

A System-level Exploration of Superconducting Supercomputers for Emerging Workloads

Daniel Peralta Velazquez*, Siyuan Ma*, Sairam Sri Vatsavai, Saikat Chakraborty, Lingda Li, Anna Herr, Manu Perumkunnil, Quentin Herr, George Biros, Jaydeep Kulkarni, Lizy Kurian John, Andreas Gerstlauer

Abstract—Superconducting digital (SCD) technology has long carried promise for advancing large-scale computing leveraging the unique physical characteristics of superconducting materials and devices, such as low-voltage, high-frequency operation and zero-resistance interconnects. At the same time, SCD technology has also historically been beset with many scaling challenges. However, recent critical advances including improved power delivery, high integration density and availability of superconducting random-access memory prompt a fresh evaluation of SCD-based computing. In this paper, we study whether SCD systems can outperform conventional CMOS and cryo-CMOS for large-scale workloads even after accounting for full system overheads. Using a novel multilevel analytical modeling framework, our study explores performance implications of various SCD configurations across a range of architecture and technology assumptions in applications such as large language models (LLMs). The results show that, despite cooling overheads, a key benefit of recently proposed SCD technology is $9\times - 286\times$ better energy efficiency. By contrast, performance gains ranging between $4\times - 16\times$ are more limited and strongly depend on the availability of high-density, high-bandwidth on-chip memory that traditionally was not attainable. In general, the results show that recent technological advances are key to a promising scalable path forward, where our ablation studies highlight critical aspects of the maturity of SCD technology.

Index Terms—High-Performance Computing, Performance Modeling, Energy Modeling, Superconducting Digital Technology, Cryo-CMOS Technology

I. INTRODUCTION

THE steady advancement of computational demands especially in scientific computing and artificial intelligence (AI), e.g. using large language models (LLMs) has led to an unprecedented demand for computing resources, where traditional CMOS technology is approaching physical and economic limitations. Superconducting digital (SCD) technology [1] has fundamental advantages based on the superconducting physics of materials, including wires with negligible dissipation/diffusion, 100 GHz frequencies, and active devices operating on a pico-second time scale and sub-attoJoule energy levels [2]. However, despite its promise, SCD technology has historically suffered from many challenges; low scalability, high static power consumption, cooling requirements, low circuit density, and low-density shift register-based memories [3]

at the single-processor scale. As a result, previously proposed SCD-based system designs were limited to simpler general-purpose processors [4], [5], [6] or specialized hardware accelerators [7], [8], [9], [10].

While superconducting technology has existed for decades, recent critical technological advances in scalable power delivery networks and superconducting memory at high density [11] make it a particularly interesting time to reevaluate SCD-based computing. The basic component in superconducting compute circuits is a 2-terminal superconducting tunnel junction known as Josephson Junction (JJ). Traditionally, the scalability of superconducting logic was limited by the large transformers required for the inductive power delivery network. Recently developed pulse conserving logic (PCL) [11] operates with capacitive power delivery using dual-rail signaling, where, similar to CMOS, energy is only consumed during signal transitions. Similarly, Josephson-SRAM (JSRAM) [11] as an active array-based memory that uses the same signal levels and power distribution as logic circuits can achieve a 600-fold increase in memory density compared to legacy SCD memories. The availability of on-chip JSRAM in particular provides a critical building block for large-scale superconducting architectures that avoid data movement across cooling domains. Together, these developments prompt a fresh look into the feasibility of SCD-based large-scale computing.

In this paper, we investigate whether large-scale SCD systems are a feasible option for future energy-efficient datacenters. We aim to advance the understanding of the SCD technology landscape by trying to answer the following questions: ❶ How much system-wide benefit can the latest SCD technology deliver in practice after cooling costs and other system overheads are taken into account, ❷ how sensitive are these benefits to SCD technology assumptions, and ❸ how do SCD technology parameters impact architectural and application-level design decisions? To answer these questions, we develop a novel hierarchical modeling framework for datacenter-scale computer systems. Using this framework, we conduct extensive design-space exploration (DSE) across technology assumptions, architecture configurations and execution strategies. We compare SCD-based systems against state-of-the-art GPU clusters, as well as other CMOS and cryo-CMOS systems. Finally, we conduct an ablation study to understand the impact of the parameters of the SCD technology while providing insights for the development of future architectures and technologies. To the best of our knowledge, our work is the first to explore the system-level performance and energy impact of large-scale superconducting systems, in particular,

* Authors contributed equally to this work.

D. Peralta Velazquez, S. Ma, S. Chakraborty, G. Biros, J. Kulkarni, L. John, A. Gerstlauer are with the Chandra Family Department of Electrical and Computer Engineering at The University of Texas at Austin, Austin, TX 78712 USA. S. S. Vatsavai and L. Li are with Brookhaven National Laboratory, Upton, NY 11973 USA. A. Herr and Q. Herr are with Snowcap Compute, and were affiliated with imec, USA at the time of this work. M. Perumkunnil is with imec, Belgium.

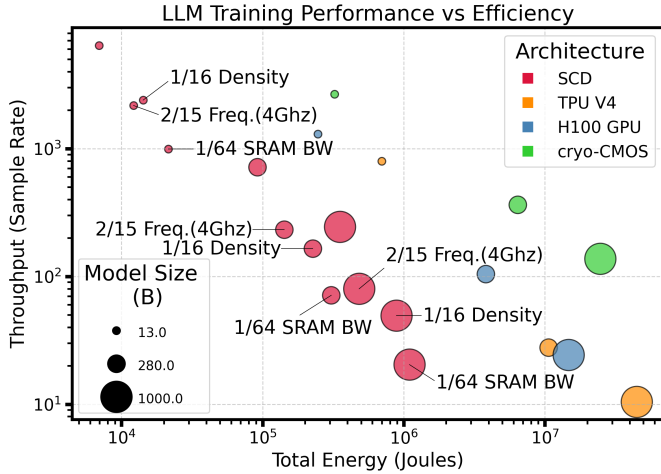


Fig. 1. LLM training performance and efficiency across multiple architectures. Labels represent ablation variations of the baseline SCD architecture as explained in Section VI.

using PCL and JSRAM technologies, with detailed modeling across full architectures and emerging workloads. Prior work has only considered node-level projections, relying largely on optimistic assumptions and stopping short of full system architectural analysis. Our work addresses this gap through the following contributions:

- We introduce a novel hierarchical framework to model performance and energy of large-scale system architectures across various technologies running emerging workloads. Our proposed framework accounts for system-level overheads while capturing the interaction between technology parameters, architectural/system configurations, and application mappings.
- We explore the architectural and system design of superconducting systems, and provide a comprehensive assessment of system performance and energy efficiency for emerging workloads such as LLMs.
- We conduct a comparative study of SCD against state-of-the-art CMOS and cryo-CMOS systems and perform in-depth analyses and ablation studies to quantify the impact of critical SCD parameters, including frequency scaling, logic/memory density, 3D stacking, and shift buffer memory basis on overall system performance and energy efficiency.

The remainder of this paper is organized as follows: Sections II and III provide a brief motivation and background on superconducting technology and systems. Sections IV and V describe our hierarchical modeling framework in detail along with its respective validation. Section VI presents our performance and energy studies of SCD systems. Lastly, Sections VII and VIII provide a discussion and conclusion on our findings.

II. MOTIVATION

Figure 1 summarizes the energy and performance of various systems modeled in this paper. Our main findings are that, despite cooling overhead, best-case SCD technology promises not only significant speedup, but crucially and somewhat surprisingly also major energy efficiency improvements compared

to H100, TPUv4, and cryo-CMOS systems scaled to the same total area. Compared to an H100 system, the SCD system achieves a $6.6\times$ and $3.9\times$ overall performance improvement, and $40\times$ and $29\times$ better energy efficiency in LLM training and inference, respectively. Those results show the potential to build future supercomputing systems with SCD technology.

Our ablation studies show that the density, frequency, and on-chip SRAM enabled by the state-of-the-art SCD technology are essential to the system’s performance. Performance and energy of the SCD system is sensitive to DRAM accesses, therefore, favor designs with high in-SRAM data locality. As such, high bandwidth superconducting SRAM is crucial to achieving performance improvements, while other factors like frequency and density are less significant, where an order of magnitude improved energy-delay product (EDP) is still achieved with reduced frequency or density. As will be detailed in the rest of the paper and in Section VI, different assumptions, including cooling technology, thus have varying impacts on overall system performance and energy.

III. BACKGROUND

In the following, we discuss superconducting technologies and prior superconducting architectures and systems.

A. Logic and Memory Technology

Operating CMOS at cryogenic temperatures has been proposed as a way to overcome energy efficiency barriers due to the end of Dennard scaling. A 5nm FinFET technology operating at 77K can achieve $\sim 1 \times 10^{-5}\times$ lower OFF current leading to ultra-low static power consumption and $\sim 50\%$ speed increase at constant operating power due to improvements in mobility compared to 300K [13]. Alternatively, supply voltage scaling ($\sim 2\times$) reduces the dynamic power ($\sim CfV_{DD}^2$) to $0.27\times$ at 77K while operating at same speed as 300K [13]. However, these metrics don’t improve further at lower temperatures, and the advantages are outweighed by exponential increase in cooling cost. Wall-plug power increases to $\sim 325\text{W/W}$ at 4K compared to $\sim 22\text{W/W}$ at 77K for >1000 PFLOPS performance [11].

By contrast, superconducting technology offers radical advantages. Superconducting circuits, composed of materials exhibiting zero electrical resistance at ultra-low temperatures (4K), can achieve ultrafast speeds ($\sim 30\text{GHz}$) and extremely low energy dissipation ($\sim 0.01\times$ compared to 7nm CMOS platform including cooling overhead) [11]. However, traditionally, scalability has been a big challenge for Josephson Junction (JJ)-based logic and memory circuits that various works over the years have aimed to overcome. Figure 2 shows a summary and overview of the evolution of superconducting technology. **Superconducting Logic:** Table III summarizes the comparison across different CMOS-based and SCD technologies. CMOS-based technologies have high scalability but suffer from energy efficiency and frequency. Rapid/Resistive single flux quantum (RSFQ) [23], [24], [25] logic achieves higher frequency, but energy efficiency is still limited by resistive (DC) power dissipation. Energy-efficient RSFQ (ERFSQ) variants [15], [16], [26] introduce JJs and inductors in place

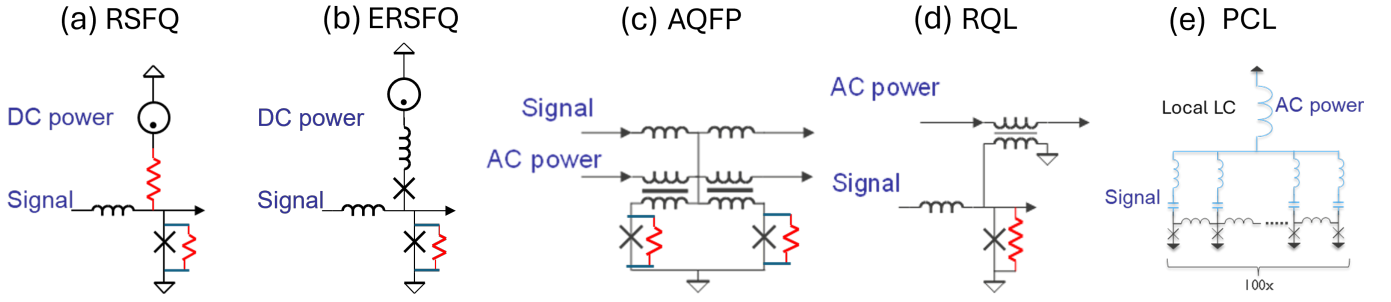


Fig. 2. Evolution of superconducting logic technology in the past 3 decades: (a) RSFQ used resistors to deliver bias current, (b) ERSFQ (Energy efficient RSFQ) replaced the bias resistors by inductors and JJs, (c) AQFP and (d) RQL logic used AC power to bias the JJs via inductively coupled transformers, and (e) PCL uses dual-rail signals and preservation of pulses while replacing the large transformers with scalable MIM capacitors for AC power delivery.

TABLE I
COMPARISON OF CMOS, CRYO-CMOS AND SUPERCONDUCTING LOGIC (INCLUDING COOLING COSTS FOR ENERGY PER MAC OPERATION).

	CMOS [12]	Cryo-CMOS [13]	RSFQ [14], [7]	ERSFQ [15], [16]	QFP [17]	RQL [18], [19]	PML [20]	PCL [11]
Temperature	300K	77K	4K	4K	4K	4K	4K	4K
Cooling cost	1 W/W	22 W/W	325 W/W	325 W/W	325 W/W	325 W/W	325 W/W	325 W/W
Tech. node	7nm	7nm	1 μ m	1 μ m	$\sim$ um	$\sim$ um	$\sim$ um	28nm
Power supply	DC	DC	DC	DC	AC	AC	AC	AC
MAC energy	0.4 pJ	2.3-8.8 pJ	0.6 pJ	1.2 pJ	High	High	High	14 fJ
Frequency	4GHz	4-5.6GHz	50 GHz	50 GHz	$\sim$ 1 GHz	$\leq$ 10GHz	$\sim$	30 GHz

TABLE II
COMPARISON OF CMOS, CRYO-CMOS (ISO- I_{ON} AND ISO- I_{OFF} V_{TH} OPTIMIZATION) AND SUPERCONDUCTING JSRAM.

	300K SRAM [21]	77K Cryo-SRAM [22]	12K JSRAM [11]
Tech Node	5nm FinFET	5nm FinFET	Si barrier JJ, NbTiN
Power Supply	0.75V	0.4V	AC bias 4 mA
Memory Density	4 MB/mm ²	4 MB/mm ²	0.04 MB/mm ²
Cycle time	2 GHz	2 GHz	30 GHz
Access time	200 ps/kB	200 ps/kB	4 ps/cell (zero R interconnect)
Avg. Read/Write Power	1 μ W/bit	0.25 μ W/bit * 22 (cooling)	$\sim$ 1.94 nW/bit * 325 (cooling)
Static Power	10 nW/cell	10 nW/cell * 22 (cooling)	$\sim$ 0 (AC bias)
Avg. Read/Write energy	0.5 fJ/bit	0.125 fJ/bit * 22 (cooling)	$\sim$ 0.065 aJ/bit * 325 (cooling)

of the bias resistors thus eliminating static power dissipation at the expense of higher dynamic energy consumption and lower density compared to RSFQ. Adiabatic quantum flux parametron (QFP) [17], [27], [28], [29], reciprocal quantum logic (RQL) [18], [19], [30], and phase mode logic (PML) [20] achieve high energy efficiency, but are limited in scalability and frequency.

Recently developed pulse conserving logic (PCL) [11] has shown potential to rival conventional CMOS technology in both integration density and performance, especially when fully leveraging advanced fabrication processes. PCL operates with dual-rail signaling, similar to CMOS, where energy is only consumed during signal transitions, and there is no added penalty in clock rate or latency for logical inversion [11]. The pulse conservation aspect of PCL enables the implementation of more complex logical operations with reduced latency [11]. In addition, integration of up to 400 million AC-biased Josephson junctions (JJs) per cm² [31] has been demonstrated, supported by higher routing density, JJ density and bias-tap capacitor density due to advancements in materials.

Superconducting Memory: The ability to architect large-scale superconducting computer architectures strongly depends on on-chip superconducting memories, as otherwise data has

to move across cooling domains. Early (R)SFQ-based logic variants relied heavily upon shift-register based sequential-access memory, which affects memory bandwidth and consumes more dynamic energy, due to the absence of a dense superconducting random access memory. This has changed with the advent of JJ-based SRAM (JSRAM).

In SFQ-based memory, state retention is naturally achieved in superconducting loops, but implementing X-Y addressing has proven challenging due to the high fanout of signals required. As a result, the memory density of SFQ systems has been relatively low [32], [33], [34].

By contrast, recently reported JSRAM [11], being an active array-based memory [35], [36] that uses same signal levels and power distribution as logic circuits, can achieve 600-fold increase in memory density than the legacy SFQ-based random access memories [3], [33], [34]. The enhanced density is made possible through the efficient utilization of fabrication resources across the entire stack, including bias taps, junctions, and inductive wiring.

Table III compares 5nm FinFET based SRAM [21], 77K cryo-SRAM [22] and JJ-array based JSRAM [11] as cache memory for superconducting computing. Due to $\sim 2 \times V_{DD}$ scaling possible with V_{TH} optimization at iso- I_{ON} and iso- I_{OFF}

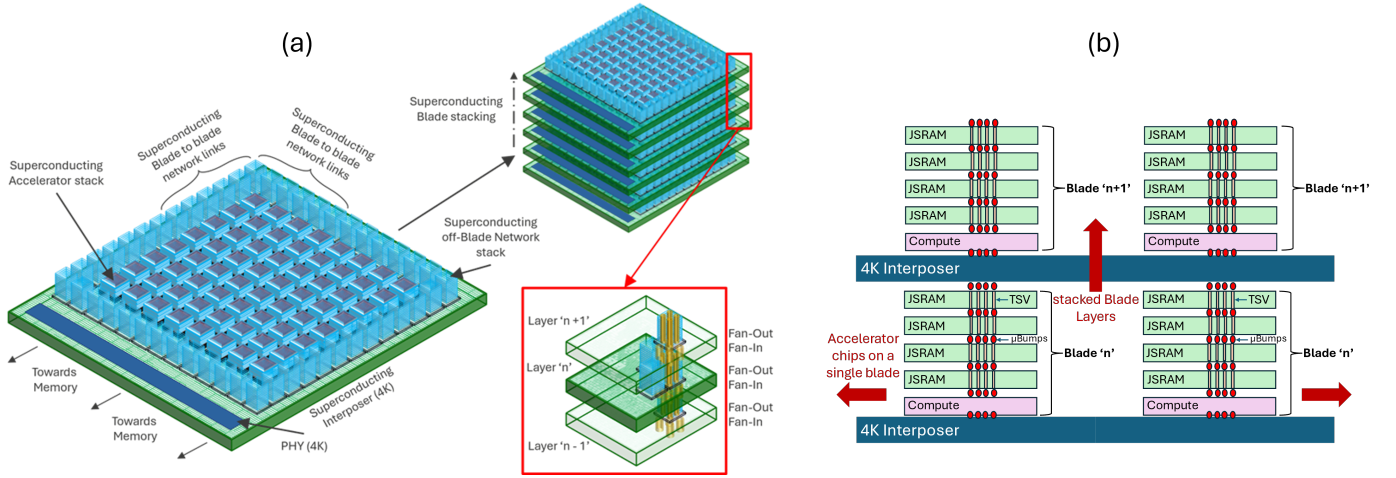


Fig. 3. A multidimensional and hierarchical superconducting interconnect network between different superconducting accelerator stacks achieves high volumetric compute density in an efficient manner. Each accelerator stack (3D die to wafer/die to die) is connected to others via 2.5D integration on a silicon interposer (at 4K) and the network stack at the edges of the blade allow for blade to blade (3D package on package) integration: (a) Overall architecture, and (b) cross-section cartoon diagram of superconducting accelerator stack and blade stacking.

condition, cryo-SRAM gains $4\times$ in terms of read/write power consumption compared to 300K operation. On the other hand, JSRAM benefits from $50\times$ lower access time and $20\times$ lower read/write power (including cooling overhead) compared to cryo-SRAM, but suffers $100\times$ lower memory density.

Cooling Technology: Modern commercial cryo-coolers using Helium liquefaction from companies such as Linde can have overheads as low as 315 W/W with capacities of 100W to 2.5kW at 4.5K in a single refrigerator [37]. However, cooling efficiency diminishes with refrigerator size favoring large systems. We use cooling factors of 325W/W and 22W/W at 4K and 77K, respectively, as baselines for our analysis.

B. Architectures and Systems

Superconducting Architectures: Superconducting computer architectures have similarly evolved jointly with technology to take advantage of benefits or account for limitations. Early SCD microprocessor with SFQ technology [4] were demonstrated in 2007, followed by support of stored program [5] and adoption of fine-grain, gate-level pipelining [6]. To work around logic limitations, an approximate stochastic computing-based accelerator [10] in AQFP technology was proposed and achieved very high energy efficiency for deep learning. Similarly, temporal logic architectures [8] were proposed to improve logic density of SFQ, followed by designs of accelerator blocks, including multiply-add, dot product units and Finite Impulse Response filters [9]. All of the above designs either lack on-chip memory or use shift-register-based memory. SuperNPU [7] demonstrated the first systolic-array accelerator that architecturally tackles the performance shortcomings of the shifting memory. However, shifting memory still introduces excessive energy cost as all the entries in the shifting buffer need to be shifted even only one entry is accessed. Furthermore, the scales of all the above designs are too small, which lowers the cooling efficiency [37] and their competitiveness against CMOS systems. **Superconducting**

Systems: While superconducting architectures have been studied broadly in the past, how to construct complete computing systems out of them has received little attention. A “data center in a shoe box” system was recently proposed [38] using SCD chips in PCL technology that are 3D stacked on blades and placed into a shoe box sized cryogenic refrigerator. It is the first proposal to scale SCD devices to the datacenter scale, enabled by 3D-stacking of chips with dense PCL logic and JSRAM, multidimensional and hierarchical superconducting interconnect, high-bandwidth cryo HBM DRAM and efficient system-level cooling (Figure 3). The “shoe box” datacenter features a hierarchical design. At the lowest level, it is composed of accelerator stacks, and each stack includes multiple compute logic and SRAM dies interconnected vertically. Similar to SuperNPU, a compute die contains a systolic array to accelerate general matrix-matrix multiplication (GEMM), while the main component of an SRAM die is high-density JSRAM to provide data for the systolic array. Figure 3(b) illustrates four accelerator stacks. Multiple accelerator stacks are connected in a 2D mesh to form a blade, and multiple blades are integrated vertically to compose an SCD system (top right of Figure 3(a)). A large volume of HBM memory is attached to each blade and shared by all accelerators on the blade. Note that while accelerator stacks reside in the 4K domain, HBM is placed at 77K due to its limited benefits and high cooling cost in 4K. As a result, HBM accesses cross the interface between 4K and 77K at the blade and not the individual accelerator level, different from current GPU and TPU systems. The compact 3D design of the “shoe box” datacenter is both constrained and made possible by current cryo coolers [37]. This architecture forms the template for studying SCD and other systems in this paper.

IV. HIERARCHICAL SYSTEM MODELING FRAMEWORK

In this section, we present a detailed overview of our multi-level system modeling framework as illustrated in Figure 4. Our approach is based on a generic four-level hierarchical

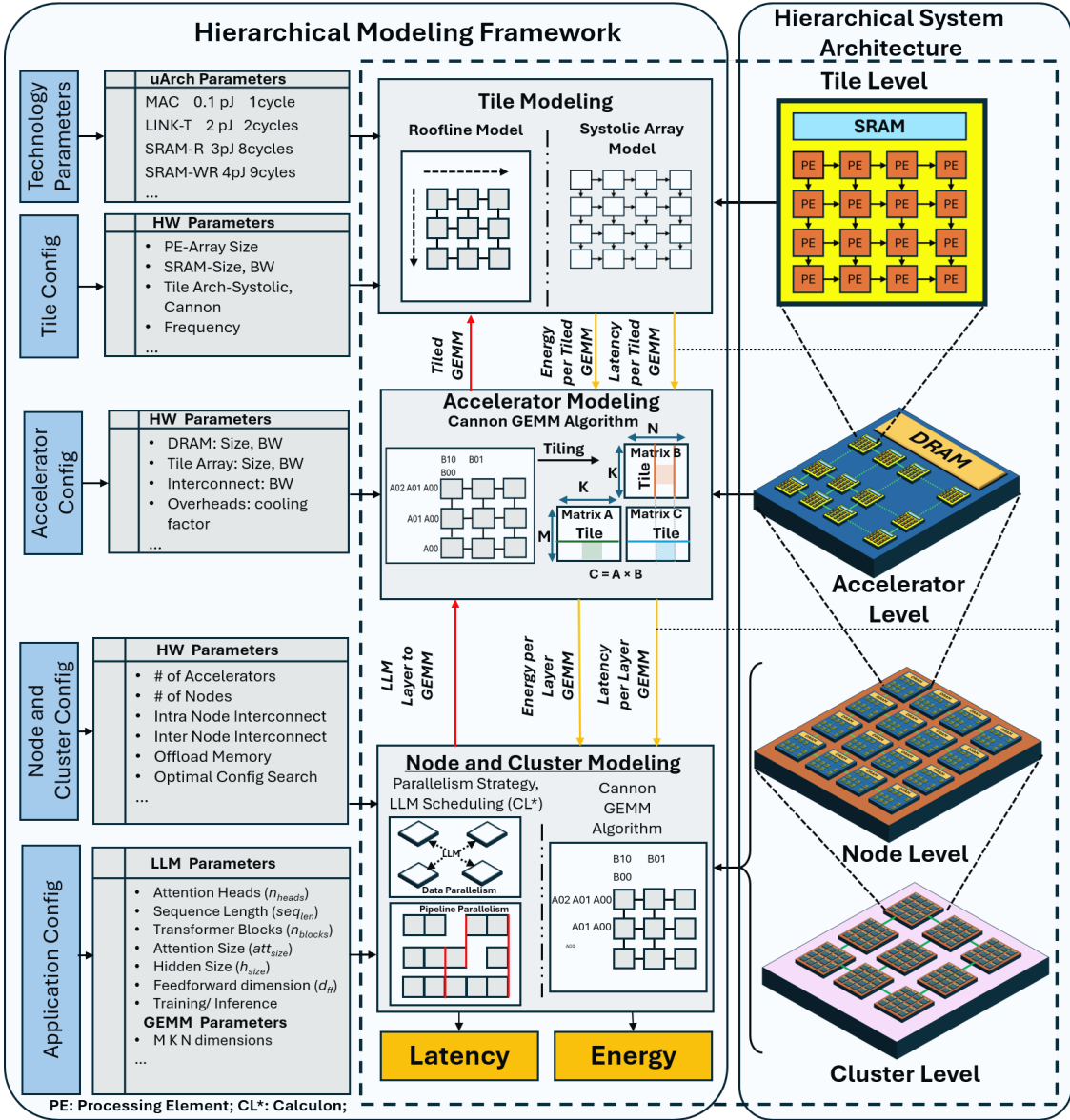


Fig. 4. Hierarchical system modeling framework for large-scale computing systems—including CMOS, cryo-CMOS, superconducting, and other emerging technologies. Organized from tile to cluster level, the framework captures application, technology, and hardware parameters for comprehensive full-system performance and energy analysis.

system architecture template. The lowest tile level corresponds to a chip containing an array of Processing Elements (PEs) and on-chip SRAM. Above tiles is the accelerator level, which can be a PCB, blade, or system-in-package (SiP) that includes multiple chips and DRAM memory. Multiple accelerators are then interconnected to form a node, and multiple nodes are interconnected to form a cluster. Additional memory hierarchies can be integrated at the node and cluster levels to model scenarios involving memory offloading.

Our hierarchical framework supports the modeling of heterogeneous compute, memory, and interconnect across CMOS, cryo-CMOS, SCD and other emerging technologies. For example, to represent the SCD system described in Section III-B, each tile could be an SCD chip containing PE and JSRAM. Each accelerator might include a cryo-HBM and an array

of tiles connected by superconducting switches and links. A node could represent a cryogenic refrigerator that houses these accelerators, while a cluster could comprise multiple cryogenic refrigerators interconnected via room-temperature links, including traditional or optical interconnects.

Our framework builds on Calculon [39]’s system model for LLMs, which captures the most important compute-memory tradeoffs at the node and cluster level. We replace Calculon’s GPU model with analytical GEMM models for systolic or flat architectures at the accelerator and tile levels, which in turn rely on table-based plugins capturing key technology parameters at the PE and SRAM level.

The rest of this section describes details of our framework for workloads targeted in this paper, i.e. GEMMs and LLMs.

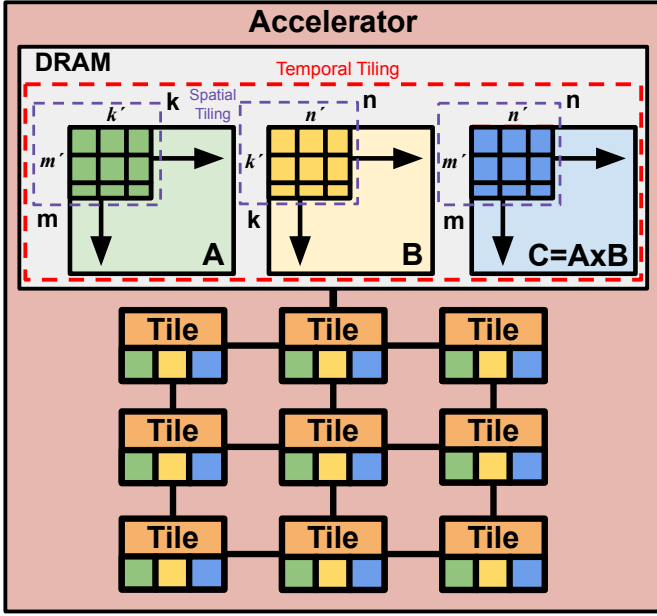


Fig. 5. Accelerator-level GEMM hierarchical tiling and mapping strategy.

A. Tile/Accelerator-Level Modeling

We use Cannon’s GEMM algorithm [40] to recursively map GEMMs of different sizes across tile and accelerator hierarchies. Cannon-GEMM is efficient in both memory accesses and interconnect communications, allowing the full use of the performance benefits across different technologies. Figure 5 provides an overview of our hierarchical tiling and mapping strategy at the accelerator level. In particular, we introduce two types of tiling: *temporal* and *spatial* tiling. Temporal tiling breaks a larger GEMM into smaller operations that fit within the aggregated memory at the next lower level to be processed sequentially. Spatial tiling then further distributes the computation across components at the next lower level, exploiting spatial parallelism.

Our heuristic for temporal tiling is to maximize local memory utilization and thus minimize the number of temporal loop iterations at each hierarchy level. As shown in Figure 5, temporal tiling partitions matrices A , B , and C along the dimensions m , k , and n into tiles of size m' , n' , and k' so that they completely fill the aggregated buffers of the next level, in this case the tile-level SRAMs. For spatial tiling, our heuristic is to fully leverage available compute resources to maximize spatial parallelism. In Figure 5, the temporally tiled dimensions m' , n' , and k' are further partitioned into m'' , n'' , and k'' so that all accelerator tiles are utilized for maximum spatial parallelism. The following formulas present our model for estimating GEMM latency for a problem of size (m, k, n) on hardware hierarchy i .

Temporal Tiling: At each hierarchy, a temporal tiling is performed to fit tiled matrices into aggregated buffers of the next hierarchy. To maintain maximum locality, matrices (m, k, n) are tiled to m', k', n' such that the number of temporal iterations (C_t) is minimized while fitting within buffer size (S_{buff}). We use an exhaustive search to find the

optimized tile sizes:

$$m', k', n' = \arg \min_{m', k', n'} \left\lceil \frac{m}{m'} \right\rceil \times \left\lceil \frac{k}{k'} \right\rceil \times \left\lceil \frac{n}{n'} \right\rceil \quad (1)$$

$$\text{s.t. } m'k' + k'n' + m'n' \leq S_{\text{buff}}$$

$$C_t = \left\lceil \frac{m}{m'} \right\rceil \times \left\lceil \frac{k}{k'} \right\rceil \times \left\lceil \frac{n}{n'} \right\rceil \quad (2)$$

Spatial Tiling: Matrices of m', k', n' are further tiled to m'', k'', n'' according to the spatial mesh dimension. We explore the maximum spatial parallelism by utilizing all tiles in the mesh interconnect with mesh dimensions $P \times P$:

$$m'', k'', n'' = \left\lceil \frac{m'}{P} \right\rceil, \left\lceil \frac{k'}{P} \right\rceil, \left\lceil \frac{n'}{P} \right\rceil \quad (3)$$

Latency Modeling: The Cannon-GEMM model estimates 4 categories of latencies: T_{prep} , T_{comm} , T_{comp} and T_{store} .

T_{prep} is the latency to load GEMM data from memories at level i into the next hierarchy $i+1$ to prepare for execution. We model this latency by taking the roofline of two latencies: (1) the latency for each component at hierarchy $i+1$ to load tiled matrices of size $m''k''$ and $k''n''$ into its local memory, dictated by the bandwidth of the interconnect network (BW_{nw_i}) and the bandwidth of local memory ($BW_{mem_{i+1}}$), and (2) the latency to read the matrices $m'k'$ and $k'n'$ from memory at hierarchy i , dictated by its memory bandwidth (BW_{mem_i}):

$$T_{\text{prep}} = \max \left(\frac{m''k'' + k''n''}{\min(BW_{nw_i}, BW_{mem_{i+1}})}, \frac{m'k' + k'n'}{BW_{mem_i}} \right) \quad (4)$$

T_{comm} is the latency for all tiles in the hierarchy $i+1$ to exchange data in a shifting manner. This is bounded by the interconnect or memory bandwidth. We assume concurrent horizontal and vertical shifting so the maximum of 4 terms is taken:

$$T_{\text{comm}} = \frac{\max(m''k'', k''n'')}{\min(BW_{nw_i}, BW_{mem_{i+1}})} \quad (5)$$

T_{comp} is time for each tile in the next hierarchy to finish executing of its tiled GEMM (m'', k'', n'') . At accelerator or higher levels, we recursively evaluate $T_{\text{comp}_{i+1}} = L_{i+2}$ as the latency of a Cannon-GEMM model at the next lower modeling level following Eq. 9. In the case of our lowest (tile) level, we adopt analytical models that either model a systolic array architecture similar to Scale-Sim [41] or a flat architecture similar to Timeloop [42]. In case of a systolic array, each tiled GEMM of dimensions (m'', k'', n'') executes on a $d \times d$ systolic array using an output stationary dataflow. We consider three cases depending on the relationship between the tile dimensions and the systolic array size, where for all cases but $m'' = d$ and $n'' = d$, the systolic array is underutilized due to partial mapping along one or more dimensions:

$$T_{comp} = \begin{cases} T_{tile}, & \text{if } m'' \leq d \text{ and } n'' \leq d \\ \lceil \frac{m''}{d} \rceil T_{tile}(d, k'', n'') \\ \quad + T_{tile}(m'' \bmod d, k'', n''), & \text{if } m'' > d \text{ and } n'' \leq d \\ \lceil \frac{n''}{d} \rceil T_{tile}(m'', k'', d) \\ \quad + T_{tile}(m'', k'', n'' \bmod d), & \text{if } n'' > d \text{ and } m'' \leq d \\ \lceil \frac{m''}{d} \rceil \lceil \frac{n''}{d} \rceil T_{tile}(d, k'', d) \\ \quad + \lceil \frac{m''}{d} \rceil T_{tile}(d, k'', n'' \bmod d) \\ \quad + \lceil \frac{n''}{d} \rceil T_{tile}(m'' \bmod d, k'', d) \\ \quad + T_{tile}(m'' \bmod d, k'', n'' \bmod d), & \text{if } m'' > d \text{ and } n'' > d, \end{cases} \quad (6)$$

where T_{tile} is the latency for a fully utilized number of compute resources in the array:

$$T_{tile} = \max \left((2m'' + n'' + k'' - 2) T_{cycle}, \frac{dk'' + k''d + d^2}{BW_{sram}} \right)$$

Alternatively, in addition to systolic array modeling, our framework supports a roofline model representing a flat architecture where the execution time is bounded by either the available compute throughput or the memory bandwidth:

$$T_{comp} = \max \left(\frac{m''n''k''}{d^2} T_{cycle}, \frac{D}{BW_{sram}} \right), \quad (7)$$

where D is the total amount of data to be transferred to perform the GEMM. Note that a systolic tile model is used for all of our experiments.

Finally, T_{store} is the latency to retrieve output matrices from memories of the hierarchy $i + 1$ and store them into memory of hierarchy i . It is the maximum of hierarchy i memory store latency, hierarchy $i + 1$ memory read latency, and interconnect communication latency:

$$T_{store} = \max \left(\frac{mn}{BW_{mem_i}}, \frac{m''n''}{BW_{mem_{i+1}}}, \frac{m''n''}{BW_{nwi}} \right) \quad (8)$$

The overall latency L to perform a GEMM (m, k, n) on hierarchy i is therefore given by a combination of the 4 latencies above. We assume perfect overlapping of memory access, compute, and communication. Therefore, the overall latency is the maximum of 3 terms times the number of temporal iterations. In each temporal iteration, compute is performed P times, where P is mesh dimension, and communication is performed $P - 1$ times, according to Cannon's GEMM algorithm:

$$L = C_t \times \max(T_{prep} + T_{store}, P \times T_{comp}, (P - 1) \times T_{comm}) \quad (9)$$

Energy Modeling: Our energy model sums up the energy consumption of each hardware component during workload execution. We collect counters like the number of MAC operations, bytes of memory accesses and interconnect communication at each hierarchy level to derive the total dynamic energy. The total static energy is calculated by the total latency times system-wide static power. All energy and power estimates include cooling costs as shown in Tables III and III.

B. Node/Cluster-level Modeling

At cluster and node levels, our framework supports modeling of large GEMMs or LLMs. For GEMMs, we extend the Cannon model described in Section IV-A hierarchically up to nodes and clusters. LLM workloads are characterized by parameters such as the number of attention heads (n_{heads}), the number of transformer blocks (layers) (n_{blocks}), sequence length (seq_{len}), hidden size (h_{size}), attention size (att_{size}), and feed-forward dimension (d_{ff}). Often, these workloads exceed the memory and computational limits of a single accelerator, necessitating their distribution across multiple accelerators and nodes. Using these parameters, each transformer layer is decomposed into its constituent GEMM operations, including feed-forward projections and attention computations (QK^T , softmax, score.V), which are then mapped through the hierarchical tiling and latency model described in Section IV-A. Different mapping strategies are explored to efficiently distribute both inference and training workloads across individual nodes and accelerators. We utilize Calculon's validated analytical model, which captures key performance aspects to evaluate a wide range of parallelism strategies, including data parallelism (DP), tensor parallelism (TP), and pipeline parallelism (PP), as well as various optimization techniques. Calculon also accounts for factors such as inter- and intra-node communication overhead, parallelism overhead, memory offloading, and activation recomputation. For each system and workload combination, we exhaustively search the space of parallelism configurations combining TP, PP, and DP along with optimizations such as ZeRO-style optimizer state partitioning to identify the optimal strategy.

While Calculon's analytical model is well suited for GPU systems, we have extended its capabilities in three key areas to effectively model a wider range of systems including SCD-based architectures. First, we incorporate SCD-specific overheads, such as cooling efficiency, cryo-interface latencies, and other factors inherent to SCD. Second, we increase the depth of the modeling to the accelerator and tile levels by integrating our custom GEMM models (described in the previous section), allowing us to analyze how technology parameters at the tile level affect the overall performance of the system. Finally, we implemented energy consumption calculations within our extended framework to evaluate a comprehensive breakdown of energy components, including static power consumption, compute operations, communication overhead, memory access, and inter-domain interface cost. Building on tile- and accelerator level models, our energy calculations are implemented by summing individual energies across all relevant components to provide the total energy consumption for a given workload and system configuration. This multilevel approach allows us to capture both high-level LLM mapping strategies and low-level performance and energy characteristics with high fidelity.

Table III summarizes the modeling capabilities of the framework used in this paper and described in this section. A combination of all these features is used in conjunction to model the cluster/node level and accelerator/tile level. Calculon's model captures parallelization strategies, workload distribution and

TABLE III
ATTRIBUTES AND LIMITATIONS OF MODELING METHODOLOGY

Characteristic	Calculon [39]	OURS
Analytical model	✓	✓
Energy prediction	○	✓
Node level (server/host partitioning)	✓	✓
Accelerator level (chip-level throughput/latency)	✓	✓
Tile level (systolic tile, PE array)	✓	✓
Layer-to-GEMM translation (operator mapping)	✓	✓
GEMM partitioning strategy (Cannon/2D/3D)	✓	✓
Systolic/dataflow modeling (timing, reuse)	✓	✓
Latency modeling (end-to-end op latency)	✓	✓
Sample Throughput	✓	✓
Memory hierarchy modeling (SRAM/DRAM/tiling)	○	✓
On-chip interconnect / NoC modeling	✓	✓
Network / inter-node communications	✓	✓
Value Precision effects	✓	✓
Multi-node / multi-accelerator scaling	✓	○
Workload characterization for LLMs (tokens, layers)	✓	○
Scheduler	○	○
Validation against hardware/silicon measurements	○	✓
Sensitivity analysis (param sweeps, ablation)	✓	✓
Area / physical-resource estimation	✓	✓
Open-source	✓	✓

Legend: ✓ = Supported ○ = Added / Partial Blank = Not supported

inter-accelerator communication while abstracting details of the accelerator architecture. We adapt our modeling to expand beyond these abstractions by incorporating accelerator-level compute, memory hierarchy partitioning, and dataflow/tile-level execution allowing system-level scheduling to be influenced by accelerator hardware performance and energy.

V. MODEL VALIDATION

A. Tile-level Validation

There are currently no SCD processors or large-scale systems based on them available that would allow for a direct validation of modeling results. However, our analytical modeling aligns with established and separately validated AI/ML system exploration methodologies. At the cluster and node levels, we base our models on Calculon, which has been validated against real-world large-scale GPU systems [39]. In our framework, we replace Calculon’s GPU models at lower accelerator and tile levels with GEMM models that we hierarchically validated by comparing against established open-source modeling tools [42], [41], [43].

At the tile level, our framework can leverage a roofline throughput versus data transfer model or a systolic streaming dataflow model. For validation, we set up micro-benchmarks with GEMMs of different squared and non-squared dimensions, yielding a diverse number of MAC operations and memory accesses. The tile architecture consists of a 200 by 200 PE array with 20 MB of memory. We compare results from our roofline and systolic models against closely matching design configurations for Timeloop [42] and SCALE-Sim [41]. Timeloop maps matrix multiplication onto a flat PE array with spatial and temporal tiling and models data movement through on-chip global buffers and direct PE links. As such, we use Timeloop as a comparison tool to validate our roofline model. Our systolic model is an adaptation of SCALE-sim’s model, making it a direct comparison point for fine-grained timing effects and stalls that a systolic tile model predicts.

Figure 6 presents the latency and energy validation results of our systolic and roofline tile models, compared to the station-

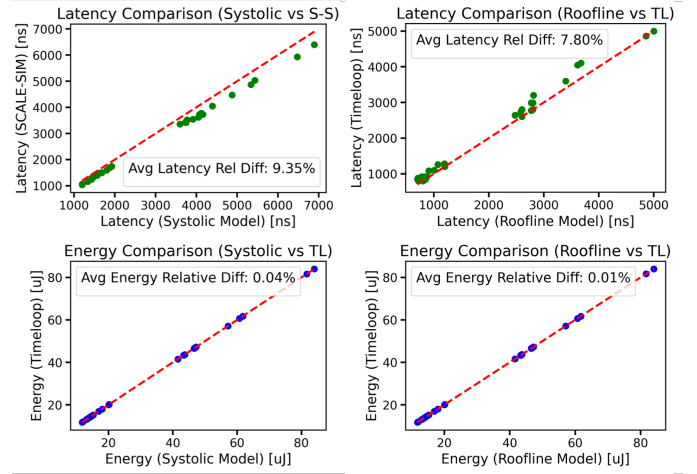


Fig. 6. Latency and energy validation of tile-level models.

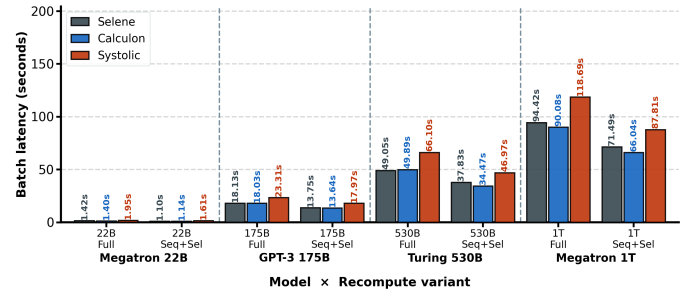


Fig. 7. Latency validation of system-level model.

ary output data flow configurations in SCALE-Sim (S-S) and optimally mapped GEMMs in Timeloop (TL), respectively, for the same micro-benchmarks. Both existing tools report latency in cycles. Hence, we convert them to time using our target frequency. In the case of our systolic model, SCALE-Sim does not directly report energy. Instead, a comparison was performed against Accelergy-enhanced Timeloop [43]. Our validation results indicate that both of our models closely track the modeling estimations of the respective existing tools. However, we observe some outlier GEMM sizes, where our roofline model exhibits a 7.8% average relative difference in latency compared to Timeloop, while our systolic model shows a 9.35% average relative difference against SCALE-Sim.

B. System-level Validation

To validate our framework’s ability to predict large-scale system performance, we compare our model against the Selene A100-based supercomputer used to validate Calculon [39]. We mimic the setup in [39] using several LLM training configurations (22B to 1T parameters). We created a configuration to match A100 specifications, abstracting the GPU architecture into an equivalent peak-throughput (312 TFLOPs) systolic architecture (Table IV). However, unlike peak roofline models, our model accounts for systolic penalties such as pipeline fill/drain latencies and tiling overheads for irregular matrix shapes.

As shown in Fig 7, our model yields conservative batch latency estimates, with Model Flops utilization (MFU) ranging

TABLE IV
A100 CONFIGURATION FOR SYSTEM VALIDATION.

System config. (Accel., Interconn.)	Accelerator config. (Compute, DRAM)	Tile config. (Compute, SRAM)	Freq.
$N \times 8^* @ 350\text{GBps}$	1 tile, 80GB@2TBps	333x333, 40MB@4.8TBs	1.41Ghz

* N scales with LLM size following [39].

from 23% to 45%, where performance degradations are caused by temporal tiling and small K-dimensions in attention layers. Overall, our approach tracks performance trends, demonstrating the ability of our framework to provide realistic projections for our hierarchical systems based on well-established acceleration principles.

VI. PERFORMANCE AND ENERGY EXPLORATION

We evaluate the performance and energy efficiency of an SCD system in comparison to multiple CMOS and cryo-CMOS-based systems for LLM inference and training workloads. Table V shows the detailed specifications of six systems evaluated in this paper. For CMOS-based architectures, we include *H100* and *TPUv4* clusters as representative state-of-the-art production systems that are widely-adopted accelerators in datacenters. These systems are organized in clusters with 8 accelerators per node, connected by NVlink (*H100*) and optical interconnects (*TPUv4*) within nodes, and InfiniBand (50 Gbps) for inter-node communication. For cryogenic implementations, we consider hierarchical cryo-CMOS systems similar to the SCD system. Two variations in cryo-CMOS technology are included: an energy-optimized (-E) variant and a performance-optimized (-P) variant of the 77K CMOS. They differ in their operating voltage, resulting in a tradeoff between energy and operation frequency. In addition, we evaluate a hierarchical 2.5D/3D room-temperature (300K) CMOS system that closely follows the SCD and cryo-CMOS systems, assuming 3D stacking of two layers. All systems are scaled and normalized to iso-area constraints, i.e. to have approximately the same total die area to ensure fair comparisons. All energy evaluations include cooling costs assuming 22W/W and 325W/W cooling efficiency for cryo-CMOS and SCD systems, respectively. We select GPT3-13B, GPT3-175B, Gopher-280B, and Megatron-1T as representative LLM workloads with 13 billion to 1 trillion parameters. We use a fixed batch size of 1024 for both training and inference.

Through our experiments, we aim to answer the following questions presented in Section I:

- Q1: What level of performance and energy efficiency remains in SCD systems after accounting for cooling and technology-derived architectural overheads, and how do system-level results compare to state-of-art CMOS systems and other cryogenic-based technologies?
- Q2: Under what conditions do SCD technologies offer tangible benefits and how would the benefit be impacted under less optimistic SCD technology assumptions?
- Q3: How do the implications of SCD technology translate to architectural design choices and, in turn, to application-level decisions?

A. Q1: System-wise Benefits of SCD

We model the iso-area performance and energy consumption of LLM workloads across all six systems. For *H100*, we make use of Calculon’s *H100* model based on peak throughput and power. We create a similar model configuration in Calculon for the *TPUv4*. All other systems are modeled using the framework from Section IV. For all systems, the reported latency corresponds to the wall-clock time for one training or inference step processing a single batch, including one forward pass, backward pass, and gradient synchronization for training, or the total time to process one batch through all transformer layers for inference. We characterize the latency and energy breakdown to show insights on where benefits and bottlenecks originate.

Figure 8 highlights end-to-end LLM inference and training comparisons across *H100*, *TPUv4*, CMOS, cryo-CMOS, and SCD-based large-scale systems. For LLM inference workloads, we observe that SCD systems, based on the geometric mean across 4 LLM models, achieve $4.0\times$, $10.1\times$, $8.3\times$ and $3.7\times$ lower latency, and $33\times$, $108\times$, $9\times$, $55\times$ and $195\times$ better energy efficiency compared to iso-area *H100*, *TPUv4*, CMOS, and cryo-CMOS energy and performance-optimized systems, respectively. Similarly, for LLM training, SCD achieves $6.6\times$, $16.0\times$, $8.5\times$, $4.8\times$ and $4.4\times$ lower end-to-end latencies, and $50\times$, $141\times$, $14\times$, $81\times$ and $286\times$ better energy efficiency compared to the same systems. At the same time, our SCD system yields only a 16% total MFU for Megatron-1T compared to around 50% for the *H100*, reflecting a higher difficulty to feed an SCD system.

Takeaway 1: SCD systems achieve significant performance and energy efficiency improvements compared to room-temperature and cryo-CMOS systems, even with not-as-dense technology nodes and 325W/W cooling costs.

Figure 9(a) shows a breakdown of batch inference latency for GPT3-175B across multiple systems for batch sizes ranging from 64 to 1024. The latency breakdown considers six components: compute time, memory overhead, PP bubble overhead, and TP, PP, and DP communication overheads. Across all batch sizes, the SCD system achieves the lowest total batch latency. The compute time reduction stems directly from the SCD system’s superior total compute capability (9,830 PFLOPs) compared to the *H100* (760 PFLOPs), *TPU* (281 PFLOPs), CMOS (4,297 PFLOPs), and Cryo-E (4,297 PFLOPs). Equally significant is the impact of superconducting interconnects, which offer substantially higher bandwidth and lower latency than NVLink (*H100*) and optical (*TPUv4*), significantly reducing TP and PP communication overheads. In the *H100* and *TPUv4* systems, these communication costs constitute a non-trivial fraction of total execution time and become proportionally more dominant at smaller batch sizes as less computation is available to amortize the transfer latency. By contrast, the SCD system maintains consistently low communication overhead across all evaluated batch sizes.

Takeaway 2: Performance benefits of SCD systems mainly come from the significantly improved compute capability and interconnect bandwidth.

TABLE V
SPECIFICATIONS OF SYSTEMS FOR COMPARISON WITH ROUGHLY EQUAL TOTAL AREA (ACCELERATOR COUNT $\times$ DIE AREA).

System	System config. (Accel., Interconn.)	Accelerator configuration (Compute, DRAM)	Tile configuration (Compute, SRAM/JSRAM)	Tech. node	System total area	3D layers	Frequency	PFLOPs
H100	96x8 @ 450 Gbps	114 SMs, 80GB @ 3.4 TBps	4 TC, 50MB	4nm	96 x 8 x 814 mm ²	-	1.83 GHz	760
TPUv4	128x8 @ 400 GBps	2 TCs, 32GB @ 1.2 TBps	4 128x128, 128MB	7nm	128 x 8 x 600 mm ²	-	1.05 GHz	281
CMOS 300K	64 @ 900 GBps	64 tiles, 1024GB @ 3.4 TBps	512x512, 256MB @ 9.68 TBps	7nm	64 x 64 x 144 mm ²	2	4 GHz	4,297
CMOS 77K-E	64 @ 1080 GBps	64 tiles, 1024GB @ 3.4 TBps	512x512, 256MB @ 9.68 TBps	7nm	64 x 64 x 144 mm ²	2	4 GHz	4,297
CMOS 77K-P	64 @ 1080 GBps	64 tiles, 1024GB @ 3.4 TBps	512x512, 256MB @ 9.68 TBps	7nm	64 x 64 x 144 mm ²	2	5.6 GHz	6,013
SCD 4K	64 @ 1048 TBps	64 tiles, 1024GB @ 30 TBps	2x200x200, 16MB @ 73.34 TBps	28nm	64 x 64 x 144 mm ²	6	30 GHz	9,830

SM-Streaming Multiprocessor, TC-Tensor Core.

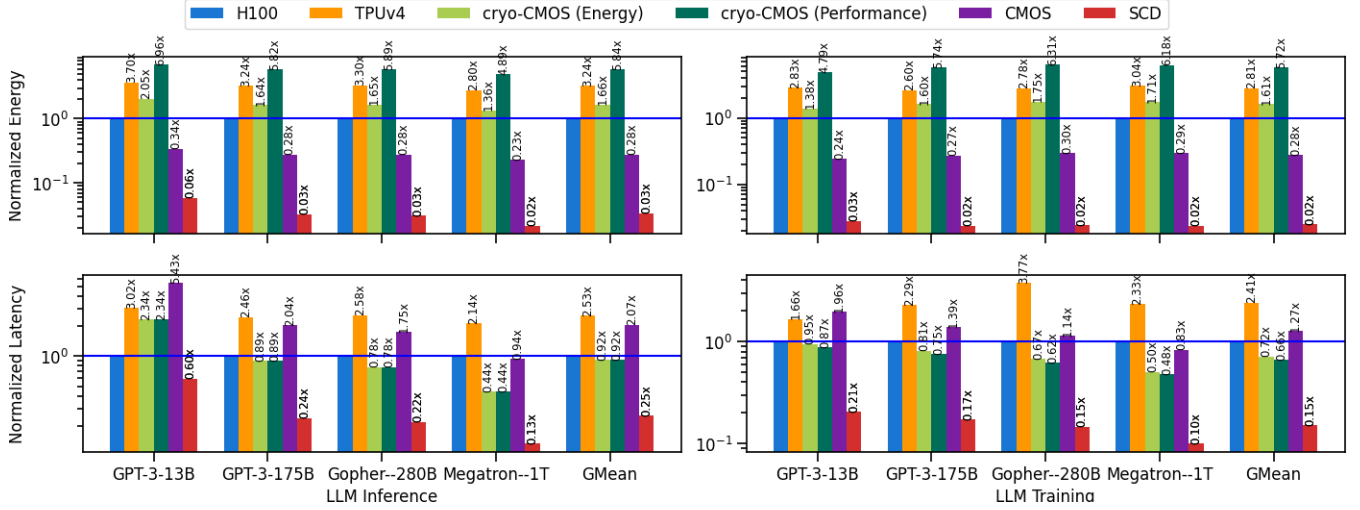


Fig. 8. Comparison of LLM training and inference energy and latency across different hardware systems. All values are normalized to H100 systems.

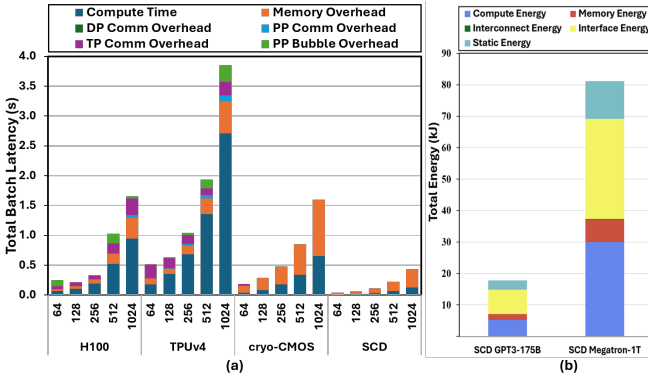


Fig. 9. End-to-end inference breakdown: (a) latency for GPT-3-175B across different hardware systems (b) energy breakdown of GPT-3-175B and Megatron-1T on SCD System.

Figure 9(b) shows the energy breakdown of LLM workloads on the SCD system. The energy of the 4K-77K interface is the biggest overhead as each DRAM access needs to go through such signal conversion. The energy of the interconnect is negligible due to resistance-free SCD wires. Even with those overheads, the total energy of SCD is still far better than other systems due to orders-of-magnitude lower compute energy (14fJ for SCD PE vs. pJ-level for CMOS PEs).

Takeaway 3: Energy savings mainly come from ultra-efficient SCD PE operations and zero-energy data movements.

B. Q2: Impacts of SCD Maturity

We conduct ablation studies of SCD features to model their performance degradations, and figure out the critical features of SCD to boost system-wide performance, and thereby gain insight into the contributions of individual technology characteristics in SCD and more specifically PCL-based technology. The ablation study evaluates the performance and energy impacts when specific aspects of SCD technology are constrained or modified. The scenarios considered include a sweep of (1) 4K and 77K cooling factors, (2) logic and memory density, (3) operational clock frequencies, and (4) JSRAM bandwidth.

Figure 10 shows normalized energy and latency for LLM training (left) and inference (right). As part of our findings, we observe degradation of energy efficiency across all ablation scenarios and performance across all scenarios except for the cooling factor. In terms of energy, although all SCD systems achieve higher energy efficiency than the H100 system, an SCD system with lower memory bandwidth shows the highest sensitivity and largest energy degradation, followed by cooling factor, density, frequency operation, and number of 3D stacked layers. The highest impact on energy is attributed to static energy caused by increased latency. Interface energy is increased in the lower density scenario as SRAM capacity becomes lower, causing degradation in data locality and increased DRAM accesses crossing 4K and 77K temperature domains.

Similarly, in terms of latency, all scenarios suffer latency degradation where the SCD systems with lower SRAM bandwidth perform worse than the H100 system. This confirms

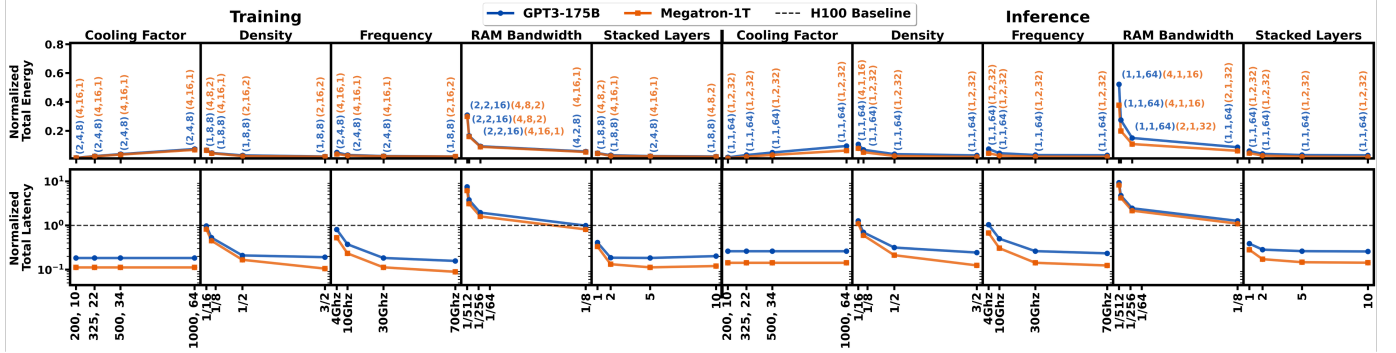


Fig. 10. SCD system ablation study for LLM training (left) and inference (right). Labels above each point represent the degree of parallelism (tensor parallelism (tp), pipeline parallelism (pp), data parallelism (dp)) used by the configuration.

on-chip SRAM technology as a critical factor for SCD system scaling. Based on our observations of the compute vs. memory time breakdown, performance degradations are otherwise mainly caused by significantly increased on-chip compute time, as lower density, lower frequency and lower SRAM bandwidth can all reduce compute throughput.

Takeaway 4: *Density, frequency and SRAM bandwidth all have essential contributions to SCD performance and energy efficiency. Of these, high-bandwidth JSRAM memory is key to achieving better performance than CMOS systems.*

We further study the impact of our ablation configurations on application-level decisions. As part of our LLM modeling, we exhaustively searched for the optimal parallelism strategy. The normalized latency plot in Figure 10 labels the optimal parallelism strategy for each configuration and model combination. The SCD baseline system takes advantage of the high frequency operation and a large number of compute elements and trades recompute of activation layers for a lower memory footprint. In larger models such as Megatron-1T, the SCD system favors a large pp degree (16). A large pipeline parallelism degree allows partitioning of very large models to further reduce memory requirements.

Takeaway 5: *SCD takes advantage of activation recompute to reduce the activation memory footprint and enable higher degrees of pipeline parallelism. If frequency or SRAM bandwidth is reduced the SCD system cannot take advantage of recompute.*

C. Q3: SCD Impacts on Architectural Design

We explore the SCD system design space of (1) PE area ratio, (2) accelerator size and (3) total DRAM capacity for GPT3-175B inference to address how SCD technology impacts the architectural and system design space. The batch size is 1024 in all cases. Other parameters including total die area and total DRAM bandwidth are kept constant across the design space, as the outcome of varying those is trivial.

PE area ratio indicates the ratio of total area dedicated to PEs in an accelerator stack, where the remaining area is used for JSRAM. We sweep 20%, 40% and 80%, ratios to explore the best SCD chip configuration. Tuning the PE area ratio explores the tradeoff between PE and SRAM area, and in turn

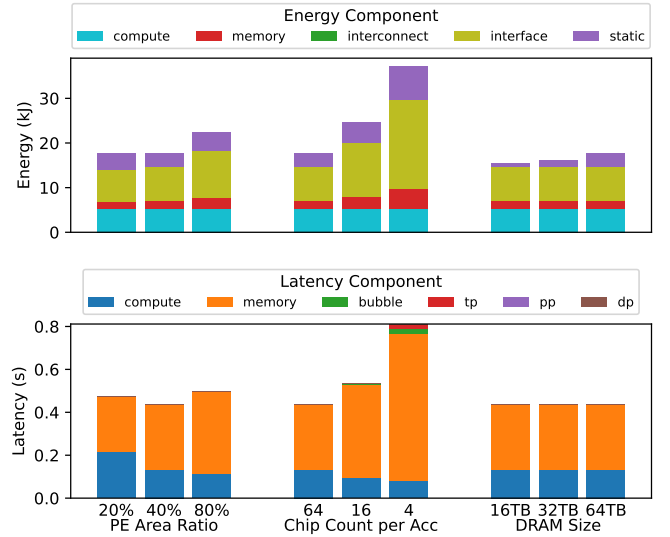


Fig. 11. Impact of SCD architectural parameters on energy and latency.

peak throughput vs. data locality. As shown in Figure 11, SCD systems with larger PE area ratios have less compute time, but can increase memory overhead as smaller JSRAM locality indicates more DRAM accesses. Increasing PE area ratio also increases the 4K-77K interface energy, as all DRAM accesses have to cross the interface. The overall best performance and energy is achieved at a PE area ratio of 40%.

Takeaway 6: *A large PE-to-SRAM ratio hurts locality and energy, while a small ratio degrades performance; an optimal ratio balances both.*

Accelerator size indicates the number of chips in each accelerator blade. We sweep accelerator sizes of 64, 16 and 4 to explore the best hierarchical SCD configuration. These correspond to 64, 256, 1024 accelerators in the system as we keep the same total area and compute power. We also scale DRAM bandwidth and capacity per accelerator linearly, but keep the total DRAM bandwidth and capacity unchanged. By sweeping accelerator size we explore the tradeoff between local and cross-accelerator communication. As shown in Figure 11, larger accelerators (64 chips) allow for more local communications and show zero bubble and communication overheads, while smaller accelerator configurations have to

additionally explore pipeline and tensor parallelism to map the workload to all accelerators, which adds additional overheads. Smaller accelerators (larger accelerator meshes) also increase the Cannon-GEMM phase, leading to increased DRAM data accesses, which significantly harm both latency and energy. The overall best configuration is 64 chips per blade.

Takeaway 7: SCD systems perform better when scaling up than scaling out, as locality is critical to both performance and energy efficiency.

Total DRAM capacity indicates the system-wide cryo-DRAM capacity across all accelerators. We sweep 3 DRAM capacities—16TB, 32TB, 64TB—to explore the tradeoff between DRAM performance and energy. In Figure 11, a larger total DRAM capacity slightly reduces latency as it allows more data parallelism instead of pipeline parallelism, which imposes bubble and communication overheads. It also slightly increase the static energy as larger cryo-DRAM needs more cooling power.

Takeaway 8: DRAM size does not have a significant impact on performance or energy, therefore can be determined based on workload sizes.

VII. DISCUSSION

In the following, we discuss the open challenges and potential roadblocks for the future adoption of SCD.

SCD fabrication: PCL makes use of CMOS fabrication methods. The fabrication of PCL building blocks has been demonstrated using NbTiN interconnects, NbTiN/ α Si/NbTiN Josephson junctions (JJs), and NbTiN/HZO/NbTiN tunable Metal-Insulator-Metal (MIM) capacitors on standard 300mm wafers [44]. The fabrication process is compatible with standard CMOS BEOL technology. Nevertheless, the yield and reliability of such technology including 3D stacking under thermal-mechanical stresses remains an open question.

Technology assumptions: For the underlying physical device- and circuit-level models, we acknowledge that the SCD technology characterization numbers used in our experiments are based on our best available knowledge (Section III). The projected performance and energy advantages are therefore conditional on realizing these assumed characteristics in practical implementations.

Large-scale integration: Smaller microarchitectural units have been demonstrated, including ALUs [45] with RSFQ operating at 50GHz. Processors with 23,713 JJs [6] have been fabricated in a 1 μ m process demonstrating 32GHz operating frequencies. A 2-by-2 systolic array at 52.6GHz was fabricated [7]. However, no larger-scale processor have been demonstrated due to DC bias current delivery. PCL technology, with AC bias current, has the potential to compete with conventional CMOS at scale [11], but large-scale integration remains under active investigation. The ISO-area assumption used throughout our evaluation should be interpreted as a research baseline for comparing technology and architectural trends under a common area budget. Real-world cost, manufacturability, packaging yield, and reliability may change the relative advantage of the evaluated systems.

Cryocooling: While modern 4K cryocoolers can handle tens to hundreds of Watts of power with higher cooling efficiency at larger scales, limited fridge volume requires tight packaging to achieve the full cooling efficiency in a single refrigerator. Therefore, 3D stacking of PE/JSRAM layers and network stacks with TSVs across interposers are necessary to leverage packaging density. Unlike conventional CMOS, where high packaging density can lead to heat dissipation problems, SCD’s extremely low energy consumption allows such packaging without heat dissipation issues. However, aforementioned 3D stacking technology and reliability questions remain.

Workloads: Due to the need to amortize the cooling cost, SCD makes the most sense for relatively large-scale and computationally intensive applications, such as LLMs. LLM behavior depends on the execution phase and configuration: training is generally more compute-intensive than inference. SCD is generally not efficient for memory- or communication-heavy applications due to limited on-chip memory densities. Therefore, applications like graph analytics or databases may benefit less. Additionally, our framework currently models attention operations as individual GEMMs rather than fused kernels such as FlashAttention. Incorporating such optimizations, along with their interaction with the SCD memory hierarchy, is an important direction for future work.

VIII. SUMMARY AND CONCLUSIONS

In this paper, to the best of our knowledge, we are the first to systematically study the performance and energy efficiency impact of state-of-the-art SCD technology on large scale supercomputing systems. We introduce a hierarchical framework for modeling data-center-scale superconducting systems and perform extensive design space exploration and technology ablation studies for emerging workloads such as LLM training and inference. Our results demonstrate that compared to H100, TPUv4, and cryo-CMOS systems scaled to the same total area, the SCD system shows significant speedup and energy efficiency improvements of 4-16 $\times$ and 14-286 $\times$ on LLM training, and 4-10 $\times$ and 9-195 $\times$ on inference, respectively. These results confirm the potential to build future supercomputing systems in SCD technology. Furthermore, our ablation studies show that the integration density and availability of on-chip superconducting JSRAM enabled by recent advances in SCD technology are essential for system performance. In future work, we plan to extend our studies by augmenting our modeling framework with semi-analytical and simulation-based components to increase accuracy and account for more complex and dynamic architectures and workload effects.

REFERENCES

- [1] C. Hamilton and F. Lloyd, “100 ghz binary counter based on dc squid’s,” *IEEE Electron Device Letters*, pp. 335–338, 1982.
- [2] K. K. Likharev, “Superconductor digital electronics,” *Physica C—superconductivity and Its Applications*, pp. 6–18, 2012.
- [3] V. K. Semenov, Y. A. Polyakov, and S. K. Tolpygo, “Very large scale integration of josephson-junction-based superconductor random access memories,” *IEEE Transactions on Applied Superconductivity*, pp. 1–9, 2019.
- [4] Y. Yamanashi, M. Tanaka, and A. e. a. Akimoto, “Design and implementation of a pipelined bit-serial SFQ microprocessor, core1 β ,” *IEEE Transactions on Applied Superconductivity*, pp. 474–477, 2007.

- [5] Y. Ando, R. Sato, and M. Tanaka, "Design and demonstration of an 8-bit bit-serial RSFQ microprocessor: CORE e4," *IEEE Transactions on Applied Superconductivity*, pp. 1–5, 2016.
- [6] K. Ishida, M. Tanaka, I. Nagaoka *et al.*, "32 ghz 6.5 mw gate-level-pipelined 4-bit processor using superconductor single-flux-quantum logic," in *IEEE Symposium on VLSI Circuits*, 2020, pp. 1–2.
- [7] K. Ishida, I. Byun, I. Nagaoka *et al.*, "SuperNPU: An extremely fast neural processing unit using superconducting logic devices," in *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2020, pp. 58–72.
- [8] G. Tzimpragos, D. Vasudevan, N. Tsiskaridze *et al.*, "A computational temporal logic for superconducting accelerators," in *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2020, p. 435–448.
- [9] P. Gonzalez-Guerrero, M. G. Bautista, and D. Lyles, "Temporal and sfq pulse-streams encoding for area-efficient superconducting accelerators," in *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2022, p. 963–976.
- [10] R. Cai, A. Ren, and e. a. Chen, Olivia, "A stochastic-computing based deep learning framework using adiabatic quantum-flux-parametron superconducting technology," in *IEEE International Symposium on Computer Architecture (ISCA)*, 2019, p. 567–578.
- [11] Q. Herr, T. Josephsen, and A. Herr, "Superconducting pulse conserving logic and josephson-sram," *Applied Physics Letters*, p. 182604, 05 2023.
- [12] R. K. Pandey and S. K. Pandey, "Analyzing the performance of 7nm FinFET based logic circuit for the signal processing in neural network," in *IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, 2020.
- [13] H. L. e. a. Chiang, "Cold CMOS as a power-performance-reliability booster for advanced FinFETs," in *IEEE Symposium on VLSI Technology*, 2020.
- [14] K. Likharev and V. Semenov, "Rsfq logic/memory family: a new josephson-junction technology for sub-terahertz-clock-frequency digital systems," *IEEE Transactions on Applied Superconductivity*, pp. 3–28, 1991.
- [15] O. A. Mukhanov, "Energy-efficient single flux quantum technology," *IEEE Transactions on Applied Superconductivity*, pp. 760–769, 2011.
- [16] A. F. Kirichenko, I. V. Vernik, M. Y. Kamkar *et al.*, "Ersfq 8-bit parallel arithmetic logic unit," *IEEE Transactions on Applied Superconductivity*, pp. 1–7, 2019.
- [17] N. Takeuchi, D. Ozawa, and Y. Yamanashi, "An adiabatic quantum flux parametron as an ultra-low-power logic device," *Superconductor Science and Technology*, p. 035010, jan 2013.
- [18] Q. P. Herr, A. Y. Herr, and O. T. Oberg, "Ultra-low-power superconductor logic," *Journal of Applied Physics*, p. 103903, 05 2011.
- [19] Q. Herr, A. Braun, and A. e. a. Brownfield, "Measurement and data-assisted simulation of bit error rate in rql circuits," *Superconductor Science and Technology*, p. 025017, jan 2022.
- [20] D. Carmean, A. Braun, and A. Y. Herr, "Phase-mode based superconducting logic," *U.S. Patent No. 9,543,959*, 10, Jan 2017.
- [21] V. Vashishtha, M. Vangala, and *et al.*, "Robust 7-nm SRAM design on a predictive PDK," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017.
- [22] S. S. Parihar, V. M. van Santen, S. Thomann *et al.*, "Cryogenic CMOS for quantum processing: 5-nm FinFET-based SRAM arrays at 10 k," *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 3089–3102, 8 2023.
- [23] Q. P. Herr and M. J. Feldman, "Error rate of a superconducting circuit," *Applied Physics Letters*, pp. 694–695, 07 1996.
- [24] Q. Herr, M. Johnson, and M. Feldman, "Temperature-dependent bit-error rate of a clocked superconducting digital circuit," *IEEE Transactions on Applied Superconductivity*, pp. 3594–3597, 1999.
- [25] N. Takeuchi, H. Suzuki, and N. Yoshikawa, "Measurement of low bit-error-rates of adiabatic quantum-flux-parametron logic using a superconductor voltage driver," *Applied Physics Letters*, no. 20, p. 202601, 05 2017.
- [26] S. V. Rylov, "Clockless dynamic sfq and gate with high input skew tolerance," *IEEE Transactions on Applied Superconductivity*, pp. 1–5, 2019.
- [27] T. Yamae, N. Takeuchi, and N. Yoshikawa, "Systematic method to evaluate energy dissipation in adiabatic quantum-flux-parametron logic," *Journal of Applied Physics*, p. 173903, 11 2019.
- [28] C. L. Ayala, T. Tanaka, R. Saito *et al.*, "Mana: A monolithic adiabatic integration architecture microprocessor using 1.4-zj/op unshunted superconductor josephson junction devices," *IEEE Journal of Solid-State Circuits*, pp. 1152–1165, 2021.
- [29] F. China, T. Narama, and N. e. a. Takeuchi, "Design and demonstration of interface circuits between rapid single-flux-quantum and adiabatic quantum-flux-parametron circuits," *IEEE Transactions on Applied Superconductivity*, pp. 1–5, 2016.
- [30] A. Y. Herr, Q. P. Herr, and O. T. e. a. Oberg, "An 8-bit carry look-ahead adder with 150 ps latency and sub-microwatt power dissipation at 10 GHz," *Journal of Applied Physics*, vol. 113, no. 3, p. 033911, 01 2013.
- [31] A. Herr, Q. Herr, and e. a. Brebels, Steve, "Scaling nbntn-based apowered josephson digital to 400m devices/cm²," 2023.
- [32] S. Tahara, I. Ishida, and Y. Ajisawa, "Experimental vortex transitional nondestructive read-out josephson memory cell," *Journal of Applied Physics*, pp. 851–856, 01 1989.
- [33] S. Nagasawa, Y. Hashimoto, and H. Numata, "A 380 ps, 9.5 mw josephson 4-kbit ram operated at a high bit yield," *IEEE Transactions on Applied Superconductivity*, pp. 2447–2452, 1995.
- [34] S. Nagasawa, H. Numata, and Y. Hashimoto, "High-frequency clock operation of josephson 256-word/spl times/16-bit rams," *IEEE Transactions on Applied Superconductivity*, pp. 3708–3713, 1999.
- [35] M. N. Dorojevets, P. I. Bunyk, and D. Zinoviev, "Flux chip: design of a 20-ghz 16-bit ultrapipelined rsfq processor prototype based on 1.75-/spl mu/m lts technology," *IEEE Transactions on Applied Superconductivity*, pp. 326–332, 2001.
- [36] R. Burnett, R. Clarke, T. Lee *et al.*, "Demonstration of superconducting memory for an rql cpu," in *Proceedings of the International Symposium on Memory Systems*, 2018, p. 321–323.
- [37] Cryogenic Electronics and Quantum Information Processing (CEQIP) International Focus Team (IFT), "International roadmap for devices and systems (IRDS): Cryogenic electronics and quantum information processing (CEQIP)," IEEE, Tech. Rep., 2020.
- [38] A. Herr and Q. Herr, "A data center in a shoebox: Imec's plan to use superconductors to shrink computers," *IEEE Spectr*, p. 37–41, Jun. 2024.
- [39] M. Isaev, N. McDonald, and L. Dennison, "Calculon: a methodology and tool for high-level co-design of systems and large language models," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2023.
- [40] L. E. Cannon, "A cellular computer to implement the kalman filter algorithm," Ph.D. dissertation, Montana State University, 1969.
- [41] A. Samajdar, J. M. Joseph, Y. Zhu *et al.*, "A systematic methodology for characterizing scalability of dnn accelerators using scale-sim," in *IEEE International Symposium on Parallel and Distributed Processing (ISPASS)*, 2020, pp. 58–68.
- [42] A. Parashar, P. Raina, Y. S. Shao *et al.*, "Timeloop: A systematic approach to dnn accelerator evaluation," in *IEEE International Symposium on Parallel and Distributed Processing (ISPASS)*, 2019, pp. 304–315.
- [43] Y. N. Wu, J. S. Emer, and V. Sze, "Accelergy: An architecture-level energy estimation methodology for accelerator designs," in *IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, 2019.
- [44] A. Pokhrel, D. P. Lozano, J.-P. Soulié *et al.*, "Nbntn based two-metal level semi-damascene interconnects, josephson junctions and capacitors for superconducting digital logic," in *IEEE International Electron Devices Meeting (IEDM)*, 2024, pp. 1–4.
- [45] G.-M. Tang, K. Takata, and M. Tanaka, "4-bit bit-slice arithmetic logic unit for 32-bit rsfq microprocessors," *IEEE Transactions on Applied Superconductivity*, pp. 1–6, 2016.



Daniel Peralta Velazquez (Graduate Student Member, IEEE) received the M.Eng degree in computer engineering from the University of Texas at Austin, Austin, TX, USA. He is currently pursuing the Ph.D. in Computer Engineering at the same institution. His work focuses on both understanding the impact of emerging technologies in energy efficiency and advancing current performance challenges.



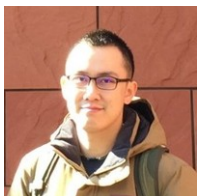
Siyuan Ma (Graduate Student Member, IEEE) received B.Eng degree in Electrical Engineering from Xi'an Jiaotong University. He is pursuing the Ph.D. in Electrical and Computer Engineering in the University of Texas at Austin, focusing on architecture exploration and performance modeling for emerging computing systems.



Sairam Sri Vatsavai is a Postdoctoral Research Associate at Brookhaven National Laboratory, where he is part of the Systems, Architectures, and Advanced Technologies group. His research focuses on leveraging AI to develop performance modeling and simulation frameworks for hardware accelerators in High-Performance Computing (HPC) systems. Sairam earned his PhD in Electrical Engineering from the University of Kentucky, USA, in 2024. He completed his Bachelor's degree at the Jawaharlal Institute of Technology, India.



Saikat Chakraborty is a PhD student at Electrical and Computer Engineering in the University of Texas at Austin. He completed his Master's degree in Microelectronics specialization from Indian Institute of Technology (IIT), Bombay. Prior to joining UT, he worked with TD-Device modelling group and TD-HV group at Globalfoundries for over 3 years. His research interests lie in modelling and development of emerging nano-scale devices and novel memory technologies including power-efficient cryogenic High Performance Computing and Ferroelectric-FET based vertical NAND FLASH technology.



Lingda Li received the PhD degree in computer architecture from Peking University in 2014. He is a group lead and research staff with the Computing and Data Sciences directorate of Brookhaven National Laboratory. He was a postdoctoral associate with the Department of Computer Science of Rutgers University between 2014 and 2016. He is generally interested in computer architecture and programming model research, with recent focuses on performance simulation/modeling, memory system, and machine learning.



Anna Herr co-founded Snowcap Compute to build the first commercially available superconducting compute platform. Anna has held senior scientific and engineering roles at imec and Northrop Grumman, where she was named a Fellow for her contributions to advanced computing technologies. Earlier, as a professor at Chalmers University of Technology and research fellow of The Royal Swedish Academy of Sciences in Sweden. Her research spans superconducting systems, VLSI circuit design, magnetic memory, and classical control for qubits. Anna holds a master's and Ph.D. in Physics from Lomonosov Moscow State University.



Manu Perumkunnil Komalan (Member, IEEE) received his integrated master's degree in nanotechnology from Amity University, India, in 2010. He pursued a Ph.D. degree in electrical engineering from Katholieke Universiteit Leuven, Belgium, and a Ph.D. degree in computer science from the Universidad Complutense de Madrid. He joined IMEC, as a Memory System Architecture Researcher in 2017. He is currently a Scientific Director within the Compute Systems Architecture Unit at IMEC, where his focus is systems design for the AI and HPC domain, including Superconducting Digital technology activities.



Quentin Herr is Chief Scientific officer of Snowcap Compute, a company he co-founded to build the first commercially available superconducting compute platform for large-scale digital and quantum applications. He previously was Scientific Director at imec and imec USA, and Northrop Grumman Fellow. He holds MA Physics and PhD Electrical Engineering degrees from the University of Rochester. His 30 years of experience in superconducting integrated circuit design, and test have spanned analog-to-digital converters, high performance digital, large-scale applications, and interconnect.



George Biros received the PhD degree in Computational Science and Engineering from Carnegie Mellon University in 2000. He is the W. A. "Tex" Moncrief chair in Simulation-Based Engineering Sciences with the Oden Institute for Computational Engineering and Sciences and has full professor appointments with the Departments of Mechanical Engineering and Computer Science (by courtesy), The University of Texas at Austin. He was a postdoctoral associate with the Courant Institute of Mathematical Sciences from 2000 to 2003. He was among a team of researchers that won the IEEE/ACM SC03 and SC10 Gordon Bell Awards.



Jaydeep Kulkarni (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2009. From 2009 to 2017, he was with Intel Circuit Research Lab, Hillsboro, OR, USA, where he worked on energy-efficient integrated circuit technologies. He is currently an Associate Professor of Electrical and Computer Engineering with the University of Texas at Austin, Austin, TX, USA, and a fellow of the Silicon Labs endowed chair in Electrical Engineering. His research focuses on machine learning accelerators, in-memory computing, emerging nanodevices, hardware security, and heterogeneous integration.



Lizy Kurian John (Fellow, IEEE) received the PhD degree in computer engineering from the Pennsylvania State University. She is the Truchard Foundation Chair in Engineering at the Department of Electrical and Computer Engineering, University of Texas at Austin. Her research interests include the areas of computer architecture, multicore processors, memory systems, performance evaluation and benchmarking, workload characterization, and reconfigurable computing. She is a Fellow of ACM, Fellow of AAAS, and a Fellow of the National Academy of Inventors (NAI).



Andreas Gerstlauer (Senior Member, IEEE) received the Ph.D. degree in Information and Computer Science (ICS) from the University of California at Irvine (UCI), Irvine, CA, USA, in 2004. He is a Cullen Trust for Higher Education Endowed Professor and an Associate Chair with the Electrical and Computer Department, The University of Texas at Austin, Austin, TX, USA. Prior to joining UT Austin in 2008, he was an Assistant Researcher with the Center for Embedded Computer Systems, UCI. His research interests cover system-level design and embedded systems.