A 65 nm Wireless Image SoC Supporting On-Chip DNN Optimization and Real-Time Computation-Communication Trade-Off via Actor-Critical Neuro-Controller

Ningyuan Cao, *Member, IEEE*, Baibhab Chatterjee, *Member, IEEE*, Jianbo Liu, *Graduate Student Member, IEEE*, Boyang Cheng, *Graduate Student Member, IEEE*, Minxiang Gong, *Graduate Student Member, IEEE*, Muya Chang, *Member, IEEE*, Shreyas Sen, *Senior Member, IEEE*, and Arijit Raychowdhury, *Fellow, IEEE*

***Abstract*— The widespread proliferation of smart sensors has led to hardware that enable edge intelligence (EI) with extreme energy efficiencies. This decreases the volume of data that is transmitted to the cloud, thus reducing: 1) processing latency; 2) communication energy; and 3) network congestion. However, this comes with an added cost of computation at the edge node. The cost (energy/latency) of edge computation and the cost of communication to the cloud vary widely depending on operating conditions, which include: 1) information content in the data; 2) algorithm selection; 3) channel conditions (noise, path-loss, etc.); 4) network size, available bandwidth; and 5) resources at the cloud. This article presents a 65 nm wireless image processing SoC for real-time computation-communication trade-off on resource-constrained edge devices. The test-chip includes: 1) an all-digital, near-memory, reconfigurable, and programmable neural-network (NN)-based systolic image processor; 2) a digitally adaptive radio-frequency digital-to-analog converter (RF-DAC)-based transceiver; and 3) a mixed-signal, time-based, actor-critic (AC) neuro-controller with compute-in-memory (CIM) and in-place weight updates that provide online learning and adaptation for efficiently controlling the computation, communication blocks separately as well as jointly. The major contributions of the proposed SoC are threefold: 1) a wireless Internet of Things (IoT) SoC architecture enabling a generic computation-communication trade-off scheme; 2) a novel CIM circuit design enabling effective AC control and online learning (0.59 pJ/MAC, 0.4 pJ/update); 3) integration of programmable deep NN (DNN) accelerator (1.05 TOPS/W) and reconfigurable transceiver (184 pJ/b @ −15 dBm) supporting versatile cloud-edge collaborations; and 4) significant system-level energy efficiency improvement (5.7×) with real-time on-chip smart control enabled by seamless chip integration and AI-enabled decision-making. Furthermore, this SoC serves as a system-level IoT prototype for next-generation context-aware EI.**

***Index Terms*— Edge intelligence (EI), edge-cloud trade-off, Internet of Things (IoT), wireless system-on-chip.**

I. Introduction

THE interplay between the Internet of Things (IoT) and artificial intelligence (AI) has made great advancement of the smart society. However, due to the constrained energy budget on the edge devices, it remains challenging to support extensive wireless data transmission, especially high-dimensional image, and video data. As a result, computation on the edge has been introduced in recent years to address this challenge [1]–[5]. The *in situ* data processing will greatly reduce the data volume of transmission, thus reducing communication energy. Such efficiency comes from the fact that computation energy is typically orders of magnitude smaller than raw data transmission [6] for conventional data processing algorithms. However, the state-of-art AI algorithms demand deep data processing: even with edge-friendly AI algorithms dedicated designed for embedded applications, such as SqueezeNet [7] and MobileNet [8], the tens of millions of computations per inference are making computation cost non-trivial. Furthermore, in the context of intelligent adaptive wireless transmission [9], [10], data transmission energy and latency are highly dynamic depending on channel noise, path-loss, sensor network size, available bandwidth, server resources, information content in the data, algorithm selection, and so on. Fig. 1 shows the simulated system-level (computation + communication) energy consumption across edge processing depth (PD) and communication cost for both SqueezeNet and MobileNet. The optimal control (denoted in the image) spreads out between full edge computation (PD = 1) and full cloud transmission (PD = 0), and it is a complex function of dynamic communication cost and algorithm selection. As such, it will

Manuscript received October 7, 2021; revised February 6, 2022; accepted March 1, 2022. This article was approved by Associate Editor Kenichi Okada. This work was supported in part by the Semiconductor Research Corporation under Grant 2720.001 and in part by JUMP CBRIC under Grant 2777.006. *(Corresponding author: Ningyuan Cao.)*

Ningyuan Cao, Jianbo Liu, and Boyang Cheng are with the Department of Electrical Engineering, University of Notre Dame, South Bend, IN 46556 USA (e-mail: ncao@nd.edu; jliu32@nd.edu; bcheng4@nd.edu).

Baibhab Chatterjee and Shreyas Sen are with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: bchatte@purdue.edu; shreyas@purdue.edu).

Minxiang Gong, Muya Chang, and Arijit Raychowdhury are with the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 46202 USA (e-mail: mxgong@gatech.edu; mchang87@gatech.edu; arijit.raychowdhury@ece.gatech.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSSC.2022.3159473.

Digital Object Identifier 10.1109/JSSC.2022.3159473

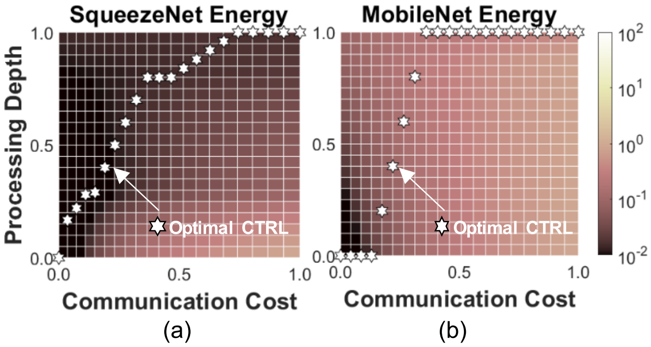


Fig. 1. Normalized systematic energy consumption (edge computation and wireless communication) across edge PD and communication cost for (a) SqueezeNet and (b) MobileNet. Simulation settings: communication cost 0.1–1 nJ/b, TX power −10 to 0 dBm, on-chip computation energy consumption 1 pJ/MAC (8 b), external memory access energy consumption 5 pJ/b.

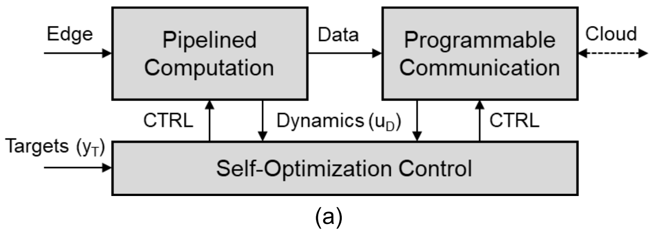
largely reduce systematic energy/delay by tracking the optimal computation-communication trade-off of a wireless edge intelligence (EI) system.

Among all the wireless IoT applications, image processing is among the most demanding use-cases for optimal computation-communication trade-off. On the one hand, image processing is at the core of many important applications, such as surveillance, recognition, behavior analysis, and so on [10]–[12]. On the other hand, the high-dimensional data volume together with extensive computation (deep neural networks (NNs), etc.) have brought about significant challenges to resource-constrained wireless IoT platforms. Both facts have motivated us to investigate chip-level solutions to address various wireless image processing challenges with systematic optimization and state-of-the-art circuit techniques.

This article presents a 65 nm wireless image processing SoC for real-time computation-communication trade-off on resource-constrained edge devices. The test chip includes: 1) an all-digital, near-memory, reconfigurable, and programmable NN-based systolic image processor at 1.05 TOPS/W (peak); 2) a digitally adaptive radio-frequency digital-to-analog converter (RF-DAC)-based transceiver with TX energy efficiency of 768 pJ/b; and 3) a mixed-signal, time-based, actor-critic (AC) neuro-controller with compute-in-memory (CIM) and in-place weight updates that provide online learning and adaptation at 0.59 pJ/MAC for efficiently controlling the computation, communication blocks separately as well as jointly.

II. Design Space Exploration

Conventionally, IoT image processing schemes either directly transmit captured image to the back-end server or process end-to-end algorithms locally without data exchange. As mentioned in Section I, both schemes lack environmental awareness and systematic optimization. The smart wireless image processing scheme proposed is shown in Fig. 2(a). There are three major building modules: pipelined computation, programmable communication, and self-optimization control. Such a system optimizes programmable system targets (y_T) according to dynamic sensed variables (u_D) through various control knobs (CTRL). The detailed variables are denoted in Fig. 2(b). A systematic overview and analysis of the three modules are discussed in this section.



Design Targets (y_T)	Sensed Variables (u_D)	Control Knobs (CTRL)
y_{TE}: Target Energy y_{TD}: Target Latency y_{TB}: Target BER	u_{PL}[1:0]: Path-Loss u_{NP}[1:0]: Noise Power u_{NS}[1:0]: Network Size u_{IC}[1:0]: Information Content	PD[1:0]: Processing Depth PA[2:0]: PA Power Out ECC: Error Correction Code

(b)

Fig. 2. (a) System diagram of self-optimizing wireless image processor. (b) Parameter table with selected design targets, sensed variables, and control knobs for the proposed platform.

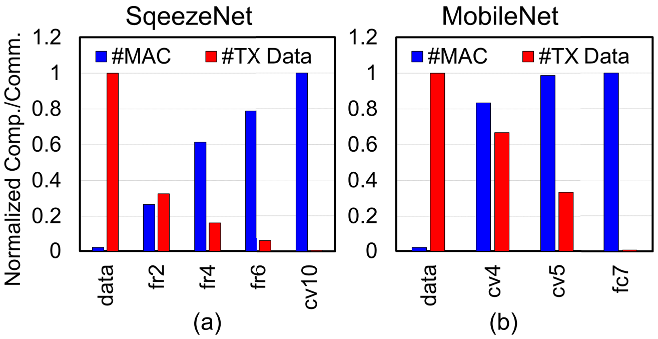


Fig. 3. Normalized computation workload and output data volume across PD for (a) SqueezeNet and (b) MobileNet.

A. DNN Processing Pipeline

Deep NN (DNN) is the state-of-the-art image processing framework and has achieved beyond-human performance in many applications [13]. Compared with shallow multi-layer perception, DNN usually has extensive cascaded/parallel convolution layers to extract features and several fully connected layers at the end to separate feature space [14]. Furthermore, people have looked into pruning techniques to sparsify NNs to maximally reduce computation/storage bottlenecks for embedded systems [15].

To understand edge DNN processing, two widely applied embedded AI DNN networks, SqueezeNet and MobileNet, are analyzed. Fig. 3 shows the output data volume and accumulated number of operations at certain layers in these DNNs. We have observed a monotonically decreased output data volume and monotonically increased computation workload with respect to deeper DNN PDs. It means that the DNN framework is inherently compatible to act as a computation-communication trade-off scheme: shallow PD for edge computation savings and deep PD for data communication savings depending on the dynamic communication cost.

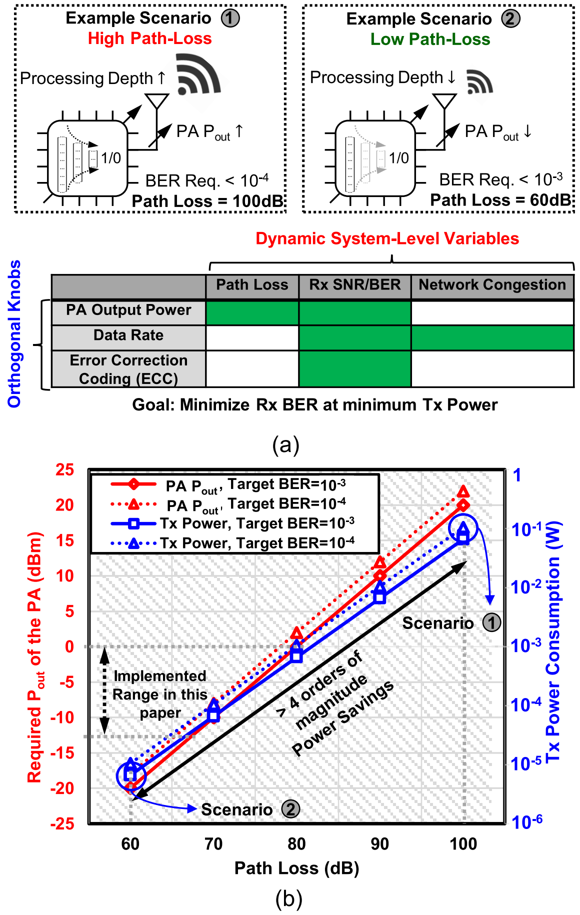
At the same time, each DNN topology has its own computation characteristics with respect to computation workload, data transfer patterns, layer specifications, and so on. The proposed DNN computation pipeline as a processor should feature not only any particular DNN, but also DNNs in general to adapt to wide future use-cases. The optimization should be at least two levels: 1) pipeline should be reconfigurable to account for workload distribution between PDs and maximize local intermediate data utilization across DNNs and 2) the processing element (PE) in the pipeline should as well be able to reconfigure for layerwise optimizations, such as convolution layers, fully connected layers, sparsely connected layers, and so on. The detailed implementation will be discussed in Section III.

B. Programmable Communication

Wireless channels present a highly dynamic environment with respect to path-loss, signal-to-noise ratio (SNR), network size, and so on. To guarantee the accuracy of data communication, transceivers (TRXs) are conventionally designed for the worst case, thereby consuming higher power than that required for specific applications in edge systems. To mitigate the communication energy bottleneck, our previous works extensively explore adaptive transmitters (Txs) and receivers (Rxs) [16]–[22]. By monitoring dynamic wireless channel conditions, the TX and RX control knobs can be tuned accordingly to provide on-the-fly zero-margin performance, thereby preserving energy.

An example of adaptive communication is illustrated in Fig. 4. In scenario 1, the channel suffers from severe path-loss (100 dB) and data accuracy is critical (BER $< 10^{-4}$). Hence, the output power of the final stage of the TX, which is usually a power amplifier (PA), needs to be high, resulting in significant TX power. However, more computation (higher PD) at the node will help in reducing the total amount of bits to be communicated, which can be helpful in reducing the total system power. On the contrary, when channel loss is moderate (60 dB) and transmission data error tolerance is high (BER $< 10^{-3}$) as in scenario 2, the transceiver can save up to 100× TX power by properly lowering PA output power in this example. In this analysis, the modulation is assumed to be quadrature phase shift keying (QPSK), resulting in a received E_b/N_0 requirement of about 7.6 dB for BER $= 10^{-3}$, and about 9 dB for BER $= 10^{-4}$ [23]. The transmitter efficiency is assumed to be 15%, while the data rate is 1 Mb/s. In Fig. 4(a), the dynamic system-level variables (such as path loss, SNR, and network congestion) are shown, and the orthogonal tuning knobs for each variable are also indicated. PA output power should be increased for high path loss and/or low SNR, data rate should be reduced in the case of low SNR and/or network congestion, while error correction coding (ECC) can be implemented for low SNR scenarios. A detailed explanation on the adaptive transceiver system implemented in this article can be found in [24].

In hardware adaptive transceiver design, it is preferred to incorporate more programmable knobs to provide high degree of freedom. Meanwhile, efficient on-chip TRX implementation is highly desired for responsive TX control. The on-chip programmable adaptive transceiver is discussed in Section III.



	Path Loss	Rx SNR/BER	Network Congestion
PA Output Power	🟩	🟩	
Data Rate		🟩	🟩
Error Correction Coding (ECC)		🟩	

(a)

(b)

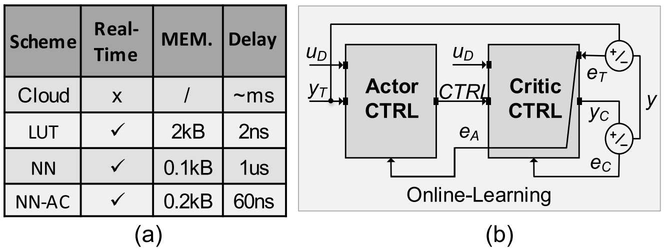
Fig. 4. Adaptive communication example showing that PA output power adapts to path-loss and BER requirements to preserver energy. (a) High path loss and low path loss scenarios, dynamic system-level variables and tuning knobs. (b) Output power requirement and power consumption in the two scenarios.

C. Optimal Control

Besides computation pipeline and programmable communication, it is crucial to optimally control the two modules independently as well as an integrated system. The controller will take design targets and sensed variables as inputs and dynamically choose control knobs as outputs. In a complex environment, both input/output dynamic range and variable size will be large, and it will consequently lead to significant policy search space and expensive real-time optimization. Furthermore, accurate modeling and control will be challenging in a sophisticated environment without online learning. The devices have to be able to calibrate offline trained models and learn in the real environment. It requires thorough investigations into the choice of the control scheme.

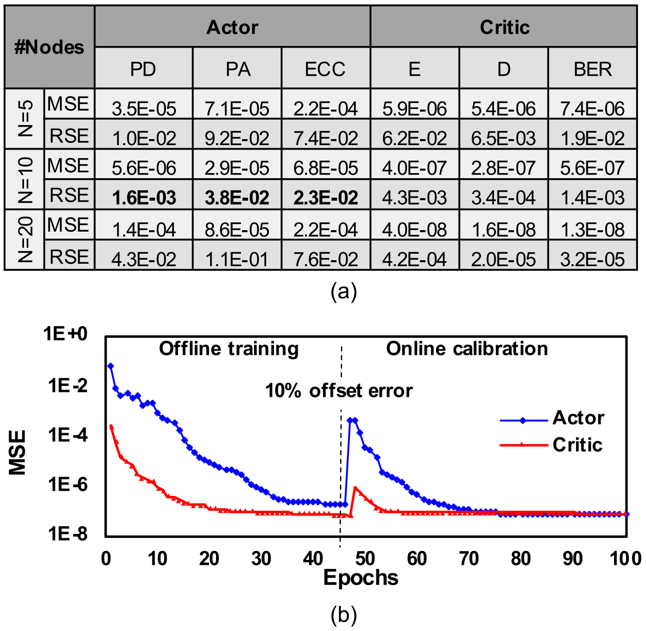
One straightforward solution is to offload control to the cloud. With immense computation resources at the backend, the controller can handle accurate environmental models. However, the latency between local dynamic and remote

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



Scheme	Real-Time	MEM.	Delay
Cloud	x	/	~ms
LUT	✓	2kB	2ns
NN	✓	0.1kB	1us
NN-AC	✓	0.2kB	60ns

Fig. 5. (a) Among different strategies for the proposed control problem. (b) NN AC controller diagram.



#Nodes		Actor			Critic		
		PD	PA	ECC	E	D	BER
N=5	MSE	3.5E-05	7.1E-05	2.2E-04	5.9E-06	5.4E-06	7.4E-06
	RSE	1.0E-02	9.2E-02	7.4E-02	6.2E-02	6.5E-03	1.9E-02
N=10	MSE	5.6E-06	2.9E-05	6.8E-05	4.0E-07	2.8E-07	5.6E-07
	RSE	**1.6E-03**	**3.8E-02**	**2.3E-02**	4.3E-03	3.4E-04	1.4E-03
N=20	MSE	1.4E-04	8.6E-05	2.2E-04	4.0E-08	1.6E-08	1.3E-08
	RSE	4.3E-02	1.1E-01	7.6E-02	4.2E-04	2.0E-05	3.2E-05

Fig. 6. (a) Performance of actor optimal control and critic chip emulation measured with mean square error (MSE) across NN hidden layer nodes. (b) NN-AC convergence for offline training and online calibration of 10% random model parameter offset error.

control could cause severe control errors in adaptive wireless systems. This delay may result in consecutive transmission failures with over-optimistic control choice. Alternatively, we could use an embedded lookup table (LUT) to implement the controller [10] for fast optimal policy indexing. However, besides extensive memory usage, LUT lacks learnability in a realistic environment. We could also choose NNs as an emulator for the platform. However, an exhaustive search is required for emulation-based optimal control. Both energy and delay overheads make it less preferable.

To address all the problems mentioned above, we have chosen NN-based AC control scheme. It has an actor NN and critic NN, one for making decisions and one for system emulation. In run-time, the actor picks optimal control knobs in a single inference time with sensed variables and design targets; and the critic emulates chip performance with sensed variables and selected controls. During training, emulation errors are collected to calibrate the critic NN, while the target errors at the output of the critic controller are back-propagated through the critic controller as control errors to train the actor NN. The AC-controller is able to provide both real-time and learnable optimal control. The control scheme comparison is shown in Fig. 5(a) and data flows are shown in Fig. 5(b).

Before the actual hardware NN-controller implementation, optimal controller network topology and its learning capability (offline training and online calibration) need thorough investigation. We take the following steps to evaluate the NN-controller scheme and observe its online environment learning capability.

1) *Dataset preparation:* Generating random environment dynamics based on prior knowledge of realistic dynamic range (path-loss, channel noise, network size, and so on); emulating chip performance (energy, delay, and bit error rate) across control knobs from measured data.
2) *Offline training:* Exploring optimal configurations of the actor and critic NN via offline training of varying network configurations (number of layers and number of nodes per layer of multi-layer perception).
3) *Online calibration:* Randomly perturbing trained network and model coefficients to emulate offline training error, and observing online calibration capability of the NN-controller.

In Fig. 6(a), it shows the actor network and critic network prediction accuracy across NN configurations. It has been found that the optimal network consists of two-layer and ten hidden nodes per layer. Furthermore, Fig. 6(b) shows that both the actor and the critic controller are able to calibrate 10% offset model errors. The implementation details will be discussed in Section III.

III. Hardware Architecture

The SoC architecture is shown in Fig. 7. There are three major blocks designed for DNN computation pipeline, programmable communication, and optimal control discussed in Section II.

1) *PE spatial array:* A 3-by-3 PE array with reconfigurable interconnections between PEs to account for various DNN architectures. Each PE has eight threads (each thread with an ALU, a 1 kb static random access memory (SRAM), and a shift register). PE is also reconfigurable for optimized layer operations.
2) *Adaptive transceiver:* On-chip digitally reconfigurable channel-aware transceiver with programmable PA gain, data rate, and error correction code mode.
3) *AC controller:* A neuro-based AC controller. Both controllers are 2-layer NN with each layer implemented with a 10-by-10 CIM module.

Besides the major building blocks, the SoC has also included an 8 kb frame buffer to store the input image, a preprocessor to infer frame difference, data/instruction caches to store temporal data/instructions, a scan chain, and a decoder.

The SoC interfaces with a camera, a power supply and management unit, and a programmable interface. The SoC will be remotely connected with the cloud server for data exchange through the on-chip transceiver. Meanwhile, the on-chip memory capacity of wireless SoC is highly constrained by both excessive area usage of RF baseband/antenna

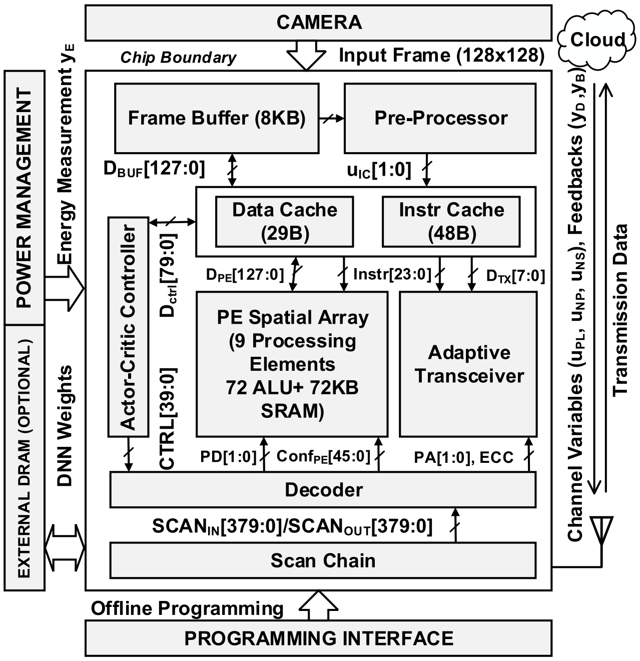


Fig. 7. Proposed wireless image processing SoC architecture.

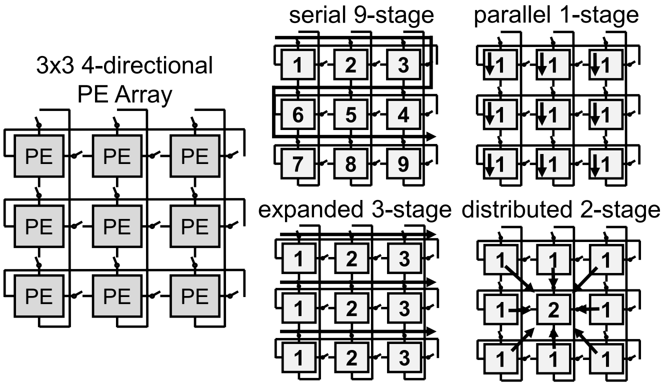


Fig. 8. Circuit diagram of the on-chip reconfigurable PE spatial array and examples of potential DNN computation pipeline configurations.

as well as a conservative selection of tech-node for analog circuit performance. As such, external DRAM will be applied to the SoC whenever the model size exceeds on-chip memory capacity.

A. Reconfigurable PE Spatial Array

The PE spatial array has nine PEs and the PEs are placed in a 3-by-3 configuration as is shown in Fig. 8. Each PE is able to reconfigure its input to any of the outputs of four adjacent PEs. At the same time, each PE can bypass the data so that one PE's output data can directly reach any PEs in the array. By controlling each PE's interconnection and bypass status, the PE array can be easily reconfigured for various pipeline topologies depending on the workload distribution and dataflow pattern. For example, in a deep sequential computation pipeline, the PE array can be reconfigured to support up to nine-stage serial pipelines. On the contrary, if a workload is highly parallel and there is minimal data exchange, the array can also be reconfigured as a fully parallel stage. And it can also form any pipeline between one stage and nine stages as is shown in Fig. 8. It should be noted that the optimal performance gain achieved from the proposed array structure are the ones whose data formats are consistent (e.g., multilayer perception). In modern DNN structure, the data generated in one layer are often unlikely to be directly applied to the next layer without proper data preparation (pooling, convolution with stride, tensor computation order). In such a case, a portion of computation resources and instruction bandwidth is required for pre/post-processing.

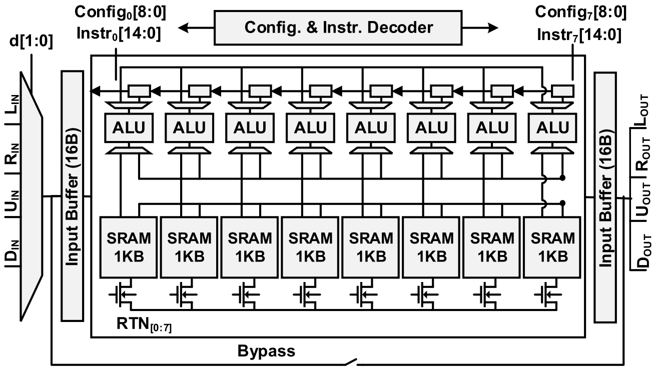


Fig. 9. Circuit diagram of the reconfigurable PE.

PE in the array (shown in Fig. 9) includes eight threads, and each thread consists of the following sub-blocks.

1) *Arithmetic logic unit:* ALU's inputs are connected with two 2-to-1 multiplexers. Input A is able to select between: 1) the data on the global bus at the output of input buffer shared by all threads and 2) the data stored in local shift register. Input B is able to select data between: 1) data on global bus at the output of any particular SRAM in the memory bank and 2) data read from the local SRAM.
2) *Retention-enabled SRAM:* The SRAM output is connected to both the ALU within the same thread and a global bus shared with all SRAM blocks in the PE. Furthermore, to reduce static power consumption of un-accessed SRAM, the PE has full control to put any SRAM blocks into the retention mode.
3) *Shift register:* The shift registers are connected to other shift registers in their neighboring threads. The first shift register is connected to the input buffer. The shift buffer chain will work as an first-in/first-out (FIFO) register array when needed and push input data forward in each cock cycle.

With proper configurations, the PE is able to optimize for various DNN layer types that differ in computation pattern and memory usage. In particular, the PE can be configured to optimize fully connected layers, convolution layers, and sparsely connected layers, which are major build modules in a typical DNN.

Fully connected layer configuration is depicted in Fig. 10(a). All threads work in parallel. The ALU selects one input from the global input bus (input data) and another input from local

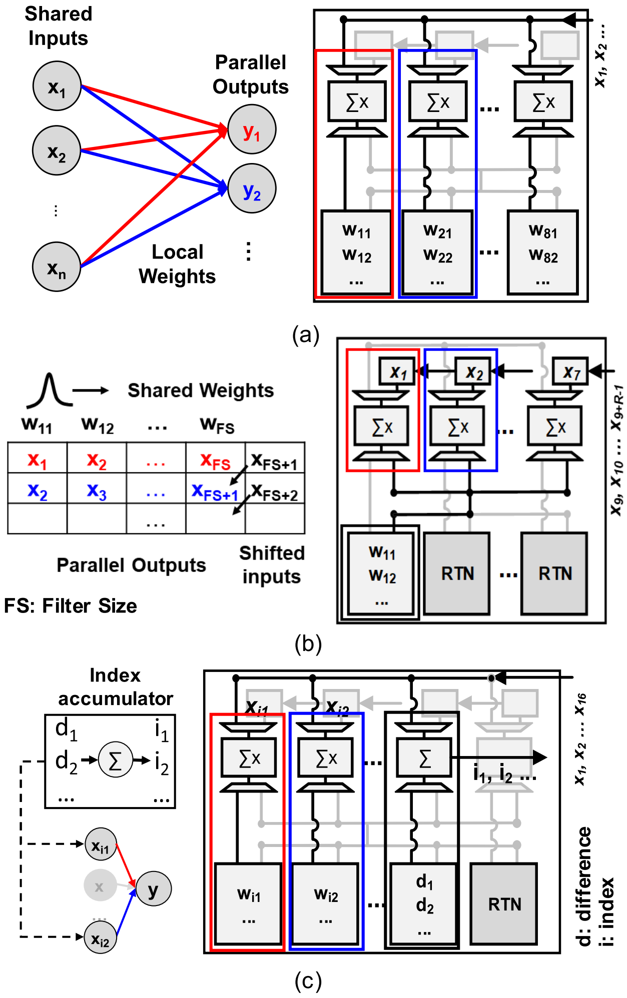


Fig. 10. PE configurations for a (a) fully connected layer, (b) convolution layer, and (c) sparsely connected layer.

memory (weights). Each thread acts as an output neuron as shown in Fig. 10(a). By feeding sequential input data to PE, a maximum of eight output neurons will be computed simultaneously through multiplication and accumulation (MAC) operations on parallel ALUs. The input sharing minimized input data access and computation parallelism improved throughput. It should be noted that fully connected layer's computation is 1-D vector product between the input and weight. As a result, such configuration also applies to 1-by-1 filter kernel in SqueezeNet and MobileNet.

Convolution layer configuration is described in Fig. 10(b). All ALUs compute MAC in parallel where one input from the local shift register (input data) and the other from the global memory bus (weight). During computation, only the SRAM stores the weight will be active, while all others in retention. All threads share kernel filters and apply the kernels to adjacent locations on the input data [shown in Fig. 10(b)]. By feeding sequential input data to PE and shift the data, a maximum of eight convolutions can be processed simultaneously. Furthermore, as filter weights are shared with all threads, un-accessed memory sub-banks are put into retention to save static energy expenditure. In modern deep learning algorithms, a convolution stride greater than 1 is usually applied to reduce data dimensions. In this case, we will not be using full threads, but portions of threads. For example, by applying continuous data into every other threads, we can achieve convolution with a stride of 2. For even larger threads, it will be more convenient and efficient to re-configure the PEs to data-stationary architecture: multiple filters operate in parallel with shared input data that is re-organized for stride greater than 1. In either case, computation efficiency will be degraded by skipped threads or data preparation.

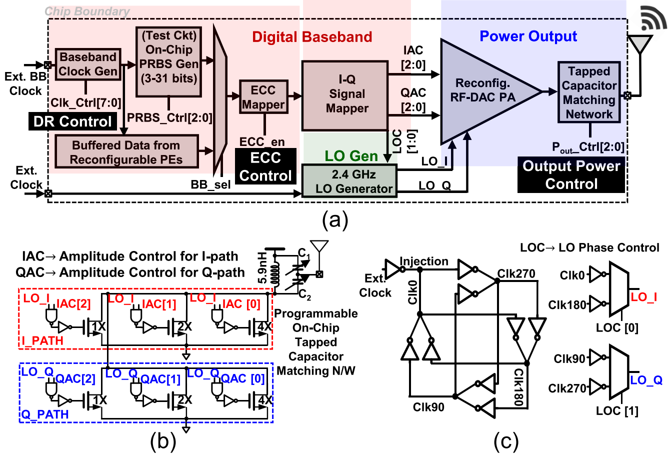


Fig. 11. (a) Reconfigurable transmitter with control on data rate (8 b), output power (3 b), and error-control coding (ECC—1 b), (b) details of the current-mode RF-DAC-based PA, and (c) details of the LO phase generation.

Sparsely connected layer's PE configuration is shown in Fig. 10(c). The threads will be assigned to either MAC or accumulation tasks, respectively. The ones assigned to MAC tasks will collect input data from the global data bus in a pipelined manner. The thread to perform accumulation will be acting as an index accumulator to compute which input should be fetched for computation. The accumulator will read from its local memory of index difference and accumulate them for an actual index. Unused SRAMs are put into the retention mode. Index differences, instead of actual indexes, are stored to save memory footprint.

B. Reconfigurable RF-DAC TX and ULP OOK Rx

To have energy-efficient communication with an external hub, a digitally reconfigurable, data-rate, and channel-aware 2.4 GHz transceiver (see Fig. 11) is designed on the same SoC that demonstrates the effectiveness of computation-communication trade-offs through adapting to various data rates and channel conditions. The transmitter consists of a digital baseband, and an RF-DAC-based power delivery to the antenna. The input data for the testing purpose may come from an on-chip PRBS generator (3–31 bits). Alternatively, real data from the on-chip compute units (reconfigurable PEs) are utilized as input, which can be selected by the baseband select mux. The data rate control is achieved by changing the clock rate from 40 kHz to 10 MHz in 256 steps using an 8-bit control. ECC can be enabled by one control bit that turns on [8, 4] Hamming codes in the digital baseband. From the baseband, 3 bits of *I*-path (in-phase) amplitude

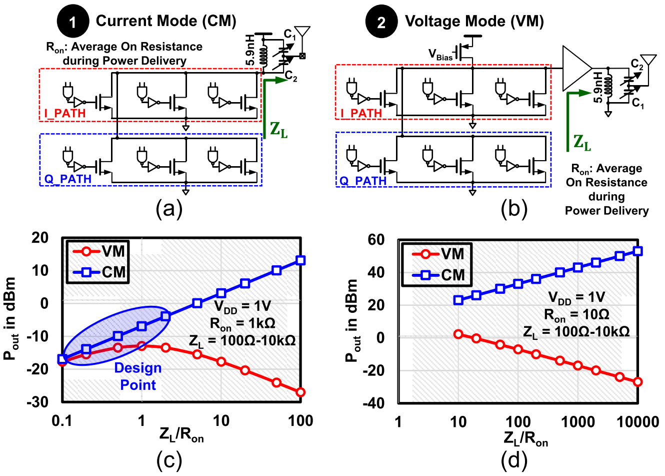


Fig. 12. Current mode and voltage mode choices for the RF-DAC. (a) and (b) Circuit architectures. (c) Output power for $R_{\mathrm{ON}} = 1$ kΩ. (d) Output power for $R_{\mathrm{ON}} = 10$ Ω.

control and 3 bits of Q-path (quadrature) amplitude control determines how many legs of the RF-DAC would be turned on (which determines the amplitudes for different modulation schemes), while 2 bits of LO control generates appropriate local oscillator phases. The possible delay mismatches in the LO generation circuit were kept within acceptable limits using a symmetric common-centroid based layout, the analysis of which will be presented in a future work. In Section IV, we shall show that the error vector magnitudes (EVMs) are $<3\%$, which proves that these delay mismatches are not significant.

The power delivery subsystem, consisting of the reconfigurable RF-DAC-based PA and a fully on-chip tapped capacitor-based matching network with reconfigurable capacitor banks is shown in Fig. 11(b). The RF-DAC-based PA combines the DAC, mixer, and PA operations in a single module through a digital-friendly architecture that switches on or off an appropriate number of current-carrying legs of the module and can support modulation schemes including on off keying (OOK), QPSK, 16-QAM, or 64-QAM. The output power control is achieved using 3 bits that alter the capacitor banks present in the matching network. As mentioned earlier, 3 bits of I-path amplitude control and Q-path amplitude control determine how many legs in the I-path and Q-path would be turned on. The choice of a current-mode (CM) RF-DAC is explained in Fig. 12. The output power (P_{out} and drain efficiency (η) expressions for power delivery with CM RF-DAC and voltage mode (VM) RF-DAC + PA structures are derived as presented in [25] and [26]. For CM implementation, the AC current through the switching transistor i_d is a square wave, while the voltage across the switching transistor v_d is a sine wave (since only the fundamental of the square-wave current passes through the tuned load and creates a sinusoidal voltage). Using the Fourier series expansion of a square wave, i_d can be written as

$$i_d = \frac{V_{\mathrm{DD}}}{2R_{\mathrm{ON}}} \times \left[1 + \frac{4}{\pi}\sum_{k=odd}\frac{\sin k\theta}{k}\right] \tag{1}$$

where R_{ON} average on resistance of the switching transistors in the RF-DAC. The load voltage, $v_L = V_{\mathrm{DD}} - v_d$ (across the load impedance, Z_L) then becomes

$$\begin{aligned} v_L &= -i_{d,\text{fundamental}} \times Z_L \\ &= -\frac{2}{\pi} \times \frac{V_{\mathrm{DD}} Z_L \sin\theta}{R_{\mathrm{ON}}}. \end{aligned} \tag{2}$$

The negative sign signifies the 180° phase difference between i_d and v_L. The output power $P_{\text{out,CM}}$ can thus be written as

$$P_{\text{out,CM}} = \frac{v_{L,\text{rms}}^2}{Z_L} = \frac{2}{\pi^2} \times \frac{V_{\mathrm{DD}}^2 Z_L}{R_{\mathrm{ON}}^2}. \tag{3}$$

The dc current in CM can be written as

$$I_{\mathrm{dc}} = \frac{V_{\mathrm{DD}} - v_{L,\text{avg}}}{R_{\mathrm{ON}}} = \frac{V_{\mathrm{DD}}}{R_{\mathrm{ON}}} \times \left(1 + \frac{2}{\pi^2}\frac{Z_L}{R_{\mathrm{ON}}}\right). \tag{4}$$

Thus, the dc power consumption becomes

$$P_{\text{dc,CM}} = V_{\mathrm{DD}} I_{\mathrm{dc}} = \frac{V_{\mathrm{DD}}^2}{R_{\mathrm{ON}}} \times \left(1 + \frac{2}{\pi^2}\frac{Z_L}{R_{\mathrm{ON}}}\right) \tag{5}$$

which results in

$$\eta_{\mathrm{CM}} = \frac{P_{\text{out,CM}}}{P_{\text{dc,CM}}} = \frac{1}{1 + \frac{\pi^2}{2}\frac{R_{\mathrm{ON}}}{Z_L}} \approx \frac{1}{1 + 5R_{\mathrm{ON}}/Z_L}. \tag{6}$$

From (3) and (5), it is evident that both $P_{\text{out,CM}}$ and $P_{\text{dc,CM}}$ keep on increasing as Z_L/R_{ON} increases (or simply, R_{ON} reduces with a fixed Z_L, making the switching transistor consume higher power). This makes CM power delivery a suitable option for a wide range of output power. For VM implementation, $P_{\text{out,VM}}$, $P_{\text{dc,VM}}$, and η_{VM} can be found as

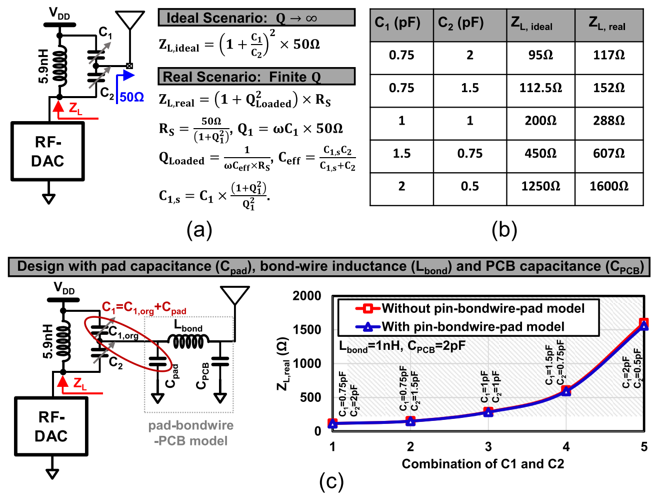
$$P_{\text{out,VM}} = \frac{2}{\pi^2} \times \frac{V_{\mathrm{DD}}^2 Z_L}{(R_{\mathrm{ON}} + Z_L)^2} \tag{7}$$

$$P_{\text{dc,VM}} = \frac{2}{\pi^2} \times \frac{V_{\mathrm{DD}}^2}{(R_{\mathrm{ON}} + Z_L)} \tag{8}$$

and

$$\eta_{\mathrm{VM}} = \frac{1}{1 + R_{\mathrm{ON}}/Z_L} \tag{9}$$

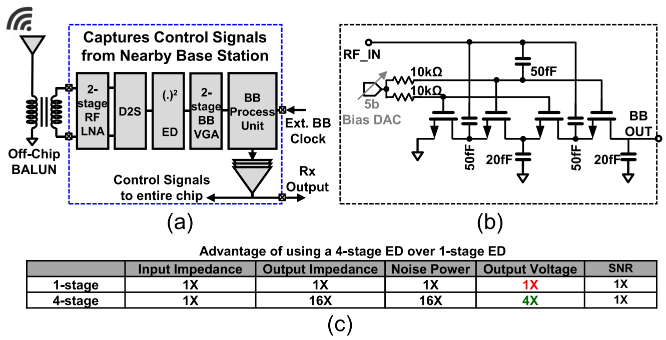
where Z_L is the load impedance as seen by the VM power delivery network (which is a class D PA) and R_{ON} is the on resistance of the PMOS (or NMOS) in the PA. From (7) and (8), it can be shown that both $P_{\text{out,VM}}$ and $P_{\text{dc,VM}}$ saturate to $(2/\pi^2) \times (V_{\mathrm{DD}}^2/Z_L)$ as Z_L/R_{ON} becomes a large number. P_{out} for both CM and VM is plotted with respect to the ratio Z_L/R_{ON} for two different values of R_{ON} (1 kΩ and 10 Ω) in Fig. 12(c) and (d). CM offers a higher P_{out} than VM for both scenarios. However, CM with $R_{\mathrm{ON}} = 1$ kΩ offers output power in the range -20 to 0 dBm, which is suited for most short-range communication and IoT-based applications. A higher R_{ON} also helps in reducing the dc power consumption according to (5). It is interesting to note that VM with $R_{\mathrm{ON}} = 10$ Ω can also support output power in the range -20 to 0 dBm [see Fig. 12(d)]. However, to achieve a low R_{ON}, large transistors need to be used, leading to higher parasitics and driving power. As a result, we have adopted the CM topology with a higher R_{ON}. As will be shown in Fig. 13, the achievable range of Z_L is 117–1600 Ω (through selection of



V_{DD}
5.9nH
C_1
C_2
50Ω
Z_L
RF-DAC
Ideal Scenario: $Q \rightarrow \infty$
$Z_{L,ideal} = \left(1 + \frac{C_1}{C_2}\right)^2 \times 50\Omega$
Real Scenario: Finite Q
$Z_{L,real} = \left(1 + Q_{Loaded}^2\right) \times R_S$
$R_S = \frac{50\Omega}{(1+Q_1^2)}$, $Q_1 = \omega C_1 \times 50\Omega$
$Q_{Loaded} = \frac{1}{\omega C_{eff} \times R_S}$, $C_{eff} = \frac{C_{1,s} C_2}{C_{1,s}+C_2}$
$C_{1,s} = C_1 \times \frac{(1+Q_1^2)}{Q_1^2}$.
(a)
(b)
Design with pad capacitance (C_{pad}), bond-wire inductance (L_{bond}) and PCB capacitance (C_{PCB})
C_1=$C_{1,org}$+C_{pad}
$C_{1,org}$
L_{bond}
C_{pad}
C_{PCB}
pad-bondwire -PCB model
Without pin-bondwire-pad model
With pin-bondwire-pad model
L_{bond}=1nH, C_{PCB}=2pF
$Z_{L,real}$ (Ω)
2000
1500
1000
500
0
C_1=0.75pF C_2=2pF
C_1=0.75pF C_2=1.5pF
C_1=1pF C_2=1pF
C_1=1.5pF C_2=0.75pF
C_1=2pF C_2=0.5pF
1
2
3
4
5
Combination of C1 and C2
(c)

C_1 (pF)	C_2 (pF)	$Z_{L,\ ideal}$	$Z_{L,\ real}$
0.75	2	95Ω	117Ω
0.75	1.5	112.5Ω	152Ω
1	1	200Ω	288Ω
1.5	0.75	450Ω	607Ω
2	0.5	1250Ω	1600Ω

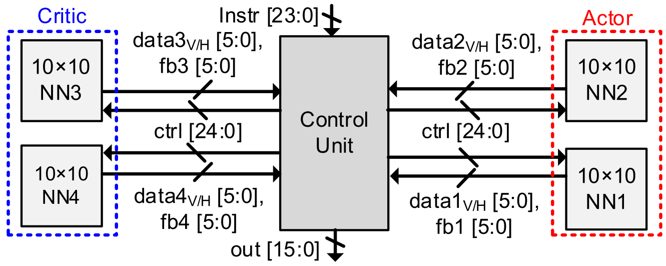
Fig. 13. Tapped capacitor matching network for the RF-DAC. (a) Circuit and equations. (b) Reconfigurable capacitor choices and corresponding load impedances. (c) Effect of pad-bondwire-PCB model, and the inclusion of the fixed pad capacitance into C_1.



Advantage of using a 4-stage ED over 1-stage ED

	Input Impedance	Output Impedance	Noise Power	Output Voltage	SNR
1-stage	1X	1X	1X	1X	1X
4-stage	1X	16X	16X	4X	1X

Fig. 14. Design of the ULP OOK receiver. (a) Block diagram. (b) Circuit diagram of the ED. (c) Advantage of using a 4-stage ED.



Critic
Instr [23:0]
Actor
10×10 NN3
$data3_{V/H}$ [5:0], fb3 [5:0]
$data2_{V/H}$ [5:0], fb2 [5:0]
10×10 NN2
Control Unit
ctrl [24:0]
ctrl [24:0]
10×10 NN4
$data4_{V/H}$ [5:0], fb4 [5:0]
$data1_{V/H}$ [5:0], fb1 [5:0]
10×10 NN1
out [15:0]

Fig. 15. Circuit diagram of the NN-based AC controller.

different capacitors in the tapped capacitor matching network), leading to a Z_L/R_{ON} ratio of 0.1–1.6 in Fig. 12(c). The tapped capacitor-based matching network is preferred over having a programmable load current going into a fixed matching network because it is much easier to keep the devices in the correct region with a fixed current, while $>50\ \Omega$ effective input impedance of the matching network ensures low power consumption.

The 2.4 GHz LO generation [see Fig. 11(c)] for both I and Q paths is performed by an on-chip LO generator. Four LO phases are selected based on the 2 bit LO control as obtained from the I–Q signal mapper.

A reconfigurable tapped capacitor-based matching network [27] with a 50 Ω antenna has been used to tune the Z_L as seen by the RF-DAC. By choosing different capacitor ratios, Z_L/R_{ON} can be reconfigured for a fixed R_{ON}, leading to different output power levels. As shown in Fig. 13, the effect of finite Q of the on-chip inductor is considered during the design of the matching network, which modifies the well-known formula obtained in the ideal scenario that considers an infinite Q. Five different values of C_1 and C_2 are considered and are placed on a chip as a part of two different capacitor banks that cover a matching network impedance from about 117 Ω to about 1600 Ω, when limiting the bigger capacitor to 2 pF due to area constraints. While designing the matching network, the effects of pad capacitance, bond-wire inductance, and printed circuit board (PCB) capacitance are also considered, and the effect of the additional capacitances are included in the value of the on-chip capacitor, C_1. The design of the tapped capacitor matching network is described in Fig. 13(a)–(c).

An ultralow-power (ULP) OOK receiver [see Fig. 14(a)] is designed on the same chip that captures control signals from a nearby base station to achieve closed-loop control on the 8 clock control bits, 1 ECC control bit, and 3 Pout control bits. The receiver consists of two stages of RF LNA, a differential to single-ended converter (D2S), an envelope detector (ED), two stages of baseband variable gain amplifier (VGA), and a baseband comparator. For the ED [see Fig. 14(b)], a 4-stage gate-biased structure is used which increases the output voltage by 4× as compared to the 1-stage ED, thereby compensating for the loss incurred during envelope detection [28]. The SNR, however, remains constant as we increase the number of stages. More stages can improve the output voltage further, thereby making the decision-making process easier at the comparator, at the cost of additional chip area.

C. NN-Based AC Controller

The large control space across computation and communication is learned using a low overhead (5% power, 2.5% area) AC-NN controller (see Fig. 15). The AC-NN takes both design targets and sensed variables as inputs and learns to optimally control the control knobs. These are listed in Fig. 2.

The controller features four (two for the actor, two for the critic) 10 × 10 memory sub-banks with time-based CIM modules. A central control unit is used for communication between CIM modules. The system architecture of the CIM module is shown in Fig. 16. At the core, 100 thermometer-encoded storage elements (SEs) form a 10 by 10 storage array. Each SE has vertical and horizontal connections for both wordlines (WLs) and bitlines (BLs). The array can be read both row-wise as well as column-wise providing a seamless design for transposing the weight matrix during back-propagation. This also enables in-place online learning without requiring reads and write-backs (baseline designs). At the edge of the storage array, digital to time converters (DTCs), ADCs,

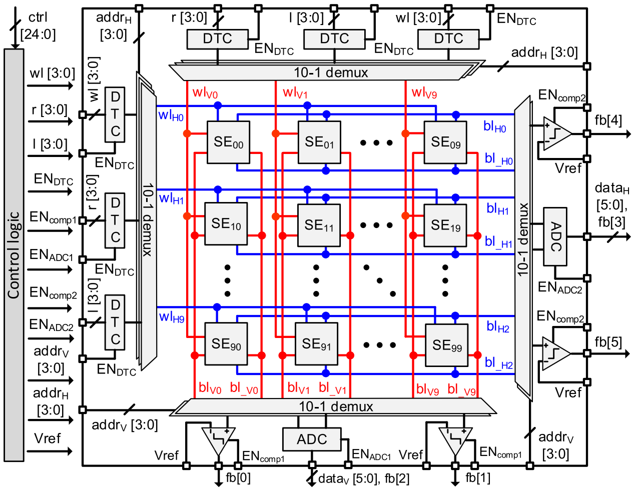


Fig. 16. Circuit of 10-by-10 compute-update-in-memory (CUIM) module.

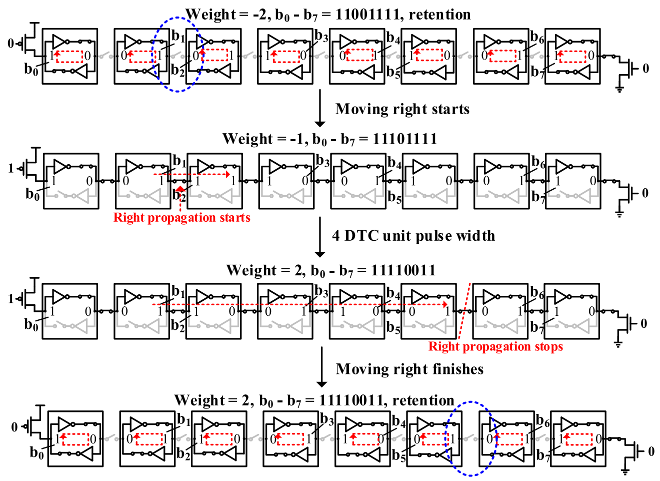
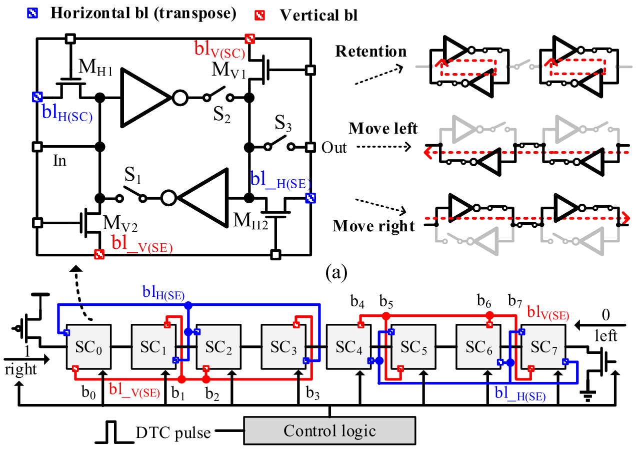

Weight = -2, b0 - b7 = 11001111, retention
Moving right starts
Weight = -1, b0 - b7 = 11101111
Right propagation starts
4 DTC unit pulse width
Weight = 2, b0 - b7 = 11110011
Right propagation stops
Moving right finishes
Weight = 2, b0 - b7 = 11110011, retention

Fig. 18. Example of *in situ* weight update in SE.



(a)

(b)

$b_0 - b_7$	00001111	10001111	11001111	11101111	11111111	11110111	11110011	11110001	11110000
weight	-4	-3	-2	-1	0	1	2	3	4

(c)

Fig. 17. (a) 1 bit SC circuit. (b) 8 b SE. (c) Thermometer encoding of SE.

and comparators form the peripheral. One control logic is used to control all peripherals and the storage array.

To achieve in-memory computing and weight update, a new 1-bit storage cell (SC) based on SRAM is proposed [see Fig. 17(a)]. In addition to standard 6T SRAM, two WL transistors (M_{H1}, M_{H2}) are added for matrix transpose. Moreover, three transmission gates (S_1, S_2, S_3) are used to enable data movement between SCs. The SC has three operation modes: 1) retention; 2) move left; and 3) move right. In the retention mode, S_1 and S_2 are closed, while S_3 is open. The data in SC is stored the same as SRAM. When moving right or left, S_3 is closed to transmit data to adjacent SCs. Depending on the direction, either S_1 or S_2 is closed.

The circuit diagram of the 3-bit thermometer-encoded SE is shown in Fig. 17(b). Eight SCs are sequentially connected with a pull-up transistor on the left and pull-down transistors on the right. A control logic controls each SC according to the input DTC pulse from peripherals. SC_0–SC_3 are connected to $\mathrm{bl\_}_{V(\mathrm{SE})}$ and $\mathrm{bl}_{H(\mathrm{SE})}$, and SC_4–SC_7 are connected to $\mathrm{bl}_{V(\mathrm{SE})}$ and $\mathrm{bl\_}_{H(\mathrm{SE})}$. $\mathrm{WL}_{H(\mathrm{SC})}$ or $\mathrm{WL}_{V(\mathrm{SC})}$ of all SCs are connected together, respectively. In each SC, either $\mathrm{bl}_{H(\mathrm{SC})}$ or $\mathrm{bl\_}_{H(\mathrm{SC})}$ and either $\mathrm{bl}_{V(\mathrm{SC})}$ or $\mathrm{bl\_}_{V(\mathrm{SC})}$ are used to represent the weight bits. The thermometer encoding of SE is shown in Fig. 17(c). The number of "0"s for b_0–b_3 represents a negative value and the number of "0"s for b_4–b_7 represents a positive value. 0 is encoded by all "1"s.

During inference, DTCs allow pulsewidth-modulated WLs (input signals) to be turned on sequentially such that the falling edge of one row triggers the rising edge of the next. The partial products are accumulated on the BL as long as the voltage on the BL is greater than a threshold. The differential SC design allows both positive and negative weights by discharging either bl (positive) or $\mathrm{bl}_{\mathrm{bar}}$ (negative). The array can be read both row-wise as well as column-wise providing a seamless design for transposing the weight matrix during back-propagation. This also enables in-place online learning without requiring reads and write-backs (baseline designs).

Besides in-memory computing, the above structure enables *in situ* weight update with low hardware and control overhead. The update is fulfilled by one single time pulse generated by DTCs and the magnitude of the update is controlled by the duration of the time pulse. Fig. 18 shows the weight update process (increase weight by 4). At first, the weight is -2 and the SE is in the retention mode. b_1 and b_2 have different values. When the update process starts, the pull-up transistor is enabled and all SCs are controlled to propagate data to the right. Starting from b_2, the data in SCs will flip from left to right sequentially. According to the DTC pulsewidth, the propagation stops at the desired place and all SCs return to the retention mode. The data in the remaining SCs will not change, and the weight in SE becomes 2. Similarly, to decrease weight magnitude, the pull-down transistor is enabled, and SCs propagate data to the left. This scheme can change the weight during the learning process without leaving the storage unit.

In CIM designs, high-resolution ADC consumes most area and energy. However, according to the data distribution simulation, more than 90% results fall in the 6-bit range, while the worst case requires 8 b resolution ADC. As we expect

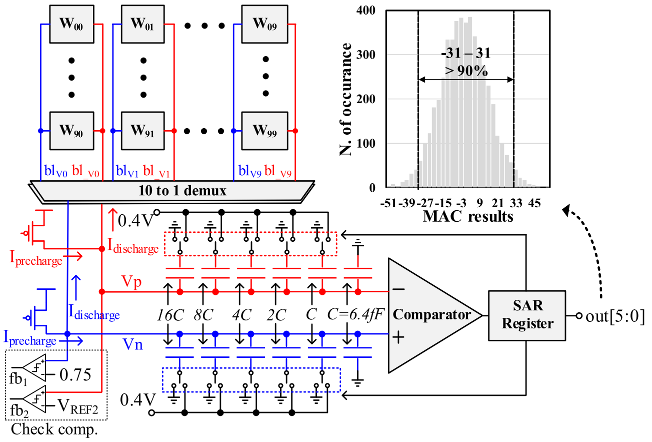


Fig. 19. Data-aware adaptive differential SAR ADC with read disturb protection comparators.

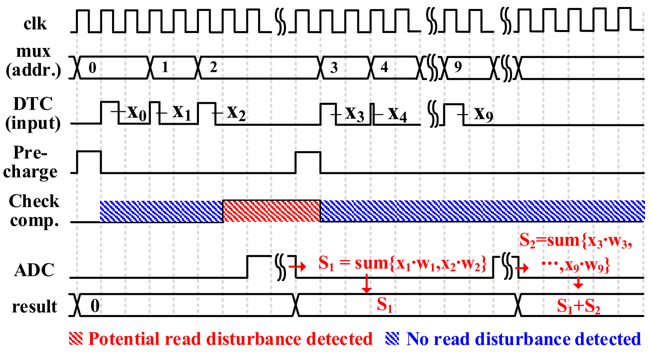
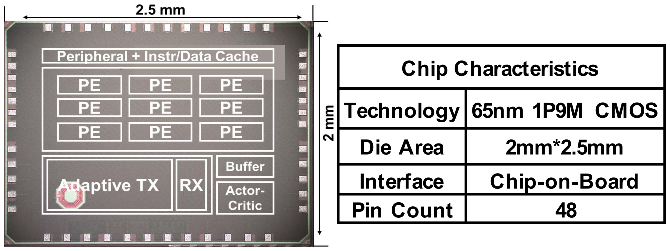

clk
mux (addr.)
0
1
2
3
4
9
DTC (input)
X0
X1
X2
X3
X4
X9
Pre-charge
Check comp.
ADC
$S_1 = \mathrm{sum}\{x_1 \cdot w_1, x_2 \cdot w_2\}$
$S_2 = \mathrm{sum}\{x_3 \cdot w_3, \cdots, x_9 \cdot w_9\}$
result
0
S_1
S_1+S_2
Potential read disturbance detected
No read disturbance detected

Fig. 20. Timing diagram showing adaptive A/D conversion scheme.

data conversion to be a major energy consumer, we decided to implement an adaptive A/D conversion scheme that uses 6 b resolution ADC for optimized energy efficiency but still supports 8 b output. The circuit diagram and data simulation distribution are shown in Fig. 19. We choose 6 b capacitor-based SAR ADC and share the capacitors with bitlines. The parasitic capacitance of each bitline is around 40 fF. During computing, 32 ADC capacitors (6.4 fF each) are connected to the bitline. Therefore, the total capacitance on the bitline is 244.8 fF. Assuming a 25% variation on bitline parasitics, it is only 4% of the total capacitance and only leads to 4% computing error. Therefore, by sharing ADC capacitors with the bitline, it improves the dynamic range and embeds the sampling process of ADC into the computing cycle. The ADC connects with weight SEs via a 10-1 multiplexer, and two additional comparators detect potential read disturbance. In addition, the monolithic switching procedure of ADC further reduces the energy [29].

The timing diagram in Fig. 20 illustrates the adaptive A/D conversion scheme. At first, BLs are pre-charged. In most cases, the 10-by-10 vector multiplication is completed before conversion. However, when the intermediate sum of the product gets close to the maximum range of ADC or may cause a read disturbance on the SC (red area in Fig. 20), the computing cycle is stopped, and the ADC starts to convert the BL voltages to digital output. After conversion, BLs are pre-charged again and continue computing the remaining cells. When all cells are computed, the outputs are accumulated in the digital domain to get the final result. The different computation engine selections for DNN processor (digital Von Neumann) and AC controller (mixed-signal compute-in-memory architecture with thermometer-coded *in situ* data update) are twofold: first, this SoC features a programmable DNN accelerator, Von-Neumann architecture is versatile for optimized operations/data-flow; on the contrary, AC-controller is mainly implemented by matrix-multiplication which is a perfect fit for CIM architecture. Second, the control algorithms require lower bit-precision (3 b) than image inference (8 b), and at the same time, it requires frequent updates to adapt to the environment. As such, the proposed CIM circuit gains energy/throughput advantages for both inference and learning.



Chip Characteristics	
Technology	65nm 1P9M CMOS
Die Area	2mm*2.5mm
Interface	Chip-on-Board
Pin Count	48

Fig. 21. Chip die micrograph and characteristics.

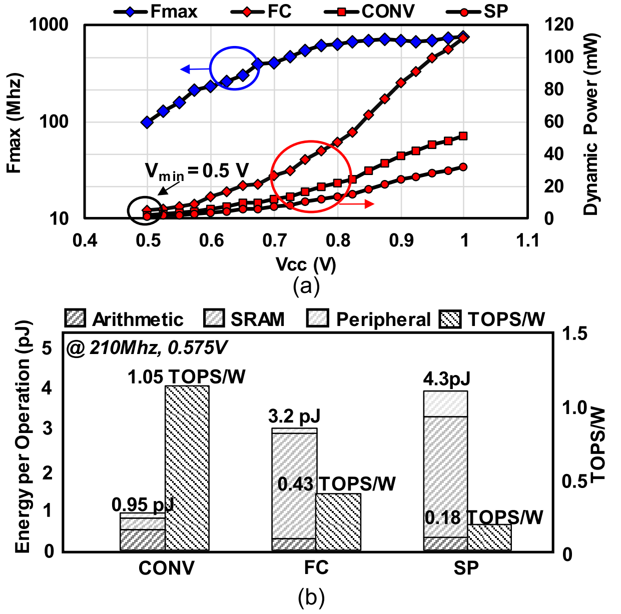


Fig. 22. Measured (a) computation pipeline frequency/power characteristics and (b) energy consumption per operations for various layers. All the measured numbers are solely for the core operations, where pooling, batch normalization, and so on are not included. CONV stands for the convolution layer with a stride of 1.

IV. Measurements

The test chip is fabricated in 65 nm technology with a total area of 5 mm^2. The chip die photograph and characteristics are shown in Fig. 21.

The measured power performance of the processing engine [see Fig. 22(a)] shows V_{MIN} of 0.5 V and

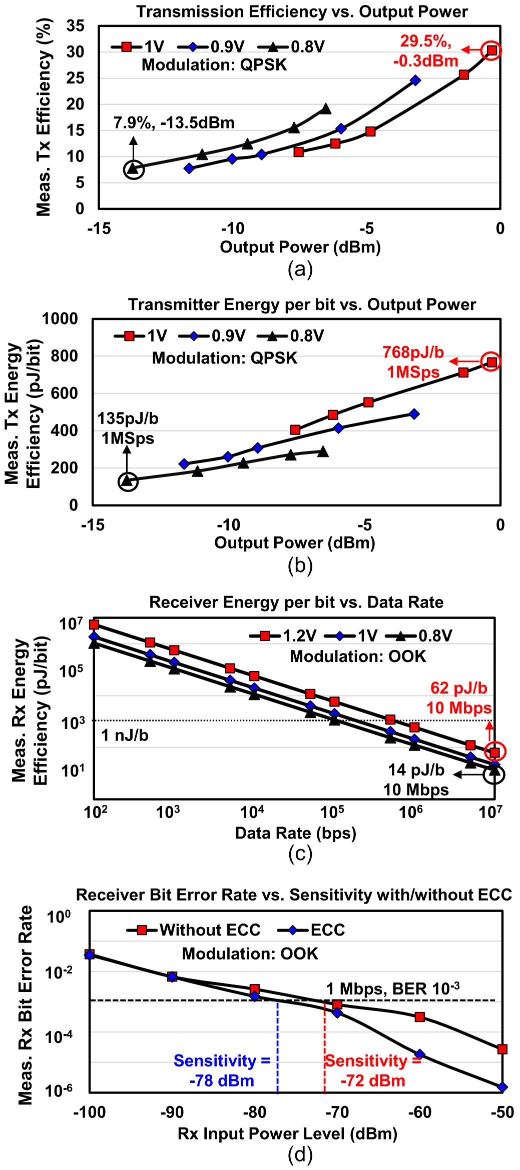
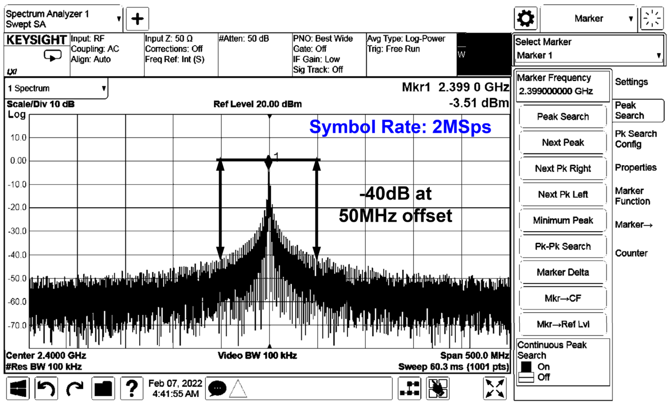


Fig. 23. Measured transceiver energy and BER performance. (a) Measured TX efficiency versus output power for different output power levels as determined by the tapped capacitor matching network. (b) Measured TX energy efficiency versus output power. (c) Measured RX energy efficiency versus data rate. (d) Measured RX BER versus input power levels.



Spectrum Analyzer 1
Swept SA

KEYSIGHT
Input: RF
Coupling: AC
Align: Auto
Input Z: 50 Ω
Corrections: Off
Freq Ref: Int (S)
#Atten: 50 dB
PNO: Best Wide
Gate: Off
IF Gain: Low
Sig Track: Off
Avg Type: Log-Power
Trig: Free Run

1 Spectrum
Mkr1 2.399 0 GHz
-3.51 dBm
Scale/Div 10 dB
Log
Ref Level 20.00 dBm

Select Marker
Marker 1
Marker Frequency
2.399000000 GHz

Center 2.4000 GHz
#Res BW 100 kHz
Video BW 100 kHz
Span 500.0 MHz
Sweep 60.3 ms (1001 pts)

Continuous Peak Search
■ On
□ Off

Feb 07, 2022
4:41:55 AM

Fig. 24. Measured TX spectrum for QPSK with 2 MSps symbol rate, 500 MHz frequency span.

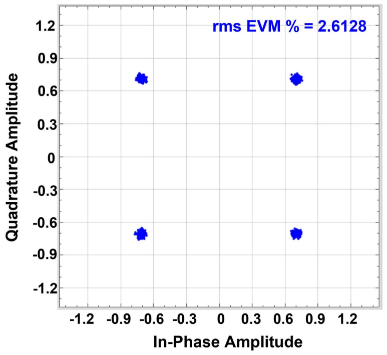


Fig. 25. Measured EVM at a distance of 2 m from the transmitter, with a TX power at −0.3 dBm (QPSK, 2 MSps).

F_{MAX} of 760 MHz. Peak arithmetic energy efficiency of 1.05 TOPS/W (0.43 TOPS/W, 0.18 TOPS/W) is measured for CONV (FC, sparse) networks at 210 MHz (0.575 V) [see Fig. 22(b)]. With the proposed weight-sharing scheme in PE's convolution configuration and fine control of the un-accessed SRAM retention mode, computation-centric convolution operation has achieved sub-pJ efficiency per operation by minimizing unnecessary memory usage.

The RF subsystem (see Fig. 23) shows a maximum TX efficiency of 29.5% at −0.3 dBm, with back-off efficiencies of 19.2% (7.9%) at −6.5 (−13.5 dBm) with QPSK. At 1 Mb/s, the TX energy efficiency is 768 pJ/bit with 1 V supply (−0.3 dBm output power). The measured energy efficiency for the OOK RX is 62 (14) pJ/bit at 1 (0.8) V supply at 10 Mb/s. Fig. 23(d) shows the effect of ECC on the RX bit error rate (BER). Without ECC, a sensitivity of −72 dBm is achieved for a BER of 10^{-3} at 1 Mb/s. An [8, 4] Hamming code-based ECC on the TX improves the RX sensitivity to −78 dBm (but halves the number of information bits). It is also interesting to note that ECC helps in achieving a significant (10×) improvement in BER when the RX input power level is in the range of −50 to −65 dBm. For lower power levels (e.g., −85 dBm or lower), burst errors are observed at the Rx, for which ECC does not have a significant advantage over the case without ECC. Fig. 24 shows the measured TX spectrum for QPSK with 2 MSps symbol rate. This result is taken near the highest output power for the PA. Fig. 25 shows the EVM at a distance of 2 m from the TX, with TX power set at −0.3 dBm for QPSK at 2 MSps. The rms EVM is only ≈2.6% (≈31.7 dB), showing that any delay mismatch in the LO generation and the RF-DAC are not significant.

The oscilloscope capture of NN 10-by-10 CIM block bitline discharge is shown in Fig. 26. By providing 1–3 unit time wordline voltage pulse, bitline discharges proportionally with constant weights.

To investigate the computation accuracy of the CIM block, we have applied random inputs to the controller at measured output result for each bitline (see Fig. 27). First, we can observe that more than 95% of final results are within −40 to 40 range. Furthermore, before digital compensation, we find an average error increase with the final computation result. That

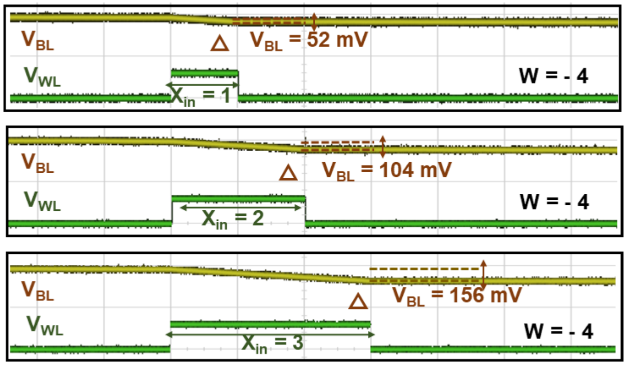


Fig. 26. Oscilloscope capture of bitline discharge of the CUIM module.

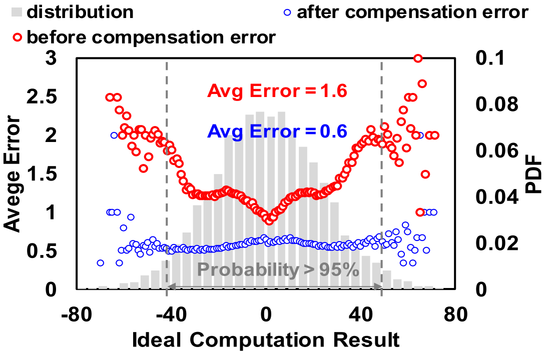


Fig. 27. Measured CUIM module nonlinearities.

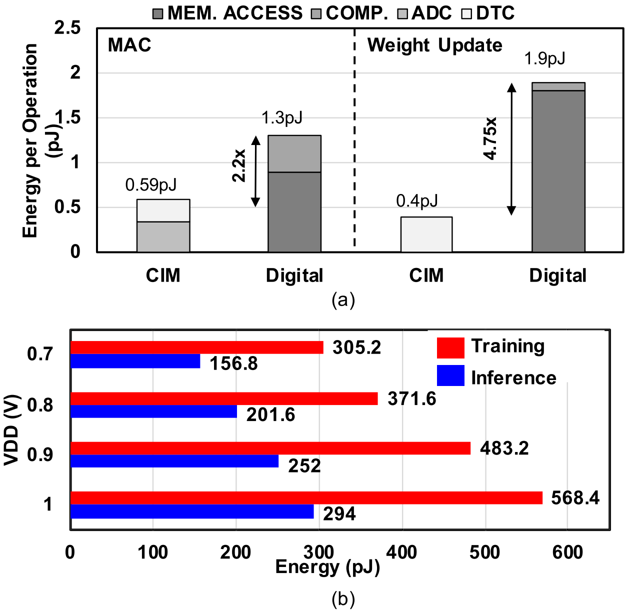


Fig. 28. Measured CUIM module energy efficiency. (a) Measured energy per operation and energy breakdown comparison between CIM and digital modules. (b) Measured training and inference energy across supply voltage.

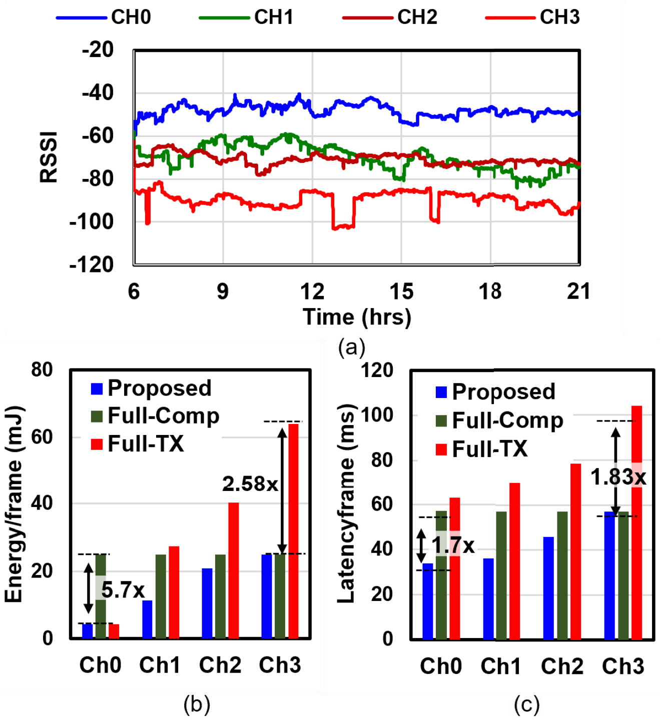


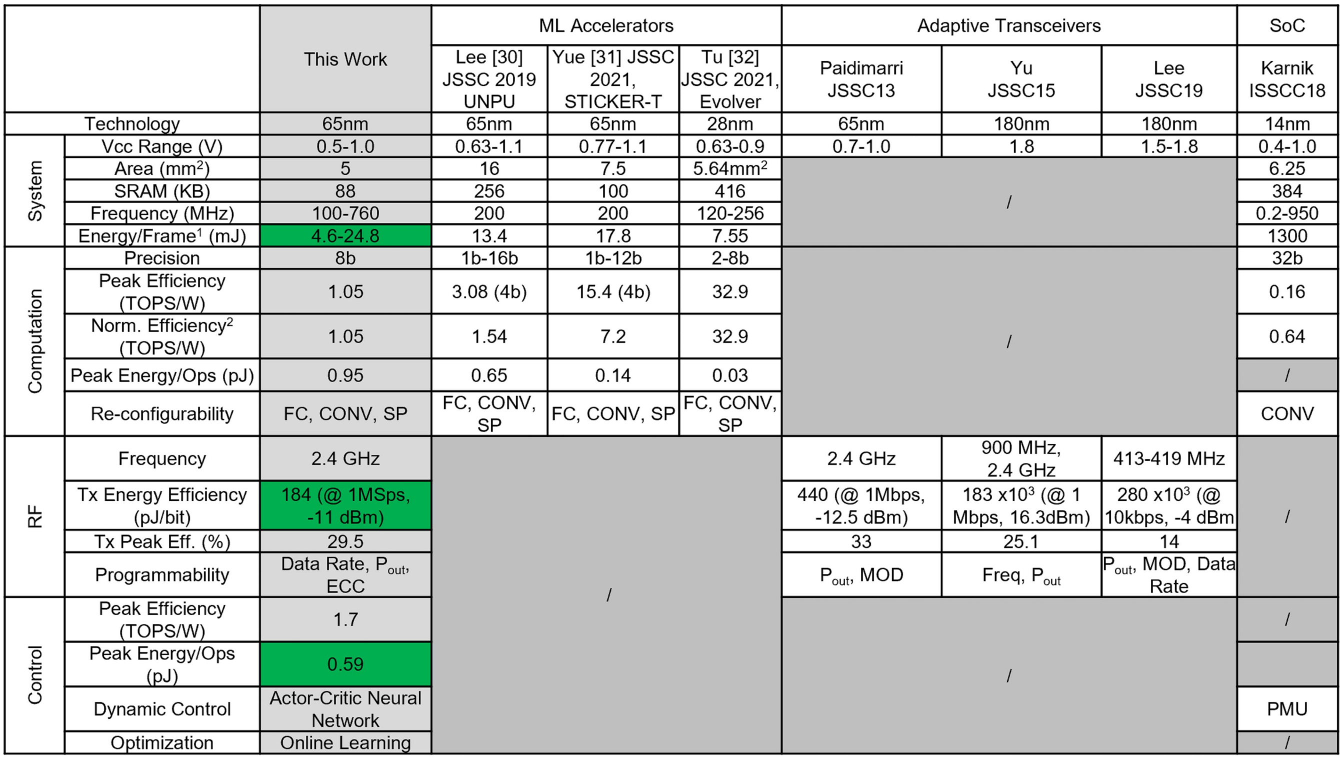
Fig. 29. (a) Real-world wireless channel measurements over 15 h across channels; (b) and (c) System-level energy/latency measurements of proposed neuro-controller with baseline control methods.

means nonlinearity errors accumulate on bitline. An average of 1.6 errors is measured. After compensation, the error is largely reduced, especially in computations where final results are significant. Average error after compensation is around 0.6.

The measured performance of the neuro-controller is shown in Fig. 28. The CIM consumes a measured 305.2 pJ (training) and 156.8 pJ (inference) at 0.7 V with less than 0.6 LSB of nonlinearity error. The peak measured energy efficiency is 0.59 pJ/MAC and 0.4 pJ for each weight update which are 2.2× and 4.75× lower than a digital counterpart (simulated).

The full system is deployed and a neuro-controller is allowed to learn online from emulated signals from the cloud and energy meters. Then it is tested for varying noise power and network sizes and the system autonomously determines the optimal PD to minimize energy, latency, or EDP. The online adaptation allows the system to learn and choose the CTRL parameters optimally. To demonstrate the effectiveness of the proposed wireless SoC in real environment, we have conducted substantial experiments on the campus of Purdue University. To account for channel variance, the experiments last for 15 h across four different channels. We test across various conditions of path-loss and the number of edge nodes (i.e., available bandwidth) and obtain a 2.58–5.7× (1.7–1.83×) improvement in average energy (latency) for a BER of 10^{-5} compared to the baseline cases while running SqueezeNet that maps to the SoC (see Fig. 29).

The proposed system is one of the first prototypes to address computation and communication trade-offs with full SoC solution. We have benchmarked our system with state-of-the-art designs and show competitive figures-of-merit (see Fig. 30) across ML accelerators [30]–[32], adaptive transceivers [27], [33], [34], and SoCs [35]. We can observe that the proposed platform achieved minimal energy per frame (4.6 mJ) in wireless environment, despite the fact that computation efficiency is suboptimal. The design presents a vertically integrated SoC featuring the first real-time NN-based adaptation for computation, communication, and their trade-offs in energy-constrained systems.



		This Work	ML Accelerators			Adaptive Transceivers			SoC
			Lee [30] JSSC 2019 UNPU	Yue [31] JSSC 2021, STICKER-T	Tu [32] JSSC 2021, Evolver	Paidimarri JSSC13	Yu JSSC15	Lee JSSC19	Karnik ISSCC18
	Technology	65nm	65nm	65nm	28nm	65nm	180nm	180nm	14nm
System	Vcc Range (V)	0.5-1.0	0.63-1.1	0.77-1.1	0.63-0.9	0.7-1.0	1.8	1.5-1.8	0.4-1.0
	Area (mm^2)	5	16	7.5	5.64mm^2	/			6.25
	SRAM (KB)	88	256	100	416				384
	Frequency (MHz)	100-760	200	200	120-256				0.2-950
	Energy/Frame^1 (mJ)	4.6-24.8	13.4	17.8	7.55				1300
Computation	Precision	8b	1b-16b	1b-12b	2-8b	/			32b
	Peak Efficiency (TOPS/W)	1.05	3.08 (4b)	15.4 (4b)	32.9				0.16
	Norm. Efficiency^2 (TOPS/W)	1.05	1.54	7.2	32.9				0.64
	Peak Energy/Ops (pJ)	0.95	0.65	0.14	0.03				/
	Re-configurability	FC, CONV, SP	FC, CONV, SP	FC, CONV, SP	FC, CONV, SP				CONV
RF	Frequency	2.4 GHz	/			2.4 GHz	900 MHz, 2.4 GHz	413-419 MHz	/
	Tx Energy Efficiency (pJ/bit)	184 (@ 1MSps, -11 dBm)				440 (@ 1Mbps, -12.5 dBm)	183 x10^3 (@ 1 Mbps, 16.3dBm)	280 x10^3 (@ 10kbps, -4 dBm	
	Tx Peak Eff. (%)	29.5				33	25.1	14	
	Programmability	Data Rate, P$_{out}$, ECC				P$_{out}$, MOD	Freq, P$_{out}$	P$_{out}$, MOD, Data Rate	
Control	Peak Efficiency (TOPS/W)	1.7				/			/
	Peak Energy/Ops (pJ)	0.59							
	Dynamic Control	Actor-Critic Neural Network							PMU
	Optimization	Online Learning							/

[1] Estimated for SqueezeNet with off-chip memory access and RF transmission energy cost included

[2] Normalized to 8b operations

Fig. 30. Comparison with the state-of-the-art.

V. Conclusion

This article presents a 65 nm wireless image processing SoC for real-time computation-communication trade-off on resource-constrained edge devices. The test chip includes: 1) an all-digital, near-memory, reconfigurable, and programmable NN-based systolic image processor at 1.05 TOPS/W (peak); 2) a digitally adaptive RF-DAC-based transceiver with TX energy efficiency of 182 pJ/b; and 3) a mixed-signal, time-based, AC neuro-controller with CIM and in-place weight updates that provides online learning and adaptation at 0.59 pJ/MAC for efficiently controlling the computation, communication blocks separately as well as jointly.

VI. Future Work

As an SoC design prototype, there exist substantial challenges and opportunities to further advance performance in individual modules, such as computation, communication, control, and their integration.

1) The online learning feature of DNN has not been enabled for the current design. It will be of practical interest to investigate the performance/cost trade-off policy for wireless IoT online learning.
2) Current DNN computation efficiency is below state-of-the-art DNN accelerators. This results from both conservative choice of technology node and under-optimized computations, such as pooling, batch normalization, and so on. These operations are currently implemented with general-purpose hardware. Future generations of the proposed SoC will target state-of-the-art DNN accelerator performance with advanced technology node and full-stack optimization of the system.
3) Current SoC design is greatly constrained by the on-chip area. In the future design, we would like to explore the 3-D integration option of major components, such as memory, digital circuit, and even antenna. Given enough computation resource, we would like to further explore optimal PE array size, number of threads per PE, and optimal SRAM capacity.

References

[1] A. Amravati, S. B. Nasir, S. Thangadurai, I. Yoon, and A. Raychowdhury, "A 55 nm time-domain mixed-signal neuromorphic accelerator with stochastic synapses and embedded reinforcement learning for autonomous micro-robots," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2018, pp. 124–126.

[2] M. Chang, L.-H. Lin, J. Romberg, and A. Raychowdhury, "OPTIMO: A 65-nm 279-GOPS/W 16-b programmable spatial-array processor with on-chip network for solving distributed optimizations via the alternating direction method of multipliers," *IEEE J. Solid-State Circuits*, vol. 55, no. 3, pp. 629–638, Mar. 2020.

[3] J.-H. Yoon, M. Chang, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, and A. Raychowdhury, "29.1 A 40 nm 64 Kb 56.67TOPS/W read-disturb-tolerant compute-in-memory/digital RRAM macro with active-feedback-based read and *in-situ* write verification," in *Proc. IEEE Int. Solid- State Circuits Conf. (ISSCC)*, Feb. 2021, pp. 404–406.

[4] N. Cao, B. Chatterjee, M. Gong, M. Chang, S. Sen, and A. Raychowdhury, "A 65 nm image processing SoC supporting multiple DNN models and real-time computation-communication trade-off via actor-critical neuro-controller," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2020, pp. 1–2.

[5] N. Cao, M. Chang, and A. Raychowdhury, "14.1 A 65 nm 1.1-to-9.1TOPS/W hybrid-digital-mixed-signal computing platform for accelerating model-based and model-free swarm robotics," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2019, pp. 222–224.

[6] B. Chatterjee, N. Cao, A. Raychowdhury, and S. Sen, "Context-aware intelligence in resource-constrained IoT nodes: Opportunities and challenges," *IEEE Des. Test*, vol. 36, no. 2, pp. 7–40, Apr. 2019.

[7] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50X fewer parameters and 0.5 MB model size," 2016, *arXiv:1602.07360*.

[8] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[9] N. Cao, S. B. Nasir, S. Sen, and A. Raychowdhury, "Self-optimizing IoT wireless video sensor node with *in-situ* data analytics and context-driven energy-aware real-time adaptation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 9, pp. 2470–2480, Sep. 2017.

[10] N. Cao, S. B. Nasir, S. Sen, and A. Raychowdhury, "In-sensor analytics and energy-aware self-optimization in a wireless sensor node," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2017, pp. 200–203.

[11] D. Kang, D. Kang, J. Kang, S. Yoo, and S. Ha, "Joint optimization of speed, accuracy, and energy for embedded image recognition systems," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2018, pp. 715–720.

[12] T. Ko, "A survey on behavior analysis in video surveillance for homeland security applications," in *Proc. 37th IEEE Appl. Imag. Pattern Recognit. Workshop*, Oct. 2008, pp. 1–8.

[13] S. P. Borra, N. Pradeep, N. Raju, S. Vineel, and V. Karteek, "Face recognition based on convolutional neural network," *Int. J. Eng. Adv. Technol.*, vol. 9, May 2020, doi: 10.35940/ijeat.D6658.049420.

[14] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.

[15] V. Radu *et al.*, "Performance aware convolutional neural network channel pruning for embedded GPUs," in *Proc. IEEE Int. Symp. Workload Characterization (IISWC)*, Nov. 2019, pp. 24–34.

[16] S. Sen, V. Natarajan, R. Senguttuvan, and A. Chatterjee, "Pro-VIZOR: Process tunable virtually zero margin low power adaptive RF for wireless systems," in *Proc. 45th ACM/IEEE Des. Auto. Conf.*, Jun. 2008, pp. 492–497.

[17] D. Banerjee, S. Devarakond, S. Sen, and A. Chatterjee, "Real-time use-aware adaptive MIMO RF receiver systems for energy efficiency under BER constraints," in *Proc. 50th ACM/EDAC/IEEE Des. Auto. Conf. (DAC)*, May 2013, pp. 1–7.

[18] S. Sen, "Invited: Context-aware energy-efficient communication for IoT sensor nodes," in *Proc. 53nd ACM/EDAC/IEEE Design Automat. Conf. (DAC)*, Dec. 2016, pp. 1–6.

[19] S. Sen, R. Senguttuvan, and A. Chatterjee, "Environment-adaptive concurrent companding and bias control for efficient power-amplifier operation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 3, pp. 607–618, Mar. 2011.

[20] S. Sen, D. Banerjee, M. Verhelst, and A. Chatterjee, "A power-scalable channel-adaptive wireless receiver based on built-in orthogonally tunable LNA," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 5, pp. 946–957, May 2012.

[21] D. Banerjee, B. Muldrey, X. Wang, S. Sen, and A. Chatterjee, "Self-learning RF receiver systems: Process aware real-time adaptation to channel conditions for low power operation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 1, pp. 195–207, Jan. 2017.

[22] J. Yang *et al.*, "Instinctual interference-adaptive low-power receiver with combined feedforward and feedback control," *IEEE Microw. Wireless Compon. Lett.*, vol. 31, no. 6, pp. 771–774, Jun. 2021.

[23] S. Haykin, *Communication Systems*, 5th ed. Hoboken, NJ, USA: Wiley, 2009.

[24] B. Chatterjee and S. Sen, "A 41.5 pJ/b, 2.4 GHz digital-friendly orthogonally tunable transceiver SoC with 3-decades of energy-performance scalability," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Mar. 2020, pp. 1–5.

[25] N.-C. Kuo *et al.*, "A wideband all-digital CMOS RF transmitter on HDI interposers with high power and efficiency," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 11, pp. 4724–4743, Nov. 2017.

[26] D. Chowdhury, S. V. Thyagarajan, L. Ye, E. Alon, and A. M. Niknejad, "A fully-integrated efficient CMOS inverse class-D power amplifier for digital polar transmitters," *IEEE J. Solid-State Circuits*, vol. 47, no. 5, pp. 1113–1122, May 2012.

[27] A. Paidimarri, P. M. Nadeau, P. P. Mercier, and A. P. Chandrakasan, "A 2.4 GHz multi-channel FBAR-based transmitter with an integrated pulse-shaping power amplifier," *IEEE J. Solid-State Circuits*, vol. 48, no. 4, pp. 1042–1054, Apr. 2013.

[28] V. Mangal and P. R. Kinget, "Sub-nW wake-up receivers with gate-biased self-mixers and time-encoded signal processing," *IEEE J. Solid-State Circuits*, vol. 54, no. 12, pp. 3513–3524, Dec. 2019.

[29] X. Hong, C. Yang, and X. Zhang, "An energy-efficient SAR ADC with a partial-monotonic capacitor switching technique," in *Proc. IEEE 2nd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Mar. 2017, pp. 2050–2054.

[30] J. Lee, C. Kim, S. Kang, D. Shin, S. Kim, and H.-J. Yoo, "UNPU: An energy-efficient deep neural network accelerator with fully variable weight bit precision," *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 173–185, Jan. 2019.

[31] J. Yue *et al.*, "STICKER-T: An energy-efficient neural network processor using block-circulant algorithm and unified frequency-domain acceleration," *IEEE J. Solid-State Circuits*, vol. 56, no. 6, pp. 1936–1948, Jun. 2021.

[32] F. Tu *et al.*, "Evolver: A deep learning processor with on-device quantization–voltage–frequency tuning," *IEEE J. Solid-State Circuits*, vol. 56, no. 2, pp. 658–673, Sep. 2021.

[33] X. Yu *et al.*, "A fully-integrated reconfigurable dual-band transceiver for short range wireless communications in 180 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 50, no. 11, pp. 2572–2590, Nov. 2015.

[34] M.-C. Lee *et al.*, "A CMOS MedRadio transceiver with supply-modulated power saving technique for an implantable brain–machine interface system," *IEEE J. Solid-State Circuits*, vol. 54, no. 6, pp. 1541–1552, Jun. 2019.

[35] T. Karnik *et al.*, "A cm-scale self-powered intelligent and secure IoT edge mote featuring an ultra-low-power SoC in 14 nm tri-gate CMOS," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2018, pp. 46–48.

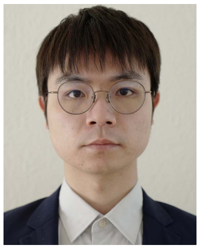
Ningyuan Cao (Member, IEEE) received the bachelor's degree from Shanghai Jiaotong University, Shanghai, China, in 2013, the master's degree from Columbia University, New York City, NY, USA, in 2015, under the supervision of Dr. Yannis Tsividis, and the Ph.D. degree in integrated circuit and algorithm design for EI from the Georgia Institute of Technology, Atlanta, GA, USA, in 2020, under the supervision of Dr. Arijit Raychowdhury.

He is an Assistant Professor with the Department of Electrical Engineering at the University of Notre Dame, Notre Dame, IN, USA. Before joining University of Notre Dame, he worked as a Research Associate with IBM T. J. Watson, New York, for one year. His research interests are in: 1) analog/mixed-signal circuit, digital architecture, and IoT system design for machine learning acceleration/distributed intelligence and 2) custom IC design automation with data-driven methods. His works have been published/presented/reported by primary journals/conferences/press in various fields of solid-state circuit design, microwave, industrial electronics, and so on (e.g., ISSCC, JSSC, TIE, IMS, CICC, TCAS-I, and so on).

Baibhab Chatterjee (Member, IEEE) received the B.Tech. degree in electronics and communication engineering from the National Institute of Technology (NIT), Durgapur, India, in 2011, and the M.Tech. degree in electrical engineering from IIT Bombay, Mumbai, India, in 2015. He is currently pursuing the Ph.D. degree with the School of Electrical Engineering, Purdue University, West Lafayette, IN, USA.

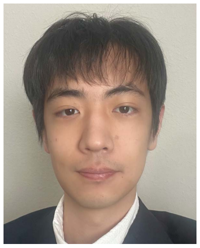
His industry experience includes two years as a Digital Design Engineer/a Senior Digital Design Engineer with Intel, Bengaluru, India, and one year as a Research and Development Engineer with Tejas Networks, Bengaluru. His research interests include low-power analog, RF, and mixed-signal circuit design for secure biomedical applications.

Mr. Chatterjee received the University Gold Medal from NIT, in 2011, the Institute Silver Medal from IIT Bombay in 2015, the Andrews Fellowship at Purdue University from 2017 to 2019, the HOST 2018 Best Student Poster Award (third), the CICC 2019 Best Paper Award (overall), the RFIC/IMS 2020 3MT Award (audience choice), and the Bilsland Fellowship at Purdue University from 2021 to 2022.



Jianbo Liu (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering from Southeast University, Nanjing, China, in 2017, and the M.S. degree in computer engineering from Northwestern University, Evanston, IL, USA, in 2019. He is currently pursuing the Ph.D. degree in electrical engineering with the University of Notre Dame, Notre Dame, IN, USA.

From 2019 to 2021, he was an FPGA Engineer with Ubiquiti Networks, Barrington, IL, USA. Since 2022, he joined Collaborative AIHW Laboratory, Notre Dame. His research interests include privacy in edge devices, accelerators for graph neural networks, and circuit topology design automation.



Boyang Cheng (Graduate Student Member, IEEE) received the B.S. degree from Southeast University, Nanjing, China, in 2020. He is currently pursuing the Ph.D. degree in electrical engineering with the University of Notre Dame, Notre Dame, IN, USA.

From 2020 to 2021, he was a Research Assistant with the National ASIC System Engineering Research Center, School of Electronic Science and Engineering, Southeast University. His research interests include low-power mixed-signal circuit and system design, and high energy-efficient accelerator design for machine learning.

Minxiang Gong (Graduate Student Member, IEEE) received the B.E. degree in information science and engineering from Southeast University, Nanjing, China, in 2017, and the M.S. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2020, where he is currently pursuing the Ph.D. degree.

Since 2018, he has been a Research Assistant with the Department of Electrical and Computer Engineering, Georgia Institute of Technology. His research interests include non-isolated high-voltage dc–dc converters and mixed-signal ICs.

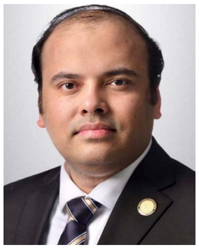
Muya Chang (Member, IEEE) received the M.S. degree in computer science and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, both in 2020.

He is currently a Post-Doctoral Fellow with the Integrated Circuits and Systems Research Laboratory, Georgia Institute of Technology, and is advised by Prof. Arijit Raychowdhury. His research interest includes energy-efficient hardware design for distributed optimizations.

Shreyas Sen (Senior Member, IEEE) received the Ph.D. degree in ECE from the Georgia Institute of Technology, Atlanta, GA, USA, in 2011.

He is currently an Associate Professor in ECE with Purdue University, West Lafayette, IN, USA. He has over five years of industry research experience with Intel Labs, Hillsboro, OR, USA; Qualcomm, Austin, TX, USA; and Rambus, Los Altos, CA, USA. He is the Inventor of the Electro-Quasistatic Human Body Communication, for which he was a recipient of the MIT Technology Review top-10 Indian Inventor Worldwide under 35 (MIT TR35 India) Award. His work has been covered by 100+ news releases worldwide, invited appearance on TEDx Indianapolis, Indian National Television CNBC TV18 Young Turks Program and NPR subsidiary Lakeshore Public Radio. He has coauthored two book chapters, over 135 journals and conference papers, and has 14 patents granted/pending. His current research interests span mixed-signal circuits/systems and electromagnetics for the Internet of Things (IoT), biomedical, and security.

Dr. Sen was a recipient of the NSF Career Award 2020, the AFOSR Young Investigator Award 2016, NSF CISE CRII Award 2017, the Google Faculty Research Award 2017, the Intel Labs Quality Award for industry wide impact on USB-C type, the Intel Ph.D. Fellowship 2010, the IEEE Microwave Fellowship 2008, and seven best paper awards including IEEE CICC 2019 and IEEE HOST in 2017, 2018, and 2019. His work was chosen as one of the top-ten papers in the hardware security field over the past six years (TopPicks 2019). He serves/has served as an Associate Editor for the *IEEE Design & Test*, an Executive Committee Member of the IEEE Central Indiana Section and Technical Program Committee member of DAC, CICC, DATE, ISLPED, ICCAD, ITC, VLSI Design, among others.



Arijit Raychowdhury (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2007.

He joined from the Georgia Institute of Technology, Atlanta, GA, USA, in January 2013, where he was an Associate Professor and held the ON Semiconductor Junior Professorship at the Department from 2013 to 2019. Prior to joining Georgia Tech, he held research positions at Intel Corporation, Portland, OR, USA, for six years and at Texas Instruments, Dallas, TX, USA, for one and a half years. He is the Steve W. Chaddick Chair and a Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology. He holds more than 27 U.S. and international patents and has published over 250 articles in journals and refereed conferences. His research interests include low power digital and mixed-signal circuit design, design of power converters, signal processors, and exploring interactions of circuits with device technologies.

Dr. Raychowdhury is currently a Distinguished Lecturer of the IEEE Solid State Circuits Society (SSCS) and a Mentor for IEEE Young Professionals and IEEE Women in Circuits. He serves on the technical program committee for key circuits and design conferences, including ISSCC, VLSI Symposium, DAC, and CICC. He is the winner of several prestigious awards, including the SRC Technical Excellence Award 2021, the Qualcomm Faculty Award 2020, the IEEE/ACM Innovator under 40 Award, the NSF CISE Research Initiation Initiative Award (CRII) 2015, the Intel Labs Technical Contribution Award 2011, the Dimitris N. Chorafas Award for outstanding doctoral research and best thesis 2007, and several fellowships. He and his students have won 14 best paper awards over the years.