This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JXCDC.2020.2983450, IEEE Journal on Exploratory Solid-State Computational Devices and Circuits

Short-Term Long-Term Compute-In-Memory Architecture: A Hybrid Spin/CMOS Approach Supporting Intrinsic Consolidation

Shadi Sheikhfaal, *Student Member, IEEE*, and Ronald F. DeMara, *Senior Member, IEEE*

***Abstract*—Biological memory structures impart enormous retention capacity while automatically providing vital functions for chronological information management and update resolution of domain and episodic knowledge. A crucial requirement for hardware realization of such cortical operations found in biology is to first design both Short-Term Memory (STM) and Long-Term Memory (LTM). Herein, these memory features are realized via a beyond-CMOS based learning approach derived from the repeated input information and retrieval of the encoded data. We first propose a new binary STM-LTM architecture with composite synapse of Spin Hall Effect-driven Magnetic Tunnel Junction (SHE-MTJ) and capacitive memory bit-cell to mimic the behavior of biological synapses. This STM-LTM platform realizes the memory potentiation through a continual update process using STM-to-LTM transfer, which is applied to Neural Networks based on the established capacitive crossbar. We then propose a hardware-enabled and customized STM-LTM transition algorithm for the platform considering the real hardware parameters. We validate the functionality of the design using SPICE simulations that show the proposed synapse has the potential of reaching ~30.2pJ energy consumption for STM-to-LTM transfer and 65pJ during STM programming. We further analyze the correlation between energy, array size, and STM-to-LTM threshold utilizing the MNIST dataset.**

***Index Terms*— Long-Term Short-Term Memory, SHE-MTJ, Capacitive Crossbar, Beyond-CMOS devices, Compute in Memory.**

I. Introduction

Neuromorphic computing offers potential advantages to various applications including high performance, robust learning capabilities, and a more efficient intrinsically-executed approach to processing. Such a computing paradigm is not limited to the separation of memory and processing, and has a high level of parallelism unlike conventional von Neumann architectures [1]. With the significant growth in neuromorphic computing research, various biologically inspired architectures and synaptic learning rules, such as Spike Time Dependent Plasticity (STDP), have been proposed [2]. However, there are still important but underexplored concepts motivated from biology, which can be emulated to improve neuromorphic designs in terms of performance and reliability. One vital example is the realization of biologically-inspired mechanisms of memory. Biological memory systems are extremely complex entities, constantly responding to a vast amount of dynamic multi-modal information. Collection and integration of temporal information is one of the fundamental parts of this system, which consists of two main storage mechanisms: Short-Term memory (STM) and Long-Term Memory (LTM) [3].

Fig. 1 shows a simplified representation of a biological memory, which consists of three different memory models. The

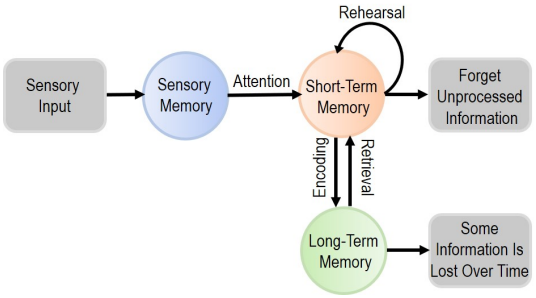

Fig. 1. The Schematic of biological multistore memory model.

sensory memory retains immediate information from the environment and is considered as the first stage of the memory, lasting only for a few milliseconds. This mechanism helps the brain to regulate the flow to avoid a flood of information. However, this information can be transferred to STM through detection and enforcement of temporal focus, once a selected stimulus has been cognitively perceived [4]. The STM can span on the order of seconds to minutes, during the interval when biological brains initiate memory formation via their molecular and cellular machinery. However, retention of information in STM can only be sustained by repeated stimulus. Repeated stimulation of synaptic structures increases the probability of STM to LTM transformation, a process termed consolidation [5]. Under requisite conditions, STM is transitioned to LTM, depending on the strength of molecular reactions and encoding. Thus, the LTM can last from months to years or become permanent, despite the attenuation which would occur otherwise without continuous stimulation [5].

From a hardware implementation perspective, emerging electronic devices can offer a viable way to mimic several plasticity measurements observed in biological synapses as opposed to conventional complementary metal-oxide semiconductor (CMOS) circuits [6]. Memristors with resistive coupling have been widely exploited to implement synapses in addition to the integrate-and-fire capability of a McCulloch–Pitts model neuron [7]. However, since the accessible signal gain and endurance in such fully-memristive networks are limited, other resistive paradigms such as spintronic devices have been taken into consideration [8]. There are a variety of hybrid arrangements of device technologies that can exploit alternative mechanisms, such as capacitive synapses used in place of resistive coupling, which feature an ultra-small static power dissipation [9-13]. In [11], a capacitive neural network has been proposed that utilizes a charge-based capacitor crossbar to perform multiply-and-accumulate (MAC) operation. Such designs realize the weighted summation of inputs through capacitive coupling and voltage division and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JXCDC.2020.2983450, IEEE Journal on Exploratory Solid-State Computational Devices and Circuits

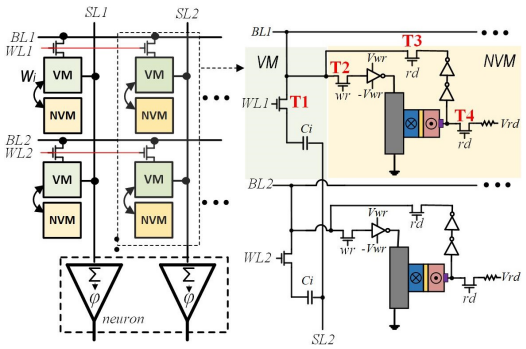

Fig. 2. The proposed STM-LTM memory architecture with VM and NVM components.

generates the output in a read-like operation. Nevertheless, most of the research to realize synapse plasticity change in response to neuron spiking trains has been so far limited to long-term plasticity [14-16], while the volatility of biological memory has been overlooked.

In [17] and [18] the authors show the functional resemblance of two different emerging devices to the short-term to long-term memory transition. In [18] the authors demonstrate that stimulating a memristor device with repeated voltage pulses can result in an effect analogous to memory transition in biological systems. A similar approach has been taken in [17] with a Magnetic Tunnel Junction (MTJ), where a sufficient input stimulus can change its magnetization. Both of these works have focused on implementing the memory transition process with a single emerging device module. Although a homogenous device technology approaches aim at the same behavior as biological memory, it does not allow data undergoing consolidation to be used in computation until such a transition has completed. Consequently, a mechanism is sought which not only exhibits this behavior of biological memory, but can also utilize the introduced data efficiently. This can be achieved by designing separate modules for STM and LTM in the memory architecture. As in [19], the researchers proposed such a design implemented by two separate spin Hall effect-driven Magnetic Tunnel Junctions (SHE-MTJs), in which the STM synapse potentiates the inputs with a greater probability and forgets at a higher rate than the LTM synapse. However, the biological STM-to-LTM transition process was not addressed in detail nor optimized for efficient processing.

In this work, we propose an energy-efficient and biologically-inspired long-term and short-term memory architecture, to mimic both biological STM and LTM synaptic connections and timing dependencies of the stimuli, via volatile and non-volatile hybrid spin-CMOS devices with respect to the synaptic memory reinforcement. The key contributions of this work are: (1) We propose a new binary STM-LTM platform with composite synapse of SHE-MTJ and a capacitive memory bit-cell to mimic the behavior of biological synapses. Our design realizes the memory potentiation through continual update using STM-to-LTM transfer. (2) We present a hardware-enabled STM-LTM transition algorithm for the platform considering the hardware parameters. (3) We explore the efficiency of the platform running the STM-LTM transition algorithm considering the correlation between energy, array size, and STM-to-LTM threshold.

The remainder of the paper is organized as follows. Section II presents the biologically-inspired STM-LTM architecture and embedded memory operations. Section III delineates the STM-LTM transition algorithm. Section IV details the simulation setup and results. Section V concludes the paper.

II. Biologically Inspired STM-LTM Architecture

The proposed biologically-inspired binary STM-LTM memory architecture, shown in Fig. 2, consists of a 2-D array of memory components leveraging a pair of Volatile Memory (VM) and Non-volatile Memory (NVM) as the memory bit-cell to realize STM and LTM, respectively. The VM utilizes a capacitor, controlled by an access transistor, in a fashion analogous to a DRAM structure. The NVM is designed with a SHE-MTJ [20]. Each memory bit cell is connected to a *Bit-Line (BL)*, *Word-Line (WL)*, and *Source-Line (SL)* managed by the control unit's voltage driver. The *BL* and *WL* are shared amongst the cells within the same row and the *SL* is shared between cells within the same column, as shown in Fig. 2, to allow the architecture operate in three distinct modes as explained in subsection II.B.

A. Memory Units

1) Capacitor as STM: Conventional DRAM is the most abundant, low-cost and simple type of memory offering relatively high speed and density, consisting of one access transistor and one capacitor as the storage element. Recently, several works have explored the potentials of such capacitor-based memories in neural network applications [11, 21]. Training neural networks to high degrees of accuracy requires consecutive, small changes in weights, which NVMs are not ideal for them due to limited speed and endurance. Thus, DRAM offers a suitable mechanism for online (in situ) training due to its relatively high speed and symmetrical read/write with infinite endurance, which is a critical aspect for networks that necessitate constant training in an extended period such as IoT edge devices [12, 22].

In digital capacitor-based accelerators [21, 23], every memory bit-line can perform bitwise digital Boolean logic operations, where each capacitor stores a binary synaptic weight and so a low-bit-width and parallel computation has been realized. These accelerators typically do not require large peripheral circuits such as ADC, DAC, and router contrary to resistive NVM accelerators [14]. Recently, the analog capacitive cross-bar networks have been demonstrated greatly-reduced static power dissipation to near-zero levels compared with the weighted sum of currents in a resistively coupled network [12, 22]. However, for such networks, the volatility of the capacitor can be a huge disadvantage as it will require the training to start over upon losing power. Thus, leakage and the resulting volatility will increase energy consumption while processing delay can be less than or equal to the total training time.

Here, we aim to implement a capacitive crossbar enhanced with a non-volatile memory in a new fashion based on the STM-LTM features inspired from biology. Each memory bit-cell's capacitor represents a binary synaptic weight ('1' or '0') stored as the "charged" or "discharged" capacitor states. The STM's

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JXCDC.2020.2983450, IEEE Journal on Exploratory Solid-State Computational Devices and Circuits

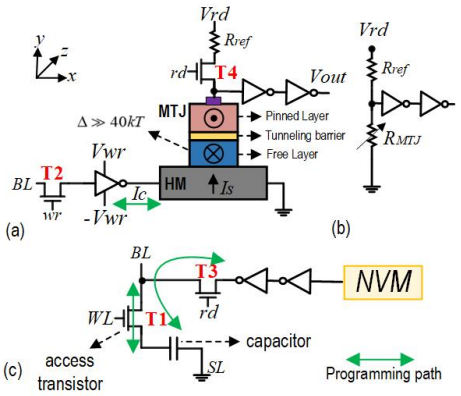

Fig. 3. (a) Structure of a SHE-MTJ as NVM, (b) Resistive equivalent read circuit of SHE-MTJ, (c) VM structure programming path.

access transistor (T1 in Fig. 3(c)) is controlled by *WL* enabling selective write/read operation on the cells located within one row. Storing the network weights in the STM (through a write operation) and strengthening the memory (through STM-to-LTM transfer) are two crucial tasks that need to be carried out. For both operations, the capacitor is initially in the Precharged State (P.S.), i.e. the *BL* voltage is preset to $\sim\frac{VDD}{2}$ by the voltage driver. To save a weight on a capacitor as tabulated in Table 1, the memory decoder first activates the corresponding *WL* and the *BL* is set to high (VDD) or low voltage (GND). This will provide enough bias voltage to change the capacitor data in a DRAM fashion. The synaptic weight representing STM will be then used to perform the computation or STM-to-LTM transfer.

Table 1: The operation modes of the STM-LTM architecture

Operation	*BL*	*WL*	*SL*	*wr*	*rd*
STM Write (1 or 0)	V_{DD} or 0	V_{DD}	0	0	0
Computation	V_{neuron}	V_{DD}	I_{sum}	0	0
STM to LTM	V_{DD} /2	V_{DD}	0	V_{DD}	0
LTM to STM	V_{DD} /2	V_{DD}	0	0	V_{DD}

2) SHE-MTJ as LTM: The NVM element in the STM-LTM memory architecture is a spintronic device named SHE-MTJ that uses a stable nanomagnet (Δ>>40kT), with two CMOS inverters to amplify the output, as shown in Fig. 3(a). A SHE-MTJ is a 3-terminal device, with isolated paths for write and read operations with lower switching energy compared to STT-MTJs. It consists of a Heavy Metal (HM) nanowire beneath an MTJ with two ferromagnetic layers, called the *pinned* and *free* layers, separated by a thin oxide barrier [24]. The MTJ free layer has two different magnetization orientations, called *parallel (P)* and *antiparallel (AP)*, that provide two different levels of resistance for this device. The HM can be made of β-tungsten (β-W) or β-tantalum (β-Ta) [20] with different electrical characteristics. Due to the higher positive Spin Hall angle achieved with tungsten [20], we modeled our device with this material. In order to store the data in the SHE-MTJ, the free-layer magnetization should be manipulated. This is accomplished by injecting a charge current (I_c) to HM in the +x (/ −x) direction as shown in Fig. 3(a). Due to spin Hall effect, I_c will cause an accumulation of oppositely-directed spin vectors on both surfaces of the HM that then generate a spin current (I_s) and further a Spin-Orbit Torque (SOT) in +y (/ −y) direction. The spin current will change the magnetization configuration of

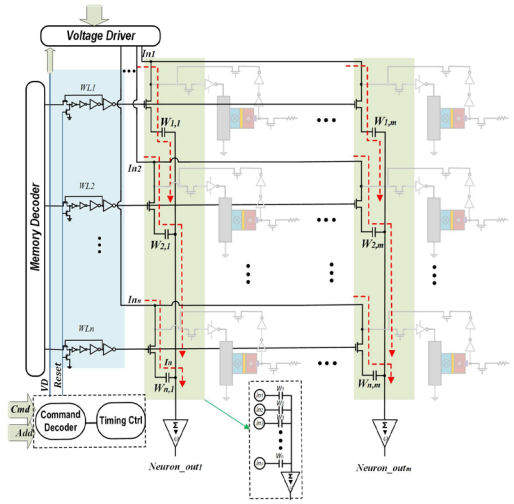

Fig. 4. Realization of the capacitive network [12] within the proposed LTM-STM memory architecture.

the free layer in the ±z direction according to the direction of the charge current [25]. The spin Hall injection efficiency (P_{SHE}) can be expressed as:

$$P_{SHE}=\frac{I_s}{I_c}=\theta_{SH}\frac{A_{FM}}{A_{HM}}\left(1-\operatorname{sech}\left(\frac{t_{HM}}{\lambda_{sf}}\right)\right) \quad (1)$$

where A_{FM} and A_{HM} denote the adjacent free layer area and the cross-sectional area of HM, respectively. In equation (1), θ_{SH} represents the spin Hall angle, as the ratio of generated spin current density to the charge current density. Also, t_{HM} and λ_{sf} denote the thickness of HM substrate and the spin flip length, respectively [26]. Fig. 3 (b) shows an equivalent read circuit of a SHE-MTJ. To read out the data from the SHE-MTJ, a read voltage is applied to sense the resistance of the device through realizing a resistive voltage divider. We have considered 3 access transistors to control the functionality of the SHE-MTJ with respect to our volatile element as shown in Fig. 2. The T3 and T4 transistors are devised to activate the read path and T2 is to control NVM and VM data transfer.

B. Circuit Architecture

1) Computing mode using crossbar operation: In this mode, by activating multiple *WL*s simultaneously (T1 is ON in Fig. 2) and applying input voltages on *BL*s, VMs can modulate the input and realize the weighted summation of inputs using a capacitive voltage divider circuit and send it to the output neuron via SL, while NVM is deactivated (T2-T4 are OFF in Fig. 2). The control signals required for this operation are tabulated in Table 1. The realization of an $n\times m$ capacitive network inspired by [11, 12] is shown in Fig. 4. The memory decoder outputs are enhanced by the inverter chain (blue shaded area) to activate multiple *WL*s simultaneously. The controller governs the timing of the signal going through the crossbar by controlling the memory address and assigning suitable input voltages through the voltage driver. The input signals are encoded as voltage pulse and simultaneously charge the array in each capacitive node. In order to perform MAC operation, by applying the *Vin* as input signal to each row, the charges in capacitors will be redistributed and averaged by a reference capacitance and finally the output voltage can be written as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JXCDC.2020.2983450, IEEE Journal on Exploratory Solid-State Computational Devices and Circuits

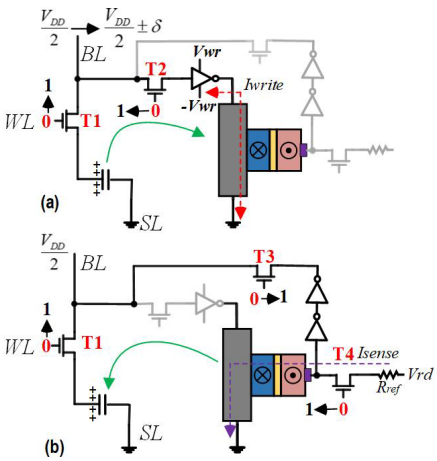

Fig. 5. (a) STM to LTM transfer and (b) LTM to STM transfer modes.

$V_{out} = \frac{\sum_{i=1}^{input} C_{i,j} V_{in,i}}{C_{ref}}$ through voltage division between the cells located in the same column [27].

2) STM to LTM transfer: One of the most significant aspects of memory in biological systems is STM into LTM consolidation after repeated use. To realize this, controller readily keeps the count of input voltages applied to a specific *BL,* which is implemented using a counting unit within the controller. Accordingly, the controller determines the reinforcement ratio of the synapses. As shown in Fig. 5(a), for STM to LTM transfer, at initial state, the *BL* voltage is precharged to $\sim\frac{VDD}{2}$, while *SL* is grounded. Now, activating the *WL* (T1: ON), the selected cell (storing V_{DD} or 0) shares its charge with the *BL* leading to a small deviation in the initial voltage of *BL* ($\frac{VDD}{2}\pm\delta$). Then, by activating the T2 transistor by *wr* signal, the SHE-MTJ's write circuit amplifies the δ of the *BL* voltage toward bipolar write voltage (V_{wr} or $-V_{wr}$) through voltage amplification. It is worth pointing out that *wr* signal is shared among the cells located in the same row and controlled by voltage driver to guarantee the simultaneous STM-to-LTM transfer for synapses connected to one particular neuron. Here the flow of write charge current through the Spin Hall Magnet switches the magnetization through SOT mechanism. If the capacitor is charged-'1' (/discharged-'0'), the SHE-MTJ write terminal is set to $-V_{wr}$ (/V_{wr}) write voltage. This allows adequate charge current to flow from the write circuit output to the ground (/ground to the inverter output), changing the MTJ state to High-R_{AP} (/Low-R_P).

3) LTM to STM transfer: To retrieve the data stored in SHE-MTJ for crossbar computation, an LTM-to-STM mode is considered in the architecture. As shown in Fig. 5(b), for this transfer, the *BL* voltage is first set to $\frac{VDD}{2}$, while *SL* is grounded. Now, activating the *WL* (T1: ON), the resistance states i.e. High-R_{AP} (/Low-R_P) can be readout by a sensing circuit. The controller activates T3 and T4 transistors and a small read voltage is applied on the SHE-MTJ realizing a voltage divider between its resistance state and a fixed reference resistor. The amplified readout data can accordingly charge (/discharge) the bit-cell capacitor with regard to the control signals in Table 1.

Algorithm 1: Memory Transition

Memory Transition Based on the Time Interval: Store the high frequency data in the specified time interval into LTM, retrieve the data from LTM at the end of the refresh interval.
input: *PI*: Pulse Interval, N_{th}: Threshold (STM to LTM)
output: Storing data in LTM or retrieving data to STM

```
Initialization: PI (min), N_th
for i ←0 to i ≤ W_k /*Iterate over all the word lines (W_k)*/
   if (!RI)  /*  capacitive network has not reached a refresh */
      (PI (In_k), N_st) ← ctrl (In_k) /* counts the stimulations*/
      if (PI (In_k) ≤ PI (min) && N_st ≥ N_th)
        M[LTM] ← M[STM]    /*Store data in LTM */
      end if
    else
      M[STM] ← M[LTM]    /*Retrieve data from LTM */
    end if
end for
```

III. STM-LTM TRANSITION

The proposed STM-LTM architecture is optimized to perform two specific tasks. First, the STM-to-LTM transition is realized with timing constrained by the hardware parameters; existing capacitive networks refresh all cells at a rate determined by the leakiest cell in the device, which is typically around 64ms. Second, LTM-to-STM transition is achieved for computing purposes. To efficiently mimic the biological memory, the sub-array controller should actively keep the count of stimuli (inputs-In_k) received at every *BL*. Therefore, we define an STM-to-LTM threshold (N_{th}) that can be readily adjusted for energy and performance tradeoffs. Algorithm 1 indicates the required procedure to accomplish STM-to-LTM transition and LTM-to-STM retrieval based on a defined time interval for the STM-LTM sub-array controller. The algorithm starts iterating on all the sub-array rows storing binary weights (W_k). As long as the capacitive network has not reached a Refresh Interval (*RI*), the controller counts the input data (In_k) applied to each row and then this data is used to analyze the number of stimuli (N_{st}) with regards to a specified Pulse Interval (*PI*). For example, Fig. 6 shows a sample *PI (min)* of 20ns for STM-LTM controller and number of stimuli recorded by it (N_{st}=3) [28]. When N_{st} reaches the preset N_{th}, the STM-to-LTM transition is accomplished for each synaptic weight according to the mechanism explained in Section II.B.2. Therefore, the data will be stored in LTM only when both conditions are met, first the pulse interval of the input is equal or less than the specified minimum pulse interval (*PI (min)*), meaning we are analyzing the data in a specific timeframe and second, the number of stimuli is equal or greater than the specified

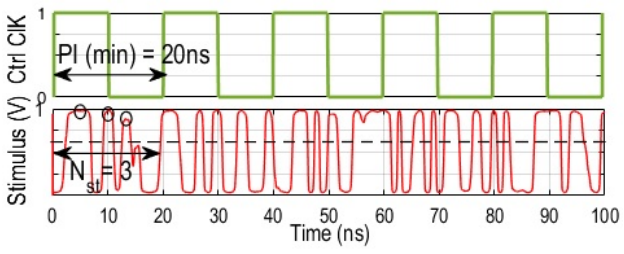

Fig. 6. A sample pulse interval (*PI (min)*) of 20ns and number of stimuli recorded by STM-LTM memory controller. When N_{st} reaches the preset N_{th}, STM-to-LTM transition is accomplished.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JXCDC.2020.2983450, IEEE Journal on Exploratory Solid-State Computational Devices and Circuits

threshold. On frequent stimulations, the STM-to-LTM transfer can be successfully accomplished according to rehearsal (reinforcement) shown in Fig. 1. Additionally, memory decay (forget) is realized by capacitor charge leakage over time. In the last step, upon arrival of the capacitor refresh interval, the data in LTM will be used to retrieve the capacitor's data according to the mechanism explained in Section II.B.3. This data will be later used for crossbar computation.

IV. Simulation Results

A. Evaluation Setup

We developed a bottom-up simulation framework to evaluate the STM-LTM architecture and estimate its energy and performance tradeoffs. We use STM cell parameters from the Rambus power model [29] with access transistor W/L = 90nm/55nm and capacitance 22fF to evaluate the functionality and performance of our design. We modeled the leakage in SPICE considering a capacitor in parallel with a relatively large-value resistor ($R_{leakage}$) and an equivalent resistance in series (R_{ESR}). The SHE-MTJ electrical model was developed in Verilog-A, which incorporates the Landau-Lifshitz-Gilbert (LLG) equation to model the free layer magnetization dynamics and Non-Equilibrium Green's Function (NEGF) to calculate the resistance range (R_P, R_{AP}) with the device simulation parameters tabulated in Table 2. To analyze the VM and NVM modules functionality, we co-designed them in SPICE. Thus, we obtain an analytical approximation to the time-averaged behavior of the full circuit characteristics in 45nm technology node. The controller unit is also simulated by Synopsis Design Compiler [30] with the same technology node. We then modified the NVSIM [31] evaluation tool to report the performance parameters in array-level.

Table 2: SHE-MTJ simulation Parameters

Parameter	Value
MTJ Dimension $W_{MTJ} \times L_{MTJ} \times T_{MTJ}$	$40 \times 120 \times 1.5$ nm^3
SHM Dimension $W_{SHM} \times L_{SHM} \times T_{SHM}$	$120 \times 80 \times 2.8$ nm^3
Demagnetization Factor D_x, D_y, D_z	0.066, 0.911, 0.022
Gilbert Damping Factor, α	0.007
Spin Flip Length, λ_{sf}	1.4 nm
Saturation Magnetization, M_s	850 kA/m
Gyromagnetic Ratio, γ	1.76×10^{11} Am^2/Js
Spin Hall Angle, θ_{SHM}	0.3
Oxide Thickness, t_{ox}	1.3 nm
Energy Barrier, E_a	42 kT
RA Product, RA_p / TMR	22.33 Ω · μm^2 / 187.2%
Resistivity, ρ_β-w	200 μΩ · cm
Supply Voltage	1 V
CMOS Technology	45 nm

B. Results

1) Circuit Design: Figure 7 shows the transient simulation results of moving data ('0' and '1') from STM to LTM. The *BL* is initially precharged to $\sim\frac{VDD}{2}$ prior to turning on the *WL*. In order to transfer the data into the SHE-MTJ, the controller turns on the corresponding *WL* and the *wr* signals, leading to charge sharing between the *BL* and STM's capacitor. The deviation on the *BL* voltage ($\frac{VDD}{2}\pm\delta$) will be then amplified using the write circuit with bipolar write voltage during Sense Amplification state (S.A.) as shown in Fig. 7, to provide the corresponding write voltage for the SHE-MTJ. Such voltage allows sufficient charge current to flow in the SHE-MTJ's write terminals and changes free layer magnetization in *z*-axis from +1 to -1 or vice versa, after ~30ns with our memory configuration. Therefore, the VM data is successfully transferred to NVM.

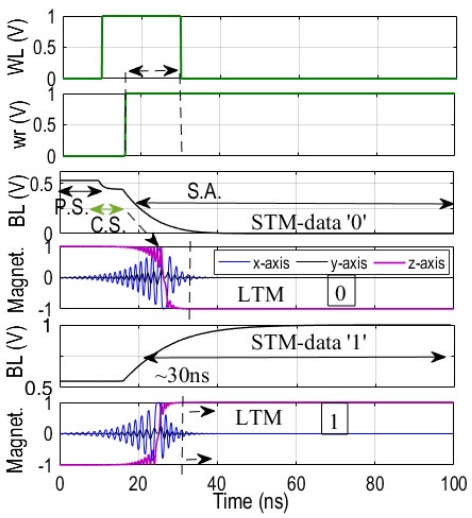

Fig. 7. The transient simulation results of moving data from STM to LTM. Glossary: P.S., C.S., and S.A. stand for Precharged State, Charge Sharing state and Sense Amplification state.

We analyze the STM-to-LTM transition algorithm performance in Section III with the real random inputs from a probabilistic spin logic neuron referred to as a p-bit device [28]. Such activation function is connected to memory *BLs*. We investigate the transient probability from STM to LTM with different parameters. We first increase the N_{th} from 10 to 90 under a constant *PI* (=40ns) plotted in Fig. 8. We observe that by increasing the N_{th} the probability of transferring data from STM to LTM reduces. For example, when N_{th}=10, the transition probability is ~75%. However, N_{th}=60 reduces transition probability to ~17% when a larger threshold is desired. Thus, the threshold can be accurately set w.r.t. the application requirements. We then explore the impact of different *PIs* on STM-to-LTM transition by increasing the expected time from 40ns to 90ns. It can be observed that in a certain N_{th}, by increasing the *PI*, the transition probability will increase.

2) Energy vs. Array Size: In order to compute the energy consumption of the design, we use four different fixed-size capacitive networks (32×32, 64×64, 128×128, and 256×256) leveraging 32, 64, 128 and 256 p-bit output neurons, respectively, to explore the energy consumption of the STM-LTM platform and yield a fair estimate. We analyze the MNIST

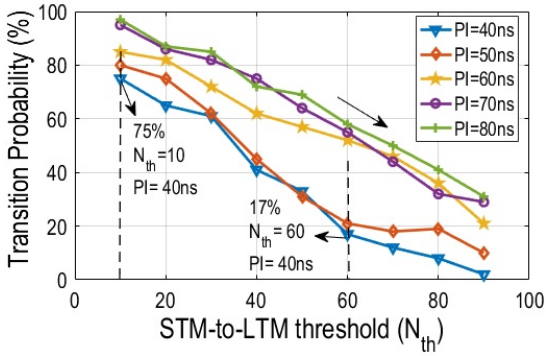

Fig. 8. The transition probability versus STM to LTM threshold under different pulse intervals.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JXCDC.2020.2983450, IEEE Journal on Exploratory Solid-State Computational Devices and Circuits

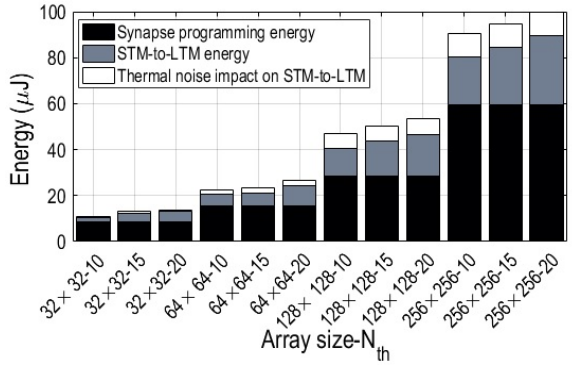

Fig. 9. The breakdown of energy consumption for different array sizes with the impact of thermal noise.

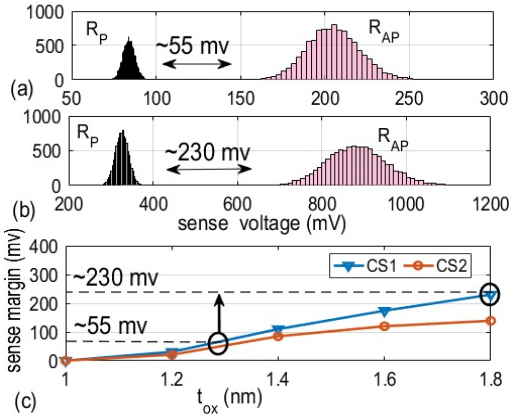

Fig. 10. (a) Monte-Carlo simulation of sense voltage of SHE-MTJ with (a) t_{ox} =1.3nm (b) t_{ox} =1.8nm, (c) Voltage margin of SHE-MTJ vs. thickness of MTJ oxide in two case studies.

data-set of handwritten digits with a two-layer perceptron with a net configuration (784×128 as layer 1 and 128×10 as layer 2) developed in MATLAB. To assess raw performance, we haven't used any optimization algorithm to map the data into the sub-arrays, so the estimation is solely based on the number of used capacitive crossbars whose performance is given through a bottom-up analysis using our simulation platform. We calculated the average programming energy of the network by dividing the energy of network by total time period per epoch for all training images. The average programming energy of 65pJ is achieved per synapse for a 32×32 crossbar. Thus, the power dissipation of 39pW per synapse is incurred by the network for 1500 images over a time period of 1.1msec per epoch. Fig. 9 depicts the programming energy as well as STM-to-LTM transfer energy (including controller counting unit) for different array sizes under three various N_{th}. Our first observation is that by increasing the array size under a fixed N_{th}, a larger programming energy is required and the STM-to-LTM energy increases almost linearly. The second observation is that by increasing N_{th}, the STM-to-LTM energy increases due to redundant counting operations. For example, by changing N_{th} from 10 to 15 in 32×32 array, the STM-to-LTM energy increases by ~1.8x. With Fig. 8 and Fig. 9, the designer can observe the trade-offs between array size, energy, STM-to-LTM transition probability, etc. to adjust system parameters.

3) Process Variation: We modeled the thermal effects on STM-to-LTM transfer by a randomly fluctuating field, H_{noise} on LTM module, with x, y, and z components from a Gaussian distribution with standard deviation $\sqrt{2\alpha K_B T/\gamma M s V \Delta t}$ [32] and zero mean. Here, α denotes Gilbert damping factor, K_B represents Boltzmann's constant, V denotes the volume of free layer, Ms denotes the saturation magnetization, γ is the gyromagnetic ratio, and Δt represents the time step for solving LLG equation [32, 33]. We carried out the Monte-Carlo simulations with 1,000 iterations introducing a Gaussian spread (σ = 5%) in the SHE-MTJ device parameters Ms and α and thermal effects (300K) in the standard deviation. Under the effect of thermal noise, the switching behavior of the SHE-MTJ changes for different samples. Such change has no adverse impact on the transition probability of STM-LTM. Based on our observation, the thermal noise increases the energy budget for STM-to-LTM transfer. This energy consumption overhead after applying thermal noise and device variations is shown in Fig. 9. This comes from the increase in the number of unsuccessful STM-to-LTM transfer.

To assess the variation tolerance of the LTM for different parameters specifically oxide thickness (t_{ox}), we run the Monte-Carlo simulation with 1,000 iterations with 2% Gaussian variation on the Resistance-Area product (RAP) and 5% process variation on the Tunneling-Magnetoresistance Ratio (TMR) and profile the voltage margin between two different resistance level (R_{AP} and R_P), as shown in Fig. 10a. We then increased t_{ox}, from the original 1.3nm to 1.8nm to show how t_{ox} variation impacts the sense margin (Fig. 10b). We observe the same trend experimentally demonstrated in [34], where the increase in the t_{ox} leads to a higher voltage margin that will considerably enhance the reliability of LTM operation. To further explore the impact of t_{ox} variation, we plotted the voltage margin of SHE-MTJ vs. thickness of MTJ oxide from 1nm to 1.8nm in two case studies (CSs). The CS1 is under RAP (2%)-TMR (5%) and CS2 is under RAP (5%)-TMR (5%) variation.

C. Energy/Delay Comparison

Table 3 compares the STM-LTM platform herein with existing designs in terms of technology, applicability and potentials of a single synapse unit. The listed designs use

Table 3: Comparison between STM-LTM architectures

	Sengupta et al. [17]	Srinivasan et al. [19]	Chang et al. [18]	Herein
STM-LTM synapse technology	MTJ	SHE-MTJ/CMOS	Memristor	SHE-MTJ/CMOS
Memory implementation	No	Yes	No	Yes
Separate LTM/STM modules	No	Yes	No	Yes
Compute with STM	No	Yes	No	Yes
Refresh required	No	No	No	Yes
Synapse programming energy (*pJ*)	110	23.7	92.4	65
STM-to-LTM Delay (ns)	~30 on constant stimulation	N/A**	~80 on constant stimulation	~30
STM-to-LTM energy (*pJ*)	165*	N/A**	122.7	~30.2
LTM endurance	$10^{10} - 10^{15}$	$10^{10} - 10^{15}$	$10^{5} - 10^{10}$	$10^{10} - 10^{15}$

* With the 5 input stimulus magnitude of 100μA with 3ns duration

** The STM-to-LTM transfer mechanism is not realized, so the performance cannot be reported.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JXCDC.2020.2983450, IEEE Journal on Exploratory Solid-State Computational Devices and Circuits

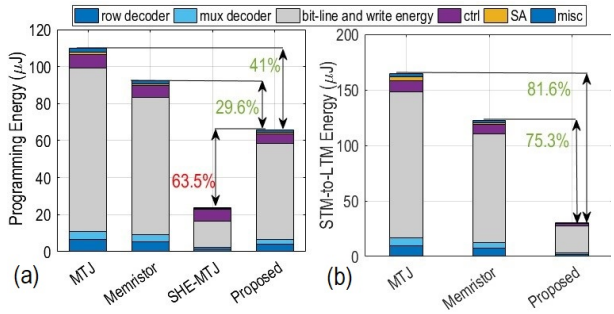

Fig. 11. The breakdown of (a) Synapse programming energy and (b) STM-to-LTM energy reported in Table 3.

different methods to implement the STM-LTM transition so different comparison metrics are appropriate. While the MTJ-based [17] and memristor-based [18] synaptic designs demonstrate a single MTJ and memristor mimicking long-term potentiation according to the magnitude, duration, and frequency of input stimulus, the crucial STM state is only a transient state to get to LTM state and not practically useful. The aforementioned designs do not present any circuit implementation to support utilization of STM during computation. Srinivasan et al. [19] presents a fully-functional binary synaptic element that uses two separate SHE-MTJ driven by a relatively different read voltage to improve the synaptic learning efficiency. Separate modules for LTM and STM provides the design with faster and more reliable functionality. To the best of our knowledge, the SHE-MTJ design in [19] is the only design that proposes a practical STM. However, the biological STM-to-LTM transition process was not addressed in detail nor optimized for efficient processing. Our STM-LTM platform brings a solution to make the STM state even more like biological memory by being practically available in the computation phase. Table 3 compares different designs in terms of a single synapse programming energy and STM-to-LTM energy. We designed a proper write/read circuitry for MTJ- and memristor-based designs to make them comparable. All designs are implemented with 45 nm technology as well. Based on our evaluation, our design herein consumes ~30.2pJ energy for STM-to-LTM (VM-to-NVM) transfer and ~65pJ for programming (of VM) purposes. The proposed design improves the synapse programming energy consumption by ~29.6% and ~41% compared with memristor and MTJ designs, respectively. The SHE-MTJ design in [19] achieves the least synapse programming energy consumption (23.7pJ) between all designs. It should be noted that the STM state in our design still incurs capacitive network refresh power. The design herein improves the STM-to-LTM energy over memristor and MTJ by 75.3% and 81.6%, respectively. From STM-to-LTM transition delay perspective, our design requires ~30ns as depicted in Fig. 7, while memristor and MTJ designs require 80ns and 30ns, respectively, on constant stimulation.

Fig. 11 shows the breakdown of energy consumption for both programming and STM-to-LTM operations, where the colored legend indicates the contribution of each hardware component to the total programming energy. The synapse programming energy can be mainly translated to write energy for different platforms, as shown in Fig 11a. The SHE-MTJ intrinsically requires lower write energy compared to the MTJs and Memristors [26]. From STM-to-LTM transition perspective (Fig. 11b), our design utilizes distinct modules, while the memristor and MTJ-based designs work with consecutive stimulations in the same component leading to a lower STM-to-LTM energy. The two primary influences that impact energy consumption of the proposed STM-LTM design are reading the capacitor's voltage and writing that to the SHE-MTJ.

V. Conclusion

Intrinsic computing capabilities provided by hybrid device technology designs offer novel approaches for realizing biologically-inspired features such as consolidation mechanisms present in STM-LTM. The design proposed herein utilizes distinct modules for STM and LTM to realize a synapse contrary to previous designs. This follows biological principles wherein transfer of information to LTM is facilitated through repeated access while providing faster and more reliable functionality. We then presented a hardware-enabled STM-LTM transition algorithm for the platform considering the real hardware parameters. Our simulations showed the proposed design has the potential of reaching pico-Joule energy level for STM-to-LTM transfer and STM programming.

Acknowledgment

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 National Science Foundation through NSF-CCF-1739635.

References

[1] G. Indiveri *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers in neuroscience,* vol. 5, p. 73, 2011.

[2] N. Caporale and Y. Dan, "Spike timing–dependent plasticity: a Hebbian learning rule," *Annu. Rev. Neurosci.,* vol. 31, pp. 25-46, 2008.

[3] R. Lamprecht and J. LeDoux, "Structural plasticity and memory," *Nature Reviews Neuroscience,* vol. 5, no. 1, pp. 45-54, 2004.

[4] I. Winkler and N. Cowan, "From sensory to long-term memory: evidence from auditory memory reactivation studies," *Experimental psychology,* vol. 52, no. 1, pp. 3-20, 2005.

[5] J. L. McGaugh, "Memory-a century of consolidation," *Science,* vol. 287, no. 5451, pp. 248-251, 2000.

[6] B. L. Jackson *et al.*, "Nanoscale electronic synapses using phase change devices," *ACM Journal on Emerging Technologies in Computing Systems,* vol. 9, no. 2, pp. 1-20, 2013.

[7] G. S. Snider, "Spike-timing-dependent learning in memristive nanodevices," *IEEE international symposium on nanoscale architectures*, 2008, pp. 85-92: IEEE.

[8] A. Sengupta, Z. Al Azim, X. Fong, and K. Roy, "Spin-orbit torque induced spike-timing dependent plasticity," *Applied Physics Letters,* vol. 106, no. 9, p. 093704, 2015.

[9] U. Cilingiroglu, "A purely capacitive synaptic matrix for fixed-weight neural networks," *IEEE transactions on circuits and systems,* vol. 38, no. 2, pp. 210-217, 1991.

[10] W. E. Engeler, "Neural net using capacitive structures connecting input lines and differentially sensed output line pairs," ed: Google Patents, 1993.

[11] D. Kwon and I.-Y. Chung, "Capacitive Neural Network Using Charge-Stored Memory Cells for Pattern Recognition Applications," *IEEE Electron Device Letters,* vol. 41, no. 3, pp. 493-496, 2020.

[12] Z. Wang *et al.*, "Capacitive neural network with neuro-transistors," *Nature communications,* vol. 9, no. 1, pp. 1-10, 2018.

[13] Q. Zheng *et al.*, "Artificial Neural Network Based on Doped HfO 2 Ferroelectric Capacitors With Multilevel Characteristics," *IEEE Electron Device Letters,* vol. 40, no. 8, pp. 1309-1312, 2019.

[14] P. Chi *et al.*, "Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory," *ACM SIGARCH Computer Architecture News,* vol. 44, no. 3, pp. 27-39, 2016.

[15] R. Zand, K. Y. Camsari, S. D. Pyle, I. Ahmed, C. H. Kim, and R. F. DeMara, "Low-energy deep belief networks using intrinsic sigmoidal spintronic-based probabilistic neurons," in *Proceedings of the 2018 on Great Lakes Symposium on VLSI*, 2018, pp. 15-20.

[16] S. D. Pyle, R. Zand, S. Sheikhfaal, and R. F. Demara, "Subthreshold Spintronic Stochastic Spiking Neural Networks With Probabilistic Hebbian Plasticity and Homeostasis," *IEEE Journal on Exploratory Solid-State Computational Devices Circuits,* vol. 5, no. 1, pp. 43-51, 2019.

[17] A. Sengupta and K. Roy, "Short-term plasticity and long-term potentiation in magnetic tunnel junctions: Towards volatile synapses," *Physical Review Applied,* vol. 5, no. 2, p. 024012, 2016.

[18] T. Chang, S.-H. Jo, and W. Lu, "Short-term memory to long-term memory transition in a nanoscale memristor," *ACS nano* vol. 5, no. 9, pp. 7669-7676, 2011.

[19] G. Srinivasan, A. Sengupta, and K. Roy, "Magnetic tunnel junction based long-term short-term stochastic synapse for a spiking neural network with on-chip STDP learning," *Scientific reports,* vol. 6, p. 29545, 2016.

[20] C.-F. Pai, L. Liu, Y. Li, H. Tseng, D. Ralph, and R. Buhrman, "Spin transfer torque devices utilizing the giant spin Hall effect of tungsten," *Applied Physics Letters,* vol. 101, no. 12, p. 122404, 2012.

[21] S. Li, D. Niu, K. T. Malladi, H. Zheng, B. Brennan, and Y. Xie, "Drisa: A dram-based reconfigurable in-situ accelerator," in *2017 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2017, pp. 288-301: IEEE.

[22] Y. Li *et al.*, "Capacitor-based cross-point array for analog neural network with record symmetry and linearity," in *2018 IEEE Symposium on VLSI Technology*, 2018, pp. 25-26: IEEE.

[23] S. Angizi and D. Fan, "ReDRAM: A Reconfigurable Processing-in-DRAM Platform for Accelerating Bulk Bit-Wise Operations," in *2019 (ICCAD)*, 2019, pp. 1-8: IEEE.

[24] L. Liu, T. Moriyama, D. Ralph, and R. Buhrman, "Spin-torque ferromagnetic resonance induced by the spin Hall effect," *Physical review letters,* vol. 106, no. 3, p. 036601, 2011.

[25] A. Roohi, R. Zand, D. Fan, and R. F. DeMara, "Voltage-based concatenatable full adder using spin hall effect switching," *IEEE transactions on computer-aided design of integrated circuits systems,* vol. 36, no. 12, pp. 2134-2138, 2017.

[26] S. Manipatruni, D. E. Nikonov, and I. A. Young, "Energy-delay performance of giant spin Hall effect switching for dense magnetic memory," *Applied Physics Express,* vol. 7, no. 10, p. 103001, 2014.

[27] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 31, no. 7, pp. 994-1007, 2012.

[28] K. Y. Camsari, S. Salahuddin, and S. Datta, "Implementing p-bits with embedded MTJ," *IEEE Electron Device Letters,* vol. 38, no. 12, pp. 1767-1770, 2017.

[29] (2010). *DRAM Power Model*. Available: https://www.rambus.com/energy/

[30] (2011). *Design Compiler, Ncsu eda freepdk45*. Available: https://www.eda.ncsu.edu/wiki/FreePDK45:Contents

[31] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *IEEE TCAD,* vol. 31, no. 7, pp. 994-1007, 2012.

[32] J. Sankey *et al.*, "Mechanisms limiting the coherence time of spontaneous magnetic oscillations driven by dc spin-polarized currents," *Physical review B,* vol. 72, no. 22, p. 224427, 2005.

[33] D. Fan, S. Maji, K. Yogendra, M. Sharad, and K. Roy, "Injection-locked spin hall-induced coupled-oscillators for energy efficient associative computing," *IEEE Transactions on Nanotechnology,* vol. 14, no. 6, pp. 1083-1093, 2015.

[34] S. Yuasa, T. Nagahama, A. Fukushima, Y. Suzuki, and K. Ando, "Giant room-temperature magnetoresistance in single-crystal Fe/MgO/Fe magnetic tunnel junctions," *Nature materials,* vol. 3, no. 12, pp. 868-871, 2004.

Shadi Sheikhfaal (S'19) is currently pursuing her Ph.D. degree in computer engineering at University of Central Florida, Orlando, FL, USA. She received her M.Sc. degree in computer engineering and computer systems architecture from Science and Research Branch of Azad University, Tehran, Iran, in 2014 and her B.Sc. degree in computer engineering from Azad University, Ardebil, Iran, in 2012. Her current research interests include biologically inspired computing, Neuromorphic computing and Spin-based computing.

Ronald F. DeMara (SM'10) received the Ph.D. degree in Computer Engineering from the University of Southern California in 1992. Since 1993, he has been a full-time faculty member at the University of Central Florida where he is a Professor of Electrical and Computer Engineering, and joint faculty of Computer Science, and has served as Associate Chair, ECE Graduate Coordinator, and Computer Engineering Program Coordinator. His research interests are in adaptive computer architectures with emphasis on reconfigurable and post-CMOS devices, evolvable and intelligent hardware, resilient and energy-aware logic design, and the digitization of STEM education. On these topics, he has completed over 300 publications, 49 funded projects as PI or Co-PI including sponsorship of NSF, NASA, Army, Navy, Air Force, DARPA, and NSA, with one patent granted and one provisional patent. He has completed 50 graduates as Ph.D. dissertation or M.S. thesis advisor and was previously an Associate Engineer at IBM and a Research Scientist at NASA Ames, in total for four years. He is an Associate Editor of IEEE Transactions on Emerging Topics in Computing. He has served as Topical Editor of IEEE Transactions on Computers and as Associate Editor of IEEE Transactions on VLSI Systems, Microprocessors and Microsystems, and as Guest Editor of various Transactions, and serves on various IEEE conference program committees including ISVLSI, NVMSA, SSCI, etc. He received the IEEE Joseph M. Bidenbach Outstanding Engineering Educator Award in 2008.