A System Level Performance Evaluation for Superconducting Digital Systems

Joyjit Kundu*, Debjyoti Bhattacharjee*, Nathan Josephsen†, Ankit Pokhrel*, Udara De Silva†, Wenzhe Guo*, Steven Van Winckel*, Steven Brebels*, Quentin Herr†, Anna Herr†, and Manu Perumkunnil*
imec (*Leuven, Belgium; †USA-Florida)

Abstract– Superconducting Digital (SCD) technology offers significant potential for enhancing the performance of next generation large scale compute workloads. By leveraging advanced lithography and a 300 mm platform, SCD devices can reduce energy consumption and boost computational power. This paper presents a cross-layer modeling approach to evaluate the system-level performance benefits of SCD architectures for Large Language Model (LLM) training and inference. Our findings, based on experimental data and Pulse Conserving Logic (PCL) design principles, demonstrate substantial performance gain in both training and inference. We are, thus, able to convincingly show that the SCD technology can address memory and interconnect limitations of present day solutions for next-generation compute systems.

I. Introduction

The ever increasing demand for Generative Artificial Intelligence is shaping the roadmap of next generation compute systems [1], [2]. Large language models (LLMs) in particular have dominated this domain given their versatility and performance scaling [3]. However, training typical LLMs is very compute intensive and requires processing a huge amount of data with significant compute time resulting in an unprecedented level of carbon footprint [4]. For instance, training a GPT-3 model costs around 10M USD with an estimated energy consumption of $\sim$1300 MWh [5]. While LLM inference is computationally cheaper compared to training, the cumulative cost is extremely high in the long run, as multiple users queries are serviced [6]. This is obviously not sustainable given the trend of increasing LLM size [7]. Furthermore, the astronomical total cost of ownership (TCO) for advanced CMOS technology [8] used to build training and inference hardware, only serves to exacerbate the problem. Finally, semiconductor technology constraints like memory and interconnect scaling walls [9], [10] adversely affect future AI models that will become increasingly memory or communication bound [7], [11], requiring massive bandwidths to feed compute. Thus, there is a desperate need for a paradigm shift in compute system technology to tackle this global challenge.

Superconducting Digital (SCD) Electronics [12]–[15] has been one of the promising beyond-CMOS technologies with a potential for tremendous improvements across the whole stack. Unique physics and material advantages facilitate active devices with 'sub-attoJoule' energy scales (at 'ps' time scales), quantum accurate encoding of digital information [12]–[14], wires with negligible dissipation and dispersion up to 100's GHz frequencies. The energy dissipation per switching event in SCD does not depend on the process node (like CMOS) but instead is relative to thermal noise. Given these benefits, superconducting architectures stand out by being able to operate at $\sim$20$\times$ higher frequencies for a fraction of the on-chip power (100$\times$ less) [16], [17] and have 10000$\times$ more energy efficient communication at the on-chip clock rate [18], [19]. Cryogenic cooling can be applied to the entire system as opposed to individual dies, resulting in extremely high volumetric packaging density with all components of a system being both physically and electrically close [20].

Despite such promising device data, it is unclear how such technology primitives would pan out when estimating overall performance while designing a full system for future AI or HPC workloads. It is imperative to perform early cross-layer system level evaluations to understand the competitiveness of SCD technology, compared to existing mature technologies. Our key contributions here are as follows:

- Construction of SCD-based system-architecture for LLM training and inference, via parametric architectural building blocks in a bottom-up manner, from SCD devices, pulse-conserving logic (PCL), superconducting Josephson SRAM (JSRAM) and packaging solutions.
- Performance projection of the SCD system architecture for LLM training and inference based on an analytical modeling framework.
- Showcasing the potential improvements when using SCD system compared to contemporary GPU systems.

II. Background

A. Advanced SCD process technology

The fabrication of `Nb` based SCD devices have been reported previously. However, these have a temperature budget of $\leq$200°C which limits effective scaling to lower dimensions for advanced process nodes, thus not meeting the process specifications required for complex integrated circuits [21]–[23]. To address these challenges, we have developed `NbTiN` based SCD building blocks that allow advanced integration (meeting the requirements for advanced digital systems). Compared to the SOA `Nb`, `NbTiN` is a stable material, compatible with temperature budget of 420°C and is scalable down to critical dimensions (CDs) of $<$50nm [24] [25]. Specifically, we use a legacy semi-damascene [26] [27] integration process

978-3-9826741-0-0/DATE25/© 2025 EDAA

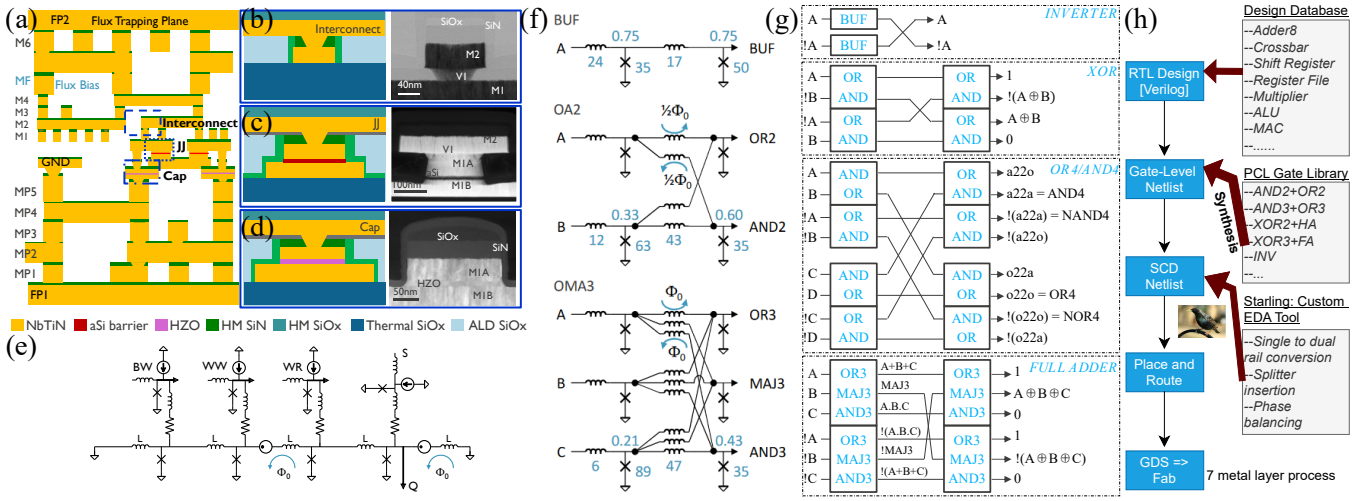


Fig. 1: *(a)Schematic of target 16ML SCD stack. (b)Scheme of 2ML BEOL interconnects and its TEM image. (c)Scheme of JJs with αSi barriers and its TEM image. (d)Scheme of HZO MIM capacitor its TEM image. (e)HD JSRAM 1R/1W unit cell with 8 JJs (f)Building blocks of the PCL logic family. (g)Dual rail logic gates in the PCL cell library. (h)Outline of RTL-GDS automated flow.*

and report the fabrication of NbTiN based BEOL interconnects, NbTiN/αSi/NbTiN Josephson junctions (JJs), and NbTiN/HZO/NbTiN tunable Metal-Insulator-Metal (MIM) capacitors that are the fundamental blocks of SCD technology. The stack utilizes a standard interlayer dielectric process and 193i lithography (suitable for 40nm and 28nm) to achieve a device density of 400M JJs/cm^2. Measurements have demonstrated the scaling of BEOL interconnects to CDs of 50-600 nm, Amorphous Si (αSi) based JJs to diameters between 210-500nm (CD control of $\sigma < 2\%$ across 300mm wafer), and tunable HZO MIM capacitors to diameters between 195-600nm (CD control of $\sigma < 2\%$ across 300 mm wafer). The resonant AC network for power distribution is enabled by NbTiN wires and HZO MIM capacitors [28]. The process and device specifications are detailed in Fig. 1 and Table. I.

B. Pulse Conserving Logic and JSRAM

Digital circuits have been constructed in various low-power superconducting logic families like RSFQ, RQL, and AQFP [15]. These efforts have been held back by the lack of large scale design automation targeting superconducting logic. Superconducting circuits that are DC-powered waste energy in power distribution, while circuits that are AC-powered face challenges in static timing analysis and automated synthesis.

Pulse-conserving logic (PCL) [17] is a SCD logic family that enables automated circuit design using commercially available tools. In a PCL circuit, each digital signal comprises two physical wires (positive and negative sense) and inversion is achieved by swapping them. This dual-rail encoding eliminates the delay in logical inversion (inherent to the data encoding in other AC powered SCD logic families) and greatly simplifies mapping of PCL gates into a standard cell library, a subset of which is shown in Fig. 1f and 1g. The PCL library is compatible with a slightly modified RTL-GDS flow. Fig. 1h illustrates the translation of an RTL design to a PCL circuit using off-the-shelf synthesis followed by a modification stage that performs phase assignment, phase matching, and single-to-dual rail conversion. A commercial place and route solution that can route wires with targeted inductance was used. JSRAM [17] is a SCD memory technology complementary to PCL, which, implemented in the advanced NbTiN/αSi/NbTiN process described in the previous subsection, enables memory density of 4 MB/cm^2, a 600× increase over older SFQ-compatible memory technology, with XY addressing analogous to CMOS SRAM. It represents a key intermediate stage of the memory hierarchy with lower density but higher locality than cryo-DRAM.

TABLE I: *Specifications for the SCD technology stack in our work*

Parameter	CMOS 5nm	This work
Operating Frequency	2GHz	30GHz
Device	FinFET	Josephson Junction
– **Device Density**	$\sim 170M/mm^2$	$\sim 4M/mm^2$
– **Voltage**	$0.7V$	$\sim$ 1.0mV
On-chip Memory	SRAM	JSRAM
– **Density** (incl. peri)	$\sim 4.5MB/mm^2$	$\sim 0.4M/mm^2$
– **HD Device Unit Cell**	6T	8JJ
(Single Port 1R/1W)	$0.021\mu m^2$	$1.86\mu m^2$
Lithography	EUV	193
ML stack layers	16	16
Interconnects	Cu	NbTiN
– **Resisitivity** (M1-M3)	$\sim 75\mu\Omega$.m	$<2\mu\Omega$.m
– **Minimum MP**	28nm/35nm	50nm
– **Power Efficiency**	1-2Gb@1pJ/bit	$\sim$200Gb@1pJ/bit

III. Design Specifications

In this section, we describe the fundamental design blocks that serve as inputs towards building the SCD blade for LLM training and inference.

High Throughput Compute Core: A regular array of bf16 MAC units is used for a TPU like high throughput compute core. Our bf16 MAC (8-bit add, 8-bit multiply and 32 bit accumulate) consists of ∼8k JJs. This regular MAC array architecture is banked for scale-out. Due to the lack of RC overhead for superconducting data transmission lines and utilization of dedicated low latency memory hierarchy (detailed below), we

achieve a very high packing density and utilization for the high throughput compute core. The peak floating point (bf16) performance achieved is ∼2.45 PetaFLOPs (30GHz×400k MACs×2Ops/MAC) at 80% utilization of the MACs (>400K) in a $144mm^2$ die footprint.

Control Complex: A simple dual core (in-order) complex manages the distribution of kernels fragments and appropriate instructions to the high throughput core. The 'Control Complex' maintains local directories for coherency for the global addressing. It also assists in power/clock gating (if any) locally.

Memory Hierarchy: The 8JJ single port (1R/1W) JSRAM configuration detailed in Fig. 1e serves as the 'High Density' (**HD**) on-chip memory variant for our SCD stack. HD JSRAM is used for the L1 and L2 data caches in our study. Multi-port variants, 2R/1W JSRAM (14 JJs) and 3R/2W (29 JJs), serve as the 'High Performance' (**HP**) on-chip memory variants. These are used for register files, high speed buffers and L1 instruction caches in our system configurations based on the required parallelism, performance and capacity specifications.

Switch: Like conventional designs, our SCD switch consists of a central crossbar that connect the input ports (+ associated buffers) to the control unit and output ports (+ associated buffers). The building block of the crossbar is in-turn the superconducting MUX based cross-point unit. Our crossbar is hierarchical in nature with a first level of cross-point units used to route each packet to the appropriate output port, and the second level serving as an aggregation point.

Main Memory Datalink: A custom interface, as shown in Fig. 2a is designed for data transfer between the 4K (Superconducting Compute) and 77K (Cryo-Memory) domains in our system. Physically, this dual-temperature interface connects the interposers in both domains via Cu transmission lines across a glass bridge. This datalink is DC coupled and thus requires no specific encoding. It performs translation for voltage levels, data rates, data direction and also protocol based on the amplification and PHY specifications at either ends(Fig. 2b). For the given baseline specifications, our datalink can achieve a peak bidirectional bandwidth of 30TBps (20TBps Downlink and 10TBps Uplink). This can be increased or decreased based on the power budget, available metal layers, channel reach, reliability, noise & dispersion etc.

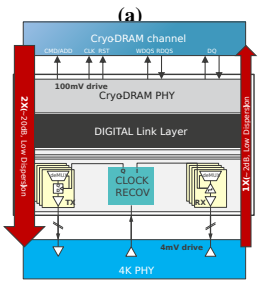
Cryo-DRAM block: The cryo-DRAM block consists of an array of cryo-DRAM packages integrated on a Si interposer. These have cryo-DRAM packages have no customizations or internal design changes and are simply regular DDR-X/LPDDR-X packages operating at 77K. As such these have inherent power benefits that have been extensively reported in [29]–[31].

A more detailed look into the power breakdown, micro-architecture and ISA lie outside the scope of this paper and will be pursued as future work.

IV. Proposed System Architecture

A. SCD Processing and Network Unit (SPU and SNU)

A single SPU (detailed in Fig. 3a), consists of a high compute throughput die (section **III**), a host controller die, multiple



(b)

Parameter	Downlink towards 4K	Uplink towards 77K
Wire Width	6.2μm	62μm
Wire Thickness	0.5μm	0.5μm
Wire Pitch	30μm	90μm
Wire Length	30mm (Cu) + 30mm (NbTiN)	30mm (Cu) + 30mm (NbTiN)
Data Rate	1Gbps	1Gbps
No. of wires	20,000	10,000
Required ML	2	8

Fig. 2: *(a) Diagrammatic representation of the datalink interface (Cu over Glass bridge) connecting the 4K (Compute) and 77K (Main Memory) domains. (b) Baseline specifications for the main memory datalink.*

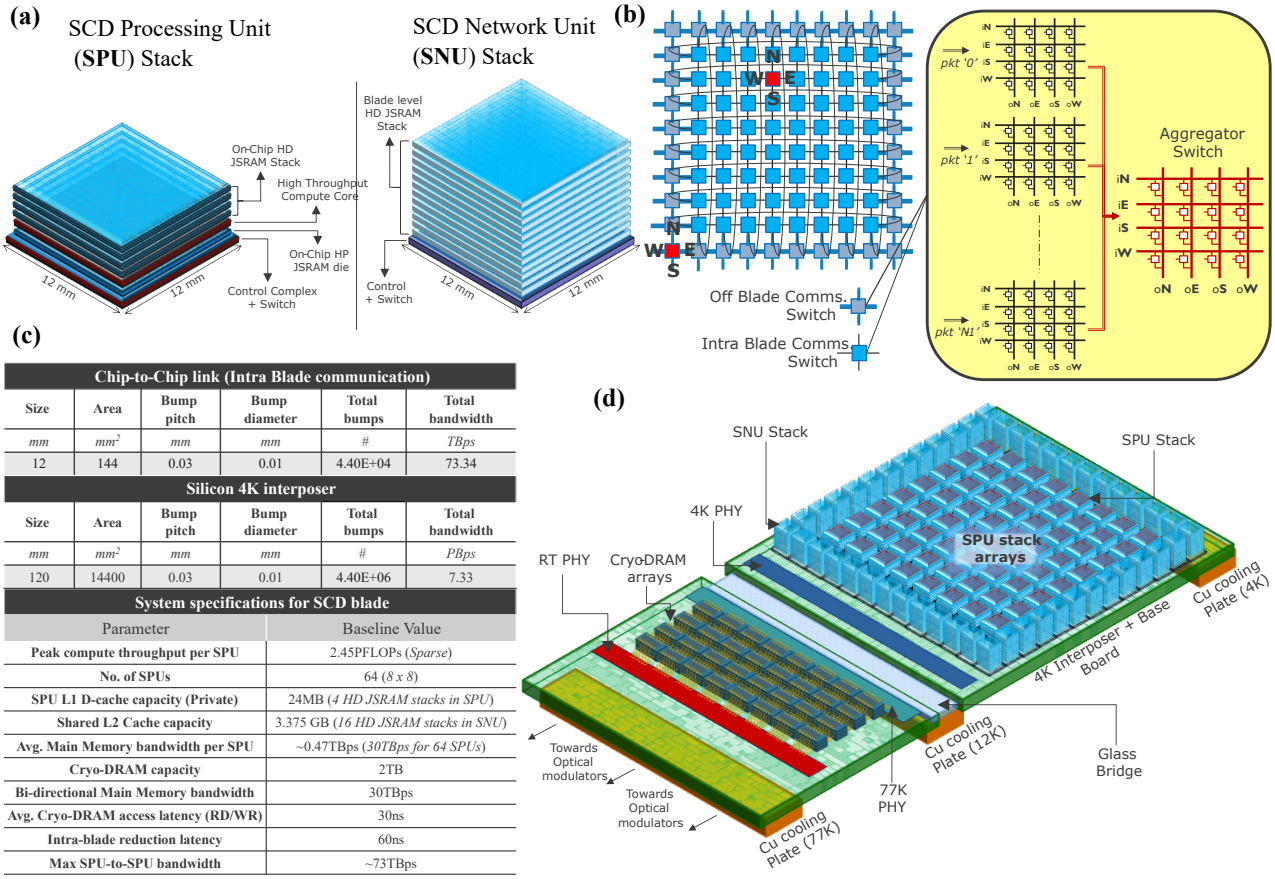
HD JSRAM based Memory dies and HP JSRAM die all of which are vertically stacked by means of `NbTiN` through-silicon vias (TSVs). The HD JSRAM dies serve both, the private L1 dcaches of the high throughput and control complex cores, as well as the shared L2 of the control complex. The HP JSRAM die contains the register files and L1 icaches for both the high throughput core and control cores. The control complex as well as the local switch lie at the base of the SPU physical stack of dies. The SNU (Fig. 3a) is another vertical stack of dies with a base die serving as switch for off-node or main memory communications. The JSRAM dies in each SNU die-stack are composed of banked HD arrays and function as slices of the shared and distributed L2 cache for all the high throughput cores (per SPU) in the blade. These help in bridging the latency gap for off-blade communication.

B. SCD blade and interconnect

We propose a SCD blade (Fig. 3d), where, a 2D array of SPUs are interconnected via their local switches to construct a 2D torus intra-node network. The SNU die-stack lies at the edges of the 2D array of SPUs, and can even enable potential vertical stacking of SCD blades. This is made possible by extending `NbTiN` TSVs in the SNU to connect to the substrate of the neighbouring blade (and optimizing its aspect ratio). Fig. 3c shows target specifications of the `Si` interposer, and intra communication interfaces. The performance numbers reported for the network interfaces are obtained considering bump density 4%, bump redundancy 40% and bandwidth per wire at 30 Gbps (30GHz operating frequency). In the proposed system, off-chip CryoDRAM operating at `77K` is connected to the compute array operating at `4K` by means of the datalink detailed in section **III** and Fig. 2a.

C. System Architecture Parameters

To assess the potential of SCDs for LLM workloads, we instantiate the system with the baseline parameters specified in Fig. 3c. Our blade consists of 8×8 SPUs (maximum ∼100 per blade, limited by the interposer stitching). The Cryo-DRAM capacity per blade is 2 TB accounting for 8×8 quad die packages of LPPDDRx/DDRx bonded on a interposer (77K) similar to the `Si` interposer housing SPU arrays. Every SPU can access the shared DRAM space at a bandwidth of 0.47



Chip-to-Chip link (Intra Blade communication)					
Size	Area	Bump pitch	Bump diameter	Total bumps	Total bandwidth
mm	*mm*^{2}	*mm*	*mm*	#	*TBps*
12	144	0.03	0.01	4.40E+04	73.34

Silicon 4K interposer					
Size	Area	Bump pitch	Bump diameter	Total bumps	Total bandwidth
mm	*mm*^{2}	*mm*	*mm*	#	*PBps*
120	14400	0.03	0.01	4.40E+06	7.33

System specifications for SCD blade	
Parameter	Baseline Value
Peak compute throughput per SPU	2.45PFLOPs (*Sparse*)
No. of SPUs	64 (*8 x 8*)
SPU L1 D-cache capacity (Private)	24MB (*4 HD JSRAM stacks in SPU*)
Shared L2 Cache capacity	3.375 GB (*16 HD JSRAM stacks in SNU*)
Avg. Main Memory bandwidth per SPU	~0.47TBps (*30TBps for 64 SPUs*)
Cryo-DRAM capacity	2TB
Bi-directional Main Memory bandwidth	30TBps
Avg. Cryo-DRAM access latency (RD/WR)	30ns
Intra-blade reduction latency	60ns
Max SPU-to-SPU bandwidth	~73TBps

Fig. 3: *(a) Physical representation of the SPU and SNU stacks (b) Network topology and switch design (c) Baseline SCD Blade specifications for explorations (d) Physical representation of the full SCD blade.*

TBps. The entire 8×8 blade can sustain a cumulative bi-directional main memory bandwidth of 30 TBps at an average read/write latency of 30 ns. Due to the unique nature of superconducting interconnects and packaging technology, we can scale both the effective DRAM and network BW as we scale the number of SPUs. The above specifications serve as the baseline for the explorations on LLM training and inference via our performance model.

V. Analytical Performance Modelling

In this section, we describe the principles of the performance modeling approach (Fig. 4). We adapt an existing and validated analytical modeling framework, Optimus [32], to predict the performance impact of the SCD system architecture. The measured SCD technology data (section **II**) is used to create the basic design blocks, which are further used to derive the high-level system architectural parameters, shown in Fig. 3c.

Given a workload (e.g., LLM training/inference), the modeling framework, Optimus, ingests a detailed task graph with the LLM model parameters such as number of layers, attention heads, hidden dimension, input/output shapes, sequence length, batch-size, working precision, etc. Using the above parameters and a chosen combination of parallelization strategies, such as data parallelism (DP), tensor model parallelism (TP) and pipeline parallel (PP), the workload is mapped onto the underlying system architecture. In DP, the model is replicated and data is sharded; in TP the model is sharded and data is replicated and in PP the model is sharded layer wise and data is divided into small chunks to inject in a pipeline fashion. Each parallelism option comes with its own advantages and trade offs [33]. At its core, Optimus relies

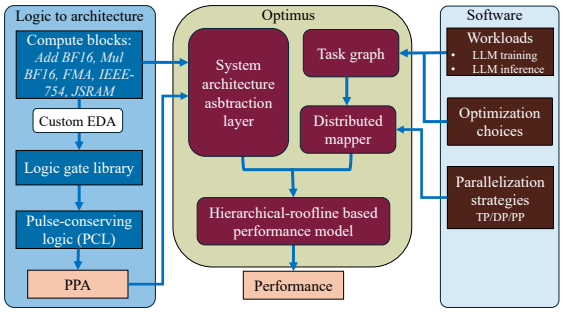


Fig. 4: *End-to-end performance analysis from logic design to system architecture, including mapping workload using task graphs for performance prediction for SCD system.*

on a hierarchical roofline model for a single accelerator to determine if a given kernel in the task graph is compute or memory (on-chip/off-chip) bound. For compute bound kernels, the execution time is primarily determined by the compute throughput, while for memory-bound kernels, it is dominated by the data transfer time from the respective memory level. For a given system architecture and workload, we assess the most

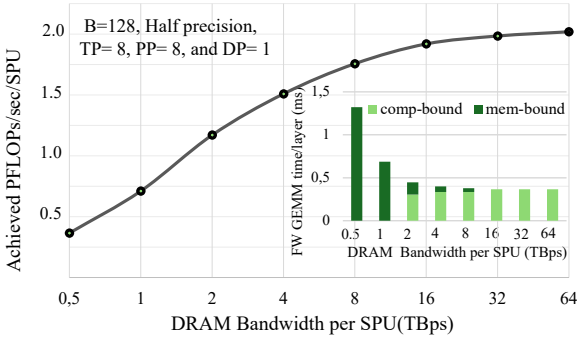


Fig. 5: *Impact of DRAM bandwidth (BW) per SPU in SCD system on the achieved throughput (PFLOPs/SPU) per batch for GPT3-76B model training. Inset shows the time breakdown of the memory versus compute-bound GEMMs per layer in the forward pass.*

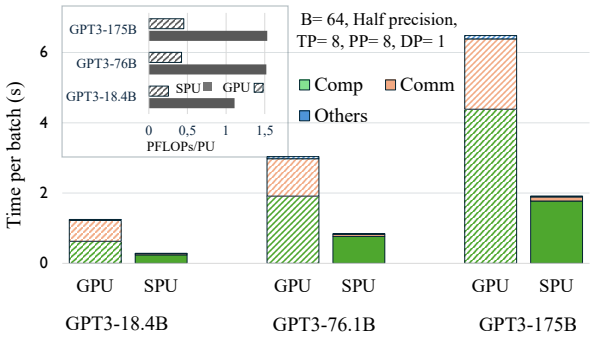


Fig. 6: *Processing time per batch with a break up of compute, communication, and remaining time for three different GPT models in training for GPU (H100) (patterned) and SPU (solid). The inset shows the corresponding throughput in PFLOPs/processing unit.*

optimal mapping, reducing communication overhead [34]. The performance modeling framework has been validated against both training and inference on different GPUs, for different parallelization strategies over multiple LLM models [32].

VI. Results and Discussion

Here, we present the projected performance numbers using the adapted analytical modeling framework. As described in the previous sections, the SCD system has a huge memory and network bandwidth advantage compared to contemporary systems, which memory bound applications can exploit. To test this hypotheses, we model both LLM training and inference on the SCD system and project its performance in comparison with equivalent number of GPUs (H100s: peak throughput of 0.9895 PFLOPs, DRAM bandwidth of 3.35 TBps).

First, we look at LLM training to investigate the potential speed-up of SCD systems. The majority of the matrix multiplication (GEMM) kernels in LLM training are compute-bound on GPUs; however, the GEMMs involved in the attention computation and non-linear operations are typically memory bandwidth bound [35] Fig. 5, gives the achieved throughput in PFLOPs/SPU for training GPT3-(76B model) on 64 SPUs, considering fixed model parallelism with a TP degree of 8 and a PP degree of 8. We perform an exploration of the total effective bandwidth available to an SPU. The minimum memory bandwidth available to each SPU is 0.47 TBps as discussed in Sec. **IV**-C (Fig. 3c). We then perform a sweep for the memory bandwidth available per SPU from0.5 TBps to 64 TBps and estimate the achieved throughput. As seen in Fig. 5, the performance scales with bandwidth monotonically and tend to saturate around 2 PFLOPs/SPU at 16 TBps of bandwidth. To further analyze this, we look at the compute versus memory-boundedness of GEMMs involved in the forward pass as a function of the memory bandwidth. As seen in the inset, at lower bandwidths, most of the GEMM time is spent in memory-bound kernels,and thus performance can improve when bandwidth is increased. We observe a gradual crossover from a memory-bound scenario to compute-bound one for a memory bandwidth $\geq$16 TBps per SPU. The modest improvement in the performance beyond 16 TBps is due to

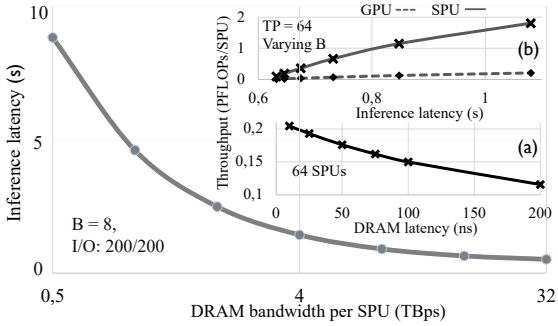


Fig. 7: *Inference latency(s) versus DRAM bandwidth for Llama-405B (precision= bf16, I/O: 200/200 tokens, DRAM latency= 30ns). Inset (a) achieved throughput versus DRAM latency (B = 8). Inset (b): inference latency versus achieved throughput while B is varied from 4 to 128. For the insets, DRAM bandwidth per SPU is set at 16 TBps.*

the remaining memory-bound operations in the task-graph, for instance, non-linear operations like, softmax, layer-norm etc.

Key takeaway: *Achieved throughput grows with available bandwidth initially and slowly saturates as most kernels turn compute-bound vs being memory-bound.*

We now compare the performance of training different GPT models on SPUs vs the same number of GPUs. Fig. 6, shows the break up of compute, communication and others (pipeline bubble + weight update time) for three different models (TP=8, PP=8, DP=1, B=64 and bf16 precision). The total bandwidth available to each SPU in a blade (64 SPUs) is set at 16 TBps. In all the three cases, we observe that the SCD system is significantly faster than the GPU system (64 H100s). The speed-up varies from $3.5\times$ – $4.4\times$ for this particular set up and model sizes. As expected, the SCD system enables both faster compute and faster communication – the primary gain coming from faster data movement. The inset exhibits the achieved throughput in terms of PFLOPs per processing units for GPU and SCD systems with the same experimental set up. We should note that in this case the maximal achieved throughput is around 1.5 PFLOPs/SPU while in the previous set up it is around 2 PFLOPs/SPU. This is primarily due the increased batch size (64 versus 128), that leads to improved throughput for both GPU and SPU. However, the SCD systems benefit more where the data transfer overhead is larger.

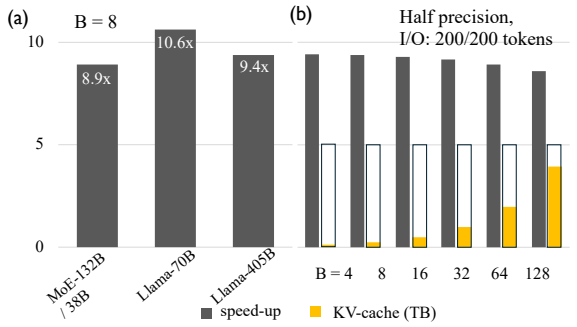


Fig. 8: *(a) Single SCD blade (64 SPUs) inference speed-up (vs the same number of GPUs) for different LLM models. (b) Impact of batch size on the speed-up and the respective KV-cache size for Llamma-405B. Height of the open bar denote the total memory capacity of 64 GPUs for reference (=64×80 GB). Memory-BW/SPU= 16TBps.*

Key takeaway: *SCD system promises to offer significantly higher throughput compared to contemporary GPU systems across different LLM models for training due to higher cumulative memory and interconnect bandwidth.*

We now move to LLM inference. Inference is known to be a memory-bound workload due to minimal data reuse in the involved kernels [32] and thus, we expect SCD systems to perform better in this case vs LLM training. In the following study, we vary the effective DRAM bandwidth per SPU from 0.5 TBps to 32 TBps to see its impact on the inference latency. Fig. 7 shows how the inference latency goes down as we increase the bandwidth seen by each SPU. Going from 0.5 TBps (8.8s) to 32 TBps (0.52s), we see a speed up of $17\times$, thus confirming the importance of memory bandwidth in inference. The performance scaling tend to saturate beyond 8 TBps since we start hitting the DRAM latency bound limit. Here, we consider the Llama-405B model with a batch size of 8, bf16 precision, summarization and prediction of 200 tokens; DRAM latency is set to 30ns. Tensor parallelism is utilized here, where the number of SPUs is same as the TP degree.

Since the collective DRAM bandwidth to a blade can be very high for the given SCD architecture, we analyze the sensitivity of the achieved throughput with respect to DRAM latency keeping the effective DRAM bandwidth for each SPU at 16 TBps. Inset (a) in Fig. 7 shows that the achieved PFLOPs/SPU goes down almost linearly with increasing DRAM latency (varied from 10ns– 200ns). We see that lower DRAM latency is crucial to achieve high performance in the case of inference. Inset (b) shows the variation of achieved throughput with inference latency (H100 versus SPU) as we vary the batch size from 4 to 128 (DRAM latency is set at 30ns). As the batch size is increased, the achieved PFLOPs/SPU increases along with the inference latency– this trade off helps determining the number of queries that can be batched without sacrificing user experience.

Key takeaway: *LLM inference performance continue to scale with available DRAM bandwidth when DRAM latency is low.*

Next, we investigate the speed-up in inference across different LLM models (MoE-132B/38B, Llama-70B, Llama-405B) for different batch sizes at a single SCD blade level compared to equivalent number of GPUs (H100s). We see a massive speed-up of $9\times$-$11\times$ depending on the LLM model (see Fig. 8a, DRAM latency is set to 30 ns and bandwidth to 16 TBps). SCD performs best for Llama-70B among these models since the fraction of communication overhead is maximal in this case. For the Mixture-of-Experts model, the speed-up is a bit lower than that for Llama models since only 4 experts are active among 16. Hence, the communication overhead is less compared to the other models. The right panel shows the robustness of the speed-up with respect to the batch size for Llama-405B model. In addition, we also show the required KV-cache size as a function of batch size. We see that, at batch size = 128, the KV-cache size is close to the maximum memory capacity of 64 GPUs, thus potentially limiting scaling up of batch sizes further for GPU systems. The speedup of the SCD system gradually decreases for large batch sizes since the compute to data-transfer ratio goes up.

Key takeaway: *The SCD system offers even more performant execution of LLM inference compared to training on GPU systems (due to memory boundedness of inference). The speed-up is robust across different LLM models and batch sizes.*

Lastly, we explore the potential use of the large L2 caches (~ 4.19 GB). The required kv-cache size for the popular llama models are, llama2-7B: 2 GB, llama2-13B: 3 GB and llama2-70B: 10 GB. Thus, one can possibly fit the entire kv-cache of the two smaller llama models (7B and 13B) onto the L2 cache of the SCD architecture. For GPU systems, the cache size is in the order of MBs (e.g., 50 MB for H100) and cannot be shared across GPUs. Thus, the GEMMs/GEMVs involving *Key* and *Value* can be accelerated by making them L2 or compute bound from a traditional DRAM bound case. Since L2 cache also comes with a huge bandwidth jump compared to DRAM, one can exploit it by properly managing the memory. Our early estimates suggest a speed-up of $\sim 2-4\times$ for the relevant GEMMs/GEMVs (depending on the software overhead of launching the kernels) if such a scheme is implemented. Since we do not have a clear estimate of the software overhead in those cases, it is hard to model the impact accurately.

VII. Conclusions and Future Outlook

In this paper[1], we present a comprehensive study on performance modeling of SCD technology for LLMs training and inference using a bottom-up approach to derive high level system architectural parameters based on superconducting process and device level data. We show that the significantly high memory and interconnect bandwidth of the SCD technology can substantially accelerate both LLM training and inference, serving as a very promising solution to the memory and interconnect wall problems. Although, we limit this study to projecting the performance of a single SCD blade, we expect the performance to scale with the number of blades. Another direction could be to look at the impact of huge JSRAM capacity on LLM inference exploiting its massive bandwidth and negligible latency. Such unusual SRAM capacity will further lead to new mapping and memory management options.

[1]This work has been funded in part by imec INVEST+ fund, Oceola County Florida USA, and NSF Central Florida Semiconductor Engine.

REFERENCES

[1] M. . Company, "What's the future of generative AI? An early view in 15 charts," https://www.mckinsey.com/featured-insights/mckinsey-explainers/whats-the-future-of-generative-ai-an-early-view-in-15-charts, 2023.

[2] A. Solon. (2023) Generative ai boosts demand for compute resources: Altman solon analysis. [Online]. Available: https://www.hpcwire.com/off-the-wire/generative-ai-boosts-demand-for-compute-resources-altman-solon-analysis/

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[4] J. Booth, "Energy efficiency scaling for 2 decades (ees2) roadmap for computing," in *28th Annual IEEE High Performance Extreme Computing Virtual Conference, Virtual, CO, US*. IEEE, 2024-07-14 00:07:00 2024.

[5] D. Patterson, J. Gonzalez, U. Hölzle, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. R. So, M. Texier, and J. Dean, "The carbon footprint of machine learning training will plateau, then shrink," *Computer*, vol. 55, no. 7, pp. 18–28, 2022.

[6] S. Samsi, D. Zhao, J. McDonald, B. Li, A. Michaleas, M. Jones, W. Bergeron, J. Kepner, D. Tiwari, and V. Gadepally, "From words to watts: Benchmarking the energy costs of large language model inference," in *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, 2023, pp. 1–9.

[7] A. Gholami, Z. Yao, S. Kim, C. Hooper, M. W. Mahoney, and K. Keutzer, "AI and Memory Wall," *IEEE Micro*, vol. 44, no. 3, pp. 33–39, 2024.

[8] N. Asia. (2023) The great nanometer chip race. [Online]. Available: https://asia.nikkei.com/Spotlight/The-Big-Story/The-great-nanometer-chip-race

[9] S. A. McKee, "Reflections on the memory wall," in *Proceedings of the 1st Conference on Computing Frontiers*, ser. CF '04. ACM, 2004, p. 162. [Online]. Available: https://doi.org/10.1145/977091.977115

[10] K. Banerjee and A. Mehrotra, "Global (interconnect) warming," *IEEE Circuits and Devices Magazine*, vol. 17, no. 5, pp. 16–32, 2001.

[11] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *CoRR*, vol. abs/2001.08361, 2020. [Online]. Available: https://arxiv.org/abs/2001.08361

[12] T. Josephsen and Q. Herr, "Pulse conserving logic," *poster presented at Applied Superconducting Conference*, 2022.

[13] M. Vesely, P. Tschirhart, B. Konigsburg, P. Farrell, S. Rahman, and E. Massaad, "A pipelined superconducting 16-bit cpu design," *Applied Superconducting Conference*, 2018.

[14] J. Egan, M. Nielsen, J. Strong, V. Talanov, E. Rudman, B. Song, Q. Herr, and A. Herr, "Synchronous chip-to-chip communication with a multi-chip resonator clock distribution network*," *Superconductor Science and Technology*, vol. 35, no. 10, p. 105010, sep 2022. [Online]. Available: https://dx.doi.org/10.1088/1361-6668/ac8e38

[15] R. Bairamkulov and G. De Micheli, "Superconductive electronics: A 25-year review [feature]," *IEEE Circuits and Systems Magazine*, vol. 24, no. 2, pp. 16–33, 2024.

[16] C. L. Ayala, T. Tanaka, R. Saito, M. Nozoe, N. Takeuchi, and N. Yoshikawa, "Mana: A monolithic adiabatic integration architecture microprocessor using 1.4-zj/op unshunted superconductor josephson junction devices," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 4, pp. 1152–1165, 2020.

[17] Q. Herr, T. Josephsen, and A. Herr, "Superconducting pulse conserving logic and josephson-sram," *Applied Physics Letters*, vol. 122, no. 18, 2023.

[18] J. Egan, M. Nielsen, J. Strong, V. Talanov, E. Rudman, B. Song, Q. Herr, and A. Herr, "Synchronous chip-to-chip communication with a multi-chip resonator clock distribution network," *Superconductor Science and Technology*, vol. 35, no. 10, p. 105010, 2022.

[19] H. Dai, C. Kegerreis, D. W. Gamage, J. Egan, M. Nielsen, Y. Chen, D. Tuckerman, S. E. Peek, B. Yelamanchili, M. Hamilton *et al.*, "Isochronous data link across a superconducting nb flex cable with 5 femtojoules per bit," *Superconductor Science and Technology*, vol. 35, no. 4, p. 045014, 2022.

[20] A. Herr and Q. Herr, "A data center in a shoebox: Imec's plan to use superconductors to shrink computers," *IEEE Spectrum*, vol. 61, no. 6, pp. 37–41, 2024.

[21] S. K. Tolpygo, V. Bolkhovsky, S. Zarr, T. J. Weir, A. Wynn, A. L. Day, L. M. Johnson, and M. A. Gouker, "Properties of unshunted and resistively shunted nb/alox-al/nb josephson junctions with critical current densities from 0.1 to 1 ma/μm2," *IEEE Transactions on Applied Superconductivity*, vol. 27, no. 4, pp. 1–15, 2017.

[22] S. K. Tolpygo, V. Bolkhovsky, R. Rastogi, S. Zarr, A. L. Day, E. Golden, T. J. Weir, A. Wynn, and L. M. Johnson, "Advanced fabrication processes for superconductor electronics: Current status and new developments," *IEEE Transactions on Applied Superconductivity*, vol. 29, no. 5, pp. 1–13, 2019.

[23] J. K. Sergey Tolpygo, Thomas Schurig, "Processing and manufacture of josephson junctions: Low-t c," in *Handbook of Superconductivity*. CRC Press, 2021.

[24] S. Steinhauer, L. Yang, S. Gyger, T. Lettner, C. Errando-Herranz, K. D. Jöns, M. A. Baghban, K. Gallo, J. Zichi, and V. Zwiller, "NbTiN thin films for superconducting photon detectors on photonic and two-dimensional materials," *Applied Physics Letters*, vol. 116, no. 17, p. 171101, 04 2020. [Online]. Available: https://doi.org/10.1063/1.5143986

[25] L. Zhang, W. Peng, L. X. You, and Z. Wang, "Superconducting properties and chemical composition of NbTiN thin films with different thickness," *Applied Physics Letters*, vol. 107, no. 12, p. 122603, 09 2015. [Online]. Available: https://doi.org/10.1063/1.4931943

[26] V. Vega-Gonzalez, D. Radisic, S. Choudhury, D. Tierno, A. Thiam, D. Batuk, G. Martinez, F. Seidel, S. Decoster, S. Kundu, D. Tsvetanova, A. Peter, H. De Coster, A. Sepulveda-Marquez, E. Altamirano-Sanchez, B. Chan, Y. Drissi, Y. Sherazi, J. Uk-Lee, I. Ciofi, G. Murdoch, N. Nagesh, G. Hellings, J. Ryckaert, S. Biesemans, E. D. Litta, N. Horiguchi, S. Park, and Z. Tökei, "Semi-damascene integration of a 2-layer mol vhv scaling booster to enable 4-track standard cells," in *2022 International Electron Devices Meeting (IEDM)*, 2022, pp. 23.2.1–23.2.4.

[27] G. Murdoch, Z. Tokei, S. Paolillo, O. V. Pedreira, K. Vanstreels, and C. J. Wilson, "Semidamascene interconnects for 2nm node and beyond," in *2020 IEEE International Interconnect Technology Conference (IITC)*, 2020, pp. 4–6.

[28] A. Herr, Q. Herr, S. Brebels, M.-S. Kim, A. Pokhrel, B. Hodges, T. Josephsen, S. ONeal, R. Bai, K. Nowack *et al.*, "Scaling nbtin-based ac-powered josephson digital to 400m devices/cm ^{2}," *arXiv preprint arXiv:2303.16792*, 2023.

[29] F. Wang, T. Vogelsang, B. Haukness, and S. C. Magee, "Dram retention at cryogenic temperatures," in *2018 IEEE International Memory Workshop (IMW)*, 2018, pp. 1–4.

[30] G.-H. Lee, D. Min, I. Byun, and J. Kim, "Cryogenic computer architecture modeling with memory-side case studies," in *2019 ACM/IEEE 46th Annual International Symposium on Computer Architecture (ISCA)*, 2019, pp. 774–787.

[31] F. Ware, L. Gopalakrishnan, E. Linstadt, S. A. McKee, T. Vogelsang, K. L. Wright, C. Hampel, and G. Bronner, "Do superconducting processors really need cryogenic memories? the case for cold dram," in *Proceedings of the International Symposium on Memory Systems*, ser. MEMSYS '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 183–188. [Online]. Available: https://doi.org/10.1145/3132402.3132424

[32] J. Kundu, W. Guo, A. BanaGozar, U. De Alwis, S. Sengupta, P. Gupta, and A. Mallik, "Performance modeling and workload analysis of distributed large language model training and inference," in *2024 IEEE International Symposium on Workload Characterization (IISWC)*, 2024, pp. 57–67.

[33] D. Moolchandani, J. Kundu, F. Ruelens, P. Vrancx, T. Evenblij, and M. Perumkunnil, "Amped: An analytical model for performance in distributed training of transformers," in *2023 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 2023, pp. 306–315.

[34] D. Narayanan, M. Shoeybi, J. Casper, P. LeGresley, M. Patwary, V. Korthikanti, D. Vainbrand, P. Kashinkunti, J. Bernauer, B. Catanzaro *et al.*, "Efficient large-scale language model training on gpu clusters using megatron-lm," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–15.

[35] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, "Flashattention: Fast and memory-efficient exact attention with io-awareness," 2022. [Online]. Available: https://arxiv.org/abs/2205.14135