EPOCHS-1: A 12 nm Highly Heterogeneous Open-Source SoC With Distributed Coin-Based Power Management and Integrated Hybrid Voltage Regulation

Joseph Zuckerman, Martin Cochet, *Senior Member, IEEE*, Maico Cassel Dos Santos, *Graduate Student Member, IEEE*, Erik Jens Loscalzo, Karthik Swaminathan, *Senior Member, IEEE*, Tianyu Jia, *Member, IEEE*, Davide Giri, Thierry Tambe, *Member, IEEE*, Jeff Jun Zhang, Alper Buyuktosunoglu, *Fellow, IEEE*, Kuan-Lin Chiu, Giuseppe Di Guglielmo, Paolo Mantovani, Luca Piccolboni, Gabriele Tombesi, *Graduate Student Member, IEEE*, David Trilla, *Member, IEEE*, John-David Wellman, *Member, IEEE*, En-Yu Yang, Aporva Amarnath, *Member, IEEE*, Ying Jing, *Graduate Student Member, IEEE*, Bakshree Mishra, Joshua Park, Vignesh Suresh, Samira Zaliasl, *Member, IEEE*, Michael Lekas, Stephen Cahill, Hesam Sadeghi, Joseph Meyer, Noah Sturcken, Sarita Adve, *Fellow, IEEE*, David Brooks, *Fellow, IEEE*, Gu-Yeon Wei, *Senior Member, IEEE*, Kenneth L. Shepard, *Fellow, IEEE*, Pradip Bose, *Life Fellow, IEEE*, and Luca P. Carloni, *Fellow, IEEE*

In memory of Davide Giri.

***Abstract*—We present EPOCHS-1, a 12 nm, 64 mm² system-on-chip (SoC) with a high degree of heterogeneity. It features four Linux-SMP-capable RISC-V cores, 14 different types of accelerators, a distributed memory hierarchy, and various peripherals. EPOCHS-1's memory hierarchy has the flexibility to support a diverse set of accelerators and can scale to support complex applications with 34% and 25% reduction in latency and energy, respectively. A subset of the SoC's 23 power and 35 clock domains is regulated with a fully-decentralized power-allocation scheme and hybrid unified voltage and frequency scaling (HUVFS) that combines an in-package switched regulator with a per-tile low dropout (LDO). Combined, these techniques achieve up to a 1.57× speedup versus a centralized power management baseline. Designed with an agile methodology, EPOCHS-1 is based on an open-source SoC architecture and features only open-source components, either third-party or newly designed, thus enabling design reuse for future research projects.**

***Index Terms*—Open-source hardware (OSH), power management (PM), system-on-chip (SoC), voltage and frequency scaling, voltage regulators.**

I. Introduction

MODERN system-on-chip (SoC) architectures feature a growing number of specialized accelerators alongside general-purpose cores. This trend, which started over a decade ago when chips like the Apple A-series SoCs began featuring dozens of specialized intellectual property (IP) blocks [1], has continued to grow: recent industry products include SoCs that combine many heterogeneous processors and accelerators [2], [3] and platforms that support the integration of in-house and third-party (TP) accelerators [4], [5]. While these designs offer state-of-the-art performance, they are obtained with high non-recurring engineering costs and the use of proprietary IPs, which limits their impact on the research community.

On the other hand, research prototypes from academia typically focus on demonstrating a particular IP as a proof-of-concept. The most complex of these SoCs are limited to a small degree of heterogeneity with one [6], [7] to three

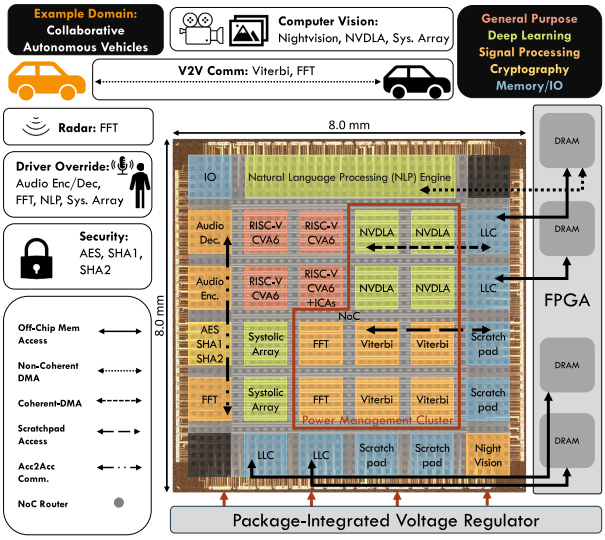


Fig. 1. Annotated die photo of EPOCHS-1, including the full-system implementation with PIVR and FPGA test harness.

Received 27 October 2024; revised 6 June 2025; accepted 11 August 2025. Date of publication 26 September 2025; date of current version 27 April 2026. This article was approved by Associate Editor Sanu Mathew. This work was supported by the Defense Advanced Research Projects Agency (DARPA). *(Joseph Zuckerman, Martin Cochet, Maico Cassel dos Santos, and Erik Jens Loscalzo contributed equally to this work.) (Corresponding author: Joseph Zuckerman.)*

Please see the Acknowledgment section of this article for the author affiliations.

Digital Object Identifier 10.1109/JSSC.2025.3602386

[8], [9], [10] accelerator types. For small teams of academic researchers, the challenges of system integration [11] are the biggest hurdle to taping out more heterogeneous SoCs. Most of these research prototypes do not consider the system-level challenges that arise from running complex software workloads across many IPs.

In this article, we present EPOCHS-1, a highly heterogeneous SoC with 14 types of accelerators (see Fig. 1). EPOCHS-1 is one of the most complex open-source heterogeneous SoCs to be presented in the literature [12]. The combination of IP reuse from open-source hardware (OSH) projects and an agile design and integration methodology allowed a small research team to complete this tape out in a period of three months [13]. Our taped-out SoC was used to characterize and address the system-level challenges that this class of SoC architectures faces. In particular, EPOCHS-1's ability to boot an SMP Linux operating system allowed for complex workloads that present significant challenges in resource contention, data movement, and power management (PM). In turn, this allowed us to demonstrate novel solutions to address these challenges and, in particular, our innovations upon existing PM techniques that are targeted toward more homogeneous SoCs.

EPOCHS-1 is based on ESP, an open-source platform for heterogeneous SoC design [14]. The ESP architecture is tile-based, with a heterogeneous grid of tiles connected by a multi-plane 2-D mesh network-on-chip (NoC). Section II details the architecture of EPOCHS-1, highlighting the main types of tiles supported by ESP and how they were modified for this specific SoC. The chip measurements presented in Section III provide a characterization of the SoC components and showcase one of the key innovations of EPOCHS-1: its scalable, flexible memory hierarchy that supports a variety of data-access modes for accelerators.

Highly heterogeneous SoCs are often used for edge devices, where energy efficiency is key due to thermal or battery constraints. However, these types of system pose additional challenges for PM. The diverse profile of different accelerators demands fine-grained control of power allocation and delivery; the implementation must also be workload agnostic to be able to integrate with a wide variety of IPs. Previous approaches, such as a programmable power-management unit (PMU) [15] or custom on-chip controller (OCC)-based designs [16], [17], leverage a centralized controller that operates sequentially. Hence, their response times scale only linearly with the system's size. In contrast, in Section IV we propose a fully decentralized PM scheme called BlitzCoin (BC) [18]. Rather than relying on software [19], BC is implemented in hardware and distributed across the tiles of the SoC. Exchanges of coins, which represent units of power, between neighboring tiles occur in parallel throughout the SoC; this results in sub-linear scaling of response time with the size of the SoC. This solution is scalable to large SoCs, responsive to short workloads, and tailored to a large degree of heterogeneity. In EPOCHS-1, we integrated BC in ten accelerator tiles that form a PM cluster.

Once BC allocates power to particular tiles, their voltage and frequency must be regulated to enforce the assigned

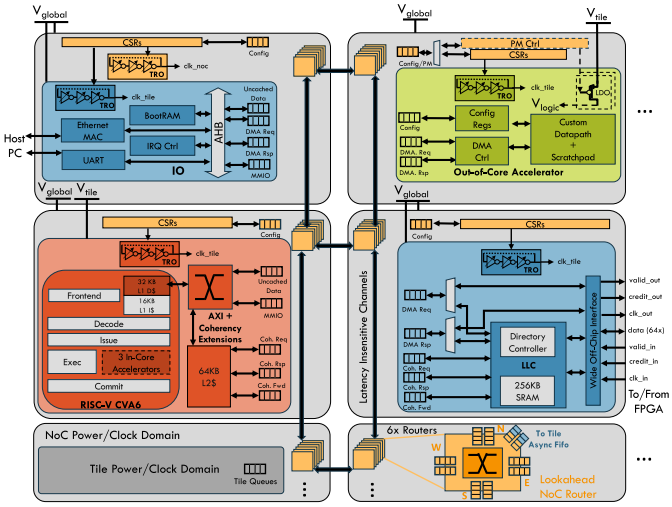


Fig. 2. Four main types of ESP tiles implemented in EPOCHS-1: IO, accelerator, CVA6, and LLC (top-left).

power. This per-tile control of voltage and frequency creates many more clock and power domains than in prior heterogeneous SoCs [6], [7], [8]. While this offers an opportunity for fine-grained voltage and frequency regulation, it also requires an easily replicable, low-overhead solution. The conventional voltage and frequency actuation strategy consists of separate closed-loop voltage and frequency actuators (e.g., a low dropout (LDO) regulator and a PLL), combined with a PM loop that adjusts the SoC voltage and frequency targets. Recent designs, instead, have proposed unified voltage and frequency scaling (UVFS) [20], [21], [22], a single-loop actuation mechanism that offers a simpler and more compact design, as well as intrinsic voltage/frequency tracking. Although LDOs offer simplicity of integration and rapid response time, they suffer in terms of conversion efficiency. In contrast, switched regulators offer high conversion efficiency, but come at the cost of higher integration complexity and slower response time [23]. A combination of these techniques can help mitigate the drawbacks of each approach [24]. EPOCHS-1 leverages a novel hybrid voltage regulation (HVR) architecture that combines the efficiency of a package-integrated voltage regulator (PIVR) with the fine granularity and fast response time of an LDO per tile on the SoC [25]. Section V presents the implementation of this hybrid UVFS (HUVFS) power enforcement, while Section VI shows the results of applying these PM techniques.

Finally, we believe that it is critically important that other small teams, particularly those in academia, have the ability to build systems like EPOCHS-1. Hence, Section VII describes the agile design methodology that enabled a small team to tape out such a complex system. Furthermore, we have released the entire synthesizable design of the SoC as OSH so that it can be a starting point for other researchers, thus supporting the growth of the OSH community [26].

II. SoC Architecture

Fig. 1 shows an annotated die photo of EPOCHS-1 that depicts our fully implemented system. The 8×8 mm SoC is an instance of the architecture provided by the open-source ESP platform for heterogeneous SoC design [14], [27]. The

TABLE I
USAGE, CONFIGURATION OPTIONS, AND KEY METRICS OF THE ESP TILES OF EPOCHS-1 AND THE IP BLOCKS THAT THEY EMBED

IP	Use/Configuration Options	Memory (KB)	Max Freq @0.6V – 1V (MHz)	Power @0.6V –1V (mW)	Speedup @0.6V-1V (vs. SW)	Energy Savings @0.6V-1V (vs. SW)	Area Utilization (%)
NoC	1 32-bit and 5 64-bit physical planes. Lookahead, dimension-ordered routers with single cycle latency.	N/A	800*-840	516*-997	N/A	N/A	N/A
CVA6	Linux capable 64-bit RISC-V core supporting the I, M, A, and C ISA extensions. With L1$ + ESP L2$.	112	640-1250	36-198	N/A	N/A	38.4
In-core FFT	Key operations of the inner-most loop of the FFT algorithm.	N/A			2-4x	2-4x	
In-core Viterbi	Key operations of the Viterbi butterfly algorithm.	N/A			2-4x	2-4x	
In-core Pop. Count	Computes hamming weight of 64-bit integers. Useful in compression and error correction.	N/A			1.3-3x	1.3-3x	
IO Tile	Various peripherals (Ethernet, UART), interrupt controller, bootROM.	128	120 (fixed)	2 (fixed)	N/A	N/A	35.1
LLC	MESI-based directory protocol with DMA handler. Synchronous off-chip link to an FPGA for DRAM.	256	1300*- 1620	69*-132	N/A	N/A	57.0
Scratchpad	Large non-coherent, software-managed on-chip storage.	1024	930*-1200	38*-76	N/A	N/A	85.6
NLP Engine	Transformer models (e.g. BERT). Can be repurposed for vision transformers.	647	303-680	23-137	34-68x	114-98x	61.2
NVDLA	CNN inference (e.g. LeNet, SimpleNet, ResNet-50, YOLO)	128	750-1340	33-207	160-160x	370-98x	62.0
Systolic Array	General matrix multiplication. Matrix size configurable.	16	440-1090	37-239	32-34x	68-18x	52.4
Audio Encoder	Spatial encoding from multiple sound sources into a single 3D soundfield with up to 16 channels.	192	440-900	3-12	6-15x	232-152x	28.6
Audio Decoder	Rotates a 3D soundfield of up to 16 channels based on a listener's orientation.	256	440-880	34-195	8-9x	19-6x	38.8
FFT	Inverse FFT supported. Number and size of FFTs configurable.	128	650-1310	6-31	38-68x	496-285x	23.4
Viterbi	Decoding of convolution-encoded WiFi messages, consistent with IEEE 802.11p.	64	850-1400	8-43	11-24x	104-71x	22.4
Nightvision	Contrast adjustment through noise filtering, histogram equalization, and discrete wavelet transform.	10	740-1540	6-37	78-141x	959-523x	10.1
AES	Encrypt and decrypt supported. ECB, CTR, and CBC modes supported. Input size configurable.	N/A	510-880	4-18	24-39x	453-244x	25.3
SHA1	Produces digests of 20 bytes.	8	480-980		8-13x	160-91x	
SHA2	Supports SHA224, SHA256, SHA384, SHA512 with digests of 28 – 64 bytes.	8	530-970		5-10x	109-68x	

ESP architecture is a heterogeneous tile grid connected by a 2-D mesh NoC. In particular, EPOCHS-1 is a 34-tile SoC that instantiates six independent physical NoC planes; three planes are dedicated to coherence messages, two to accelerators' direct memory access (DMA), and one to auxiliary functions, such as memory-mapped IO and interrupts. Fig. 2 shows the main types of tiles provided by ESP with some details of how they are implemented in EPOCHS-1. The EPOCHS-1 tiles embed a variety of different IP blocks; some of these were taken directly from the ESP release, while others are new IPs that were integrated into the ESP architecture from various sources, as described in Section VII. Table I summarizes the IPs featured in the SoC, and lists some key metrics and results that are covered further in Section III.

A. Clock and Power Domains

EPOCHS-1 has 35 clock domains and 23 power domains. Each of the 34 tiles has its own clock generated from a local tunable replica oscillator (TRO) that allows it to run at its own frequency, independently of the rest of the SoC. The clock for the synchronous NoC is generated by a TRO in the IO tile, which contains various I/O peripherals (e.g., Ethernet and universal asynchronous receiver-transmitter (UART)) and other components (e.g., the boot ROM and interrupt controller). The clock domain crossings from the NoC into the tiles are handled by asynchronous FIFOs.

The physical implementation of the tiles also includes the logic of the NoC routers serving that tile; therefore, the chip's top-level does not contain additional logic, only wires connecting the tiles. The six routers (one per physical plane), along with some control and status registers (CSRs) that must always be accessible, are placed in an always-on power domain V_{global}. Tiles with external interfaces (IO and last-level cache (LLC)) are also powered on V_{global}. The remaining tiles are powered separately from the NoC on V_{tile}. Section V covers the voltage and frequency regulation of the SoC's tiles in depth.

B. RISC-V CVA6 Tile

The processor tile of ESP comprises an open-source CPU core that hosts the system. For EPOCHS-1, we used the Linux-capable, 64-bit RISC-V CVA6 core [28]. The tile also contains a private *L*2 cache that implements ESP's distributed MESI-based coherence protocol [29]. The CVA6 core is instantiated off-the-shelf from the open-source version except for two modifications.

First, since its *L*1 cache does not support invalidations, it cannot be used natively in a coherent, multicore system. Hence, we modified the interface of the core and the *L*1 cache to receive invalidations on an AXI Coherency Extensions (ACE) bus [30]. When another core (or accelerator) needs to access a cache line in a modified state, the ESP's *L*2 cache sends an invalidation to the *L*1 cache over the ACE bus. This enables coherent multicore execution across the four instances of this tile, including booting Linux SMP. Second, the pipeline of one of the four cores was modified to include three in-core accelerators for important computational kernels of the fast Fourier transform (FFT), Viterbi, and population count algorithms [31].

C. Scalable and Flexible Memory Hierarchy

Two types of tiles, the LLC and scratchpad tiles, implement EPOCHS-1's memory hierarchy. The LLC tile, based on ESP's memory tile, contains a partition of the last-level cache and a channel to main memory (external dynamic random access memory (DRAM)). The DRAM is accessed through a wide source-synchronous link to an field-programmable gate

array (FPGA), which instantiates dual data rate (DDR) controller IPs. The LLC tiles, together with the *L2* caches in the processor tiles, implement ESP's coherence protocol, which is a variation of a traditional directory-based MESI protocol, adapted to work on a multi-plane NoC and support various degrees of coherence for accelerators [29]. The presence of multiple LLC tiles increases off-chip memory bandwidth and distributes memory traffic across the NoC, allowing the memory hierarchy to scale to complex applications that use a significant portion of the chip. The ESP architecture supports multiple LLC tiles by partitioning the global address space; each tile serves a discrete partition, and all logic to support this (e.g., address maps and routing tables) is automatically generated.

The scratchpad tile, instead, contains a large 1-MB block of on-chip SRAM. The scratchpad tiles' addresses lie outside of the coherent address space. Its usage must be explicitly orchestrated (i.e., by software) in contrast to the usage of the cache hierarchy, which is implicitly managed by the coherence protocol. Due to the high latency of reaching DRAM through an FPGA, the scratchpad tile is useful for expanding on-chip storage when workloads do not fit within the cache hierarchy.

As shown in Fig. 1, the NoC-supported memory hierarchy supports a variety of different communication modes for accelerators in the SoC. The out-of-core accelerators perform DMA to bring data into their private scratchpads. These DMA requests can be served in many ways. First, they can be served by the cache hierarchy using one or more coherence modes [32]. In the non-coherent DMA mode, accelerator requests bypass the cache hierarchy entirely to access off-chip data directly; coherence must be maintained by software through cache flushes. In the coherent-DMA mode, DMA requests are sent to the LLC, and coherence is maintained fully in hardware; data must be recalled or invalidated in the private caches of the CPU, as necessary. In a variation of coherent-DMA mode, the private CPU caches are flushed first to avoid recall and invalidate messages; this is often referred to as LLC-coherent DMA. Accelerator DMA requests can also be directed to the scratchpad tiles, in which case software must explicitly write the data to these tiles. Finally, accelerators can bypass the memory hierarchy and exchange data with each other when one accelerator's output can be used as the input of another; this can save a round-trip to main memory, reduce NoC traffic, and allow for finer-grained pipelining of multiple accelerator invocations.

D. Accelerators

In addition, the three in-core accelerators, the remaining accelerators in EPOCHS-1 are out-of-core accelerators. With the exception of a natural language processing (NLP) engine [33], [34] that spans four tiles, and three symmetric cryptography engines that share one tile, each accelerator occupies its own tile. Loosely-coupled from the host cores [35], these accelerators execute coarse-grained tasks on large data sets and are invoked from device drivers through memory-mapped registers. The accelerators are chosen from important kernels of our target application domain and span the fields of signal processing, deep learning, and cryptography. Highly optimized for their particular tasks, these fixed-function accelerators are not programmable (i.e., execute instructions) but are highly configurable.

III. SoC Measurements

A. Frequency, Power, and Performance

Table I summarizes key metrics of each of the IPs in EPOCHS-1. First, we report the maximum frequency of each IP at the SoC's $V_{\min}$ (0.6 V) and $V_{\max}$ (1 V); we can characterize each IP separately thanks to the per-tile clocking scheme. A few tiles (FFT, Viterbi, Cryptography, and NVIDIA deep learning accelerator (NVDLA)) can also run at 0.5 V, but we report the max frequency at 0.6 V for consistency. The IPs powered by V_{global} (NoC, LLC, and scratchpad) must remain at 0.8 V or above, so their frequency is reported for this range. The fastest IP is the LLC, which runs at 1.62 GHz at 1 V, while the slowest is the large NLP engine, which can run up to 680 MHz. The CVA6 core, along with the in-core accelerators, which follow its frequency, can operate at a max of 1.25 GHz. The NoC reaches a maximum of 880 MHz at 1 V.

Next, we report the power of each IP at $V_{\min}$ and $V_{\max}$. Among the IPs, the systolic array consumes the most power at 239 mW, while the audio encoder only consumes 12 mW; the others vary greatly within this range. This highlights a diverse power profile among the heterogeneous mix of accelerators. At 1V, with all components operating at their maximum frequency, the SoC consumes 4.33 W, including 1 W consumed by the NoC. With all components operating at their minimum voltage and minimum frequency of 40–80 MHz (dictated by the design of the TRO and $V_{\min}$), the SoC consumes 174 mW. This gives a possible dynamic power range of 25×; power or clock gating of unused components can give further savings.

We also report the improvements in performance and energy by offloading tasks to accelerators compared with running the same task on the CVA6 cores. We measure the performance of the accelerators while running end-to-end workloads, including the overheads of software, any required cache flushes, and data movement. Accelerators are invoked using device drivers, and in most cases, the complexity of the driver is hidden from the user space with a simple, three-function API provided by ESP [36]. The in-core accelerators yield small speedups of 1.3×–4×, while negligibly increasing the power of the core; their low invocation overhead makes them useful for small workloads. In contrast, the NVDLA accelerator has the largest speedup at 160×. As shown in Table I, a few accelerators, there is a negligible change in speedup when raising their voltage and frequency. This means that their performance is bound by either memory latency or software overheads; in these cases, the accelerator voltage and frequency can be reduced to save power while still preserving performance. While speedup is greatest at 1 V, energy efficiency gains are greatest at 0.6 V. There is a wide range of energy efficiency improvements, the highest of which is a 959× improvement from the nightVision accelerator. These performance improvements are critical for edge devices to meet constraints of real-time applications while remaining under a specified power budget.

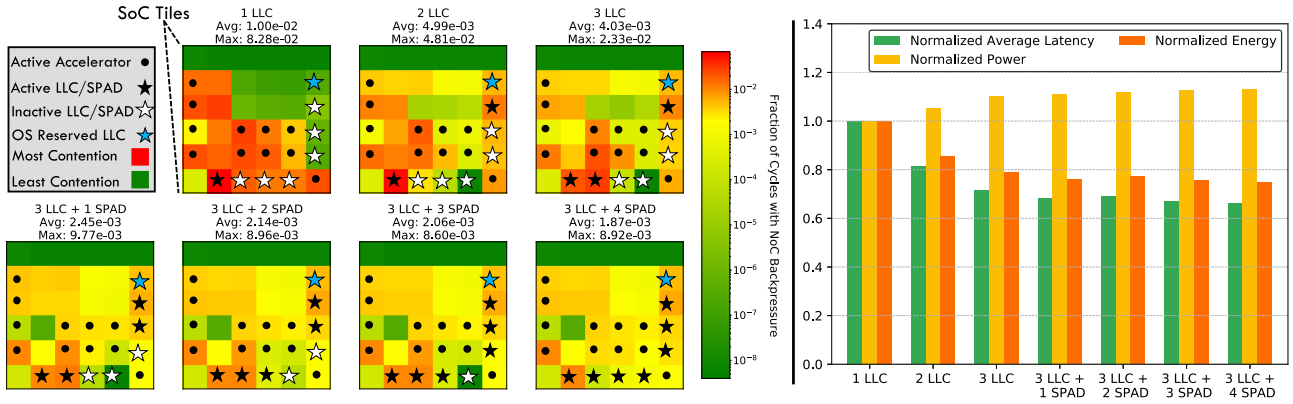


Fig. 3. Measured NoC congestion and performance, power, and energy for different configurations of the memory hierarchy.

The last column of Table I reports the area utilization of each tile type in EPOCHS-1. We chose a tile size of 1×1 mm to accommodate the tile with the highest logic density. Because the NoC logic is included in each tile, we also needed to leave enough room to route the NoC wires through each tile; this degrades the area utilization because a significant portion of the tile is occupied only by wires. The uniform tile size greatly simplifies top-level floorplanning and timing closure [13], but could lead to underutilization of the area budget for some tiles. In EPOCHS-1, the utilization of the area of the tiles ranges from 10.1% (nightVision) to 85.6% (scratchpad). While optimizing area utilization was not a primary goal for this chip, there are many techniques that can be used to minimize tile-area underutilization in tile-based SoC architectures, including, for example, clustering multiple IPs in the same tile (e.g., the cryptography tile in EPOCHS-1) and augmenting the intra-tile memory to improve performance (i.e., caches for CPUs, scratchpads for accelerators), and moving the NoC logic outside the tiles.

B. Scalable Memory Hierarchy

Fig. 3 shows how EPOCHS-1's memory hierarchy can scale up to handle complex applications using a significant portion of the chip. In this experiment, 11 accelerators, marked in black dots, run in parallel on top of a Linux-SMP operating system. We repeat the experiment while scaling the number of active memory partitions, marked with black stars, from one to seven; one LLC tile, marked with a blue star, is always reserved for the operating system. Accelerators are assigned to each memory tile with a best-effort partitioning. Each tile in Fig. 3 shows the fraction of cycles in which the NoC queues in each tile are full, and are exerting backpressure on a neighboring tile [37]. This backpressure is used as a measure of contention on the NoC, where red corresponds to the most, and green corresponds to the least.

In the one LLC scenario, we can see significant contention at the active LLC tile (bottom left of the SoC) and in the surrounding tiles. Backpressure occurs in 8% of cycles at the active LLC, and in 1% of cycles, on average, across the whole SoC. As we add additional LLC tiles, the traffic distributes across the SoC (i.e., less red and green; more orange and yellow). Adding scratchpads further helps distribute the traffic; when all memory partitions are active, peak backpressure decreases by an order of magnitude to 0.8% of cycles, and the average decreases by 5× to 0.2% of cycles. In summary, adding additional memory partitions not only improves memory bandwidth but can also help reduce contention on the NoC.

The right side of Fig. 3 shows the corresponding latency and energy improvements from scaling up the memory hierarchy. Shown in green, the average per-application latency of the 11 accelerators decreases, as expected, as additional memory partitions are added. However, unused memory partitions can be clock- or power-gated, presenting a tradeoff between performance and energy. The yellow bars show the normalized power of each configuration, while the orange bars show the normalized energy consumption (normalized power × normalized latency) from each configuration of the SoC. For this particular application, the reduction in latency by adding more memory partitions outweighs the additional power consumption, as we see an overall drop in energy. Simpler applications, which use a smaller subset of the SoC's components, will likely exhibit different trends.

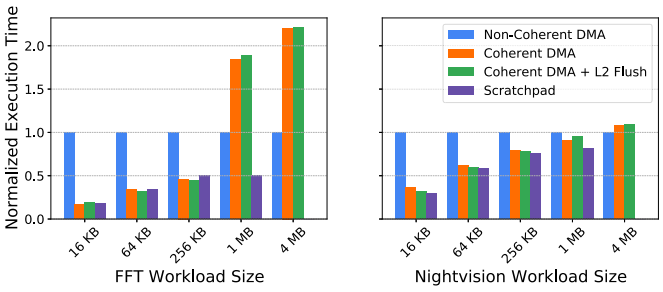


Fig. 4. Performance of different accelerators across different workload sizes when using various data-access modes.

C. Flexible NoC-Based Data Orchestration

Fig. 4 shows the importance of having heterogeneous data-access modes available to accelerators. The figure reports the execution time for the FFT (left) and nightVision (right) accelerators across a range of workload sizes from 16 kB to 4 MB. For each workload size, we report the performance of each available data-access mode, normalized to that of the non-coherent DMA mode. For the FFT accelerator, we

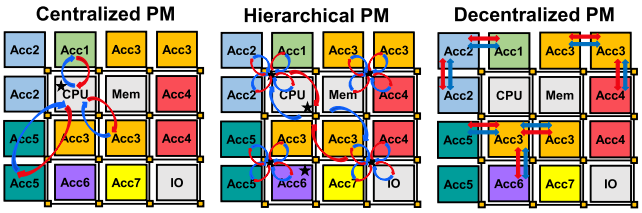


Fig. 5. Comparison of centralized, hierarchical, and decentralized PM schemes in a heterogeneous 4 × 4 SoC.

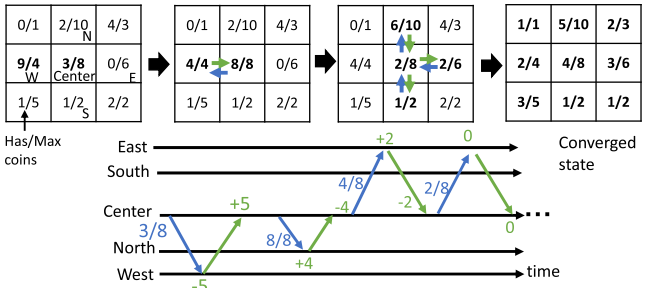


Fig. 6. Example of successive coin exchanges in a 3 × 3 SoC.

see significant variation in the performance across workload sizes. When workloads fit within a cache or scratchpad bank, there is a significant improvement from using coherent-DMA or scratchpad access modes, since the data can be fetched from on-chip storage. For the 16-kB workload, these modes are 6× faster than non-coherent DMA. When data no longer fits in the on-chip memory banks, non-coherent DMA becomes a preferable choice due to thrashing in the cache hierarchy; it outperforms the other modes by 2.2×. The nightVision accelerator shows similar trends, but the differences are less pronounced. However, this highlights that the performance of different data-access modes varies not only by workload size but also by accelerator type. This emphasizes the need for highly heterogeneous SoCs to have heterogeneous data-access modes for accelerators.

IV. BC: Decentralized Power Allocation

A. Power Allocation Approaches

SoC-level PM can enforce a global power budget under dynamic activity across the system. This power budget can be determined by the thermal design power limit in high-performance (HP) systems, battery life for portable electronics, or energy harvester output for Internet-of-Things devices. Although PM at the SoC level has been widely studied for homogeneous multicore designs, highly heterogeneous SoCs pose an additional set of challenges. Solutions must be able to scale to a large number of devices, provide a fine granularity both in terms of spatial and temporal actuation as well as power levels, and should be decoupled from the design of particular IPs. Moreover, challenges are exacerbated by the fact that, unlike CPUs, fixed-function accelerators are usually not equipped to run PM software within the tile.

Fig. 5 shows three different approaches to power allocation. Conventionally, power allocation is implemented as a software program running on the CPU or an OCC. This offers simplicity and flexibility, but suffers from a slow response time, which scales linearly with the number of tiles in the SoC, since the centralized controller must update each tile sequentially.

Another possibility is a hierarchical approach, where the SoC is partitioned into subdomains, a centralized controller allocates power to each sub-domain, which in turn has its own controller that allocates power to each tile. This can reduce convergence time, as exchanges within each sub-domain can happen in parallel. Hierarchical power allocation has worked well in homogeneous SoCs [38], where partitioning the system is straightforward. However, the lack of regularity of heterogeneous SoCs makes partitioning less efficient and could require a bespoke implementation for each heterogeneous sub-domain. Furthermore, requiring a controller in each sub-domain results in significant resource utilization overheads.

In contrast to these two approaches, BC takes a decentralized approach. As shown in Section IV-C, this approach allows for sub-linear response time scaling and low-overhead implementation.

B. Decentralized Coin-Based Power Allocation

As shown in Fig. 5, BC relies on the exchange of information between neighboring tiles. Since not all tiles can execute software, this necessitates a hardware implementation in each tile, which in turn dictates a low-overhead, easy-to-integrate solution.

Fig. 6 presents an illustration of BC's power-allocation operations. The SoC's power budget is quantized into units of power called coins. At runtime, each tile exchanges coins with its neighbors based on the number of coins it possesses (current power, called has) and the number of tiles it wants (target power, called max). Max is programmed based on an offline profiling of the tile. Specifically, each tile exchanges coins with one neighbor at a time to equalize their has/max ratios. For example, in Fig. 6, the center tile (has: 3, max: 8) will receive five coins from its west neighbor (has: 9, max: 4). This simple approach is amenable to hardware implementation, and more complex exchange mechanisms do not provide a significant improvement in the convergence time [18].

The tile's max value is set from a configuration register, which gives the software the ability to prioritize power allocation by configuring the relative target power of each tile, which BC subsequently enforces. In our experiments, setting the max value of each tile to be proportional to its power at $(V_{\max}, F_{\max})$ gave the best performance [18], but BC supports using different strategies according to workload requirements.

Local exchanges between pairs of neighboring tiles ultimately result in global convergence across all tiles. We can understand the convergence process by defining the convergence ratio $\alpha = \sum_{i=0}^{N-1} \text{has}_i / \sum_{i=0}^{N-1} \text{max}_i$, the error for a given tile $E_i = |\text{has}_i - \alpha \times \text{max}_i|$, and the global average error $E = (1/N) \sum_{n=0}^{N-1} E_i$. During each exchange, coins flow from the tile with the larger has/max ratio to the one with the smaller has/max ratio. The error E decreases or stays constant with each exchange [18]. Assuming the coin-exchange network is connected, convergence achieves the desired global minimum.

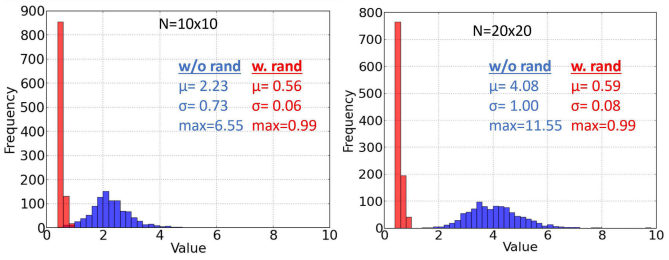


Fig. 7. Histograms of worst case absolute error across all N tiles over 1000 runs; blue bars are without random pairing and red bars are with random pairing enabled.

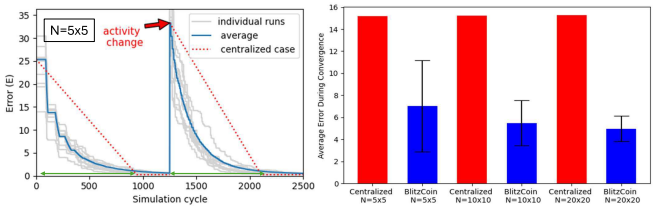


Fig. 8. Dynamic error as a function of time for ten individual Monte-Carlo runs (gray), an average of 1000 runs (blue), and a centralized sequential tile update (dashed red) (left). Average error during convergence with $3-\sigma$ error bars (right).

However, when a tile is idle, max is set to zero and the tile rejects coins, effectively interrupting exchanges through the idle tile and forming a break in the network. To ultimately ensure a global minimum is still achieved, we introduce a mechanism where tiles occasionally randomly pairs with non-neighbors throughout the whole SoC. This pairing does not need to be truly random and we simplify the implementation by having tiles iteratively select a non-neighbor after a fixed number of exchanges.

C. Analytical Results

We used a custom high-level simulator to perform a design-space exploration of BC and validate the coin-exchange algorithm on large SoCs (up to 400 tiles). Fig. 7 illustrates the worst case error $E_{i,\max}$ after reaching steady state from different random starting conditions. The blue bars show cases without random pairing, where getting stuck in a local minimum leaves a non-negligible amount of error in some tiles. Instead, with random pairing enabled (red), the error is always less than 1 and is only limited by the quantized nature of the coins, resulting in global convergence.

In addition to steady-state error, it would also be valuable to quantify the error during the convergence process. In the case of a centralized scheme, after computing the new power allocation, the controller will need to sequentially update the tiles, resulting, on average, in a reduction of the error over time following a linear trend. In contrast, during the parallel exchanges of coins, the first few exchanges contribute the most to error reduction, as the allocation of coins to immediate neighbors already leads to significantly more optimal allocations. Fig. 8 highlights this phenomenon and plots the simulated error as a function of time after SoC initialization, as well as after an activity change. The average of 1000 simulations as well as ten individual results are plotted.

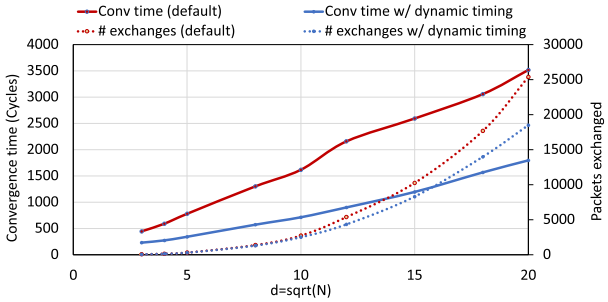


Fig. 9. Number of packets exchanged and time (NoC cycles) to reach convergence (Err < 1.0) as a function of the SoC dimension $d = (N)^{1/2}$ when comparing the conventional exchange scheme and a scheme with dynamic timing enabled.

These results confirm that the average error (in blue) decreases monotonically over time, at a convergence rate that is approximately exponential. This can be compared with a linear convergence trend for a centralized scheme (red dots). Even assuming the same convergence time to $E = 1$, the distributed exchange achieves an average error that is 51% to 63% lower than the centralized scheme during the time displayed. Therefore, not only does the distributed scheme converge faster to the desired allocation, but also has a lower transient error.

Finally, we examine the speed at which this convergence is achieved as a function of the size of the SoC, expressed as the number N of tiles. In schemes that use a centralized controller, with each activity change, the centralized controller must read the current allocation from each tile, decide on a new allocation, and sequentially pass this information back to each tile, resulting in a response time that is at best linear with N. In contrast, Fig. 9, shows that the convergence time of our scheme (solid lines) increases proportionally to $d = \text{sqrt}(N)$ due to coin exchanges that occur in parallel between neighboring tiles. Fig. 9 also compares the default implementation to one with an exponential backoff strategy (blue lines), which helps to both speed up convergence time and reduce the total number of packets exchanged (dashed lines), thus alleviating interconnect congestion concerns. It is important to note that in addition to the increase in response time with SoC size, the average time between required allocation decisions is proportional to $1/N$, making its scalability requirement even more critical [18].

D. Hardware Implementation

The power allocation is governed by a finite state machine (FSM) [18], whose implementation has a few additional practical considerations. First, we set the coin counter's precision to 6 bits, which gives a satisfactory granularity of 64 power levels per tile. At the same time, it still allows the arithmetic of the FSM to finish in one clock cycle. In particular, the coin-exchange arithmetic includes a divider that adds significant area and delay when the bitwidth of the coin count is large.

The second consideration arises from the fact that coin-exchange messages may have to compete with other message types when traveling on the NoC. Hence, a coin status/update packet can be delayed during the exchange. In some cases, a tile may receive a request for a second exchange before a first exchange is completed. This can lead to a negative coin count,

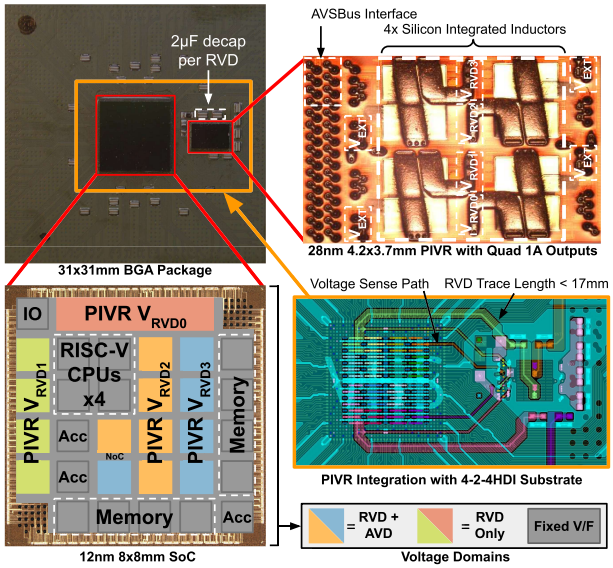


Fig. 10. Annotated package and SoC/PIVR die photographs.

which is supported by the addition of a sign bit. Although negative coin counts can appear as transient artifacts, at steady state, the coin counts are always positive.

Integrating the BC control unit into ESP required some additional changes to the SoC architecture. Typically, new functionalities for ESP tiles are added in the socket, a modular shell that provides various services (e.g., coherence, DMA, interrupts) to the encapsulated IP [39]. However, the socket is in the clock and power domains of the tile, and PM logic must be placed in the NoC domain to keep the service available even as the tile voltage and frequency change, particularly when a tile becomes fully clock- or power-gated. Since the NoC available in the ESP release only contains logic related to its core functionality, we developed a new NoC domain socket in each tile that sits in the NoC clock and power domain (i.e., before the synchronizers to the tile domain) and instantiates the BC FSM and UVFS control. It also further instantiates a set of CSRs, including configuration registers for the PM unit and the ring oscillator. The NoC domain socket only interacts with the auxiliary plane, which provides access to these CSRs and carries the new coin-exchange message type.

We can quantify the overhead of running the coin-exchange algorithm on the auxiliary NoC plane. Each coin exchange happens at a programmable rate, set nominally to 100 NoC cycles, and it requires a total of four flits, due to the limited amount of information needed for each update. This results in a maximum usage of 4% of the bandwidth of the auxiliary NoC plane during a change in activity. Our proposed dynamic exchange rate scaling [18] reduces this amount by 20× to 0.2% once a steady state is reached. Moreover, coin exchanges are carried out on a plane used for short auxiliary messages that is separate from the one used for data transfers between cores and accelerators, and hence has no impact on the data movement.

V. HUVFS-Enabled PM Cluster

The coin allocation decided by the distributed BC exchanges must be converted into a per-tile power enforcement. This

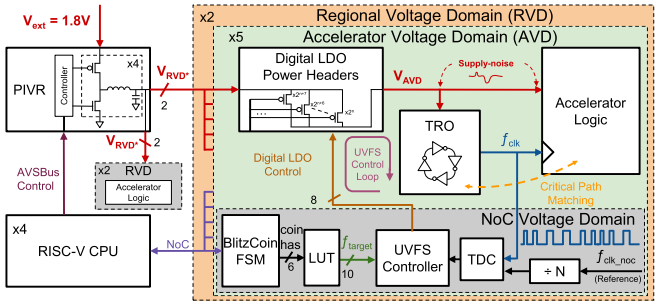


Fig. 11. HUVFS architecture block diagram.

section presents the principles of our proposed HUVFS regulation, combining an in-package switched regulator with per-tile LDO-based UVFS [25]. It then describes its implementation in the PM cluster of the EPOCHS-1 SoC.

A. Power Enforcement With Hybrid Voltage Regulation

In addition to decentralized power allocation, we present a novel power-enforcement scheme leveraging HVR. The fully-packaged system is shown in Fig. 10 with the EPOCHS-1 SoC mounted in the center of the ball grid array (BGA) package and the PIVR mounted adjacently on the right. The PIVR contains control circuits, power FETs, and back-end ferromagnetic inductors, all integrated into a 28-nm die; it is supplied with a 1.8-V input.

Fig. 11 shows the block diagram for the entire PM system. The PIVR supports four independently controlled outputs and is mounted to minimize the trace length (<17 mm) and associated parasitic inductance (<2 nH) of the V_{RVD} rails for the SoC. Each of these four outputs regulates an on-substrate regional voltage domain (RVD).

Fifteen accelerators are each assigned to one of four RVDs. Two RVDs each contain five UVFS-capable accelerator tiles: one FFT, two Viterbi, and two NVDLA. The UVFS-capable tiles contain an LDO, supplied by the PIVR-controlled RVD, which further divides the RVDs into accelerator voltage domains (AVDs).

The RISC-V CPUs can control the PIVR using the adaptive voltage scaling bus protocol (AVSBus); they can also manually set the target frequency of each accelerator. Alternatively, our UVFS control can automatically adjust the frequency based on the power and voltage dictated by BC.

B. Per-Tile UVFS Implementation

Fig. 11 further shows the block diagram of the per-tile UVFS implementation. As described in Section IV, BC logic generates the coin allocation of the accelerator. It is then converted to a target frequency f_{target} by a look up table (LUT) containing the accelerator-specific power/frequency curves. The UVFS controller then locks on to f_{target}. The local f_{clk} is driven from a TRO that can be adjusted with different amounts of delay elements to track process–voltage–temperature variations in the accelerator's critical path. A time-to-digital converter (TDC) measures the TRO frequency, which is used

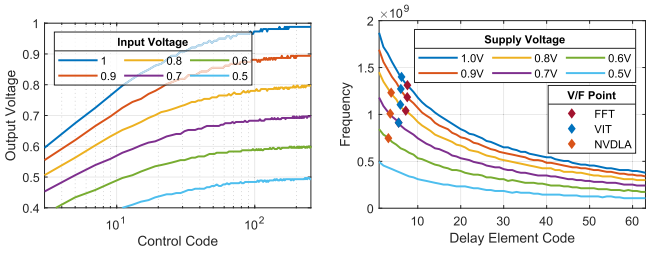


Fig. 12. LDO voltage dropout with V_{in} swept from 0.5 to 1 V (left). TRO voltage–frequency characterization (right).

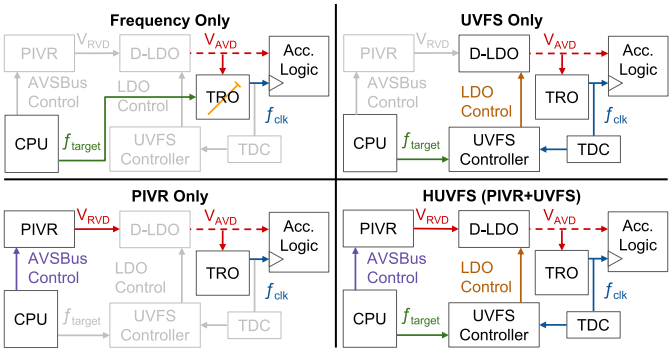


Fig. 13. Comparison of evaluated power-enforcement techniques.

as feedback to the UVFS controller, by counting the clock edges captured in a known reference clock cycle.

As shown in Fig. 11, each accelerator core consists of two physical voltage domains: a DVFS-capable AVD and a NoC domain with fixed V/F. PM control is placed in the NoC domain and consists of the BC FSM, the UVFS controller, and the TDC. The UVFS controller uses PI control, which is implemented with a Han-Carlson adder and a pipelined multiplier. The TDC is implemented with a counter that is in the NoC power domain but operates using f_{clk} from the AVD. Hence, some input and output signals must be synchronized. The reference clock divider is also programmable, so the latency and accuracy can be balanced at runtime. The CSRs located in the NoC domain are used to set the tunable parameters of the TRO, TDC, and PI controller. The UVFS controller output is then used to control the LDO.

The LDO is implemented with power-header standard cells that are meant for power gating but are repurposed for this application. The LDO has less than 0.03% area overhead, and the header cells are evenly distributed over the AVD. The LDO control signals from the UVFS controller go through the AVD, which could be power-gated, so always-on buffers are used to drive the gate of the PMOS headers. When the LDO is set for minimal dropout with the RVD voltage equal to 0.88 V, the dc voltage drop is less than 2 mV. The TRO is located in the AVD, supplied by the output of the LDO. Critical-path matching between the TRO and accelerator logic is a critical aspect of the UVFS system. To accomplish this, the TRO uses multiplexers to selectively add up to 64 voltage-sensitive delay elements to the ring-oscillator feedback path.

C. LDO and TRO Characterization

Fig. 12 (left) shows the LDO voltage dropout at a variety of input voltages between 0.5 and 1 V, while sweeping the 8-bit control code. Lower control codes correspond to higher LDO resistance. Even with the log scale on the x-axis, nonlinear behavior is observed. This created challenges in the PI controller tuning as the control effort per bit at the low end is exponentially higher than at the high end. Due to this, we typically operate the LDO at the low end of the control codes to maximize the UVFS control authority and improve the voltage slew rate.

Fig. 12 (right) plots the frequency of the TRO as a function of the control code that dictates the number of delay elements used, while also sweeping the voltage from 0.5 to 1V. This shows the range of frequencies the TRO can achieve at each supply voltage. As shown in Section III, we determined the maximum operating frequency for each accelerator across a voltage range. With perfect matching, the same TRO control code could match the maximum frequency across all voltages. Since we use the same TRO in each tile, this is difficult to do in practice. However, we can see that over certain voltage ranges for the FFT, Viterbi, and NVDLA accelerators, the maximum frequencies are fairly aligned to control codes. These points are shown as diamonds in Fig. 12 (right). To make sure the logic is free from timing violations across a range of voltages, we choose the highest delay element code from this set of points.

D. Baseline PM Techniques

In addition to HVR, we also implemented three other baseline PM techniques, shown in Fig. 13. All the sub-figures show a simplified block diagram of the HVR system with the disabled components shown in gray. To implement the frequency-only technique, the CPU sends a command to the TRO to adjust the number of delay elements, while the PIVR outputs 0.97 V and the LDO is held at a fixed resistance, which create a constant AVD voltage. This allows the CPU to perform frequency scaling using open-loop TRO control. This technique reduces dynamic power but does not impact the leakage power, so the power savings are limited. For the UVFS-only technique, the PIVR supplies a constant voltage while the UVFS system locks onto the frequency target given by the CPU by using the LDO to regulate the AVD voltage. For the PIVR-only technique, the CPU controls the PIVR to change the RVD voltage while the LDO is set at a fixed dropout voltage. The frequency is set to the target using open-loop tuning of the RVD voltage. Finally, using all the components of the HVR system, the PIVR can adjust the RVD voltage, while the LDO simultaneously adjusts the AVD voltage. The compared measured performance of the different techniques is discussed in Section VI.

VI. PM Results

We first present a characterization of the HUVFS implementation and its performance compared with the baselines. Next, we show measured results from deploying BC on the manufactured EPOCHS-1 SoC. Finally, we evaluate system-level results across multiple SoCs combining Blitzcoin-based power allocation and HUVFS-based power enforcement.

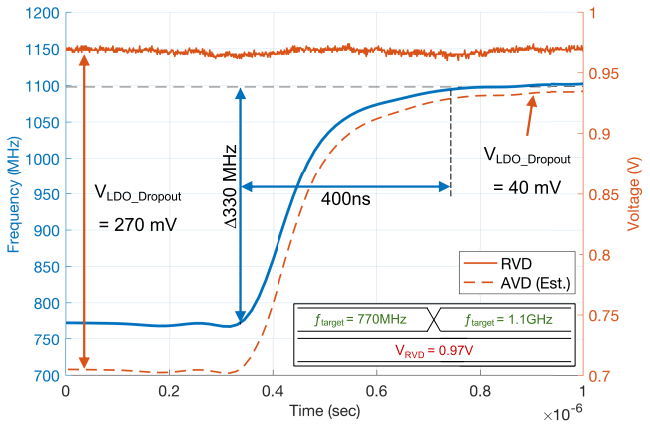


Fig. 14. Transient response of UVFS during a frequency target step.

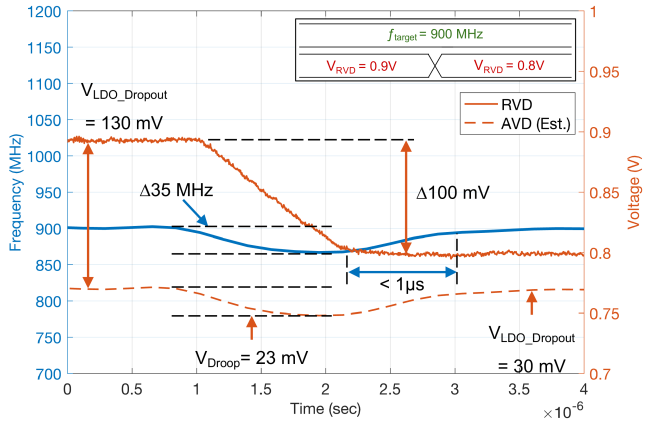


Fig. 15. Transient response of PIVR and UVFS during a V_{RVD} target step.

A. HUVFS Performance

Figs. 14–16 show measurements from our HUVFS system. In each figure, the measured RVD voltage is shown with a solid red line. The TRO frequency, shown with a blue line, is measured using an oscilloscope and then the short-time-Fourier transform is taken to plot the changes in frequency with respect to time. The AVD voltage, shown with a red dashed line, is estimated based on our voltage–frequency characterization of the TRO, as shown in Fig. 12.

First, to show the functionality and response time of our UVFS implementation, we performed a frequency target step, as shown in Fig. 14. For this experiment, the frequency target was changed from 770 MHz to 1.1 GHz, while the RVD voltage was kept at 0.97 V. Initially, the LDO had a voltage dropout of 270 mV; the UVFS system reduced the dropout to 40 mV to raise the AVD voltage and reach the higher frequency target. Notably, the frequency can change by 330 MHz in just 400 ns.

Next, we demonstrate the functionality and performance of the hybrid voltage regulator by performing an RVD voltage step, as shown in Fig. 15. During this experiment, the frequency target was kept at a constant 900 MHz while the RVD voltage was changed from 900 to 800 mV. The LDO, which initially had a dropout voltage of 130 mV, then compensated for the 100 mV reduction of RVD voltage by reducing the

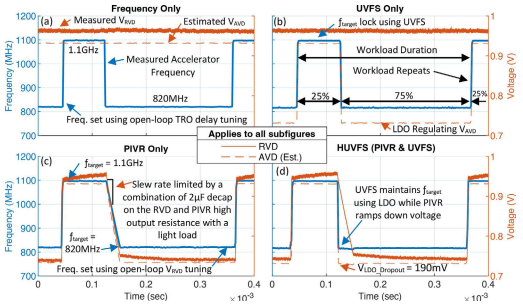


Fig. 16. Workload waveform comparison for four power enforcement techniques: (a) frequency only, (b) UVFS only, (c) PIVR only, and (d) HUVFS (PIVR & UVFS).

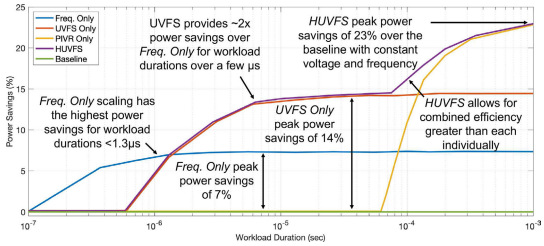


Fig. 17. Power savings of the four power-enforcement techniques.

voltage dropout to 30 mV. During this transient, the supply droop to the logic was 23 mV, which resulted in a 35 MHz (<4%) droop in the frequency. The UVFS system was able to reach a steady state at the target frequency within 1 μs of the supply voltage drop.

Finally, we quantify the HUVFS performance compared with the three presented baselines in Fig. 13. We run a periodic workload on the FFT accelerator with a target frequency of 1.1 GHz for 25% of the period and 820 MHz for the remaining 75% of the period. The workload duration was then swept from 100 ns to 1 ms while utilizing the four different PM strategies.

Fig. 16 shows measured waveforms for each PM technique. For the frequency-only technique [see Fig. 16(a)], the frequency changes between 1.1 GHz and 820 MHz, while the RVD and AVD voltage are kept constant. For the UVFS-only waveform [see Fig. 16(b)], the LDO is active, and the AVD is regulated to maintain the frequency targets, while the RVD is kept constant. This allows for power savings by utilizing the LDO. The PIVR-only technique [see Fig. 16(c)] controls the RVD voltage to indirectly set the AVD voltage, while the LDO is held at a fixed dropout voltage. Since the PIVR has 2 μF of decap per output, the slew rate is limited. This is especially noticeable on the falling edge of the frequency. Since all of the accelerators in this RVD, except the active FFT, are powered off, the load is light. With full HVR [see Fig. 16(d)], the PIVR regulates the RVD, while the LDO simultaneously regulates the AVD. This is particularly noticeable on the falling edge of the frequency target, when the PIVR is slew-rate limited but the LDO can quickly scale the AVD voltage and lock onto the frequency target with UVFS. The UVFS system continues to dynamically adjust the LDO to maintain a constant frequency until the RVD voltage settles.

Fig. 17 compares the power savings for each of the four techniques across the different workload durations. The cur-

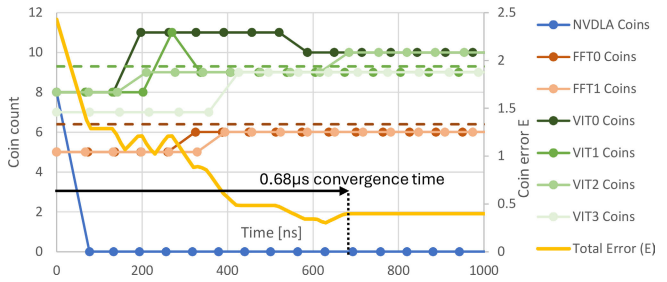


Fig. 18. Silicon-measured response time of the proposed power-allocation scheme and annotated error during convergence. Dashed lines are the allocation target for the Viterbi and FFT accelerators.

rent and voltage supplied to the PIVR were measured while running a periodic workload. The workload is run long enough for the current to stabilize to measure the average power. The workload duration, which is swept from 100 ns to 1 ms, is shown on the X-axis, while the power savings are shown on the y-axis. The frequency-only technique (blue) has the highest power savings for workload durations that are less than 1.3 μs and shows peak savings of 7% over the baseline. The UVFS-only technique (red) doubles the power savings over the frequency-only technique for workload durations over a few microseconds, with peak power savings of 14%. The PIVR-only technique (yellow) can only provide power savings for workload durations over 60 μs, but the PIVR can boost power savings significantly to 23% using the efficient buck converter. Finally, the HVR technique (purple) shows power savings greater than any technique independently for durations around 100 μs; it also matches the performance of the best other technique for all durations above 1.3 μs.

B. BC Silicon Validation

To validate the functionality of the implementation of BC in the ten-tile PM cluster of EPOCHS-1, we first performed a test to observe a redistribution of coins following an activity change in the SoC. Because the current coin count is assigned to a memory-mapped register in the SoC, it can be polled by software. We ran a bare-metal workload, in order to poll at short intervals, with seven accelerators: four Viterbi, tw FFTs, and one NVDLA. Fig. 18 shows the measured coin count in each tile over time, captured at the end of the NVDLA task. When the NVDLA completes, we can see its coins being reallocated within 0.68μs and the average error E reaching a value of 0.40 once the steady state is reached. As shown in the simulations of Fig. 8, the largest change in error happens immediately. The time to reach the error threshold $E \leq 1$ is 0.32μs, much faster than the 0.68μs taken for full settling.

Next, we evaluate the performance improvements by leveraging BC for the accelerators in the PM cluster. The top of Fig. 19 presents the measured power trace of a parallel five-accelerator workload across two cases. As a baseline, the allocation power is static, and hence each tile has a fixed power. Once one accelerator completes, the power is not reallocated to the other tiles, resulting in unused power ($P_{\text{avg}}/P_{\text{max}} = 61.7\%$). For the same 80-mW power cap, BC in conjunction with the per-tile UVFS actuator dynamically reallocates power, resulting in a much better power efficiency ($P_{\text{avg}}/P_{\text{max}} \geq 99\%$) and a 26% throughput improvement.

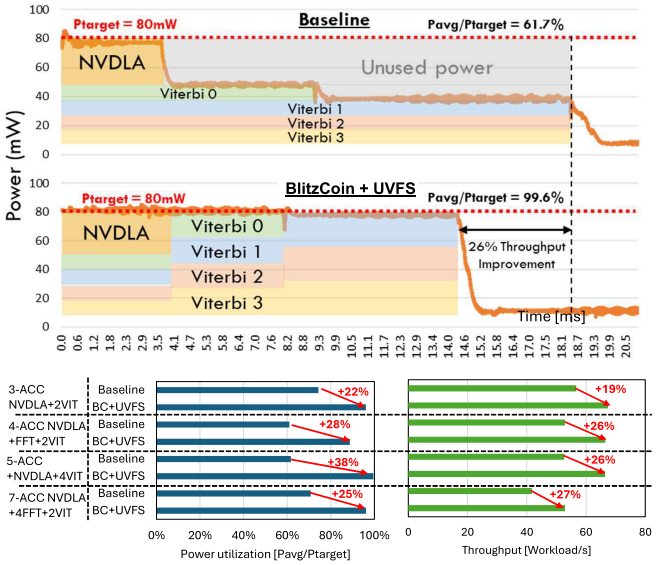


Fig. 19. Decentralized power-allocation performance improvements.

Measurements for other workloads using three, four, or seven accelerators result in similar power utilization and throughput improvements.

C. Extension to Other SoCs and Techniques

Finally, by means of register-transfer level (RTL) simulations, we evaluate two different SoC configurations, which were easily composed thanks to the modularity of ESP, to further characterize BC in addition to the silicon measurements. The first is a 3 $\times$ 3 SoC including the accelerators from the PM cluster of EPOCHS-1, which target relevant kernels of an industry-developed application for connected autonomous vehicles [40]. To highlight the reusability of BC, the second SoC includes a different set of accelerators to target computer-vision applications, namely general matrix multiplication and 2-D convolution for CNN inference, as well as the nightVision accelerator from EPOCHS-1. In both SoCs, all of the accelerators are equipped with BC, including a per-tile TRO, as in the silicon implementation. Our evaluations demonstrate that BC can be adapted to a wide range of SoCs with diverse accelerators and NoC topologies.

We mapped both SoCs to the same 12nm technology library of EPOCHS-1, and ran simulations on two types of workloads: 1) parallel (Par), where all tiles start their execution at the same time and 2) dependent (Dep), where there exist dependencies between the kernels running on each tile, represented in the form of a directed acyclic graph (DAG). We validated the behavior of BC with UVFS in these SoCs and simulated both the workload runtime and PM response time. As part of our evaluations, we compared the performance of Blitzcoin to a state-of-the-art centralized baseline for heterogeneous SoCs (C-RR) [41] and a centralized implementation of our proposed algorithm (BC-C).

Fig. 20 presents these comparative results for different power caps and workloads across the two SoCs. Compared with the C-RR baseline, BC-C provides 20%–24% throughput improvement on average, while our fully decentralized hard-

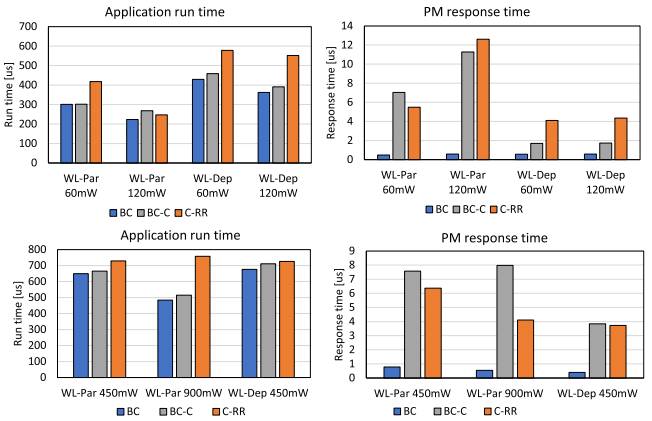


Fig. 20. Comparisons of execution time (left) and response time (right) of BC, BC-C, and C-RR over different SoCs [3 × 3 (top), 4 × 4 (bottom)], power budgets, and workload types.

ware implementation (BC) improves C-RR's response time by 8×–12× and throughput by 25%–34%. The benefit of BC's shorter PM response time becomes more significant as the SoC size increases due to the increase in activity changes, supporting up to 13× larger SoCs compared with C-RR [18].

Fig. 21 shows the system-level benefits of applying power allocation with BC in conjunction with HUVFS. We illustrate these benefits using a workload DAG similar to the WL-Dep case for the 3 × 3 SoC. To showcase the benefits of the PIVR, the tiles were split into an HP tile group (shown in red) and a high-efficiency (HE) tile group (in blue), based on each tile's peak operating voltage, as shown in Fig. 21(c). For the baseline consisting of a single voltage domain, the PIVR supplies the maximum voltage required for all of the tiles, while in the case of the two groups, it only supplies the maximum voltage required by each group. The voltage traces for the tiles are shown in Fig. 21(a), broken down by the HP (top) and HE (bottom) groups. The power breakdowns in Fig. 21(b) show the LDO power loss, buck power loss, and power of the voltage domains. The bottom figure also shows the power savings (red) obtained by using the two groups. This saves around 17% of power on average, which in turn can be re-targeted toward the tile power budget, thus resulting in further throughput improvements. The execution time for the static and dynamic PM techniques is shown in Fig. 21(d). In static power allocation, each tile is assigned a fixed power value proportional to its max power, so as to ensure the entire chip remains within the specified power budget. In the case of BC+HUVFS, the power savings shown in Fig. 21(c) are used to increase the tile power, enabling a 57% speedup over C-RR, the baseline DVFS design, and a 3× speedup over a static allocation scheme, for a fixed system-level power budget.

VII. Agile, Open-Source SoC Design

A. Collaborative SoC Design

EPOCHS-1 was taped out by a nine person team consisting of six Ph.D. students, one postdoctoral researcher, and two industry researchers in a period of three months thanks to our agile ESP design methodology, intrinsically suited to foster collaborative, multi-institution SoC design [14], [39]. In addition to its modular architecture, ESP provides several flows for designing new IPs and integrating existing TP IPs into the SoC design [42]. These flows span a range of specification languages and commercial tools; this flexibility was key to enabling the degree of heterogeneity of EPOCHS-1, which is unprecedented for an SoC designed by an academic team. Table II describes the design and integration methodology for the various IPs included in EPOCHS-1. The table shows the designer, the specification language, the commercial high-level synthesis tool used (if applicable), and whether one of ESP's design flows or the TP integration flow was used.

TABLE II
Design and Integration Approach of EPOCHS-1 IPs

IP	Designer	Language	HLS Tool	Flow	Release
CVA6	ETH Zurich[1]	SysVerilog	N/A	N/A	[43]
NLP	Harvard	SystemC	Catapult	TP	[44]
Audio Dec	Harvard/UIUC	SystemC	Catapult	ESP	[45]
Audio Enc	Harvard/UIUC	SystemC	Catapult	ESP	[45]
NVDLA	NVIDIA	Verilog	N/A	TP	[46]
Cryptography	Columbia	C++	Catapult	ESP	[43]
Systolic Array	Harvard	SystemC	Catapult	TP	[47]
FFT	IBM/Columbia	SystemC	Stratus	ESP	[43]
Viterbi	IBM	SystemC	Stratus	ESP	[43]
NightVision	Columbia	SystemC	Stratus	ESP	[43]
BlitzCoin PM	IBM/Columbia	Verilog	N/A	N/A	[43]
NoC	Columbia	VHDL	N/A	N/A	[43]

TABLE III
Comparison of Key Metrics Across Two Chip Generations

	EPOCHS-0	EPOCHS-1
Area (mm^2)	21.6	64
Tiles	16	34
Tile Types	6	14
Clock Domains	17	35
Power Domains	16	23
Person-Months	32	27

The collaborative approach was further strengthened with our agile physical design methodology [13]. By adopting a hierarchical approach, physical design for individual tiles could proceed in parallel, better utilizing the computing resources of our institutions. After all tiles were implemented, a top-level integration flow was run to produce the final GDS. Our methodology is informed, in part, by our experience with a previous tapeout: the EPOCHS-0 SoC is a smaller 16-tile SoC, also based on the ESP architecture, that was taped out a year before EPOCHS-1 [9]. Since EPOCHS-0 was taped out in the same 12-nm technology, there was less of a learning curve with the technology for the tapeout of EPOCHS-1. Still, EPOCHS-1 required fewer person-months to tape out even though it was a much more complex chip by many metrics, as shown by Table III. This highlights the scalability of our approach.

B. Compositional Verification, Emulation, and Testing

With limited available person-hours, verification can be particularly challenging in an agile SoC design cycle. Our approach focuses on verifying the integration of components into the SoC and leverages the tile-based modularity of the ESP architecture. Our tests are written in the form of bare-

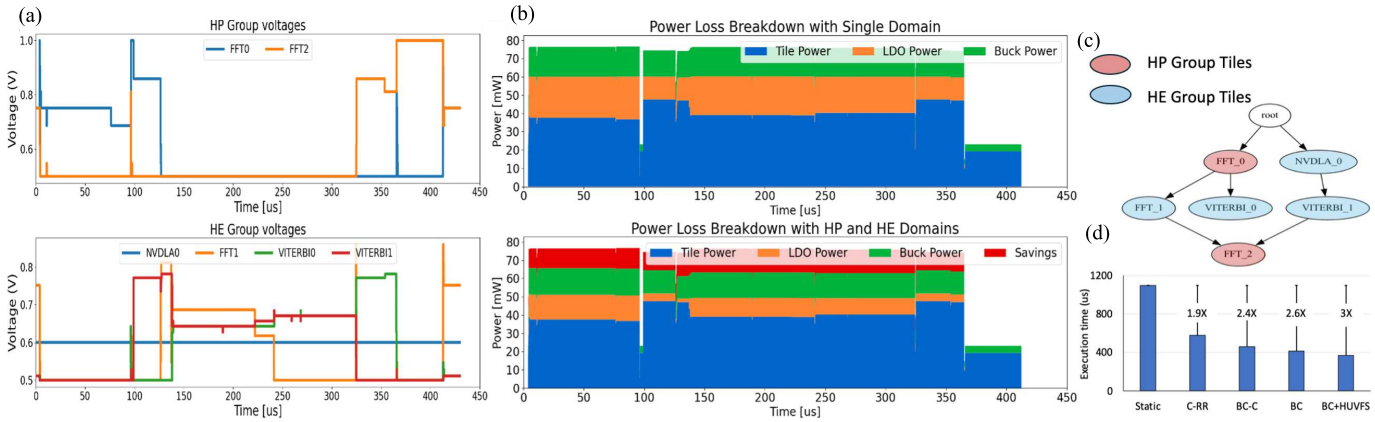


Fig. 21. (a) HP and HE group voltage traces. (b) Power breakdown with single domain (top); power breakdown and savings with HE and HP domains (bottom). (c) Workload DAG with HP and HE tile groupings. (d) Speedups over the baseline with a static power budget.

metal C programs that run on the CVA6 core to invoke a particular SoC component, such as an accelerator, and validate that its outputs are correct; the same tests can be used for RTL simulation, FPGA emulation, and chip bring-up. This software-based verification reduces the time to develop tests in contrast to cumbersome RTL testbenches. Individual components can be verified in systems that are much smaller than EPOCHS-1; for example, the integration of an accelerator can be tested in a 2 $\times$ 2 SoC with the target accelerator, a CPU tile, a LLC tile, and an IO tile. Thanks to the decoupling of the IP from the rest of the SoC provided by the tile socket [39], if the component works in a 2 $\times$ 2 SoC, it will also work in the target 6 $\times$ 6 SoC. Before the tapeout, however, we still run at least one test of each IP in the full SoC.

This compositional approach makes it possible to verify the integration of some components before others are ready, speeds up simulation time, allows us to make strong use of FPGA emulation without complex design partitioning tools, and even allows us to test individual tiles in isolation in the fabricated silicon [48]. For emulation, we rely on ESP's push-button flow for FPGA prototyping that simplifies the generation of many bitstreams, each containing a different subset of the target SoC. In addition to allowing more thorough verification through longer tests, emulation also enables early software development while awaiting the fabricated silicon. Furthermore, our ASIC test setup leverages the same connections from a host PC that we use for FPGA prototyping. This means that once: 1) basic electrical tests are conducted; 2) communication with the SoC is established; and 3) connection to the FPGA is tuned, then the exact same tests and tools that were used for FPGA emulation can be used on the SoC. This greatly speeds up chip bring-up time; after just three days of testing from receiving the packaged EPOCHS-1 dies, we were able to boot Linux SMP and run tests invoking most of the accelerators.

C. Open-Source Hardware

Building an SoC with the size, heterogeneity, and complexity of EPOCHS-1 was necessary to study critical aspects of modern SoCs. However, building an SoC, such as EPOCHS-1 is a complex task, especially for a small, mostly academic, team. Compared with the most recent heterogeneous SoCs taped out by academia [6], [7], [8], [10], EPOCHS-1 shows unprecedented levels of heterogeneity and complexity. In part, EPOCHS-1 was made possible by OSH [49]; our team alone could not have designed every component in the SoC from scratch, so IP reuse was critical. Furthermore, years of improvement of ESP through its open-source release enabled a robust integration platform for OSH components. We recognize that advancing research in SoC design requires building more systems like EPOCHS-1. To support more teams in building and doing research on complex SoCs, we have released the entire synthesizable design of EPOCHS-1 in the public domain; Table II reports the various repositories that contain the SoC's source code.

VIII. Conclusion

We presented EPOCHS-1, an SoC with unprecedented levels of heterogeneity and complexity compared with the other chips designed by academic teams in the literature. We used EPOCHS-1 to characterize and address challenges that highly heterogeneous architectures face. Different accelerators have different data-access patterns, and simultaneously running complex software applications creates resource contention issues; EPOCHS-1 addresses this with its scalable memory hierarchy and flexible NoC-based communication. The wide variety of power profiles for accelerators demands an IP-agnostic PM scheme with fast response time for a large number of tiles; EPOCHS-1 addresses this with a decentralized power-allocation scheme, BC, which has sub-microsecond response time and can scale to large systems. Finally, enforcing the allocated power across a diverse set of tiles requires a lightweight, yet efficient power delivery approach; EPOCHS-1 addresses this with hybrid voltage regulation, combining an efficient in-package regulator with a fine-grained on-chip LDO, and UVFS. Combined, these techniques show significant performance improvements of up to 57% over a state-of-the-art centralized design operating at the same power budget. Finally, we detailed the agile design methodology that enabled a small team to design, verify, and test EPOCHS-1. To foster future research in this area and enable more systems like EPOCHS-1, we have released its entire synthesizable design in the public domain as OSH.

ACKNOWLEDGMENT

The views and findings should not be interpreted as representing the official views or policies of the DoD.

Joseph Zuckerman was with Columbia University, New York, NY 10027 USA. He is now with Jump Trading, Chicago, IL 60654 USA (e-mail: jzuck@cs.columbia.edu).

Martin Cochet, Karthik Swaminathan, Alper Buyuktosunoglu, David Trilla, John-David Wellman, and Pradip Bose are with IBM Research, Yorktown Heights, NY 10598 USA.

Maico Cassel dos Santos, Kuan-Lin Chiu, Gabriele Tombesi, Kenneth L. Shepard, and Luca P. Carloni are with Columbia University, New York, NY 10027 USA.

Erik Jens Loscalzo and Paolo Mantovani were with Columbia University, New York, NY 10027 USA. They are now with Google, Mountain View, CA 94043 USA.

Tianyu Jia was with Harvard University, Cambridge, MA 02138 USA. He is now with Peking University, Beijing 100871, China.

Davide Giri, deceased, was with Columbia University, New York, NY 10027 USA.

Thierry Tambe was with Harvard University, Cambridge, MA 02138 USA. He is now with Stanford University, Stanford, CA 94305 USA.

Jeff Jun Zhang was with Harvard University, Cambridge, MA 02138 USA. He is now with Arizona State University, Tempe, AZ 85281 USA.

Giuseppe Di Guglielmo was with Columbia University, New York, NY 10027 USA. He is now with Fermilab, Batavia, IL 60510 USA.

Luca Piccolboni was with Columbia University, New York, NY 10027 USA. He is now with Microsoft, Redmond, WA 98052 USA.

En-Yu Yang was with Harvard University, Cambridge, MA 02138 USA. He is now with Google, Taipei 100615, Taiwan.

Aporva Amarnath was with IBM Research, Yorktown Heights, NY 10598 USA. She is now with AMD, Santa Clara, CA 95054 USA.

Ying Jing, Bakshree Mishra, Vignesh Suresh, and Sarita Adve are with the University of Illinois Urbana-Champaign, Champaign, IL 61820 USA.

Joshua Park was with Harvard University, Cambridge, MA 02138 USA. He is now with NVIDIA, Santa Clara, CA 95051 USA.

Samira Zaliasl, Stephen Cahill, Hesam Sadeghi, Joseph Meyer, and Noah Sturcken are with Ferric Inc., New York, NY 10001 USA.

Michael Lekas was with Ferric Inc., New York, NY 10001 USA. He is now with IBM Research, Yorktown Heights, NY 10598 USA.

David Brooks and Gu-Yeon Wei are with Harvard University, Cambridge, MA 02138 USA.

REFERENCES

[1] Y. S. Shao, B. Reagen, G.-Y. Wei, and D. Brooks, "The Aladdin approach to accelerator design and modeling," *IEEE Micro*, vol. 35, no. 3, pp. 58–70, May 2015.

[2] M. Subramony, D. Kramer, and I. Paul, "AMD Ryzen 7040 series," *IEEE Micro*, vol. 44, no. 3, pp. 18–24, May 2024.

[3] M. Ditty, "NVIDIA ORIN system-on-chip," in *Proc. IEEE Hot Chips 34 Symp. (HCS)*, Aug. 2022, pp. 1–17.

[4] Y. Yuan et al., "Intel accelerators ecosystem: An SoC-oriented perspective: Industry product," in *Proc. ACM/IEEE 51st Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2024, pp. 848–862.

[5] A. Pellegrini et al., "The arm neoverse N1 platform: Building blocks for the next-gen cloud-to-edge infrastructure SoC," *IEEE Micro*, vol. 40, no. 2, pp. 53–62, Mar. 2020.

[6] C. Schmidt et al., "An eight-core 1.44-GHz RISC-V vector processor in 16-nm FinFET," *IEEE J. Solid-State Circuits*, vol. 57, no. 1, pp. 140–152, Jan. 2022.

[7] F. Conti et al., "A 12.4TOPS/W @ 136GOPS AI-IoT system-on-chip with 16 RISC-V, 2-to-8b precision-scalable DNN acceleration and 30%-boost adaptive body biasing," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 21–23.

[8] S. K. Lee, P. N. Whatmough, M. Donato, G. G. Ko, D. Brooks, and G.-Y. Wei, "SMIV: A 16-nm 25-mm^2 SoC for IoT with arm Cortex-A53, eFPGA, and coherent accelerators," *IEEE J. Solid-State Circuits*, vol. 57, no. 2, pp. 639–650, Feb. 2022.

[9] T. Jia et al., "A 12 nm agile-designed SoC for swarm-based perception with heterogeneous IP blocks, a reconfigurable memory hierarchy, and an 800 MHz multi-plane NoC," in *Proc. IEEE 48th Eur. Solid State Circuits Conf. (ESSCIRC)*, Sep. 2022, pp. 269–272.

[10] F. Gao et al., "DECADES: A 67 mm^2, 1.46TOPS, 55 giga cache-coherent 64-bit RISC-V instructions per second, heterogeneous many-core SoC with 109 tiles including accelerators, intelligent storage, and eFPGA in 12 nm FinFET," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr. 2023, pp. 1–2.

[11] B. Khailany et al., "A modular digital VLSI flow for high-productivity SoC design," in *Proc. 55th Annu. Design Autom. Conf.*, Jun. 2018, pp. 1–6.

[12] M. C. Dos Santos et al., "A 12 nm Linux-SMP-capable RISC-V SoC with 14 accelerator types, distributed hardware power management and flexible NoC-based data orchestration," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2024, pp. 262–264.

[13] M. C. D. Santos et al., "A scalable methodology for agile chip development with open-source hardware components," in *Proc. 41st IEEE/ACM Int. Conf. Comput.-Aided Design*, Oct. 2022, pp. 1–9.

[14] P. Mantovani et al., "Agile SoC development with open ESP," in *Proc. 39th Int. Conf. Comput.-Aided Design*, Nov. 2020, pp. 1–9.

[15] B. Keller et al., "A RISC-V processor SoC with integrated power management at submicrosecond timescales in 28 nm FD-SOI," *IEEE J. Solid-State Circuits*, vol. 52, no. 7, pp. 1863–1875, Jul. 2017.

[16] J. M. Cebrión, J. L. Aragón, and S. Kaxiras, "Power token balancing: Adapting CMPs to power constraints for parallel multithreaded workloads," in *Proc. IEEE Int. Parallel Distrib. Process. Symp.*, May 2011, pp. 431–442.

[17] C. Isci, A. Buyuktosunoglu, C.-Y. Cher, P. Bose, and M. Martonosi, "An analysis of efficient multi-core global power management policies: Maximizing performance for a given power budget," in *Proc. 39th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Dec. 2006, pp. 347–358.

[18] M. Cochet et al., "BlitzCoin: Fully decentralized hardware power management for accelerator-rich SoCs," in *Proc. ACM/IEEE 51st Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2024, pp. 801–817.

[19] P. Shah, R. G. Shenoy, V. Srinivasan, P. Bose, and A. Buyuktosunoglu, "TokenSmart: Distributed, scalable power management in the many-core era," *IEEE Comput. Archit. Lett.*, vol. 20, no. 1, pp. 42–45, Jan. 2021.

[20] F. U. Rahman et al., "A unified clock and switched-capacitor-based power delivery architecture for variation tolerance in low-voltage SoC domains," *IEEE J. Solid-State Circuits*, vol. 54, no. 4, pp. 1173–1184, Apr. 2019.

[21] C.-H. Huang et al., "Improving SIMO-regulated digital SoC energy efficiencies through adaptive clocking and concurrent domain control," *IEEE J. Solid-State Circuits*, vol. 57, no. 1, pp. 90–102, Jan. 2022.

[22] X. Wang, X. Liu, and W.-H. Ki, "A self-clocked and variation-tolerant unified voltage-and-frequency regulator for in-order executed digital loads," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 70, no. 11, pp. 4627–4640, Nov. 2023.

[23] F. U. Ahmed, Z. T. Sandhie, L. Ali, and M. H. Chowdhury, "A brief overview of on-chip voltage regulation in high-performance and high-density integrated circuits," *IEEE Access*, vol. 9, pp. 813–826, 2021.

[24] J. Haj-Yahya et al., "FlexWatts: A power{-} and workload-aware hybrid power delivery network for energy-efficient microprocessors," in *Proc. 53rd Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Oct. 2020, pp. 1051–1066.

[25] E. Loscalzo et al., "A 400-ns-settling-time hybrid dynamic voltage frequency scaling architecture and its application in a 22-core network-on-chip SoC in 12-nm FinFET technology," in *Proc. IEEE Symp. VLSI Technol. Circuits (VLSI Technol. Circuits)*, Jun. 2024, pp. 1–2.

[26] Columbia SLD Group.(2024). *EPOCHS-1*. Columbia SLD Group. [Online]. Available: https://github.com/sld-columbia/epochs-1

[27] Columbia SLD Group.(2024). *ESP Website*. [Online]. Available: https://esp.cs.columbia.edu

[28] F. Zaruba and L. Benini, "The cost of application-class processing: Energy and performance analysis of a Linux-ready 1.7-GHz 64-bit RISC-V core in 22-nm FDSOI technology," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 11, pp. 2629–2640, Nov. 2019.

[29] D. Giri, P. Mantovani, and L. P. Carloni, "NoC-based support of heterogeneous cache-coherence models for accelerators," in *Proc. 12th IEEE/ACM Int. Symp. Netw.-Chip (NOCS)*, Oct. 2018, pp. 1–8.

[30] J. D. Zuckerman, P. Mantovani, D. Giri, and L. P. Carloni, "Enabling heterogeneous, multicore SoC research with RISC-V and ESP," in *Workshop Comput. Archit. Res. RISC-V (CARRV)*, 2022.

[31] D. Trilla, J.-D. Wellman, A. Buyuktosunoglu, and P. Bose, "NOVIA: A framework for discovering non-conventional inline accelerators," in *Proc. 54th Annu. IEEE/ACM Int. Symp. Microarchitecture*, Oct. 2021, pp. 507–521.

[32] J. Zuckerman, D. Giri, J. Kwon, P. Mantovani, and L. P. Carloni, "Cohmeleon: Learning-based orchestration of accelerator coherence in heterogeneous SoCs," in *Proc. 54th Annu. IEEE/ACM Int. Symp. Microarchitecture*, Oct. 2021, pp. 350–365.
[33] T. Tambe et al., "EdgeBERT: Sentence-level energy optimizations for latency-aware multi-task NLP inference," in *Proc. 54th Annu. IEEE/ACM Int. Symp. Microarchitecture*, Oct. 2021, pp. 830–844.
[34] T. Tambe et al., "A 12 nm 18.1TFLOPs/W sparse transformer processor with entropy-based early exit, mixed-precision predication and fine-grained power management," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 342–344.
[35] E. G. Cota, P. Mantovani, G. Di Guglielmo, and L. P. Carloni, "An analysis of accelerator coupling in heterogeneous architectures," in *Proc. 52nd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2015, pp. 1–6.
[36] D. Giri, K.-L. Chiu, G. Di Guglielmo, P. Mantovani, and L. P. Carloni, "ESP4ML: Platform-based design of systems-on-chip for embedded machine learning," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2020, pp. 1049–1054.
[37] L. P. Carloni, "From latency-insensitive design to communication-based system-level design," *Proc. IEEE*, vol. 103, no. 11, pp. 2133–2151, Nov. 2015.
[38] T. S. Muthukaruppan, A. Pathania, and T. Mitra, "Price theory based power management for heterogeneous multi-cores," *SIGARCH Comput. Archit. News*, vol. 42, no. 1, pp. 161–176, Feb. 2014, doi: 10.1145/2541940.2541974.
[39] L. P. Carloni, "The case for embedded scalable platforms," in *Proc. Design Autom. Conf. (DAC)*, Jun. 2016, pp. 17:1–17:6.
[40] *Mini-ERA: Simplified Version of the Main ERA Workload*. Accessed: 2025. [Online]. Available: https://github.com/IBM/mini-era
[41] P. Mantovani et al., "An FPGA-based infrastructure for fine-grained DVFS analysis in high-performance embedded systems," in *Proc. 53rd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2016, pp. 1–6.
[42] D. Giri, K.-L. Chiu, G. Eichler, P. Mantovani, and L. P. Carloni, "Accelerator integration for open-source SoC design," *IEEE Micro*, vol. 41, no. 4, pp. 8–14, Jul. 2021.
[43] Columbia SLD Group.(2024). *ESP Release*. [Online]. Available: https://github.com/sld-columbia/esp/tree/epochs
[44] Harvard Architecture, Circuits, and Compilers Group. (2024).*Edge-BERT*. [Online]. Available: https://github.com/harvard-acc/EdgeBERT
[45] Columbia SLD Group.(2024). *EPOCHS Audio Accelerators*. [Online]. Available: https://github.com/sld-columbia/epochsaudioaccelerators
[46] NVIDIA.(2021). *NVDLA Open Source Hardware, Version 1.0*. [Online]. Available: https://github.com/sld-columbia/nvdla-hw/
[47] Harvard Architecture, Circuits, and Compilers Group. (2021).*A HLS Based Systolic Array Accelerator*. [Online]. Available: https://github.com/harvard-acc/ESPSystolicArrayAccelerator
[48] G. Tombesi et al., "SoCProbe: Compositional post-silicon validation of heterogeneous NoC-based SoCs," in *Proc. IEEE Design Test*, vol. 40, Dec. 2023, pp. 64–75.
[49] G. Gupta, T. Nowatzki, V. Gangadhar, and K. Sankaralingam, "Kickstarting semiconductor innovation with open source hardware," *Computer*, vol. 50, no. 6, pp. 50–59, 2017.

Joseph Zuckerman received the Ph.D. and M.S. degrees in computer science from Columbia University, New York, NY, USA, and the S.B. degree in electrical engineering from Harvard University, Cambridge, MA, USA.

For three years, he led the development of ESP, an open-source platform for system-on-chip (SoC) design. He is currently a Hardware Engineer with Jump Trading, Chicago, IL, USA. His research interests broadly span heterogeneous computing, hardware acceleration, and system-on-chip design and programming.

Martin Cochet (Senior Member, IEEE) received the Dipl.Ing. degree from the École Polytechnique, Palaiseau, France, in 2012, the M.S. degree from Imperial College London, London, U.K., in 2013, and the Ph.D. degree in electronic engineering from Aix-Marseille University, Marseille, France, in 2016.

He is currently a Research Scientist with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA. He holds five U.S. patents. His research interests include mixed-signal circuit design for high-speed wireline communications and system-on-chip (SoC) power management.

Maico Cassel dos Santos (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering and the M.S. degree in microelectronics from the Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil, in 2008 and 2015, respectively. He is currently pursuing the Ph.D. degree in computer science with Columbia University, New York, NY, USA.

His research interests extend across multiple domains of computer architectures and hardware design, with ongoing work in areas such as integrated circuit design methodologies and heterogeneous architectures.

Mr. Santos is a recipient of the IBM Ph.D. Fellowship Award (2024).

Erik Jens Loscalzo received the B.E. degree in electrical engineering from Dartmouth College, Hanover, NH, USA, in 2019, and the Ph.D. degree in electrical engineering from Columbia University, New York, NY, USA, in 2024.

He is currently working with Google, Mountain View, CA, USA, as a silicon power engineer. His research interests are in integrated power management and adaptive clocking.

Karthik Swaminathan (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from The Pennsylvania State University, University Park, PA, USA.

He is a Senior Research Scientist with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA. His research interests include the design and optimization of resilient and power-aware processor architectures.

Tianyu Jia (Member, IEEE) received the B.S. and M.S. degrees from Beijing University of Posts and Telecommunications, Beijing, China, and the Ph.D. degree in computer engineering from Northwestern University, Evanston, IL, USA.

He was a Post-Doctoral Fellow with Harvard University, Cambridge, MA, USA, and an Assistant Research Professor with Carnegie Mellon University, Pittsburgh, PA, USA. He is currently an Assistant Professor with the School of Integrated Circuits, Peking University, Beijing, China. His research interests include domain-specific architecture, compute/processing-in-memory, and heterogeneous system-on-chip (SoC) design.

Dr. Jia received the IEEE SSCS Predoctoral Achievement Award in 2020 and the ISLPED Best Paper Award in 2025.

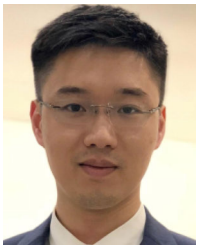
Davide Giri received the master's degree in electronic engineering from Politecnico di Torino, Turin, Italy, and the master's degree in electrical and computer engineering from the University of Illinois at Chicago, Chicago, IL, USA, and the Ph.D. degree in computer science posthumously from Columbia University, New York, NY, USA.

He made fundamental contributions to agile system-on-chip design, the design and integration of hardware accelerators, and open-source hardware.

Thierry Tambe (Member, IEEE) received the B.S. and M.Eng. degrees in electrical engineering from Texas A&M University, College Station, TX, USA, in 2010 and 2012, respectively, and the Ph.D. degree in electrical engineering from Harvard University, Cambridge, MA, USA, in 2023.

He is now an Assistant Professor with the Department of Electrical Engineering, Stanford University, Stanford, CA, USA, where he leads a VLSI Research Lab at the intersection of artificial intelligence (AI) and hardware systems.

Dr. Tambe was a recipient of the Google ML and Systems Junior Faculty Award, the NVIDIA Graduate Ph.D. Fellowship, and the IEEE Solid-State Circuits Society Predoctoral Achievement Award.



Jeff Jun Zhang received the Ph.D. degree from New York University, New York, NY, USA.

From 2020 to 2022, he was a Post-Doctoral Fellow with Harvard University, Cambridge, MA, USA. He is an Assistant Professor with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA. His general research interests include deep learning, computer architecture, embedded systems, and EDA, with particular emphasis on energy-efficient and fault-tolerant design for artificial intelligence (AI)/ML systems and hardware accelerators.

Dr. Zhang received the 2023 IEEE Top Picks in Test and Reliability and the IEEE Micro 2022 Best Paper Award, etc.

Alper Buyuktosunoglu (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Rochester, Rochester, NY, USA.

He is currently a Principal Research Scientist with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY. He has over 200 patents and has authored or co-authored more than 140 publications in these areas. He has been involved in research and development work in support of IBM z Systems and IBM POWER Systems in the areas of computer architecture and robust power management.

Dr. Buyuktosunoglu received several IBM-internal awards and IEEE Micro Top Pick recognition. He is an IBM Master Inventor.

Kuan-Lin Chiu received the B.S. degree from National Taiwan University, Taipei, Taiwan, and the M.S. degree from University of California at Los Angeles, Los Angeles, CA, USA, and the Ph.D. degree in computer science from Columbia University, New York, NY, USA.

His research focuses on system-level design for heterogeneous system-on-chip platforms, emphasizing hardware accelerators and efficient data movement. He explores intelligent scheduling and hardware/software co-design to improve performance and energy efficiency in artificial intelligence (AI)-driven image processing and edge computing.

Giuseppe Di Guglielmo received the Ph.D. degree in computer science from Università di Verona, Verona, Italy.

He was a Research Scientist with Columbia University, New York, NY, USA, and a Post-Doctoral Fellow with the University of Tokyo, Bunkyō, Japan. He is a Senior Engineer at Fermilab, Batavia, IL, USA, and an Adjunct Associate Professor of electronics and communication engineering at Northwestern University, Evanston, IL, USA, specializing in system-level design and high-level synthesis with contributions to open-source system-on-chip (SoC) design and ML acceleration. He has authored over 100 publications. His current research focuses on artificial intelligence (AI)-enriched hardware for high-energy physics and quantum readout and control.

Paolo Mantovani received the Ph.D. degree in computer science from Columbia University, New York, NY, USA, and the M.S. degree in electronic engineering from Politecnico di Torino, Turin, Italy.

He is a Staff Software Engineer at Google DeepMind, New York. He specializes in architecture design and system-level methodologies for hardware accelerators and datacenter computing platforms. At Columbia, he served as the Lead Architect for the open-source project embedded scalable platforms, a platform enabling the rapid prototyping and integration of heterogeneous, accelerator-rich system-on-chips (SoCs).

Luca Piccolboni received the B.S. degree in computer science and the M.S. degree in computer science and engineering from the University of Verona, Verona, Italy, in 2013 and 2015, respectively, and the Ph.D. degree in computer science from Columbia University, New York, NY, USA, in 2022.

He currently works at Microsoft, Redmond, WA, USA.

Gabriele Tombesi (Graduate Student Member, IEEE) received the B.S. degree in physical engineering from the Politecnico di Torino, Turin, Italy, and the joint M.S. degree in electrical engineering from the Politecnico di Torino, and also with the École Polytechnique Fédérale de Lausanne and Grenoble INP, Lausanne, Switzerland. He is currently pursuing the Ph.D. degree with the System-Level Design Group, Columbia University, New York, NY, USA.

His Research interests include high-level synthesis (HLS)-based design flows for hardware acceleration of deep learning workloads, design-for-testability techniques for heterogeneous system-on-chip architectures, and tiled hardware accelerators for multitenant artificial intelligence (AI) workloads.

David Trilla (Member, IEEE) received the Ph.D. degree in computer architecture from the Universitat Politécnica de Catalunya, Barcelona, Spain.

During his Ph.D. degree, he was with Barcelona Supercomputing Center, Barcelona. His doctoral research focused on the impact of time-randomized processors on energy consumption, security, and hardware reliability. He is currently a researcher for IBM at the Thomas J. Watson Research Center, Yorktown Heights, NY, USA. His research interests include agile hardware design, fully homomorphic encryption (FHE), field-programmable gate array (FPGA) hardware emulation, and he actively collaborates with IBM's infrastructure team on future z processor architectures.

John-David Wellman (Member, IEEE) received the Ph.D. degree in computer science engineering from the University of Michigan, Ann Arbor, MI, USA.

After his Ph.D. degree, he joined IBM Thomas. J. Watson Research Center, Yorktown Heights, NY, USA, where he has worked primarily in computer and systems design, modeling, and analysis. Throughout his career, he has provided many contributions across several generations of IBM products and research designs, and systems. As a Senior Research Scientist, he continues to develop new system designs, most recently focusing on the agile design of domain-specific systems-on-a-chip, and application-driven accelerator hardware to drive both existing and emerging workload domains.

En-Yu Yang received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2018, and the M.S. degree in computer science from Harvard University, Cambridge, MA, USA, in 2020.

He is currently with Google in Taipei, Taiwan. His research focuses on specialized architecture and hardware design for machine learning applications.

Aporva Amarnath (Member, IEEE) received the B.E. degree (Hons.) in electrical and electronics engineering from the Birla Institute of Technology and Science, Pilani, Goa, India, in 2015, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2017 and 2021, respectively.

She is currently a Senior Member of Technical Staff at AMD Research and Advanced Development, Santa Clara, CA, USA. Her research interests include hardware–software co-design for heterogeneous systems, near-memory accelerators, and architectures for data-intensive and real-time applications.

Ying Jing (Graduate Student Member, IEEE) received the B.Eng. degree in electronic engineering from Hong Kong University of Science and Technology, Hong Kong, in 2019. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Illinois Urbana-Champaign, Champaign, IL, USA.

His research focuses on computer architecture, heterogeneous computing, cache coherence, and near-memory processing.

Bakshree Mishra is currently pursuing the Ph.D. degree with the University of Illinois Urbana-Champaign, Champaign, IL, USA, advised by Prof. Sarita Adve.

Her research interests include hardware–software co-design and efficient mapping and orchestration of workloads on heterogeneous hardware.

Joshua Park received the B.A. degree in computer science and mathematics and the M.A. degree in statistics from Harvard University, Cambridge, MA, USA, in 2025.

He is currently a Software Engineer with NVIDIA, Santa Clara, CA, USA. His prior research explored quantization, sparsity, and reliability in machine learning systems, and his broader interests include high-performance computing, machine learning systems, and domain-specific architecture design.

Vignesh Suresh received the B.Tech. degree in electronics and communication engineering from Vellore Institute of Technology, Vellore, India, in 2017. He is currently pursuing the Ph.D. degree with the University of Illinois Urbana-Champaign, Champaign, IL, USA, advised by Dr. Sarita Adve.

In the past, he has worked as a Design Engineer at Intel, India, Bengaluru, India, and more recently, as a graduate intern with the Qualcomm Cloud artificial intelligence (AI) Group, San Diego, CA, USA, in 2024. His research interests include accelerator design, specifically their interaction with the memory system and software.

Samira Zaliasl (Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Tehran, Tehran, Iran, in 2008, and the M.Eng. and Ph.D. degrees from Oregon State University, Corvallis, OR, USA, in 2012 and 2015, respectively.

She held research and engineering roles at IMEC, Leuven, Belgium, and SiTime, Santa Clara, CA, USA, pioneering solutions for biomedical devices and precision timing systems. She currently serves as Vice President of engineering at Ferric Inc., New York, NY, USA, where she drives innovation in high-performance power converters. She is an inventor on multiple patents and has authored numerous publications.

Michael Lekas received the B.S. and M.S. degrees in electrical engineering from Case Western Reserve University, Cleveland, OH, USA, and the Ph.D. degree in electrical engineering from Columbia University, New York, NY, USA.

He worked on process integration, packaging design, and computational modeling of power electronics, which included the development of compact models and design automation tools for integrated dc–dc converters at Ferric Inc., New York. He is currently with IBM Research, Yorktown Heights, NY, where he develops methodologies and tools for quantum electronics design. His research interests include power electronics, process and packaging integration, and design automation for analog and mixed-signal circuits.

Stephen Cahill received the B.S. degree in electrical engineering from Northeastern University, Boston, MA, USA, and the M.S. degree in electrical engineering from the University of Massachusetts at Lowell, Lowell, MA.

He worked for Vicor Corporation, Andover, MA, USA, as a Design Engineer, creating high-performance sine-amplitude resonant converters. He is a Staff Applications Engineer with Ferric Inc., New York, NY, USA. He assists customers with component integration and develops demonstration platforms for Ferric's chip-scale power converters.

Hesam Sadeghi received the B.S. degree from Isfahan University of Technology, Isfahan, Iran, and the M.S. degree from Sharif University of Technology, Tehran, Iran, in 2012 and 2014, respectively, and the Ph.D. degree in electrical engineering from Pennsylvania State University, University Park, PA, USA, in 2020.

He is currently a Staff Analog/Mixed-Signal IC Design Engineer at Ferric Inc., New York, NY, USA. His research interests include analog, mixed-signal, RF IC, power management, and low-power medical devices.

Joseph Meyer received the B.S. degree in electrical and computer engineering from the Olin College of Engineering, Needham, MA, USA, and the M.S. degree in electrical engineering from Columbia University, New York, NY, USA, with an IC design focus.

He is the Vice President of Product for Ferric Inc., New York, a company dedicated to the creation of chip-scale power converter modules with integrated inductors. In his role, he is responsible for Ferric's product specifications and product roadmap, as well as Ferric's Applications team. He is the primary technical point of contact for Ferric's customers. Joe has been at Ferric for over 11 years.

Noah Sturcken received the B.S. degree (summa cum laude) from Cornell University, Ithaca, NY, USA, and the Ph.D. and M.S. degrees in electrical engineering from Columbia University, New York, NY.

He previously worked at AMD R&D Lab, Fort Collins, CO, USA, where he developed Integrated Voltage Regulator (IVR) Technology. He is the Founder and CEO of Ferric Inc., New York, with over 45 patents issued and 20 publications on Integrated Voltage Regulators. He leads Ferric with a focus on business development, marketing and new technology development.

Sarita Adve (Fellow, IEEE) received the B.Tech. degree from Indian Institute of Technology at Bombay, Mumbai, India, and the Ph.D. degree from the University of Wisconsin-Madison, Madison, WI, USA.

She is the Richard T. Cheng Professor of Computer Science at the University of Illinois Urbana-Champaign, Champaign, IL, USA, where she directs IMMERSE, the Center for Immersive Computing. Her research interests are in computer architecture and immersive computing systems.

Dr. Adve is a member of the American Academy of Arts and Sciences, a fellow of the ACM and AAAS, and a recipient of the ACM/IEEE-CS Ken Kennedy award, Computing Research Association (CRA) distinguished service award, and IIT Bombay distinguished alumni award.

David Brooks (Fellow, IEEE) received the B.S. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, USA.

He is the Haley Family Professor of computer science with the Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. His research interests include resilient and power-efficient computer hardware and software design for high-performance and embedded systems.

Prof. Brooks is a fellow of the ACM.

Gu-Yeon Wei (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 1994, 1997, and 2001, respectively.

He is currently a Robert and Suzanne Case Professor of electrical engineering and computer science with the Paulson School of Engineering and Applied Sciences (SEAS), Harvard University, Cambridge, MA, USA. His research interests span multiple layers of a computing system: mixed-signal integrated circuits, computer architecture, and design tools for efficient hardware.

Kenneth L. Shepard (Fellow, IEEE) received the B.S.E. degree from Princeton University, Princeton, NJ, USA, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 1988 and 1992, respectively.

From 1992 to 1997, he was a Research Staff Member and Manager with the VLSI Design Department, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA. Since 1997, he has been with Columbia University, New York, NY, where he is now Lau Family Professor of electrical engineering, Professor of biomedical engineering, and Professor of neurological sciences (in neurosurgery). He is currently Chairperson of the Board and co-founder of Ferric Inc., New York. His current research interests are primarily directed to CMOS bioelectronics.

Pradip Bose (Life Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA.

He is a Distinguished Research Scientist and Manager of the Efficient and Resilient Systems Group, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA.

Luca P. Carloni (Fellow, IEEE) received the Laurea degree (summa cum laude) in electrical engineering from the Universita' di Bologna, Bologna, Italy, in 1995, and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California at Berkeley, Berkeley, CA, USA, in 1997 and 2004, respectively.

He is Professor and Chair of computer science at Columbia University, New York, NY, USA. He has authored over 200 publications. His research interests include design automation, heterogeneous computing, system-on-chip platforms, embedded systems, and open-source hardware.

Dr. Carloni is a senior member of the Association for Computing Machinery.