	Amplifiers, transistors, capacitors, low-/high-barrier MTJs	6-input Fracturable C-LUT	Spin-based FPAA with NVM crossbar for CS applications

volatility of configuration bitstreams, and the inherently low logic density incurred by the large footprint of SRAM cells. Thus, spin-based devices provide opportunities to significantly reduce energy consumption of reprogrammable fabrics [2]. Moreover, spintronic Analog-to-Digital Converters (ADCs) can offer significant improvements in static power consumption and performance, as compared to CMOS-only designs [11]. Finally, use of spin-based stochastic oscillators for random number generation, as necessary for implementation of CS algorithms, can lead to benefits including quality of randomness, energy, and area, when compared to CMOS-based Linear Feedback Shift Registers (LFSRs) to generate pseudo-randomness [8].

In this work, the OMP algorithm is considered to determine power consumption/area/performance on a spin-device enhanced Mixed-signal FPAA (M-FPAA). M-FPAA comparisons are drawn with the same algorithm implemented on a CMOS FPGA. Section II of the paper provides further background and previous works relating to mixed-signal hardware, CS theory, the OMP algorithm, and spin-based devices. Section III presents the M-FPAA reconfigurable fabric architecture and Section IV describes the CS implementation on the proposed hardware, after which Section V provides simulation results while Section VI concludes the paper.

II. BACKGROUND AND RELATED WORKS

A. Mixed-Signal Reconfigurable Hardware Approaches

Due to the analog nature of real-world signals, the feasibility of utilizing spin-based devices has renewed interest in performing analog and digital computation concurrently within the same reconfigurable fabric. Previously, Schlottmann and Hasler [12] noted that two hurdles have slowed the broader use of analog computation: the need for programmability and the lack of robust design tools. A groundbreaking development in FPAAs was their Reconfigurable Analog Signal Processor (RASP) which provided an avenue for programmability of analog devices, and then further augmented via an integrated set of high-level tools for system-level analog design.

Since then, there has been continued development of mixed-signal arrays, encompassing both digital and analog computation. Several approaches to this problem have been explored in the literature as listed in Table 1. Wunderlich *et al.* [4] presented a Field-Programmable Mixed Array (FPMA) which interleaves analog and digital elements in a Manhattan-routable fabric. Their architecture was comprised of a network of computational analog blocks (CABs) and computational logic blocks (CLBs), interwoven via a global interconnect. The CLBs were comprised of LUTs and D Flip-Flops (D-FFs) while the CABs were comprised of elements such as capacitors, transistors, and op-amps. Additionally, each block contained a local interconnect consisting of a set of reconfigurable switches.

George *et al.* [13] proposed a similar architecture which also integrated a 16-bit microprocessor for added computational capability, thus enabling a 1,000-fold improvement in energy efficiency in addition to a 100-fold decrease in die area compared to the digital equivalent. Finally, Choi *et al.* [14] proposed an architecture which consisted of three separate arrays of CLBs, Arithmetic Logic Units (ALUs) and Time-domain Configurable Analog Blocks (TCABs), with a network of "gluing blocks" interfacing the arrays with one another as well as external input/output. TCABs allow for dynamic reconfigurability of the analog function being implemented, in contrast to CABs which only allow for reconfigurability of interconnects.

While problems in signal processing can readily benefit from analog computation, Pyle *et al.* [15] further explored the possibility of analog computation of mathematical functions, specifically, the square, square root, cube, and cube root functions. Pyle's approach was to use a Self-Scaling Genetic

Algorithm 1 Orthogonal Matching Pursuit

Inputs: The measurement matrix, $\mathbf{\Phi}$
The measurement vector, $\mathbf{y}$
The signal sparsity, k

Output: The signal vector, $\mathbf{x}$

Procedure:

1) Initialize the residual $\mathbf{r_0} = \mathbf{y}$, the index set $\Lambda_0 = \emptyset$, the column set $\mathbf{\Phi'_0} = \mathbf{0}$, and the counter $i = 1$.

while $i < k$ **do**

2) Find the index λ_i most correlated with the measurement matrix, $\mathbf{\Phi}$ by solving the optimization problem: $\lambda_i = \arg\max_{j=1,\ldots,N} |\langle \mathbf{r}_{i-1}, \phi_j \rangle|$.

3) Add λ_i to the index set: $\Lambda_i = \Lambda_{i-1} \cup \{\lambda_i\}$.

4) Augment the column set $\mathbf{\Phi'_i}$ by ϕ_{λ_i}: $\mathbf{\Phi'_i} = [\mathbf{\Phi'_{i-1}}, \phi_{\lambda_i}]$.

5) Solve a least squares problem to determine the updated solution for the signal, $\mathbf{x}_i$: $\mathbf{x}_i = \arg\min_{\mathbf{x}} \|\mathbf{y} - \Phi_i \mathbf{x}\|$.

6) Update the residual: $\mathbf{r}_i = \mathbf{y} - \Phi_i \mathbf{x}_i$.

7) Increment the counter, i.

end while

Algorithm (SSGA) to scale the function parameters to an acceptable range, at which point the computations were performed on an analog fabric and refined through a process of Differential Digital Correction (DDC), using the Cypress PSoC-5LP chip [15]. This approach was later extended to more generalized mathematical functions by Thangavel *et al.* [16] by extending these functions for Puiseux series generalization accommodating negative and fractional exponents as power series algebraic expansions.

B. Compressive Sensing (CS)

In CS, the problem is to determine an optimal solution to the problem $\boldsymbol{y} = \boldsymbol{\Phi x}$ where $\boldsymbol{y} \epsilon \mathbb{R}^M$ is the measurement vector, $\boldsymbol{\Phi} \epsilon \mathbb{R}^{M \times N}$ is the measurement matrix, and $\boldsymbol{x} \epsilon \mathbb{R}^N$ is the signal vector. The signal is said to be *k*-sparse if it has no more than *k* non-zero entries and the ratio *k/N* defines the signal sparsity rate. Moreover, the number of measurements used is significantly less than the length of the signal, i.e., $M \ll N$. As a result, these equations define an undetermined system with infinitely many solutions. The optimal solution is simply the one with lowest sparsity rate, i.e., the solution to the following minimization problem: $\hat{\mathbf{x}} = \text{argmin} \|\boldsymbol{x}\|_0$ s.t. $\boldsymbol{y} = \boldsymbol{\Phi x}$. However, this problem has been shown to be NP-hard and is therefore not practical [7]. Thus, signal reconstruction is more commonly achieved by solving the basis pursuit problem [8] for $\hat{\mathbf{x}} = \text{argmin} \|\boldsymbol{x}\|_1$ s.t. $\boldsymbol{y} = \boldsymbol{\Phi x}$, which reconstructs the original signal if $\boldsymbol{\Phi}$ satisfies the *Restricted Isometry Property (RIP)*, i.e., if for any *k*-sparse vector $\boldsymbol{x}$ then $\|\boldsymbol{x}\|_p (1-\delta) \leq \|\boldsymbol{\Phi x}\|_p \leq \|\boldsymbol{x}\|_p (1+\delta)$ for some specified *p*.

C. Orthogonal Matching Pursuit (OMP)

OMP is a greedy algorithm with lower computational complexity than basis pursuit and is an alternative approach to CS reconstruction [9]. OMP begins by choosing the column of $\boldsymbol{\Phi}$ that has the highest correlation with the residual of measurement vectors, which is initially set to $\boldsymbol{y}$. It then computes a new residual by subtracting the contribution from this column and computes an estimate of the original signal. After *k* iterations, the algorithm returns the final reconstructed signal [9, 10]. The specific steps of the OMP procedure are presented in Algorithm 1 as listed herein.

There have been previous digital-only implementations of the OMP algorithm with hardware support. Septimus and Steinberg [7] were among the first to propose such an implementation. Their approach was to use an array of multipliers to accomplish the set of vector-matrix and vector-vector multiplications in *Step 2* of Algorithm 1 in parallel. They made use of the Moore-Penrose pseudo-inverse, defined as $\boldsymbol{\Phi_i}^{\dagger} = (\boldsymbol{\Phi_i}^T \boldsymbol{\Phi_i})^{-1} \boldsymbol{\Phi_i}^T$, whereby the matrix inversion problem in *Step 5* was reduced to that of inverting the symmetric matrix, $\boldsymbol{C} = \boldsymbol{\Phi_i}^T \boldsymbol{\Phi_i}$. This inversion could then be performed in a computationally-efficient way by using the technique of Alternative Cholesky Decomposition to express $\boldsymbol{C}$ in the form $\boldsymbol{C} = \boldsymbol{LDL}^T$, where $\boldsymbol{L}$ is a lower triangular matrix and $\boldsymbol{D}$ is a diagonal matrix. These computations are then performed using the same hardware used for *Step 2*.

Stanislaus and Mohsenin [9] significantly improved the performance of Algorithm 1 by modifying it to use a thresholding process to remove certain columns of $\boldsymbol{\Phi_i}$ based on relative magnitude of the dot product. Their architecture involved separate hardware cores to perform the two optimization problems involved in the algorithm. Rabah *et al.* [10] used the same algorithm and computation approach as [7]; however, they designed a four-stage architecture aimed at maximizing the utilization of parallelism as well as reuse of hardware. Their architecture consisted of 1) inner product and comparator unit, 2) Cholesky inversion unit, 3) residual computation unit, and 4) reconstructed signal computation unit. This approach yielded an improvement in performance for large-signal analysis, compared to previous works. All of the implementations discussed in this section relied on purely-digital computation via Xilinx FPGAs: Virtex-5 components were used in [7] and [9], while Virtex-6 was used in [10].

D. Spintronic devices suitable for reconfigurable fabrics

Spintronic devices, specifically Spin Transfer Torque-based magnetic tunnel junctions (STT-MTJs), have been recently explored by researchers for applications such as nonvolatile memory due to their near-zero power consumption, area efficiency, and fast read operation [17]. STT-MTJs are comprised of two ferromagnetic layers, referred to as the fixed layer and free layer, separated by a thin oxide barrier. A bi-directional current passing through the device can change the polarization of the free layer magnetization and thus flip the device between its parallel (P) state and anti-parallel (AP) state. The device resistance depends on which state the device is in and is higher when the device is in the AP state. Specifically the P-state resistance is given by $R_P = R_{MTJ}$ and the AP-state resistance is given by $R_{AP} = R_{MTJ}(1 + TMR)$, where:

$$R_{MTJ} = \frac{t_{ox}}{Factor \times Area \sqrt{\varphi}} exp(1.025 t_{ox} \sqrt{\varphi}) \quad (1)$$

$$TMR = \frac{TMR_0}{1 + \left(\frac{V_b}{V_h}\right)^2} \quad (2)$$

where *TMR* is tunneling magnetoresistance, t_{ox} is the oxide layer thickness, *Factor* is a material-dependent parameter which depends on the resistance-area product of the device, *Area* is the surface area of the device, φ is the oxide layer energy barrier height, V_b is bias voltage, and V_h is the bias voltage at which *TMR* drops to half of its initial value.

MTJs contribute valuable properties such as non-volatility and stochasticity, allowing them to be suitable for diverse applications. One application is the clockless fracturable 6-input spin-based look-up table (C-LUT) proposed in [2]. The C-LUT's select tree consists of *D* levels of transmission gates, each controlling access to a spin-based memory cell. The memory cells consist of pairs of complementary MTJs for a wide read margin yielding reliable read operation. Furthermore, sensing is accomplished through a voltage divider circuit and a pair of inverters to amplify the signal, which eliminates the need for an external clock or large sense amplifiers. Such a design can be used for combinational logic to implement either one *D*-input Boolean function, or two (*D*-1)-input Boolean functions in parallel. This design yields an 80% reduction on standby power consumption compared to an SRAM-based LUT, which addresses a key challenge faced by CMOS designs.

In addition, the stochastic switching properties of low-energy-barrier MTJs can be used to implement a true random

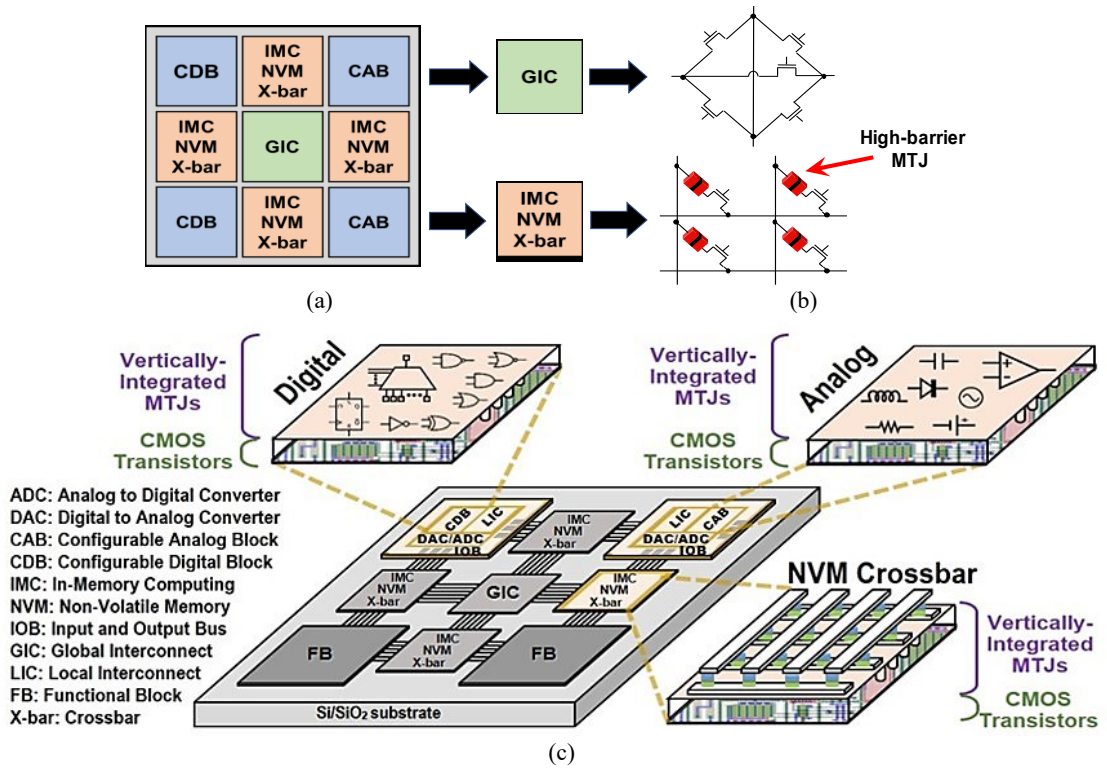

Fig. 1: (a) Single-Slice organization for proposed M-FPAA architecture, (b) M-FPAA routing and switch interconnect design, and (c) Hybrid Spin/Charge device realization as configurable blocks within the M-FPAA fabric.

number generator (TRNG) to generate an adaptive CS measurement matrix [8, 18]. This design is based on a *p-bit*, which divides the supply voltage V_{DD} between an MTJ and NMOS transistor. The MTJ is fabricated to have a low energy barrier (~1kT) between P and AP states, and hence switches due to thermal activation. The p-bit utilizes the voltage in between the two devices, which switches stochastically due to the stochastic switching of the MTJ device. The p-bit output serves as the input to a D flip-flop, which then generates a random *M*-bit stream, where each bit determines one row of the measurement matrix, for random sampling of the input signal. The TRNG used in this design was found to reduce energy consumption per bit by 9-fold on average, compared to state-of-the-art TRNGs, in addition to an average area reduction of 3-fold [8].

To support mixed-signal operation and conversion, an Adaptive Intermittent Quantizer (AIQ) is a suitable spintronic circuit. It utilizes the Voltage-Controlled Magnetic Anisotropy (VCMA) effect to dynamically control MTJ energy barriers to implement an Analog-to-Digital Converter (ADC) featuring dynamic Sampling Rate/Quantization Resolution (SR/QR) tradeoff [17]. In this design, the MTJs are arranged in a resistive-switch-ladder architecture, with the analog signal as input. Dynamically controlling the states of the switches and control over the number of active devices in the circuit allows the architecture to function at various QRs; in addition, use of an asynchronous clock allows the SR to be dynamically set as well. The SR/QR tradeoff is determined by the Signal-to-Noise (SNR) ratio of the input signal, e.g., high SNR favors high QR when sampling. As expected, this technique allows ADC at fixed bit and energy budgets, and results in considerable energy savings overall. Thus, spin-based architectures offer key benefits in power and area consumption when compared to CMOS and are promising candidates for next-generation reconfigurable fabrics.

III. M-FPAA Platform

A. Overview of Architecture

Herein, we investigate a device-level-to-architecture-level approach to integrate front-end signal processing within a low-footprint reconfigurable fabric that enables mixed-signal processing. This approach advances hybrid spin/CMOS *Mixed-signal Field Programmable Analog Arrays (M-FPAAs)*, which enable high-throughput on-chip compressive sensing via established algorithms for signal reconstruction. Mixed-signal techniques combined with in-memory computation geared to the demands of compressive sensing will be combined in a field-programmable and run-time adaptable platform.

The M-FPAA architecture is shown in Fig. 1. As shown, we describe a circuit and register-level design so that an M-FPAA slice acquires analog signals and then performs machine learning tasks via In-Memory Computing (IMC) using reduced precision/dynamic range. IMC approaches extend related works, such as Rabah's architecture [10] consisting of separate processing elements (PEs) and memory elements (MEs). The proposed architecture develops analog computable memories, or analog computing arrays, where instead of storing the analog values to be used by external computing elements, IMC is utilized. This cross-cutting beyond von Neumann architecture explores the use of dense emerging *Non-Volatile Memory (NVM)* arrays to perform *Vector Matrix Multiplication (VMM)* necessary for execution of CS signal reconstruction algorithms such as OMP.

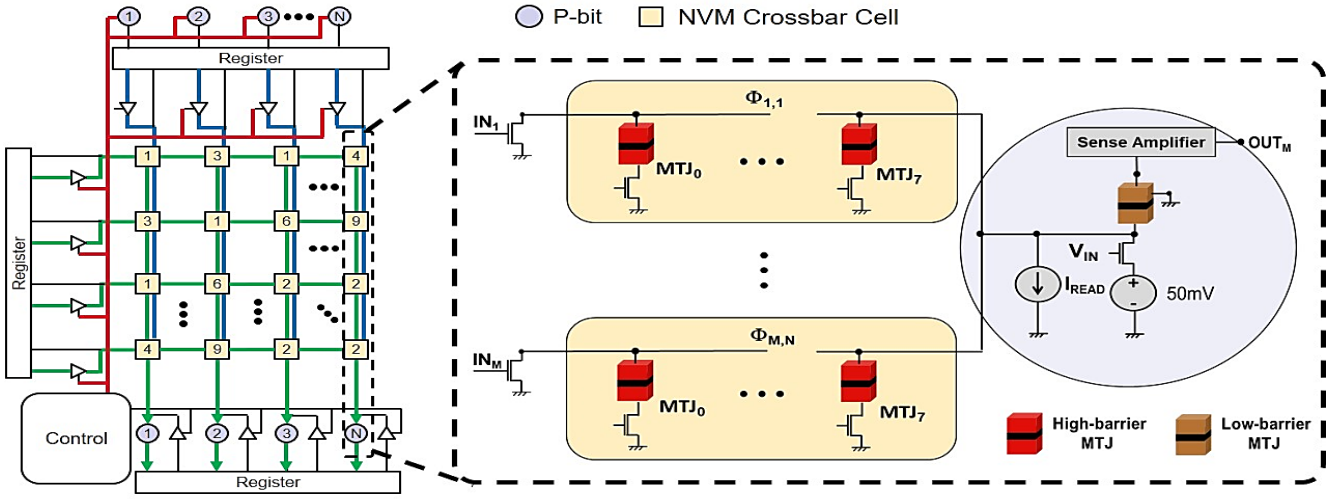

Fig. 2: M-FPAA NVM Crossbar consisting of 1 MTJ per cell for In-Memory Computing, where red signals show the configuration flow, the blue signals depict the path for populating the measurement matrix and green signals illustrate the path for VMM operation.

Low energy barrier MTJs are used as compact TRNGs for generation of the CS measurement matrix, as justified within previously-published work [8]. Our proposed M-FPAA is composed of two types of Functional Blocks (FBs): *Configurable Digital Blocks (CDBs)* and *Configurable Analog Blocks (CABs)*, similar to CABs and CLBs used in previous CMOS-based FPMAs [4, 13]. These FBs are connected via the embedded NVM Crossbar Arrays which perform VMM. Furthermore, within the CDBs the recently-published MTJ-based Look-Up Table (LUT) [2] is used to implement Boolean functions via IMC. Additionally, hybrid spin-CMOS ADCs [11] are used within CABs.

Thus, MTJs are investigated for selected processing roles to simultaneously reduce area and energy requirements while providing stochasticity and non-volatility needed by the OMP algorithm. M-FPAAs can advance a unified platform on a single die accommodating a continuum of information conversion losses and costs targeting compressive sensing applications. Design of such a mixed-signal reconfigurable fabric can enable feasible hardware approaches that can execute CS algorithms more efficiently than digital FPGA-based or CPU-based implementations, which can then be extended to low-energy miniaturization for IoT sensing applications.

B. NVM Crossbar

The proposed M-FPAA architecture utilizes a 50×50 global interconnect crossbar (GIC) as well as 50×50 NVM crossbar arrays connecting the analog and digital blocks. The NVM crossbar arrays consist of deterministic bit cells, along with probabilistic low-energy barrier p-bits to realize energy- and area-efficient implementation of CS applications.

As previously mentioned, p-bits enable true random number generation based on thermally unstable MTJs. In this design, the probabilistic behavior of the device is tunable. Our approach requires just a single p-bit and a D-FF to quantize the output to a 1 or 0. Whereas the tunable stochastic voltage range of p-bits is only ±50mV, a current-summation approach is used to perform the matrix multiplication of the input vector with the weight matrix that corresponds to the measurement matrix of the CS algorithm. By utilizing a collection of programmable resistive elements for each weight with a fixed read current, we can tune the voltage applied to a p-bit, which in turn adjusts the probability of reading a 1 or 0. Therein, an MTJ device with a high energy barrier, such as 40kT, maintains the CS matrix data in a non-volatile manner, as shown in Figure 2.

The M-FPAA crossbar operates by applying inputs to either the rows or columns and reading the resulting node states, which allows the M-FPAA to efficiently realize CS applications. Figure 2 depicts a possible implementation of the NVM Crossbar. MTJs are the targeted devices for adjusting the voltage applied to the input of the output p-bit device given a fixed current. According to detailed analysis, a write voltage with ±50mV range can provide the desired probabilistic switching behavior. The positive and negative voltage range is achieved through connecting one of the write terminals to a fixed voltage of 50mV, while the other terminal can alter from 0V to $V_{IN\text{-}MAX}$ = 100mV. The read current, I_{READ}, is defined based on the size of the array, as elaborated in Equation 3:

$$I_{READ} = \frac{V_{IN-MAX} \times Number\ of\ Array\ Rows}{R_{MTJ}} \qquad (3)$$

where R_{MTJ} is the MTJ resistance in the anti-parallel state, and $V_{IN\text{-}MAX}$ is the maximum input voltage allowed to ensure the designed probabilistic behavior for the p-bit device. The total power consumption of the array during the read process can be calculated using Equation 4:

$$P_{READ} = I_{READ} \times V_{DD} \times Num.of\ Array\ Columns \qquad (4)$$

Within this array, the input voltage range only depends on the TMR value of the MTJ, as expressed by Equation 5:

$$\frac{V_{IN-MAX}}{1+TMR} < V_{IN} < V_{IN-MAX} \qquad (5)$$

so that the total read energy consumption of the array is determined by $E_{READ} = P_{READ} \times T_{SW}$ where T_{SW} is the switching time of the p-bit device, which is on the order of 10ps based on simulation results. However, T_{SW} is lower than the time required for MOS transistor switching, thus our energy consumption is limited by the circuit clock frequency.

C. CDB Architecture

Figure 3(a) shows the proposed CDB design, similar to the architecture proposed by Wunderlich *et al.* [4]. Each CDB takes *N* inputs and *M* outputs; for CS applications, a choice of N=50 and M=25 would be a suitable choice of values. The building block of the CDB is the C-LUT, described earlier in Section II. As shown in Figure 3(b), each fracturable C-LUT can provide two 5-input Boolean logic function or one 6-input function. Consequently, each C-LUT contains $2^6 = 64$ memory cells. The CDB is able to interface with the analog inputs/outputs of the NVM Crossbar through analog-digital and digital-analog conversion. Herein, the aforementioned spin-based AIQ is used for signal conversion while the C-LUT is configured to realize a LUT-based encoder as shown in Figure 3(b) [17]. The latter transforms the output of the AIQ ADC into a suitable binary representation for OMP's matrix inversion step.

D. CAB Architecture

The proposed CAB design is shown in Figure 4. The CAB elements include 4 Operational Transconductance Amplifiers (OTA), 4 PMOS/NMOS transistors, 4 capacitors, and both high energy barrier and low energy barrier MTJs. The CAB utilizes local interconnect dimensions of 50×25. Local routing interconnects are programmed to configure CABs to implement analog computing functions such as calculating square/square root, which is used during least squares minimization of OMP, as depicted in Figure 4(b) which is described later in detail.

IV. Fabric-Based Compressive Sensing (CS) Realization

As outlined in Section II, Compressive Sensing (CS) requires a measurement matrix, $\boldsymbol{\Phi}$, which multiplies the signal vector $\boldsymbol{x}$ to yield the compressed measurement vector, $\boldsymbol{y}$. Often the signal vector will contain a region of interest (RoI) sampled at a higher rate than the rest of the signal. To accomplish this, the columns in $\boldsymbol{\Phi}$ which coincide with the RoI should have a higher concentration of nonzero elements than the other columns. As proposed by Salehi *et al.* [8] the measurement matrix can be generated using a spin-based crossbar architecture as shown in Figure 2. In this approach, p-bits located at the top of each column are used to populate their respective columns. The input voltages to the p-bit at each column allows for tunable stochasticity of the output which can be utilized to generate the CS measurement matrix adaptively according to the signal characteristics such as noise, sparsity rate, and region of interest. The p-bit enables a tunable TRNG, in which higher input voltage yields a higher probability of nonzero values being generated. The p-bit output is amplified via a CMOS inverter and fed into a power-gated D-FF to generate a digital output string, and these values are written into the measurement matrix row-by-row, i.e., one row per clock cycle. As shown in Figure 2, the red lines show the configuration flow, the blue lines depict the path for populating the measurement matrix and the green lines illustrate the path for the VMM operation.

After the measurement matrix is generated, and values are stored in the NVM array, Algorithm 1 is used for signal reconstruction. Several key operations involved in carrying out the algorithm can be implemented directly on the NVM array. These include VMM, maximization/minimization, matrix inverse, and matrix transpose. The NVM array allows for VMM in the usual way with input vector fed in along the rows and output vectors read along the bottom columns. At the edge of the array, the p-bit devices read the outputs, which can then be readily maximized/minimized using a winner-take-all/loser-take-all approach, consistent with the OMP algorithm. Calculation of matrix transpose then amounts to replacing data in the NVM Crossbar which can be achieved by reprogramming the array using the lowermost element in Figure 4(a).

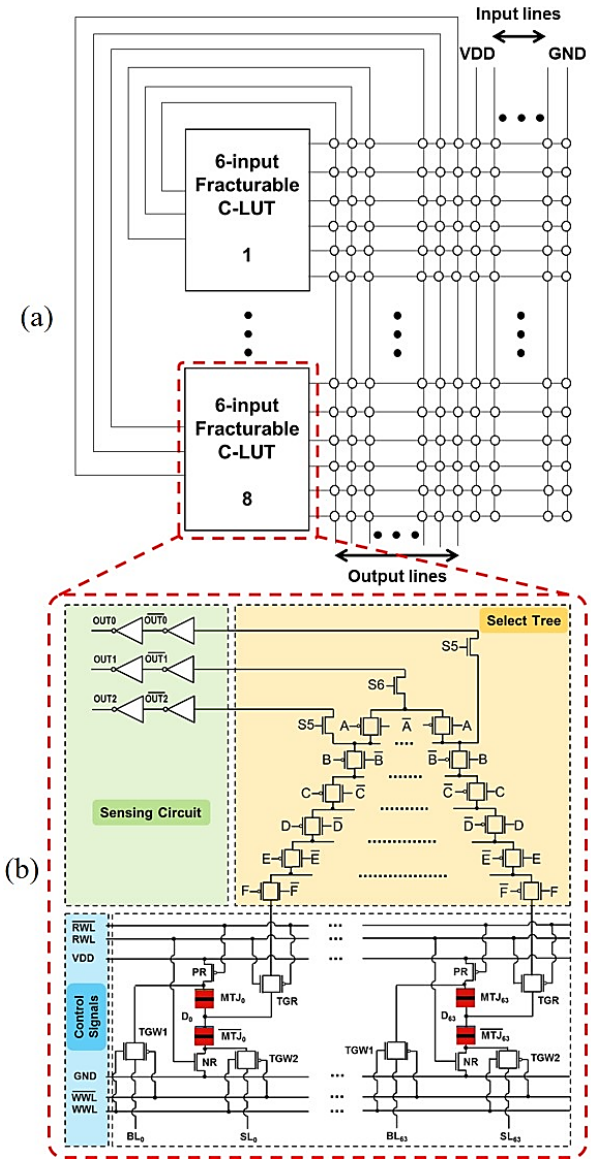

Fig. 3: (a) M-FPAA CDB structure and (b) C-LUT circuit components utilized for CDB logic select/retrieval [2].

In addition to the above-mentioned operations, performing least-squares minimization, i.e., *Step 5* of Algorithm 1, requires calculation of vector norm and the matrix inverse. Calculating the norm of a vector requires the use of square/square root operations which can be efficiently implemented in analog. Squaring requires direct use of an analog multiplier, having its two inputs ganged together. Calculation of square is accomplished via the circuit shown in Figure 4(b). This proposed CAB has all elements necessary to implement these circuits as shown in Figure 4.

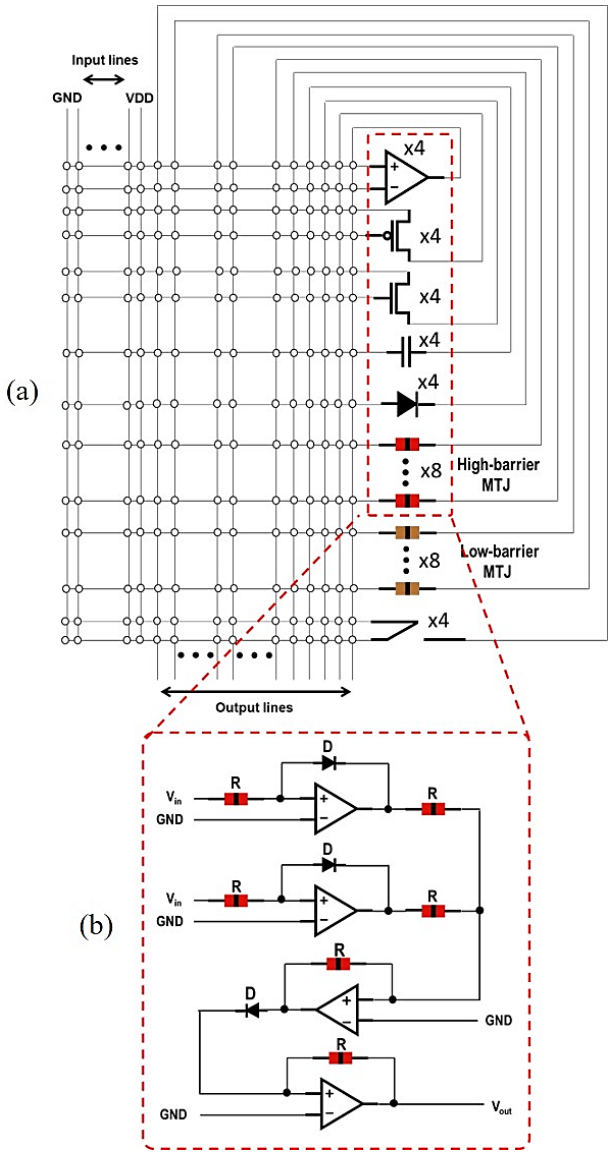

Fig. 4: (a) M-FPAA CAB structure and (b) configuration of an analog multiplier circuit using CAB elements.

Finally, matrix inversion operations are accomplished using the Moore-Penrose pseudo-inverse, which reduces the problem to that of inverting a symmetric matrix as mentioned in Section II, and thus it is performed using Alternative Cholesky decomposition. As Septimus and Steinberg pointed out [7], this process can be accomplished digitally by using 32-bit multipliers combined with multiplexers, and thus readily accomplished in the M-FPAA fabric using sufficient CDBs.

V. Simulation Results

We utilized the HSPICE circuit simulator to validate the functionality of the C-LUT using the 14nm HP-FinFET Predictive Technology Model (PTM) libraries, the STT-MRAM model developed by Kim *et al.* in [19], the VCMA-STT-MRAM model developed by Kang *et al.* in [20], and the p-bit model developed by Camsari *et al.* in [21] to validate the functionality of the CDB and CAB elements used in our proposed M-FPAA. Previous hardware-based CS implementations have included stochastic CMOS [22] and hybrid CMOS-memristor designs [23], as well as CMOS FPGAs for signal reconstruction [7, 9, 10]. For instance, reconstruction time using a CMOS FPGA was found to be 24 μs in comparison to 68ms using a CPU implementation and 37.6ms on a GPU [7]. However, CMOS-based designs suffer from significant area and leakage power overheads, as well as limited quality of randomness from linear feedback shift registers (LFSRs), in comparison to emerging device TRNG approaches [8].

Table 2: Comparison of energy needed for VMM in CMOS Crossbar vs. proposed NVM Crossbar.

Array Size	CMOS X-bar Energy	NVM X-bar Energy	Energy Improvement
100×25	1,177 pJ	240 pJ	~5X
200×50	4,708 pJ	968 pJ	~4.8X
400×100	18,832 pJ	3840 pJ	~4.9X

To estimate the energy reduction of our approach over a pure-CMOS approach, we consider the necessary CMOS elements required to implement a 100×25 single-cycle parallel weighted sum operation using 8-bit weights, which is comparable to the computation performed within the analog array of a 100×25 matrix. Each weight would require eight SRAM cells to store the 8-bit weight as well as eight AND gates and eight 1-bit Full Adders to multiply the input bit with the weight. This yields a total of 20,000 SRAM cells consuming 1,050pJ in-total [24], along with 20,000 Full Adders consuming 106pJ [25, 26] in aggregate, and 20,000 AND gates consuming roughly 21pJ collectively. Thus, a grand total of 1,177pJ per operation is consumed by the CMOS-only design, which is roughly 5-fold more energy for computation than in the proposed M-FPAA's NVM Crossbar. Additionally, a spin-based approach offers non-volatility, as opposed to volatile SRAM cells. Moreover, the CMOS-only approach requires 640,287 transistors, while our approach utilizes just 20,000 MTJ devices each having an access transistor, which achieves a ~26-fold device reduction contributing considerable area savings per the results listed in Table 2.

Simulation results indicate that the average read energy consumption of the C-LUT is 21.9fJ while the write energy consumption of the C-LUT is 155.2fJ. Additionally, according to the results, the C-LUT achieves more than 80% standby power consumption reduction while providing around 25% reduced area footprint compared to a CMOS-based LUT. Moreover, the p-bit TRNG only consumes 0.23fJ for generating each random output bit. Additionally, the area of the p-bit TRNG is 0.4μm^2. Finally, the AIQ ADC consumes 1pJ per sample on average while eliminating the need for an external Flash memory or latch to store the data after each sampling operation due to the non-volatile nature of the MTJ devices.

Furthermore, the OMP algorithm involves calculating norms of vectors of length M. This operation includes M squaring operations and one square root operation. In order for the squaring operations to be performed in parallel, M analog multipliers are required. For instance, considering $M = 25$, and 1 analog multiplier per CAB, 25 CABs are required for this task. Moreover, in the approach taken by Septimus and Steinberg for matrix inversion operation [6], four parallel

multipliers are utilized. Each C-LUT accommodates 6 inputs and 1 output, thus, each multiplier occupies 6 C-LUTs. Considering 8 C-LUTs per each CDB, the matrix inversion operation requires 4 multipliers, which occupies 3 CDBs. Table 1 lists relevant measures for comparable approaches previously proposed in the literature versus the platform developed herein, including Schlottmann *et al.* [27], and others.

VI. Conclusion

The M-FPAA developed herein provides a palette of analog and digital functional blocks sufficient to realize adaptive sampling and quantization rate based compressive sensing algorithms within a compact and reduced-energy reconfigurable fabric. Each CAB within the M-FPAA fabric can realize 1 analog multiplier/square unit. Meanwhile, each CDB can realize eight 6-input fracturable LUTs sufficient to implement matrix inversion. Finally, the NVM Crossbar performs energy- and area-sparing vector-matrix multiplication in analog. Simulation results with 14nm CMOS and STT-based 2-terminal spintronic device libraries indicate that M-FPAAs can offer a promising pathway towards new classes of mixed-signal computation. Specifically, the intrinsic computational strengths of specific post-CMOS devices are leveraged via hybrid analog/digital processing within a reconfigurable fabric.

Acknowledgements

This work was supported in part by the Center for Probabilistic Spin Logic for Low-Energy Boolean and Non-Boolean Computing (CAPSL), one of the Nanoelectronic Computing Research (nCORE) Centers as task 2759.006, a Semiconductor Research Corporation (SRC) program sponsored by the NSF through CCF-1739635, and by NSF through ECCS-1810256.

References

[1] J. Huang, M. Parris, J. Lee, and R. F. Demara, "Scalable FPGA-based architecture for DCT computation using dynamic partial reconfiguration," ACM Transactions on Embedded Computing Systems, vol. 9, no. 1, p. 9, 2009.

[2] S. Salehi, R. Zand, and R. F. DeMara, "Clockless Spin-based Look-Up Tables with Wide Read Margin," in Proceedings of the 2019 on Great Lakes Symposium on VLSI, 2019: ACM, pp. 363-366.

[3] R. S. Oreifej, C. A. Sharma, and R. F. DeMara, "Expediting GA-Based Evolution Using Group Testing Techniques for Reconfigurable Hardware," in *Proceedings of the International Conference on Reconfigurable Computing and FPGAs*, San Luis Potosi, Mexico, Sept. 20–22, 2006.

[4] R. B. Wunderlich, F. Adil, and P. Hasler, "Floating gate-based field programmable mixed-signal array," IEEE Transactions on Very Large Scale Integration Systems, vol. 21, no. 8, pp. 1496-1505, 2012.

[5] Y. Huang, "Hybrid Analog-Digital Co-Processing for Scientific Computation," Columbia University, 2018.

[6] B. Rumberg and D. W. Graham, "A low-power field-programmable analog array for wireless sensing," in Sixteenth International Symposium on Quality Electronic Design, 2015: IEEE, pp. 542-546.

[7] A. Septimus and R. Steinberg, "Compressive sampling hardware reconstruction," in Proceedings of 2010 IEEE International Symposium on Circuits and Systems, 2010: IEEE, pp. 3316-3319.

[8] S. Salehi, A. Zaeemzadeh, A. Tatulian, N. Rahnavard, and R. F. DeMara, "MRAM-based Stochastic Oscillators for Adaptive Non-Uniform Sampling of Sparse Signals in IoT Applications," in Symposium on VLSI Circuits, 2019.

[9] J. L. Stanislaus and T. Mohsenin, "Low-complexity FPGA implementation of compressive sensing reconstruction," in 2013 International Conference on Computing, Networking and Communications (ICNC), 2013: IEEE, pp. 671-675.

[10] H. Rabah, A. Amira, B. K. Mohanty, S. Almaadeed, and P. K. Meher, "FPGA implementation of orthogonal matching pursuit for compressive sensing reconstruction," IEEE Transactions on very large scale integration Systems, vol. 23, no. 10, pp. 2209-2220, 2014.

[11] S. Salehi and R. F. DeMara, "SLIM-ADC: Spin-based Logic-In-Memory Analog to Digital Converter leveraging SHE-enabled Domain Wall Motion devices," Microelectronics Journal, vol. 81, pp. 137-143, 2018.

[12] C. Schlottmann and P. Hasler, "FPAA empowering cooperative analog-digital signal processing," in 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012: IEEE, pp. 5301-5304.

[13] S. George et al., "A programmable and configurable mixed-mode FPAA SoC," IEEE Transactions on Very Large Scale Integration Systems, vol. 24, no. 6, pp. 2253-2261, 2016.

[14] Y. Choi, Y. Lee, S.-H. Baek, S.-J. Lee, and J. Kim, "CHIMERA: A Field-Programmable Mixed-Signal IC With Time-Domain Configurable Analog Blocks," IEEE Journal of Solid-State Circuits, vol. 53, no. 2, pp. 431-444, 2017.

[15] S. D. Pyle, V. Thangavel, S. M. Williams, and R. F. DeMara, "Self-Scaling Evolution of analog computation circuits with digital accuracy refinement," in 2015 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), 2015: IEEE, pp. 1-8.

[16] V. Thangavel, Z.-X. Song, and R. F. DeMara, "Intrinsic evolution of truncated Puiseux series on a mixed-signal field-programmable soc," IEEE Access, vol. 4, pp. 2863-2872, 2016.

[17] S. Salehi, M. B. Mashhadi, A. Zaeemzadeh, N. Rahnavard, and R. F. DeMara, "Energy-Aware Adaptive Rate and Resolution Sampling of Spectrally Sparse Signals Leveraging VCMA-MTJ Devices," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 8, no. 4, pp. 679-692, 2018.

[18] S. Salehi, R. Zand, A. Zaeemzadeh, N. Rahnavard, and R. F. DeMara, "AQuRate: MRAM-based Stochastic Oscillator for Adaptive Quantization Rate Sampling of Sparse Signals," in Proceedings of the 2019 on Great Lakes Symposium on VLSI, 2019: ACM, pp. 359-362.

[19] J. Kim, A. Chen, B. Behin-Aein, S. Kumar, J.-P. Wang, and C. H. Kim, "A technology-agnostic MTJ SPICE model with user-defined dimensions for STT-MRAM scalability studies," in 2015 IEEE custom integrated circuits conference (CICC), 2015: IEEE, pp. 1-4.

[20] W. Kang, Y. Ran, Y. Zhang, W. Lv, and W. Zhao, "Modeling and exploration of the voltage-controlled magnetic anisotropy effect for the next-generation low-power and high-speed MRAM applications," IEEE Transactions on Nanotechnology, vol. 16, no. 3, pp. 387-395, 2017.

[21] K. Y. Camsari, S. Salahuddin, and S. Datta, "Implementing p-bits with embedded MTJ," IEEE Elec. Dev. Letters, vol. 38, no. 12, 2017.

[22] Y. Oike and A. El Gamal, "CMOS image sensor with per-column ΣΔ ADC and programmable compressed sensing," IEEE Journal of Solid-State Circuits, vol. 48, no. 1, pp. 318-328, 2012.

[23] F. Qian, Y. Gong, G. Huang, K. Ahi, M. Anwar, and L. Wang, "A memristor-based compressive sensing architecture," in 2016 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), 2016: IEEE, pp. 109-114.

[24] A. Biswas and A. P. Chandrakasan, "A 0.36 V 128Kb 6T SRAM with energy-efficient dynamic body-biasing and output data prediction in 28nm FDSOI," in ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference, 2016: IEEE, pp. 433-436.

[25] I. Hassoune, D. Flandre, and I. O'Connor, "ULPFA: A new efficient design of a power-aware full adder," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 8, pp. 2066-2074, 2008.

[26] A. T. Mahani and P. Keshavarzian, "A novel energy-efficient and high speed full adder using CNTFET," Microelectronics Journal, vol. 61, pp. 79-88, 2017.

[27] C. R. Schlottmann, S. Shapero, S. Nease, and P. Hasler, "A digitally enhanced dynamically reconfigurable analog platform for low-power signal processing," IEEE Journal of Solid-State Circuits, vol. 47, no. 9, pp. 2174-2184, 2012.