AIMMI: Audio and Image Multi-Modal Intelligence via a Low-Power SoC With 2-MByte On-Chip MRAM for IoT Devices

Zichen Fan, *Graduate Student Member, IEEE*, Qirui Zhang, *Member, IEEE*, Hyochan An, Boxun Xu, Li Xu, *Member, IEEE*, Chien-Wei Tseng, *Graduate Student Member, IEEE*, Yimai Peng, *Member, IEEE*, Andrea Bejarano-Carbo, *Graduate Student Member, IEEE*, Pierre Abillama, *Graduate Student Member, IEEE*, Ang Cao, *Member, IEEE*, Bowen Liu, Changwoo Lee, Zhehong Wang, Hun-Seok Kim, *Senior Member, IEEE*, David Blaauw, *Fellow, IEEE*, and Dennis Sylvester, *Fellow, IEEE*

***Abstract*— In this article, we present an ultra-low-power multi-modal signal processing system on chip (SoC) [audio and image multi-modal intelligence (AIMMI)] that integrates a versatile deep neural network (DNN) engine with audio and image signal processing accelerators for multi-modal Internet-of-Things (IoT) intelligence. In order to get high energy efficiency under resource-constrained IoT scenarios, AIMMI features three efficiency-boosting techniques: 1) 2-MB on-chip non-volatile magnetoresistive RAM (MRAM) to store all DNN weights with MRAM-cache microarchitecture that incorporates dynamic power gating to reduce both leakage and dynamic power consumption; 2) a deliberate power management scheme that enables optimized power modes under different operating situations; and 3) a novel reconfigurable neural engine (NE) with energy-efficient dataflow for comprehensive DNN instructions. Fabricated in TSMC 22-nm ultra-low leakage (ULL) technology with MRAM, AIMMI achieves up to 3–10-TOPS/W peak energy efficiency and consumes only 0.25–3.84 mW. It demonstrates convolutional neural network (CNN), generative adversarial network (GAN), and back-propagation (BP) operations on a single accelerator SoC for multi-modal fusion, outperforming state-of-the-art DNN processors by 1.4×–4.5× in energy efficiency.**

***Index Terms*— Machine learning, non-volatile memory, signal processing, system-on-chip (SoC), ultra-low power.**

I. Introduction

IN RECENT years, the Internet of Things (IoT) has become increasingly prevalent, with widespread applications including smartphones, voice assistants, smart surveillance, robotics, and more. Image and audio signal processing has emerged as essential components within these applications, attracting significant interest. As deep neural networks (DNNs) gain popularity, DNN-based image and audio processing has become integral to intelligent IoT systems [1], [2], [3], [4], [5]. However, the majority of current energy-/power-constrained IoT devices primarily perform relatively simple tasks such as image white balancing or audio denoising. Computationally intensive DNN-based analysis typically occurs in the cloud [6], [7], [8]. This conventional processing flow [see Fig. 1 (top)] not only demands high data transmission bandwidth for (raw) data but also suffers from high energy consumption and weak privacy protection for wireless data offloading.

To address these challenges, smart image and audio sensors have been proposed to reduce data transmission bandwidth by performing simple pre-processing tasks such as motion detection [9] for images and voice activity detection (VAD) [10] for audio signals. This approach ensures that only relevant data are transmitted and analyzed, reducing overall latency and power consumption. However, in-sensor/near-sensor pre-processing is limited in its ability to execute complex DNN-based algorithms due to the fixed sensor circuit structure and computing limitations [11], [12], [13], [14], [15]. Consequently, we aim to develop a fully-at-edge processing flow [see Fig. 1 (bottom)] by constructing an intelligent processor capable of end-to-end DNN-based analysis entirely at the IoT edge. Ultimately, only a small quantity of useful, encrypted information is sent to the cloud for further processing, resulting in reduced transmission bandwidth requirements, lower energy consumption, and enhanced privacy protection.

In recent years, there has been a significant surge in the development of low-power intelligent edge processors. These processors are particularly aimed at enhancing image and audio applications through edge computing. For image applications, An et al. [1] introduced an ultra-low-power vision processor capable of performing tasks such as person detection, face detection (FD), and face recognition (FR), operating at 170 μW. However, this processor has a limitation

Manuscript received 7 June 2023; revised 11 January 2024 and 20 May 2024; accepted 25 May 2024. Date of publication 20 June 2024; date of current version 26 September 2024. This article was approved by Associate Editor Meng-Fan Chang. This work was supported by Sony Semiconductor Solutions Corporation/Sony Electronics Inc. *(Corresponding author: Zichen Fan.)*

Zichen Fan, Qirui Zhang, Chien-Wei Tseng, Andrea Bejarano-Carbo, Pierre Abillama, Ang Cao, Changwoo Lee, Hun-Seok Kim, David Blaauw, and Dennis Sylvester are with the Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: zcfan@umich.edu).

Hyochan An is with Apple, Inc., Cupertino, CA 95014 USA.

Boxun Xu is with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106 USA.

Li Xu is with NVIDIA Corporation, Santa Clara, CA 95051 USA.

Yimai Peng is with Qualcomm, Inc., Raleigh, NC 27617 USA.

Bowen Liu is with Zoom, Inc., San Jose, CA 95113 USA.

Zhehong Wang is with Meta Platforms, Inc., Menlo Park, CA 94025 USA.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSSC.2024.3410306.

Digital Object Identifier 10.1109/JSSC.2024.3410306

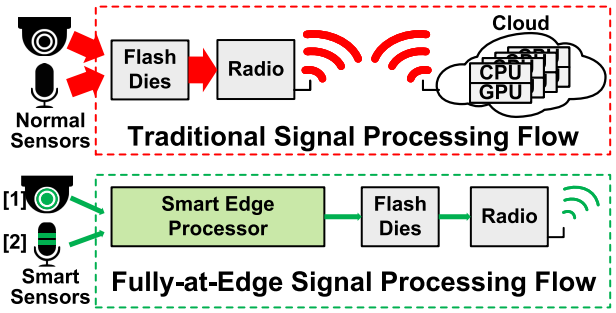


Fig. 1. Traditional signal processing flow versus fully-at-edge signal processing flow.

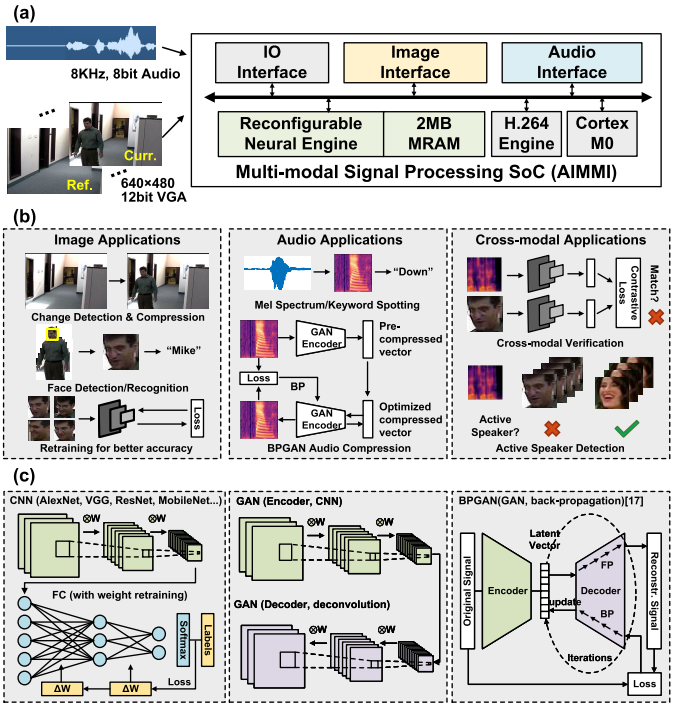


Fig. 2. (a) Multi-modal signal processing SoC overview. (b) Supported applications by AIMMI: image applications, audio applications, and cross-modal applications. (c) Supported neural network algorithms, including CNN, FC, GAN, and their BPs.

in its processing speed, which is only at 0.2 frames/s. A prior work [3] developed a sub-mW edge processor tailored for face analysis. This processor utilizes a binary decision tree (BDT) for FD and a convolutional neural network (CNN) for FR. However, it lacks an image pre-processing interface, which constrains its application in various scenarios. In the domain of audio processing, the design in [16] shows a 141-μW edge processing with VAD and speech recognition. This processor, however, is limited to supporting only binary neural networks, which hinders its effectiveness in more complex scenarios, such as audio compression (AC). In addition, Giraldo et al. [4] show a 10-μW audio processor designed for keyword spotting and speaker verification. However, only long short-term memory (LSTM)-like neural network structure is supported, which lacks the programmability for the users.

Unlike most aforementioned edge signal processors, which are limited to handling a single type of signal [1], [2], [3], [4], our smart IoT processor is capable of simultaneously receiving and processing both image and audio data. This enables multi-modal signal fusion for enhanced scene understanding. Fig. 2 presents our multi-modal processing system on chip (SoC) [audio and image multi-modal intelligence (AIMMI)] alongside its supported applications and algorithms. Designed for resource-constrained IoT scenarios, AIMMI features μW-level stand-by (STB) power and mW-level active power consumption while achieving state-of-the-art energy efficiency. The design details and techniques are elaborated on in Sections III and IV of this article.

This article is organized as follows. Section II provides an overview of the AIMMI SoC, including its working scenario, supported applications, and supported algorithms. Section III presents the SoC design architecture. Section IV introduces three main novel techniques employed in the AIMMI SoC to enhance the system energy efficiency, while Section V discusses the real chip measurement results and compares them with state-of-the-art processors. Finally, Section VI concludes this article.

II. Application Overview

The AIMMI SoC can simultaneously receive 12-b per pixel video graphics array (VGA) images and 8-kHz 8-b per sample audio signals using dedicated interfaces, as illustrated in Fig. 2(a). The SoC primarily comprises an audio and image interface for signal fetching and pre-processing, a neural engine (NE) with a magnetoresistive RAM (MRAM) macro for intelligent signal post-processing, an H.264 engine for image compression, and other auxiliary units for system and power control. The detailed architecture will be explained in Section III. Owing to this structure, the SoC can perform image- or audio-only applications, as well as (conditionally) execute image-audio fusion applications to further enhance detection and recognition credibility, as shown in Fig. 2(b).

For image applications, the image interface first stores a reference frame in JPEG compressed format and performs change detection by comparing the current frame with the reference frame to identify altered image parts. Once the changed region is detected, the neural network-based FD and FR are executed in the NE. In addition, the NE can support on-chip re-training to improve accuracy as more data and labels are obtained.

For audio applications, the audio interface features hardware support for efficiently computing fast Fourier transform (FFT)/Mel-spectrum to obtain the audio's frequency domain information. Utilizing the spectrum, the NE can perform keyword spotting. Another audio application supported by AIMMI is back-propagation GAN (BPGAN)-based AC, which employs a similar algorithm to BPGAN [17] using generative adversarial network (GAN) and back-propagation (BP) to compress audio sequences. Initially, the original signal passes through the GAN encoder to obtain a compressed latent vector. This latent vector is then fed through the GAN decoder to produce the reconstructed signal. Comparing the original and reconstructed signals yields a loss, which allows BP through the GAN decoder to optimize the compressed latent vector. Consequently, only the compressed latent vector is updated via BP without altering any weights in the GAN decoder. After

several iterations, a better compressed latent vector is obtained for storage and transmission. The compression ratio can reach up to 32× on 8-kHz, 8-bit audio sequences with negligible audio quality loss.

For audio–image cross-modal fusion applications, we demonstrate cross-modal verification (CMV) by considering both human face and voice, modifying the algorithm from [18] to detect if the sound and face are matched. The audio spectrum and face image pass through two separate CNNs, and the contrastive loss between the two feature vectors indicates whether the face and audio match. Moreover, active speaker detection [19] can identify the speaker using several face image frames when more than one person is present. In our implementation, all three types of applications are evaluated in a person-of-interest (PoI) identification scenario, which will be presented in Section V.

To support various NN-based applications ranging from detection and recognition to signal compression, the NE accommodates different types of DNN operations, as depicted in Fig. 2(c). First, it supports various CNN model structures with reconfigurable channel sizes and kernel dimensions. In addition to conventional operations such as convolution (CONV), fully connected (FC), pooling, and nonlinear functions, the AIMMI NE can perform element-wise operations for ResNet [20], depth-wise separable CONV for MobileNet [21], and FC weight updates for on-chip training. A typical GAN structure involves a CNN-based encoder and a decoder that requires deconvolution (DECONV). The AIMMI NE supports a dedicated instruction to efficiently perform DECONV operations by skipping zero multiplications. For BPGAN, all operations related to BP are supported in the AIMMI NE. Our configurable NE employs dedicated efficient dataflows for different NN operations, as detailed in Section IV.

III. Architecture

Fig. 3 shows the overall SoC architecture. All sub-blocks communicate with each other using an advanced high performance bus (AHB). The Cortex-M0 is used as a central controller to program and launch different accelerators and to perform data pre-processing such as image resizing and interpolation. The instructions and data of Cortex-M0 are stored in the 64-kB M0 memory. There are four main accelerators (functional blocks: the audio interface is for audio sequence stream-in and spectrum generation). The image interface is for image frame stream-in, change detection, white balancing, and JPEG compression. The NE enables efficient DNN-based signal post-processing and on-chip learning. The H.264 engine is used for customized H.264 intra-frame compression on an arbitrary (non-rectangular) shaped region-of-interest (RoI) of images. After the signal processing, useful data are extracted and sent out through the flash interface and stored in off-chip flash memories. For the power supply, we use different voltages for logics (VDD_LOGIC, 0.44–1.0 V), static random-access memory (SRAM)/MRAM core (VDD_CORE, 0.6–1.0 V), and MRAM IO (VDIO_MRAM, 1.8–3.3 V). The design details of every block and power domain are illustrated in Sections III-A–III-C.

A. Neural Engine

The NE is a programmable DNN accelerator that supports various operations including (depth-wise) CONV, DECONV, FC, BP, and so on. In this section, we introduce NE architecture from three perspectives: computation architecture, memory architecture, and instruction architecture.

1) NE Computation Architecture: The NE supports DNN weights both in uncompressed (8 b/weight) and Huffman-compressed (around 2 b/weight) formats [22]. When uncompressed, weights are directly read from the weight cache (WC) SRAM and stored in the row buffer during computation. For the compressed weights, the Huffman decoder decodes the compressed weight on the fly when loaded from the memory without stalling the computation. The decision to utilize 8-bit precision for both weights and activations in neural networks is primarily driven by the empirical evidence that suggests a negligible loss in accuracy when these networks are quantized to 8-bit representations [23]. The main computation unit is an $8 \times 8 \times 8$ processing element (PE, each with an 8-bit multiply accumulate (MAC) unit) array, which enables activation and weight reuse via inter-PE connections. The detailed dataflow is explained in Section IV. There are two types of PEs: the top $1 \times 8 \times 8$ PEs are multi-functional PEs (MPEs) that support both MAC and max/average pooling operations. Since the data reuse rate of pooling is less than CONV, the rest of $7 \times 8 \times 8$ PEs only support MAC operation for saving logic complexity. The non-linear unit is responsible for supporting non-linear functions in forward inference and BP operations including rectified linear unit (ReLU), Sigmoid, and Tanh in a single-instruction-multiple-data (SIMD) way. The bias unit is for bias addition, where the bias values are read from the bias memory. The batch-normalization (BN) unit performs BN with pre-calculated matrix mean, variance, and coefficients.

2) NE Memory Architecture: The NE incorporates a 2-MB MRAM macro for the on-chip weight storage and a 1.5-MB multi-bank SRAM activation memory to store all activations/feature maps for BP. For simpler applications such as FR and keyword spotting with low memory footprints requiring only partial activation memory, unused SRAM banks are power-gated to reduce leakage power. The MRAM macro is paired with an MRAM cache implemented with SRAMs, enabling a dynamic power-gating scheme (details provided in Section IV). In addition, ping-pong memory structures for local memory and row/col buffers facilitate non-blocking pipelining between data movement and computation, thereby increasing PE utilization.

3) NE ISA: As shown in Fig. 2, the NE supports various neural network layers. Consequently, we designed a custom complex instruction set computer (CISC) architecture depicted in Fig. 4 to include different NN operations. The NE instruction set architecture (ISA) encompasses both forward propagation and backward propagation for CONV, depth-wise CONV (DWCONV), FC layer [dense fully connected (DFC)], pooling (POOL), element-wise operations (ELEMENT), non-linear functions (NONLINEAR), and DECONV. It also features data movement instructions, such as moving activations from activation memory to local memory (MOV), loading Huffman

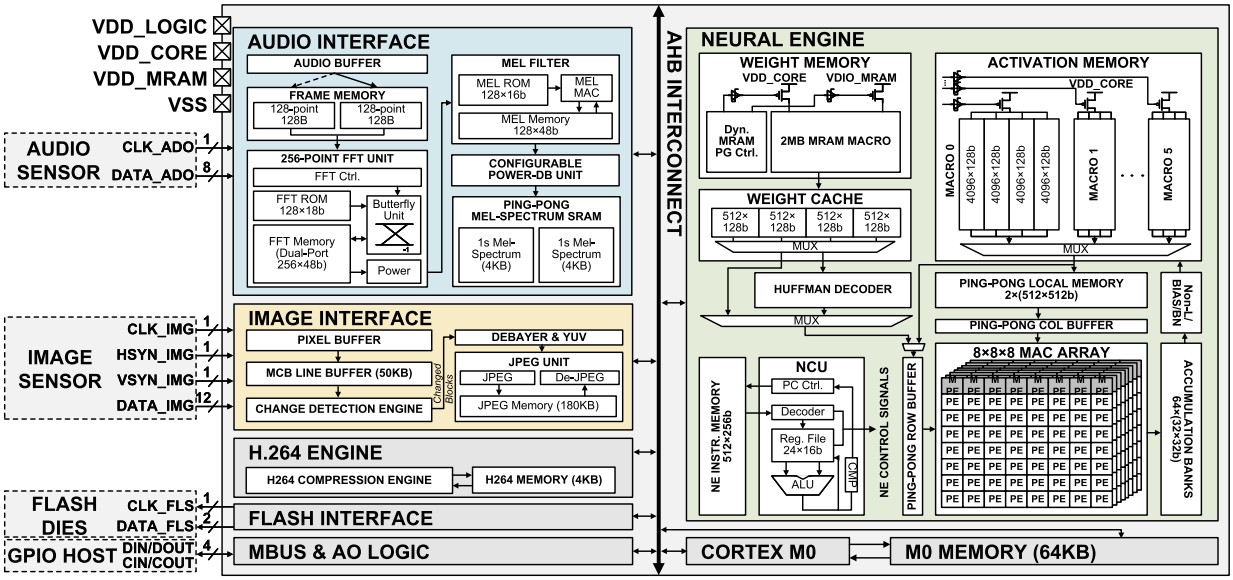


Fig. 3. Multi-modal processing SoC architecture overview.

OPCODE	OPERATION	DESCRIPTION
1	MOV	Move data
2	CONV	Configurable convolution (k, s···)
3	POOL	Max/average pooling
4	DECONV	Deconvolution
5	DWCONV	Depth-wise convolution
6	DFC	Dense fully-connected
7	CONV_BP	Convolution backprop.
8	S_CONV_BP	S=2 conv backprop.
9	POOL_BP	Pooling backprop.
10	DECONV_BP	Deconvolution backprop.
11	DWCONV_BP	Depth-wise conv backprop.
12	S_DWCONV_BP	S=2 dwconv backprop.
13	DFC_BP	Dense FC backprop.
14	DFC_W_UPT	Dense FC weight update
15	ELEMENT	Element-wise add/mult
16	NONLINEAR	ReLU, tanh, sigmoid, ReLU_BP...
17	LD_HUFF	Load huffman table
18	LD_BIAS	Load bias to bias mem
19	LD_WEIGHT	Load weight to w cache
31	NCU_{}	NCU RISC instructions

NCU RISC instruction set

sOP	INSTR.	sOP	INSTR.	sOP	INSTR.
1	ADD	9	OR	17	BNE
2	ADDI	10	XOR	18	BLT
3	SUB	11	NAND	19	BGT
4	SUBI	12	LSR	20	BLE
5	MULT	13	LSL	21	BGE
6	MULTI	14	LDS	31	HALT
7	MULTS	15	STS	0	NOOP
8	AND	16	BEQ		

Fig. 4. NE ISA.

tables for weight decompression (LD_HUFF), loading weights (including bias) from MRAM to WC (LD_WEIGHT), and loading bias from WC to bias memory (LD_BIAS). Each NE CISC instruction is 214-bit long and includes an instruction/operation ID and control fields, such as kernel size, input/output channel size stride, row/column/channel starting/ending point, and write-back address.

To control the NE independently without the direct involvement of the Cortex-M0 core, we designed a lightweight NE control unit (NCU) to fetch and decode instructions from the NE instruction memory. The NCU has its own reduced instruction set computer (RISC)-like instructions [see Fig. 4 (right)], including arithmetic, branch, for-loop, and other control flow-related instructions that simplify the control and reduce the total number of instructions. Each NCU instruction is 31-bit long, and therefore, each 256-bit memory word can store up to eight NCU RISC instructions. Algorithm 1 presents an example NCU program to execute a CONV layer. Based on our MAC array architecture, a single CONV instruction can generate a maximum output feature map size of $8 \times 8 \times$ [number of output channels (OCs)]. To support a larger feature map, for-loop can be used to run multiple CONV instructions. Our custom assembler converts an assembly code (e.g., Algorithm 1) to executable bitstreams, which are stored in the NE instruction memory. The Cortex-M0 and the NCU are used for control purposes rather than computation. All neural network-based computations are performed in the NE's PE array. As shown in Fig. 5, the control overhead is minimal and has negligible impact on the utilization of the NE.

B. Audio Interface

The audio interface (see Fig. 3, blue block) performs audio feature extraction to generate FFT/Mel-spectrum for different audio applications. The input audio sequence is 8 kHz, and incoming samples are divided into 256-sample (32 ms) long windows/frames. The ping-pong frame buffers enable continuous processing of the audio and also enable half-window overlapping between two consecutive windows. After framing, Radix-2 FFT is executed on each 256-point audio window. Since the on-the-fly Mel-spectrum generation only requires less than 16-ms latency (16k cycles with 1 MHz) for each audio window, we only use one arithmetic unit to perform all butterfly calculations to save area and power. This approach requires 2048 cycles for one 256-point FFT

Algorithm 1 NE Code Example for CONV

```
1:  procedure CONVOLUTION LAYER
2:      #Load weights from MRAM to weight cache
3:      LD_WEIGHT(addr_MRAM, addr_WeightCache, ...)
4:  convolution_loop:
5:      #Load activation from activation memory to local
        memory. Then do convolution.
6:      MOV(row, col, addr_ActMem, addr_LocalMem, ...)
7:      CONV(row, col, channel, kernel_size, stride, ...)
8:      #Register control in NCU for for-loop
9:      NCU_ADDI, row, block_size
10:     NCU_BLT  row, row_size, convolution_loop
11:     NCU_MOVI, row, initial_row
12:     NCU_ADDI, col, block_size
13:     NCU_BLT  col, col_size, convolution_loop
14:     NCU_MOVI, col, initial_col
```

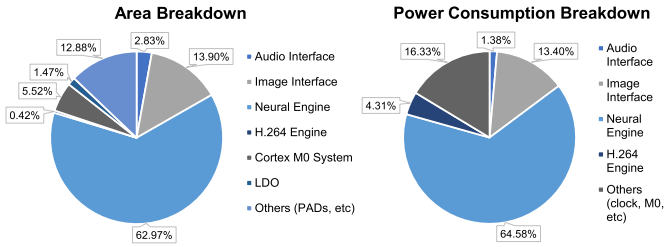


Fig. 5. Area and power consumption breakdown (in average) of AIMMI processor. The area breakdown is based on the layout of the SoC, and the power consumption breakdown is based on post-P&R simulation results.

calculation. Each FFT coefficient is 18 bits (9 bits for each real and imaginary part), and we implement a 1.5-kB dual-port FFT memory to store the intermediate and final results. Mel filtering is a simple multiply-accumulation operation, and our mel filter implementation can support 32 or 64 mel channels for different accuracy requirements. Finally, the mel-filtered results go through a log-2 power-to-dB unit producing the mel spectrum results that are stored in an 8-kB ping-pong memory for the subsequent NN processing.

C. Image Interface, H.264 Engine, and Flash Interface

In this work, we adopt the same architecture of the image interface and H.264 engine employed in [1]. The image interface features a change detection and on-the-fly JPEG compressed-memory [1] to temporarily store VGA frames in a compressed format. Only the change-detected macro-blocks are stored and processed as RoIs. The H.264 engine performs image compression [1] on non-rectangular RoIs for compact storage in on-/off-chip memory. The flash interface is for storing final processed results, such as compressed audio vectors, to off-chip flash memories rather than for storing the weights of the DNN algorithms. All the weights and intermediate activations required for the DNN processing are stored on-chip.

IV. Energy Efficiency-Boosting Techniques

Since the SoC is specifically designed for edge IoT applications, improving energy efficiency is our primary goal. We implement three system energy efficiency-boosting techniques in the proposed SoC. The first is an optimized memory architecture using MRAM, the second is the multi-mode system-level power management scheme, and the last is the energy-efficient dataflow for the NE.

A. MRAM in Low-Power System

1) MRAM Challenges and Opportunities: A non-volatile memory such as MRAM has become a promising candidate for the on-chip weight storage memory of neural network-based IoT devices for its high density and non-volatility. It eliminates the need to reload data from off-chip after the chip is powered-up. Compared to other embedded non-volatile memories [such as resistive random-access memory (RRAM)], MRAM offers distinct advantages, including longer endurance and lower read-out, as well as STB power [24], [25]. These attributes make MRAM a more suitable choice for our weight memory needs in various applications. The embedded MRAM used in our system is a foundry-provided proprietary IP with a total of 2-MB capacity, fabricated using a 22-nm ultra-low leakage (ULL) logic process. This MRAM demonstrates a read speed with an access time of 10 ns and a read power of 0.8 μA/MHz/bit. Moreover, the MRAM macro is characterized by a write endurance of 100k cycles and a data retention period of up to ten years [25].

Recently, several SoCs implemented embedded MRAM as the instruction/data memory, demonstrating its advantage as a non-volatile on-chip memory [26], [27], [28], [29], [30], [31]. However, from the energy perspective in resource-constrained IoT scenarios, MRAM-based memory system optimization is still understudied. First, MRAM features a very high write power (around 700 pJ/bit) and a slow write speed (1.5 kByte/ms) [25]. Therefore, MRAM in our system is used as NE weight memory for read-only operations during chip processing after one-time programming. Another main issue is that MRAM exhibits significant leakage power when it stores all weights on the chip for IoT devices. Fig. 6 illustrates a simulation comparison between SRAM and MRAM of equivalent sizes in the context of an NE weight memory. In this simulation, it is assumed that the NE performs CONV with its active time accounting for 20% of the total simulation time. Initially, we employed always-on (AO) 2-MB SRAM as the weight memory, which led to substantial leakage, resulting in an average total power exceeding 800 μW. To counteract the elevated leakage power, power gating the SRAM when the NE is idle was considered. However, this strategy has its pitfalls as reactivating the SRAM necessitates reloading weights from an off-chip memory such as DRAM, thereby incurring significant energy costs due to off-chip data transactions. One possible solution is substituting SRAM with MRAM, but this approach presents complications, primarily a heightened dynamic readout power and significant leakage power. Therefore, maintaining MRAM in an AO state fails to offer energy savings as shown in the third case in Fig. 6. To address these concerns, we propose an MRAM-cache architecture with dynamic power gating that effectively reduces both leakage and dynamic power. A more

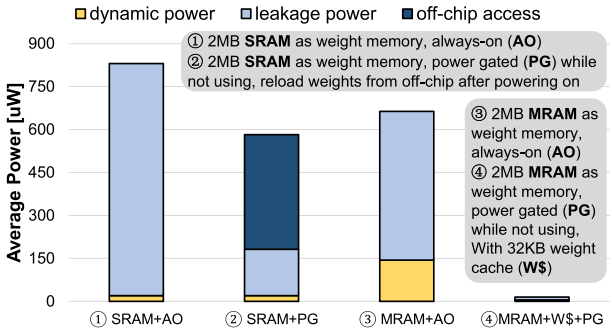


Fig. 6. SRAM and MRAM average power comparison in NE weight memory under different design settings.

detailed explanation of this proposed design will follow in Sections IV-A2 and IV-A3.

2) MRAM-Cache Architecture and Dynamic Power Gating: To address the aforementioned MRAM-related challenges, our SoC employs an MRAM-cache architecture and dynamic power-gating scheme depicted in Fig. 7. Given that 2-MB MRAM exhibits a relatively higher dynamic read power compared to SRAM, our design strategy ensures weights are not directly read from MRAM but rather from a smaller 32-kB SRAM WC, as illustrated in Fig. 7(a). This methodology capitalizes on the weight reuse characteristics inherent in DNN algorithms, enabling a weight fetched once from MRAM to be utilized (read) multiple times from the WC. The size of the WC is determined based on the layer structures of the neural network, as detailed in Table I. Opting for a smaller WC size would yield benefits in terms of reduced read-out power. However, its limited capacity would require more frequent access to the MRAM for weight reloading, consequently increasing the overall power consumption. Conversely, a larger cache size could decrease the number of MRAM accesses, but this comes with a trade-off of higher read-out and leakage power. After careful consideration, a 32-kB WC size was chosen. This size strikes a balance between low read-out power and overall system leakage power while still being capable of storing an entire layer of neural network weights. Thanks to the regularity and determinism of weight access in neural network processing, the required weights are fully identified in advance of computation. The weight movement instruction (LD_WEIGHT) can directly set start and end MRAM addresses. Consequently, the hit rate of the WC is 100%. As depicted in Fig. 7(c), this architecture drastically diminishes dynamic energy consumption.

Nevertheless, the significant leakage power associated with MRAM remains unaddressed. To mitigate this, we exploit the non-volatile nature of MRAM to implement a dynamic power-gating scheme, allowing the MRAM to be power-gated whenever it is advantageous to do so to reduce the overall power consumption. Fig. 7(b) shows the instruction waveform, power control signals, and measured MRAM transient current using our dynamic power-gating scheme. Within this scheme, MRAM is power-gated until the NE executes the weight loading (LD_WEIGHT) instruction. During the LD_WEIGHT operation, MRAM powers up and the weights are read and loaded into the WC, a process that can be triggered during NE instruction decoding in NCU. The measured MRAM VDIO

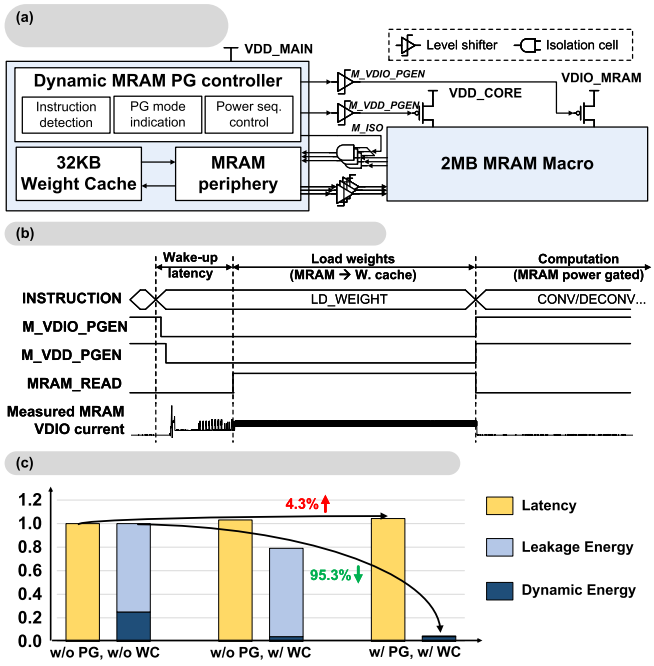


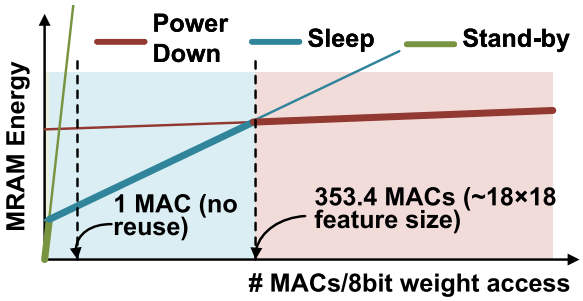
Fig. 7. MRAM-cache architecture and dynamic power-gating scheme. (a) MRAM-cache architecture and dynamic power gating. (b) Control waveform and measured MRAM current. (c) MRAM-cache system normalized performance (simulation).

current waveform reflects the entire MRAM-cache sequence. During NN processing, weights are read from the WC (while MRAM is in either sleep (SLP) or power down (PD) state), thus decreasing memory readout power in comparison to direct MRAM accesses. Based on realistic neural network execution simulation, which is presented in Fig. 7(c), the combination of weight caching and power gating results in a reduction of weight readout power by 95.3%, with only a marginal increase (4.3%) in operation time due to the latency in MRAM wake-up and the overhead of cache loading time.

3) MRAM Power-Gating Mode Selection: The MRAM macro offers various power-gating/saving modes, each with distinct power-up energy and leakage power characteristics. These are: 1) PD mode, where both the peripherals and MRAM array are power-gated; 2) SLP mode, where only the MRAM array is powered off while peripherals remain on; and 3) STB mode, which involves no power gating. The optimal power-gating mode selection is intended to minimize the overall MRAM energy, accounting for both power-gating leakage energy and power-up overhead energy, as depicted in Fig. 8. The boundary decision diagram demonstrates that the STB mode is never selected, as it exhibits higher energy consumption even when no weight reuse occurs under our test scenario. The choice between SLP or PD mode is contingent upon the reuse factor of the cached weight. Our analysis concludes that the PD mode is preferable when each cached weight is reused for $\geq$353.4 MAC operations. Otherwise, the SLP mode is advantageous due to its lower power-up energy overhead, which offsets its higher leakage.

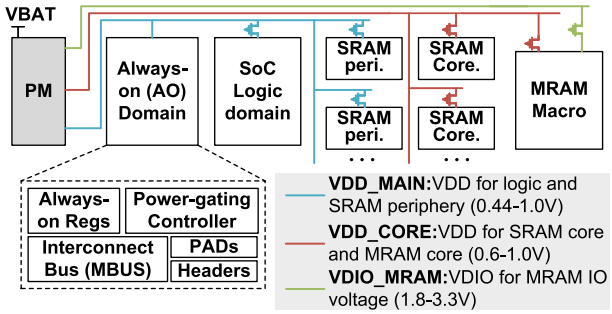
B. Overall Power Domain Design

Acknowledging the sparse nature of most events such as an intruder detection by a security system as detailed



Power-gating Mode	VDD_ CORE	VDIO_ MRAM	Power-up Energy	PG leakage power
Power down (PD)	OFF	OFF	130nJ	0.2μW
Sleep (SLP)	OFF	ON	4nJ	22.5μW
Stand-by (STB)	ON	ON	0nJ	519μW

Fig. 8. MRAM power-gating mode selection and boundary decision.



Power Mode	AO domain	Logic domain	M0 SRAM	Other SRAMs	MRAM	Leakage Power
Deep Power Down	ON	OFF	OFF	OFF	OFF	0.46uW
Sleep Mode	ON	ON	ON	OFF	OFF	70uW
Working Mode	ON	ON	ON	ON	OFF	360uW
MRAM Access Mode	ON	ON	ON	ON	ON	879uW

Fig. 9. System power domain design and power mode comparison. The measured leakage power under nominal voltage settings (VDD_LOGIC = 0.6 V, VDD_CORE = 0.8 V, and VDIO_MRAM = 2.5 V) of different power modes is shown in the table.

in [2], not all parts in the SoC need to be turned on during processing. Therefore, we carefully designed the power domains and power modes for the AIMMI SoC. Fig. 9 presents the SoC's power domain design. It comprises three different voltages: for logics (VDD_LOGIC, 0.44–1.0 V), for the SRAM/MRAM core (VDD_CORE, 0.6–1.0 V), and for MRAM IO (VDIO_MRAM, 1.8–3.3 V). The table in Fig. 9 provides an in-depth depiction of the power domain and power modes. The AO domain contains AO registers, a power-gating controller, an interconnect bus to other chips, pads, and headers. It exhibits only 0.46-μW leakage power consumption, making it a desired mode for scenarios with extremely sparse activation events. However, due to the power gating of SRAM, the AO mode fails to preserve Cortex-M0 programs in the SRAM, which need to be reloaded upon wake-up. For the faster resumption of the Cortex-M0 programs, an SLP mode has been implemented. This mode maintains power to the logic and Cortex-M0 instruction SRAM, thus preserving the initial data and instructions. In comparison to the working mode where all SRAMs are active, SLP mode can curtail STB leakage power by up to 81% (from 360 to 70 μW). In the working mode, data SRAMs are powered on, while the MRAM is predominantly power-gated to mitigate overall leakage power, all the while preserving the NE weight data. The MRAM is only activated (MRAM access mode) when the NE requires weight transfer to the WC. When the MRAM is active, the system's maximum leakage power surges to 879 μW from 360 μW of the working mode. The increased system power of 519 μW is primarily attributed to the MRAM leakage power. Recognizing this issue, our MRAM-cache architecture and dynamic power gating approaches focus on reducing the duration for which the MRAM operates in this high-power mode.

C. Efficient Dataflow in Neural Engine

1) Output Stationary Dataflow: Fig. 10 shows the energy-efficient, computation-skipping dataflow scheme implemented in NE for different instructions. The base dataflow is output stationary, which is used in CONV, stride convolution, and DWCONV, as shown in Fig. 10 (left). For our 8 × 8 × 8 PE array in NE, 64 input weights are shared within eight PEs along PE array rows, and 64 input activations are shared within eight PEs across the OCs direction. In one CONV kernel computation (3 × 3 for example), the partial multiplication output is stored in registers in each PE, and the partial result of each PE is accumulated along the input channel direction. Finally, 64 final CONV sums (in 8 × 8 × 8 PEs) are sent to the accumulation memory in parallel.

2) Zero-Skipping Dataflow: It is discussed in [32] and [33] that an output stationary dataflow designed for regular CONV is inefficient for DECONV; hence, it requires a modification for efficient DECONV execution. DECONV primarily involves two steps: 1) zero-padding the input feature map by inserting zeros in every other row and column and 2) performing a standard CONV with a stride of 1 on the expanded feature map. As a result, numerous ineffective computations are carried out due to the presence of zero operands. Fortunately, zero computations have regular patterns that provide an opportunity for us to skip those. Fig. 10 (mid) shows four different (zero) computation patterns. For the first CONV (green), only A1, A2, B1, and B2 are non-zero operands that contribute to the result. The following CONV after sliding the kernel filter horizontally by one step involves only two effective operands A2 and B2 and so on. We observe that the CONV operations in the DECONV are the repetition of those four computation patterns. Furthermore, the non-zero weights of the kernel filter are exclusive among these modes. Therefore, for DECONV, we implement a zero-skipping dataflow that exploits the deterministic repetitive pattern of zero padding/skipping to increase throughput by 4× by only computing non-zero values in each PE. Since the stride CONV BP, S_CONV_BP, uses a similar computation scheme (first zero-padding then normal CONV), we also use the zero-skipping dataflow for executing S_CONV_BP.

3) Zero-Gating Dataflow: The ineffective zero multiplication not only lies in DECONV but also happens in

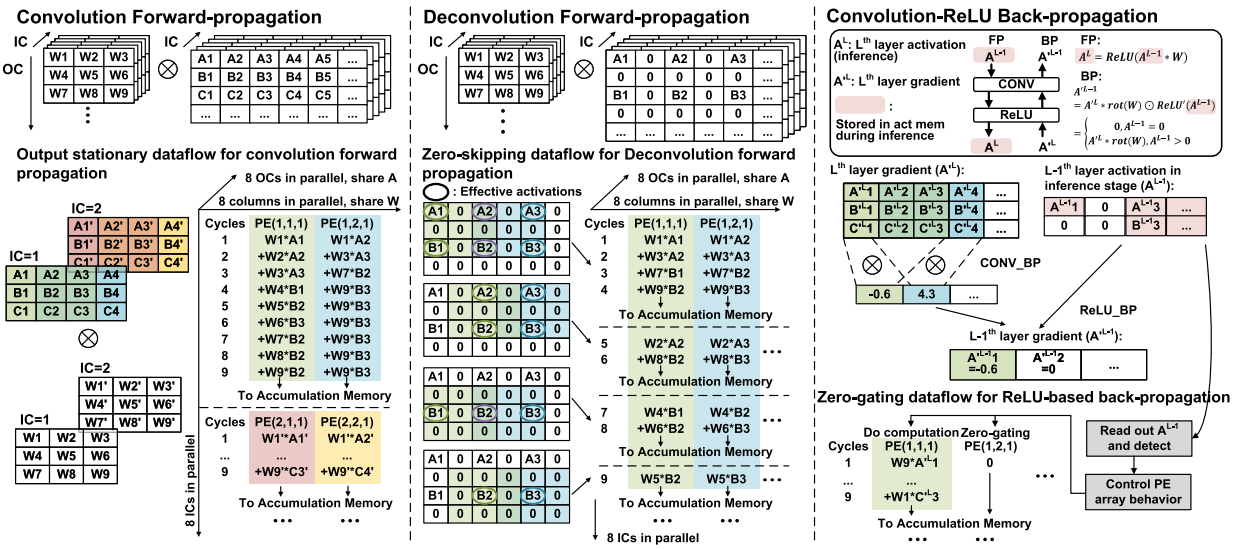


Fig. 10. Efficient NE dataflow for different neural network operations. Left: Base output stationary dataflow. Mid: Zero-skipping dataflow. Right: Zero-gating dataflow.

ReLU-based BP operations. As shown in Fig. 10 (right), the gradient in BP is zero when the related inference activation is zero in the forward propagation (i.e., inference) because of the element-wise multiplication with ReLU in gradient calculation. During the inference in NE, the previous inference results are all stored in the activation memory as we designed the activation memory large enough for our supported applications in Fig. 2. During BP and calculating $L-1$th layer's gradient (A'^{L-1}), the Lth layer's gradient (A'^{L}) is read out from local memory waiting for computation. In the meantime, the stored $L-1$th layer's activation (A^{L-1}) is directly read out from activation memory for the gradient computation. If zero is detected, the related PE column is set to idle (inactive) and just bypasses zero to the accumulation memory to save energy.

V. Measurement Results

A. Chip Implementation

The AIMMI SoC is fabricated in a 22-nm ULL technology incorporating MRAM, with a total area of 12 mm^2. To quantify the performance and efficiency scaling with the supply voltage, we evaluated AIMMI using CONV operations to identify the optimal energy efficiency point. Fig. 11 shows the voltage–frequency–efficiency tradeoff of the chip for CONV operations. The SoC attains peak energy efficiency at a VDD_MAIN of 0.46 V and 1.2 MHz with a total system power consumption of 387 μW. The minimum functional VDD_MAIN is 0.44 V with the lowest power consumption of 0.25 mW. The gray line in Fig. 11 represents the energy efficiency of AIMMI without MRAM dynamic power gating, illustrating that the proposed MRAM dynamic power gating technique substantially enhances energy efficiency, particularly at lower voltages, by reducing leakage significantly. The V–F scaling curve is presented up to 10 MHz, as this range primarily focuses on the analysis to find the energy-efficient operating point, which is considerably slower than the highest supported frequency of 70 MHz. Operating at frequencies

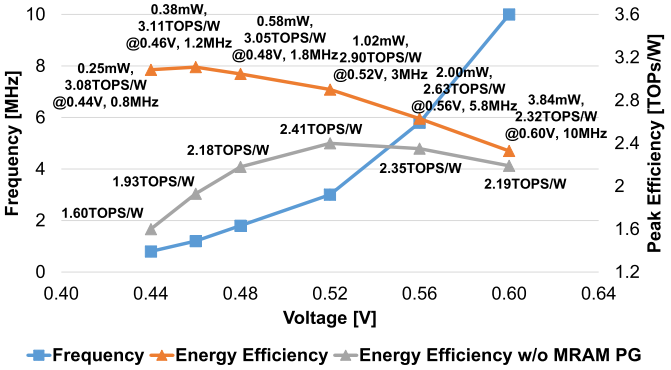


Fig. 11. Voltage–frequency–efficiency scaling for CONV.

lower than 10 MHz is sufficient for the intended applications, which will be elaborated on in Fig. 12 in Section V-B.

Fig. 13 lists the peak energy efficiency for various NN instructions. The peak energy efficiency is obtained by measuring voltage–frequency scaling, as shown in Fig. 11. For convolution/deconvolution/stride-convolution-backpropagation (CONV/DECONV/S_CONV_BP), the efficiencies are 3.1/10/10 TOPS/W, respectively. These computation efficiency numbers include the energy related to memory accesses. The DECONV and S_CONV_BP efficiencies are higher due to the proposed zero-skipping dataflow. For the DECONV operation, the activation sparsity is 75% due to the zero insertion of the input feature map. For the S_CONV_BP, the stride is 2, which also makes the input activation sparsity 75%. CONV BP can achieve 3.7 TOPS/W utilizing the zero-gating dataflow. We assume 50% sparsity in the original inference activation due to the ReLU function. DWCONV and fully connected layer (FCL) exhibit lower efficiencies because of the limited data-sharing/reuse opportunities resulting in only 1/8 utilization of all available PEs in the PE array.

Fig. 14 shows the die photograph and specifications of the AIMMI SoC. The total AIMMI area is 12 mm^2 including

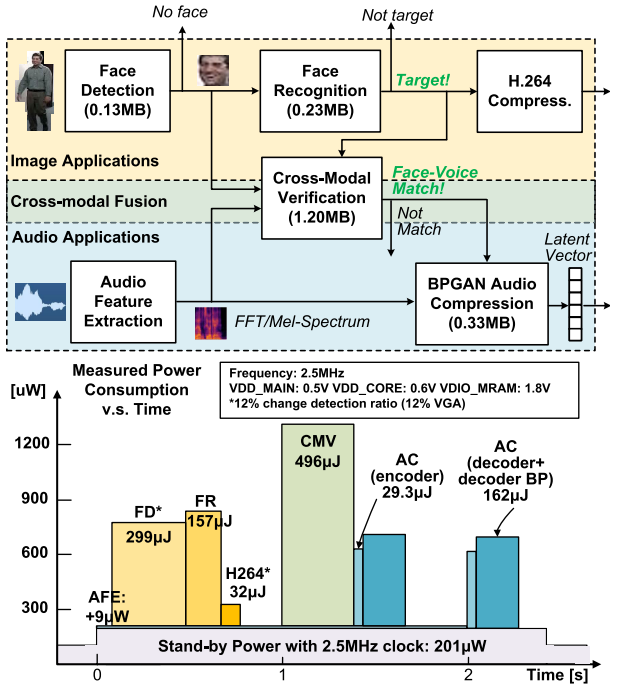


Fig. 12. PoI tracking scenario and its measured transient power consumption.

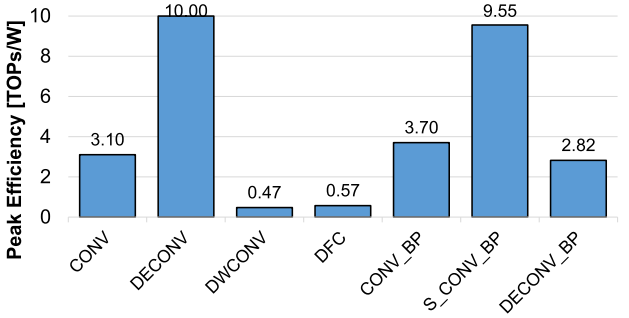
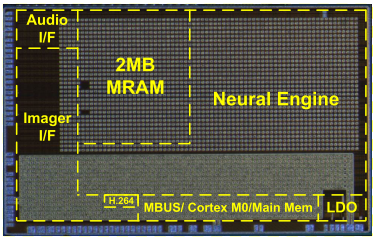


Fig. 13. Peak energy efficiency of AIMMI for different instructions.



	Specifications
Technology	TSMC 22nm ULL
Die Area	4.35mm*2.75mm (12mm^2)
On-chip Memory	1.98MB SRAM + 2MB MRAM
Stand-by Power	460nW@AO 70uW@Stand-by
Active Power Consumption	0.25mW@0.8MHz, 3.84mW@10MHz
Performance	3.1-10TOPS/W

Fig. 14. Die photograph of AIMMI SoC and its characteristics.

4-MB on-chip memory, which is comprised of 1.98 MByte of SRAM and 2 MByte of MRAM. The SRAM has a sufficient storage capacity to accommodate all activations for BP in the BPGAN algorithm under our target scenarios. With the maximum voltage configuration, the chip can achieve a processing frequency of up to 70 MHz, which is the maximum MRAM frequency. The logic can go up to this frequency by increasing supply voltage to 1.1 V. Power consumption varies depending on the operational scenario reaching up to 10-TOPs/W peak system efficiency.

Fig. 5 shows the area and power breakdown for AIMMI. The power breakdown varies under different working loads and voltage selections. Thus, we use the average number to plot the power breakdown in Fig. 5. Overall, the NE (with MRAM) takes the largest portion of total area and power consumption. The control overhead (Cortex M0 and so on) only contains a small portion. The audio and image interfaces consume approximately 15% of the total energy, representing a relatively low overhead. Depending on application requirements, these interfaces can be managed off-chip.

B. System Evaluation on Real Scenarios

As depicted in Fig. 2, AIMMI supports different AI applications from image- or audio-only inference to cross-modal inference. Fig. 12 demonstrates a PoI tracking scenario and chip performance for that cross-modal intelligence scenario. In order to showcase real-time processing capabilities while conserving energy, we configured the voltage settings as follows: VDD_MAIN at 0.5 V, VDD_CORE at 0.6 V, and VDIO_MRAM at 1.8 V. Under these conditions, the chip runs at 2.5 MHz. We assume an external ultra-low-power VAD chip such as [10] to detect human voice activities and wake up our SoC from the low-power STB mode (75 μW) to begin the audio feature extraction process. The overall power consumption during this audio feature extraction phase is 210 μW. In parallel, change detection on the incoming image frame is performed by the image interface in the SoC. Then, the FD neural network is executed on the change detected region of the image. Assuming that the change detected region constitutes 12% of the entire VGA frame, the energy consumption amounts to 299 μJ for FD. If no face is detected, the system returns to the low-power STB mode waiting for the next VAD trigger (off-chip).

In the event a face is detected, FR is subsequently performed. If the recognized face is not the target PoI, the system returns to STB mode. Otherwise, H.264 image compression is applied to the change detected region containing the face. The complete image processing sequence, comprising FD, FR, and H.264 compression, takes less than 1 s. During this 1-s interval, the audio interface completes Mel-spectrum feature extraction in parallel.

Using both the face image and audio Mel-spectrum features, the SoC performs the CMV. If the face and voice do not correspond to one another, indicating that the audio does not belong to the PoI, the process is terminated. Conversely, if a match is found, the BPGAN AC algorithm compresses the audio sequence.

The number of parameters (after 8-bit quantization and compression) is shown in Fig. 12 (top). AIMMI is designed for resource-constrained edge IoT systems, where models are usually tiny. In our applications, all weights are stored in 2-MB MRAM. Users can define their own models as well; the proposed solution can still be useful for applications that require large models when the system employs a relatively small network with all on-chip parameters as a pre-processing/wake-up step to enable larger network models only when it is necessary based on the event detected by the all-on-chip small network. This hierarchical intelligence approach using the combination of small and larger models is also discussed in [1]. Another way to benefit from our approach

TABLE I
NETWORK STRUCTURES FOR DIFFERENT APPLICATIONS

Algorithm		Dataset	layers	Weight dimension (oc,ic,k1,k2) (s)	Accuracy (INT8)
Face detection		COCO 2017 [34]	conv1	(16,1,5,5) (1)	94.5%
			conv2	(32,16,3,3) (1)	
			fc1	(64, 2048)	
			fc2	(2,64)	
Face recognition		COCO 2017 [34]	conv1	(48,1,5,5) (1)	92.7%
			conv2	(96,48,3,3) (1)	
			conv3	(192,96,3,3) (1)	
			conv4	(128,192,3,3) (1)	
			conv5	(128,128,3,3) (1)	
			fc1	(256,512)	
			fc2	(8,256)	
BPGAN Audio Compression	Encoder	TIMIT Speech Dataset [35]	conv1	(64,1,7,7) (2)	MSE loss: 4.849e-3
			dwconv1	(64,1,3,3) (2)	
			conv2	(128,64,1,1) (1)	
			dwconv2	(128,1,3,3) (2)	
			conv3	(256,128,1,1) (1)	
			conv4	(4,256,3,3) (1)	
	Decoder		conv1	(256,4,3,3) (1)	
			conv2	(128,256,1,1) (1)	
			deconv1	(128,128,3,3) (2)	
			conv3	(256,128,1,1) (1)	
			conv4	(64,256,1,1) (1)	
			deconv2	(64,64,3,3) (2)	
			conv5	(128,64,1,1) (1)	
			conv6	(32,128,1,1) (1)	
			deconv3	(32,32,3,3) (2)	
			conv7	(64,32,1,1) (1)	
			conv8	(1,64,3,3) (1)	
	Decoder BP		conv_bp8	(1,64,3,3) (1)	
			conv_bp7	(64,32,1,1) (1)	
			deconv_bp3	(32,32,3,3) (2)	
			conv_bp6	(32,128,1,1) (1)	
			conv_bp5	(128,64,1,1) (1)	
			deconv_bp2	(64,64,3,3) (2)	
			conv_bp4	(64,256,1,1) (1)	
			conv_bp3	(256,128,1,1) (1)	
			deconv_bp1	(128,128,3,3) (2)	
			conv_bp2	(128,256,1,1) (1)	
			conv_bp1	(256,4,3,3) (1)	
Cross-modal verification	Audio Mel-spectrum	VoxCeleb [36]	conv1	(48,1,5,5) (1)	73.8%
			conv2	(96,48,3,3) (1)	
			conv3	(192,96,3,3) (1)	
			conv4	(128,192,3,3) (1)	
			conv5	(128,128,3,3) (1)	
			fc1	(256,512)	
	Face Image		conv1	(48,1,5,5) (1)	
			conv2	(96,48,3,3) (1)	
			conv3	(192,96,3,3) (1)	
			conv4	(128,192,3,3) (1)	
			conv5	(128,128,3,3) (1)	
			fc1	(256,512)	

is to employ a multi-chip solution as in [37], which can scale the system by adding more chips.

Table I shows the layer-wise neural network model structures for FD, FR, BPGAN-based AC, and CMV. OC, input channel, kernel size, and stride size for each layer are shown in Table I. We use the COCO2017 [34] dataset for FD and FR, the TIMIT speech dataset [35] for AC, and the VoxCeleb [36] dataset for CMV. Each model is quantized to 8-bit weight and 8-bit activation, and the accuracy aligns with the real chip test accuracy. All DNN weights for the aforementioned steps (FD, FR, CMV, and AC) are stored in the 2-MByte MRAM, while all activations are housed in the 1.5-MByte SRAM activation memory, eliminating the need for off-chip access.

Fig. 12 (bottom) shows the energy consumption and latency of each step in this process. We also show the layer-wise measured performance result for the FR neural network in Table II. Each CONV layer contains a non-linear layer (ReLU) and a max pooling layer. The "Input" and "Output" columns in Table II show the feature map height, width, and channel size. The (K, S, MP) column shows the kernel size (height and width), stride size, and max pooling parameters. The efficiency of the first convolutional layer, fifth CONV layer, and FC layers is relatively lower due to the under-utilization of PE arrays. In the first layer, the input channel is only 1 and the fifth CONV layer has a feature map size of 2×2, which is smaller than the PE array dimension (equal to 8). The fifth CONV layer also includes the overhead of converting the CONV feature map ($2 \times 2 \times 128$) to a (512-D) vector for the subsequent FC layer. Hence, the efficiency of those layers is relatively low. The efficiency of the other layers is closely aligned with the observed peak performance, resulting in an overall average power efficiency of 1.74 TOPs/W.

TABLE II
LAYER-WISE MEASURED RESULT IN FR

Layer	Input	Output	K,S,MP	Latency	Efficiency
conv1	(64,64,1)	(32,32,48)	(5,5,1,2)	51.0 ms	0.35 TOPs/W
conv2	(32,32,48)	(16,16,96)	(3,3,1,2)	47.5 ms	2.45 TOPs/W
conv3	(16,16,96)	(8,8,192)	(3,3,1,2)	48.4 ms	2.99 TOPs/W
conv4	(8,8,192)	(4,4,128)	(3,3,1,2)	20.5 ms	2.07 TOPs/W
conv5	(4,4,128)	(2,2,128)	(3,3,1,2)	12.8 ms	0.44 TOPs/W
fc1	(512)	(256)	-	4.45 ms	0.03 TOPs/W
fc2	(256)	(8)	-	0.28 ms	0.003 TOPs/W
Total	-	-	-	184.9 ms	1.74 TOPs/W

TABLE III
COMPARISON WITH MRAM-BASED PROCESSORS

	This Work (AIMMI)	NV-MCU [30]	Vega [26]	MRAM-CIM [31]
MRAM technology	22nm	40nm	22nm	28nm
MRAM size	2MB	0.063MB	4MB	0.25MB
MRAM density	1.02MB/mm2	0.02MB/mm2	1.11MB/mm2	0.08MB/mm2
System Architecture	MRAM+ Cortex M0+ Neural Engine	MRAM+ Cortex M0+ FPGA	MRAM+ RISC-V+ Neural Engine	MRAM CIM Macro
Application	Domain specific	General purpose	General purpose	Domain specific
Performance	3.1TOPS/W (8bit)	-	1.3TOPS/W (8bit)	22.4TOPS/W (1bit)

C. Comparison With MRAM-Based Processors

Table III shows the comparison with recent processors that use MRAM. NV-MCU [30] and Vega [26] are general-purpose processors utilizing MRAM as CPU instruction and data caches. The incorporation of MRAM in these systems enhances efficiency thanks to its non-volatile nature that enables ultra-low AO power by power gating. In contrast to these general-purpose processors, AIMMI is specifically designed to integrate MRAM with an NE, employing MRAM primarily for all-on-chip weight memory storage. This domain-specific architecture allows AIMMI to achieve higher energy efficiency, leveraging the unique strengths of MRAM in a more focused application. Recently, Cai et al. [31] demonstrated an MRAM-based compute-in-memory (CIM) macro design

TABLE IV
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART PROCESSORS

		This Work (AIMMI)	JSSC'20 [1]	VLSI'21 [3]	JSSC'20 [4]	JESTCS'20 [5]	JSSC'22 [37]	JSSC'21 [33]	JSSC'23 [27]
Applications		Face detection / recognition, audio compression, cross-modal verification	Person detection, face detection, face recognition	Face detection, face recognition	Keyword spotting, speaker verification	Super resolution	NN inference & training	Face manipulation	Edge NN inference
Algorithms	CNN/FC	✓	✓	✓	✓ (LSTM)	✓	✓	✓	✓
	GAN	✓	-	-	✓	✓	-	✓	✓
	BP	✓	-	-	-	-	✓ (for FC)	✓	-
Image Processing		Change detection, JPEG, H264	Change detection, JPEG, H264	-	-	-	-	-	-
Audio Processing		FFT, Mel spectrum	-	-	FFT, MFCC	-	-	-	-
Technology [nm]		22	40	22	65	65	40	65	22
Die Area [mm^2]		12.0	27.0	3.4	2.6	16.0	29.2	32.4	6.3
Non-volatile Memory		2MB MRAM	-	-	-	-	2MB RRAM	-	0.5MB MRAM
On-chip SRAM [MB]		1.98	1.13	1.2	0.1	0.56	0.50	0.66	0.66
Off-chip memory access		No	No	-	No	Yes	No	Yes	No
Precision		INT8	INT8	INT1/16	INT8	INT8	INT8	FP8/FP16	INT2/8
Voltage [V]		0.44-1.0	0.58-0.7	0.65	0.6-1.2	0.75-1.1	1.1	0.7-1.1	0.4-0.9
Max Frequency [MHz]		70	0.15	180	12.5	200	200	200	150
Throughput [GOPs]		71.4	0.153	5.76	-	-	920	1080	17.6
Operating Power [mW]		0.25-3.84	0.17	16.7	0.02	31-211	126	58-647	0.13-20
Sys. Peak Eff. [TOPs/W]		3.1-10	1.5	1.1	0.7	1.9	2.2	1.8	2.5

that combines MRAM cells with computation logic. The CIM design can reduce overall energy consumption by eliminating MRAM read-out power. Meanwhile, AIMMI also significantly reduces the MRAM read-out power by the proposed MRAM-cache architecture, which uses a small SRAM cache for weight reuse. A notable drawback of the CIM design is its reduced MRAM density, which is over ten times lower than that of standard MRAM macros, compromising one of MRAM's key advantages.

D. Comparison With State-of-the-Art Processors

Table IV provides a detailed comparison between AIMMI and other state-of-the-art processors. Unlike the image-only processing SoC [1] and the audio-only processing SoC [4], AIMMI stands as the first IoT edge processing SoC to accommodate image, audio, and cross-modal applications on a single chip. Furthermore, it supports the most versatile set of operations such as CNN, FC, GAN, and BP. From a memory standpoint, AIMMI integrates a 2-MByte non-volatile MRAM. Compared to CHIMERA [37] and TinyVers [27], which either integrates a 2-MByte non-volatile RRAM and 0.5-MByte MRAM, AIMMI employs dynamic power gating and an MRAM-cache micro-architecture to enhance system energy efficiency. Exhibiting the largest total on-chip memory capacity (1.98-MByte SRAM + 2-MByte MRAM), AIMMI eliminates the need for off-chip memory accesses, outperforming specialized accelerators such as super resolution neural processing unit (SRNPU) [5] and generative adversarial network processing unit (GANPU) [33], which rely on double data rate (DDR) off-chip transactions due to small on-chip memory. In terms of DNN execution, AIMMI achieves lower power consumption and higher energy efficiency.

VI. CONCLUSION

We introduce the AIMMI SoC in this article, an ultra-low-power multi-modal signal processing solution designed for IoT intelligence applications. It integrates a versatile DNN engine with audio and image interface accelerators, efficiently managing multi-modal data. Utilizing 2-MB MRAM for on-chip storage, MRAM-cache, and dynamic power gating, AIMMI achieves outstanding energy efficiency (3–10 TOPS/W) and low power consumption (0.25–3.84 mW). As the first SoC to demonstrate CNNs, GANs, and BP on a single ultra-low-power accelerator, it outperforms existing low-power DNN processors by 1.4–4.5 times in energy efficiency, setting a new benchmark for multi-modal IoT devices.

REFERENCES

[1] H. An et al., "An ultra-low-power image signal processor for hierarchical image recognition with deep neural networks," *IEEE J. Solid-State Circuits*, vol. 56, no. 4, pp. 1071–1081, Apr. 2021.

[2] H. An et al., "A 170μw image signal processor enabling hierarchical image recognition for intelligence at the edge," in *Proc. IEEE Symp. VLSI Circuits*, 2020, pp. 1–2.

[3] P. Jokic, E. Azarkhish, R. Cattenoz, E. Türetken, L. Benini, and S. Emery, "A sub-mW dual-engine ML inference system-on-chip for complete end-to-end face-analysis at the edge," in *Proc. Symp. VLSI Circuits*, Jun. 2021, pp. 1–2.

[4] J. S. P. Giraldo, S. Lauwereins, K. Badami, and M. Verhelst, "Vocell: A 65-nm speech-triggered wake-up SoC for 10-μ w keyword spotting and speaker verification," *IEEE J. Solid-State Circuits*, vol. 55, no. 4, pp. 868–878, Apr. 2020.

[5] J. Lee, J. Lee, and H.-J. Yoo, "SRNPU: An energy-efficient CNN-based super-resolution processor with tile-based selective super-resolution in mobile devices," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 10, no. 3, pp. 320–334, Sep. 2020.

[6] N. P. Jouppi et al., "In-datacenter performance analysis of a tensor processing unit," in *Proc. 44th Annu. Int. Symp. Comput. Archit.*, 2017, pp. 1–12.

[7] N. P. Jouppi et al., "A domain-specific supercomputer for training deep neural networks," *Commun. ACM*, vol. 63, no. 7, pp. 67–78, Jun. 2020.

[8] N. P. Jouppi et al., "TPU v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings," 2023, *arXiv:2304.01433*.

[9] K. D. Choo et al., "Energy-efficient motion-triggered IoT CMOS image sensor with capacitor array-assisted charge-injection SAR ADC," *IEEE J. Solid-State Circuits*, vol. 54, no. 11, pp. 2921–2931, Nov. 2019.

[10] S. Oh et al., "An acoustic signal processing chip with 142-nW voice activity detection using mixer-based sequential frequency scanning and neural network classification," *IEEE J. Solid-State Circuits*, vol. 54, no. 11, pp. 3005–3016, Nov. 2019.

[11] Q. Li et al., "NS-FDN: Near-sensor processing architecture of feature-configurable distributed network for beyond-real-time always-on keyword spotting," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 5, pp. 1892–1905, May 2021.

[12] H. Xu et al., "Senputing: An ultra-low-power always-on vision perception chip featuring the deep fusion of sensing and computing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 1, pp. 232–243, Jan. 2022.

[13] T.-H. Hsu et al., "A 0.5-V real-time computational CMOS image sensor with programmable kernel for feature extraction," *IEEE J. Solid-State Circuits*, vol. 56, no. 5, pp. 1588–1596, May 2021.

[14] T. Hsu et al., "A 0.8 V intelligent vision sensor with tiny convolutional neural network and programmable weights using mixed-mode processing-in-sensor technique for image classification," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 1–3.

[15] J. Choi, S. Lee, Y. Son, and S. Y. Kim, "Design of an always-on image sensor using an analog lightweight convolutional neural network," *Sensors*, vol. 20, no. 11, p. 3101, May 2020.

[16] S. Yin et al., "A 141 UW, 2.46 PJ/neuron binarized convolutional neural network based self-learning speech recognition processor in 28NM CMOS," in *Proc. IEEE Symp. VLSI Circuits*, 2018, pp. 139–140.

[17] B. Liu, A. Cao, and H.-S. Kim, "Unified signal compression using generative adversarial networks," in *Proc. ICASSP IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 3177–3181.

[18] A. Nagrani, S. Albanie, and A. Zisserman, "Learnable PINs: Cross-modal embeddings for person identity," *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 71–88.

[19] T. Afouras, A. Owens, J. S. Chung, and A. Zisserman, "Self-supervised learning of audio-visual objects from video," in *Proc. Eur. Conf. Comput. Vis.* Glasgow, U.K.: Springer, 2020, pp. 208–224.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, vol. 16, 2016, pp. 770–778.

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

[22] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*.

[23] J. L. McKinstry et al., "Discovering low-precision networks close to full-precision networks for efficient embedded inference," 2018, *arXiv:1809.04191*.

[24] S. Ikegawa, F. B. Mancoff, J. Janesky, and S. Aggarwal, "Magnetoresistive random access memory: Present and future," *IEEE Trans. Electron Devices*, vol. 67, no. 4, pp. 1407–1419, Apr. 2020.

[25] Y.-D. Chih et al., "13.3 A 22 nm 32mb embedded STT-MRAM with 10ns read speed, 1m cycle write endurance, 10 years retention at 150°C and high immunity to magnetic field interference," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2020, pp. 222–224.

[26] D. Rossi et al., "Vega: A ten-core SoC for IoT endnodes with DNN acceleration and cognitive wake-up from MRAM-based state-retentive sleep mode," *IEEE J. Solid-State Circuits*, vol. 57, no. 1, pp. 127–139, Jan. 2022.

[27] V. Jain, S. Giraldo, J. D. Roose, L. Mei, B. Boons, and M. Verhelst, "TinyVers: A tiny versatile system-on-chip with state-retentive eMRAM for ML inference at the extreme edge," *IEEE J. Solid-State Circuits*, vol. 58, no. 8, pp. 2360–2371, Aug. 2023.

[28] Z. Fan et al., "Audio and image cross-modal intelligence via a 10TOPS/W 22nm SoC with back-propagation and dynamic power gating," in *Proc. IEEE Symp. VLSI Technol. Circuits*, 2022, pp. 18–19.

[29] Q. Zhang et al., "A 22 nm 3.5TOPS/W flexible micro-robotic vision SoC with 2MB eMRAM for fully-on-chip intelligence," in *Proc. IEEE Symp. VLSI Technol. Circuits (VLSI Technol. Circuits)*, Jun. 2022, pp. 72–73.

[30] M. Natsui et al., "A 47.14-μW 200-MHz MOS/MTJ-hybrid nonvolatile microcontroller unit embedding STT-MRAM and FPGA for IoT applications," *IEEE J. Solid-State Circuits*, vol. 54, no. 11, pp. 2991–3004, Nov. 2019.

[31] H. Cai et al., "33.4 A 28 nm 2mb STT-MRAM computing-in-memory macro with a refined bit-cell and 22.4–41.5 TOPS/W for AI inference," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2023, pp. 500–502.

[32] Z. Fan, Z. Li, B. Li, Y. Chen, and H. Li, "RED: A ReRAM-based deconvolution accelerator," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2019, pp. 1763–1768.

[33] S. Kang et al., "GANPU: An energy-efficient multi-DNN training processor for GANs with speculative dual-sparsity exploitation," *IEEE J. Solid-State Circuits*, vol. 56, no. 9, pp. 2845–2857, Sep. 2021.

[34] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Comput. Vis. ECCV 13th Eur. Conf.*, Zurich, Switzerland. Cham, Switzerland: Springer, Sep. 2014, pp. 740–755.

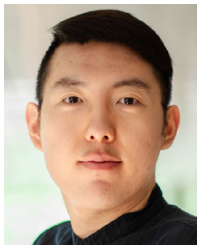
[35] J. S. Garofolo, "Timit acoustic phonetic continuous speech corpus," in *Proc. Linguistic Data Consortium*, 1993.

[36] A. Nagrani, J. Son Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," 2017, *arXiv:1706.08612*.

[37] K. Prabhu et al., "CHIMERA: A 0.92-TOPS, 2.2-TOPS/W edge AI accelerator with 2-MByte on-chip foundry resistive RAM for efficient training and inference," *IEEE J. Solid-State Circuits*, vol. 57, no. 4, pp. 1013–1026, Apr. 2022.

Zichen Fan (Graduate Student Member, IEEE) received the B.S. degree from Tsinghua University, Beijing, China, in 2019. He is currently pursuing the Ph.D. degree with Michigan Integrated Circuit Laboratory, University of Michigan, Ann Arbor, MI, USA.

His current research interests include machine-learning accelerator design and efficient AI algorithm design, including model quantization, model pruning, and low-power VLSI digital system design.



Qirui Zhang (Member, IEEE) received the B.S. degree (Hons.) from the School of Microelectronics, Shanghai Jiao Tong University, Shanghai, China, in 2018. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Michigan, Ann Arbor, MI, USA.

His research interests include efficient algorithm-hardware co-designs, very-large-scale integration (VLSI) architectures, and integrated circuits for emerging applications, including artificial intelligence and quantum computing.

Mr. Zhang was a recipient of the Best Paper Award at the 2022 tinyML Research Symposium and the 2023 IEEE International Conference on Application-Specific Systems, Architectures and Processors. He was a Finalist of the 2023 Qualcomm Innovation Fellowship (North America).

Hyochan An received the B.S. degree in electrical and computer engineering from Sungkyunkwan University, Seoul, South Korea, in 2014, and the Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA, in 2022.

From 2014 to 2017, he was an Engineer with Samsung Electronics, Hwasung, South Korea. His current research interests include energy-efficient accelerator design and systems.

Dr. An was a recipient of the Doctoral Fellowship from the Kwanjeong Educational Foundation in South Korea.

Boxun Xu received the B.S. degree in electronic engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2019, and the M.S. degree in electrical and computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2021. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of California at Santa Barbara, Santa Barbara, CA, USA.

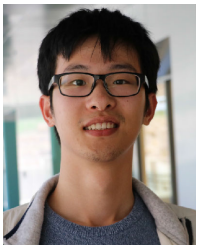
His current research focuses on bio-inspired machine-learning algorithms, algorithm-hardware co-design, and energy-efficient architectures for deep learning.

Li Xu (Member, IEEE) received the B.Eng. degree in automation from Tongji University, Shanghai, China, in 2009, the M.S. degree in electrical and computer engineering from Northeastern University, Boston, MA, USA, in 2016, and the Ph.D. degree in electrical and computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2021.

From 2009 to 2011, he was an IC Design Engineer with Ricoh Electronic Devices Shanghai Company, Ltd., Shanghai, where he worked on LDO and dc/dc converter projects. In summer 2015, he was a Design Intern with Linear Technology Corporation, Colorado Springs, CO, USA. In summer 2020, he was a Research Intern with NVIDIA Corporation, Santa Clara, CA, USA, where he is currently a Research Scientist. His current research interest is energy-efficient mixed-signal circuit design.

Chien-Wei Tseng (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2011 and 2014, respectively, and the Ph.D. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2024.

From 2014 to 2019, he was with MediaTek Inc., Hsinchu, researching cellular RFIC design. He is currently with Nvidia Inc., Santa Clara, CA, USA, developing high-speed mixed-signal circuits. His research interests include high-frequency analog/RF circuit design, clocking circuit design, and digital signal processing with machine learning.



Yimai Peng (Member, IEEE) received the B.Eng. degree in electrical and computer engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2015, and the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2017 and 2022, respectively.

In 2022, he joined Qualcomm, Inc., Raleigh, NC, USA, as a Senior Engineer to work on circuit research and development for higher energy efficiency and robustness, for various processors, servers, and IoT devices. His research interests include low-power circuit and system designs in power management, energy harvesting, analog front end, and intelligent micro-robot systems.

Dr. Peng was a recipient of the Dwight F. Benton Fellowship at the University of Michigan.

Andrea Bejarano-Carbo (Graduate Student Member, IEEE) received the M.Eng. degree in electrical and electronic engineering from the University of Bristol, Bristol, U.K., in 2019, and the M.Sc. degree in electrical and computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2022, where she is currently pursuing the Ph.D. degree.

Her research interests lie in low-power and area-constrained intelligent devices for Internet-of-Things applications.

Ms. Bejarano-Carbo was a recipient of the Best Paper Award at the 2022 tinyML Research Symposium.

Pierre Abillama (Graduate Student Member, IEEE) received the B.Sc. degree in computer engineering from the University of Minnesota, Twin Cities, Minneapolis, MN, USA, in 2020, and the M.Sc. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2022, where he is currently pursuing the Ph.D. degree in electrical engineering.

His current research interests include low-power and energy-efficient hardware accelerators and processors for deep learning and edge intelligence.

Ang Cao (Member, IEEE) received the B.S. degree in electronics engineering from Wuhan University (WHU), Wuhan, China, in 2018, and the master's degree in electrical and computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2020, where he is currently pursuing the Ph.D. degree in computer science and engineering.

Mr. Cao received the Rollin M. Gerstacker Foundation Fellowships in 2020.

Bowen Liu received the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2018 and 2024, respectively.

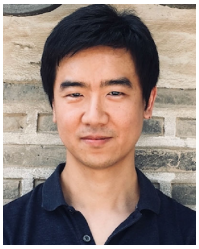
He joined Zoom Video Communications Inc., San Jose, CA, USA, in 2024. His research interest lies in deep learning, computer vision, signal processing, generative AI, and their applications in low-power systems.

Changwoo Lee received the B.S. and M.S. degrees in electronic engineering from Hanyang University, Seoul, South Korea. He is currently pursuing the Ph.D. degree in electrical engineering and computer science with the University of Michigan, Ann Arbor, MI, USA.

His research interests include efficient deep learning.

Zhehong Wang received the B.E. degree in electronics and information engineering from Zhejiang University, Hangzhou, China, in 2016, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2019 and 2022, respectively.

He is currently an ASIC Engineer at Reality Labs, Meta Platforms, Inc., Menlo Park, CA, USA, working on high-efficiency machine-learning accelerators for AR/VR edge devices.



Hun-Seok Kim (Senior Member, IEEE) received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 2001, and the Ph.D. degree in electrical engineering from the University of California at Los Angeles, Los Angeles, CA, USA, in 2010.

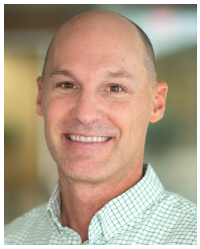
He is currently an Associate Professor with the University of Michigan, Ann Arbor, MI, USA. His research focuses on system analysis, novel algorithms, and VLSI architectures for low-power/high-performance wireless communications, signal processing, computer vision, and machine-learning systems.

Dr. Kim was a recipient of the DARPA Young Faculty Award in 2018 and the National Science Foundation (NSF) CAREER Award in 2019. He is an Associate Editor of IEEE TRANSACTIONS ON MOBILE COMPUTING.

David Blaauw (Fellow, IEEE) received the B.S. degree in physics and computer science from Duke University, Durham, NC, USA, in 1986, and the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 1991.

Until August 2001, he worked for Motorola, Inc., Austin, TX, USA, where he was the Manager of the High Performance Design Technology Group and received the Motorola Innovation Award. Since August 2001, he has been with the faculty of the University of Michigan, Ann Arbor, MI, USA, where he is currently the Kensall D. Wise Collegiate Professor of EECS. He has authored over 600 articles and holds 65 patents. He has researched ultra-low-power wireless sensors using subthreshold operation and low-power analog circuit techniques for millimeter systems. This research received the MIT Technology Review's "One of the Year's Most Significant Innovations." His research group introduced the so-called near-threshold computing, which has become a common concept in semiconductor design. Most recently, he has pursued research in cognitive computing using analog, in-memory neural networks for edge devices and genomics for precision health.

Dr. Blaauw received the 2016 SIA-SRC Faculty Award for lifetime research contributions to the U.S. semiconductor industry. He has received numerous best paper awards. He was the General Chair of the IEEE International Symposium on Low Power and a member of the IEEE International Solid-State Circuits Conference (ISSCC) Analog Program Subcommittee.



Dennis Sylvester (Fellow, IEEE) received the Ph.D. degree from the University of California at Berkeley, Berkeley, CA, USA.

He has held research staff positions at the Advanced Technology Group of Synopsys, Mountain View, CA, USA, and Hewlett-Packard Laboratories, Palo Alto, CA, USA, and visiting professorships at the National University of Singapore, Singapore, and Nanyang Technological University, Singapore. He is currently the Edward S. Davidson Collegiate Professor of electrical and computer engineering at the University of Michigan, Ann Arbor, MI, USA, where he is also the Founding Director of Michigan Integrated Circuits Laboratory (MICL), a group of ten faculty and 70+ graduate students. He co-founded Ambiq, Austion, TX, USA, a fabless semiconductor company developing ultra-low-power mixed-signal solutions for compact wireless devices. He has authored over 550 articles along with one book and several book chapters. He holds 53 U.S. patents. His research interests include the design of millimeter-scale computing systems and energy-efficient near-threshold computing.

Dr. Sylvester is a fellow of the National Academy of Inventors. He received the NSF CAREER Award, the Beatrice Winner Award at ISSCC, an IBM Faculty Award, an SRC Inventor Recognition Award, and 16 best paper awards and nominations. He was named a top-five Contributing Author all-time at ISSCC in 2023 and a Most Prolific Author at the IEEE Symposium on VLSI Circuits and received the University of Michigan Henry Russel Award for distinguished scholarship. He previously served on the administrative committee for the IEEE Solid-State Circuits Society and was an Associate Editor of IEEE JOURNAL OF SOLID-STATE CIRCUITS, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS (TCAD), and IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. He is the current Editor-in-Chief of the IEEE JOURNAL OF SOLID-STATE CIRCUITS. He served as an IEEE Solid-State Circuits Society Distinguished Lecturer.