The Advantage of the GPU as a Real-Time AI Accelerator*

Joshua Bakita and James H. Anderson

Department of Computer Science, University of North Carolina at Chapel Hill.

Contributing authors: jbakita@cs.unc.edu; anderson@cs.unc.edu;

Abstract

Integrating AI into real-world systems such as autonomous vehicles or interactive assistants requires the use of compute accelerators. Traditional processors such as x86 or ARM CPUs are insufficient. Unfortunately, real-world systems have responsiveness requirements, and research is underdeveloped on guaranteeing such responsiveness for accelerator-using systems. One constraint has been uncertainty about what sort of accelerator is best for such systems. In this paper, we argue that researchers should focus on the GPU as the accelerator of choice for embedded real-time AI workloads. We argue that GPUs are already being widely adopted, provide leading compute density, and are architecturally well-suited for real-world, real-time systems.

Keywords: real-time systems, compute acceleration, graphics processing units

1 Introduction

As AI tasks demand ever-more processing power, compute accelerators—*e.g.*, a Graphics Processing Unit (GPU), Tensor Processing Unit (TPU), or other matrix processor—have become critical to completing AI tasks within reasonable timescales.

*This version of the article has been accepted for publication, after peer review (when applicable) but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/s11241-025-09447-7. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms.

In systems with real-world interactivity requirements, such as autonomous vehicles or intelligent assistants, embedded AI tasks must complete within bounded periods of time (*e.g.*, in an autonomous vehicle, detect obstacles within 200 ms of obtaining a camera frame). The field of *real-time systems* is concerned with making such response-time guarantees, and these guarantees depend on assumptions about the hardware to be used, such as the choice of compute accelerator.

Unfortunately, ambiguity about the best accelerator to use for real-time AI tasks has led researchers to make conservative assumptions about accelerators. This causes problems when making response-time guarantees, as conservative assumptions—*e.g.*, assuming that accelerators do not support preemption—can make it impossible to provide practical response-time guarantees. For example, if a best-effort task cannot be preempted by a time-critical one, the critical task may be delayed indefinitely, leading to an unbounded response time. This problem could be avoided by the use of a GPU (as GPUs support preemption), but can we assume that?

We argue that GPUs are and will be the accelerator of choice for enabling AI tasks in real-time, embedded systems. We base our argument on the current rapid adoption of GPUs in real-world systems, and on architectural reasons including compute density, flexibility, and real-time suitability. By focusing on a specific platform, the real-time systems community could use more specific, platform-derived assumptions, and utilize those additional capabilities to craft more broadly-applicable schedulers and tighter response-time analyses.

2 GPUs are *the* AI Accelerator

Adoption. Despite many vendors entering the accelerator market in recent years, this has not meaningfully slowed the adoption of GPUs—especially NVIDIA GPUs—as the accelerator of choice. On a revenue basis, NVIDIA outsells Intel, AMD, ARM, Qualcomm, and NXP *combined*.[1] In other words, NVIDIA outsells the entire x86 CPU market, the entire FPGA market, and much of the ARM ecosystem *combined*. This dominance is not limited to cloud computing. From the perspective of high-performance embedded systems, *every* company approved to operate autonomous vehicles without a safety driver in California—Waymo, Zoox, WeRide, Nuro, Baidu, and AutoX—use NVIDIA GPUs in their vehicles (Herger 2024; WeRide.ai 2022; NVIDIA 2024a,b, 2023, 2021; Baidu 2024).

This market dominance shapes what AI researchers are building models to run on, what systems researchers are considering to optimize, and what is available on the market. Historically, the most widely available platform has become the defacto standard, even when other, arguably better, options were available (*e.g.*, the history of x86 CPUs). However, GPUs have fundamental strengths beyond momentum.

Flexibility. NVIDIA GPUs are general-purpose accelerators, allowing them to be used for a diversity of compute-intensive tasks that a complex embedded system may need to perform (*e.g.*, a classical graph-search-based planner alongside a

[1] As of each company's last-available quarterly revenue on February 27th, 2025; NVIDIA: $39B, Intel: $14B, AMD: $7.6B, ARM: $1.0B, Qualcomm: $12B, and NXP: $3.2B.

Table 1 Compute Density of Commercial Off-The-Shelf Processors

	GPU	CPU	Special-Purpose Accelerator		
Model	NVIDIA H100 SXM	AMD EPYC 9755	Cerebras WSE-3	Tesla FSD-2	Google TPU v5p
Date Available	Mar 2023	Oct 2024	Mar 2024	Feb 2023	Oct 2023
BF16 TFLOP/s*	989	22†	125,000‡	61§	459
Die Size (mm^2)	814	1,564**	46,225	361**	729**
GFLOP/s/mm^2	**1,216**	**14**	**134**	**168**	**629**

*Trillions of dense 16-bit floating-point fused-multiply-add (FMA) operations per second.
†EPYC FLOP/s are based off a peak BF16 FMA throughput of 64/cycle per core when using AVX-512 (Yee 2024).
‡Cerebras only publicly quotes sparse FLOP/s. Their dense FLOP/s (shown here) are one-tenth the sparse rate (Lie 2023).
§For FSD-2, BF16 is unsupported; this is half the INT8 rate (Patel and Kostovic 2023).
**Size estimated, based off available die shots (AMD 2025; green 2023; Vengineer 2024).

neural-network-based perception system in an autonomous vehicle). Even if a system's compute-intensive workloads are entirely AI-based, the GPU's general-purpose capability allows updating the AI system over its lifetime (*e.g.*, switching from convolutional- to transformer-based neural networks). Special-purpose accelerators may not be capable of such a switch.

Density. GPUs are efficient at turning die area into compute capacity. Such *density* is critical to real-world systems. As the die is the single most expensive part of a processor, its density shapes the cost and compute capability of the system. Table 1 compares the throughput (TFLOP/s), die size (mm^2), and density (throughput per unit of die area; bold) for several concurrently released high-end processors. Note how none of the non-GPU processors match the density of NVIDIA's H100 GPU. The CPU has especially poor density due to the large caches that must be used to hide memory latency; accelerators that hide latency in other ways can use more of their die for compute units. Note that the only special-purpose AI accelerator we find to be density-competitive with the GPU is Google's TPU. Unfortunately, this is only available as a datacenter rack-mount unit—even Waymo (Google's self-driving car division) uses NVIDIA GPUs instead of Google's TPUs (Herger 2024).

This underwhelming set of non-GPU accelerator options is unlikely to change, as it has historically been difficult to deliver on lofty goals when developing special-purpose hardware. Consider the story of ray-tracing at NVIDIA. NVIDIA began a project in 2013 to develop a special-purpose unit within NVIDIA GPUs to accelerate ray-tracing operations by 100×. After five years of work, this project successfully shipped as the "ray-tracing cores" in NVIDIA's Turing-generation GPUs. They mostly met their goal—ray-tracing on the special-purpose unit was 96× faster than it was in software *at the time they started*. As the general-purpose GPU compute cores increased in speed concurrently with the development of the special-purpose units, the ray-tracing cores were only 6-8× faster than software by the time they shipped (Luebke 2022b). If a company with NVIDIA's resources and experience succeeded at a 96× absolute improvement that shipped as only a 6× relative improvement, the smaller companies developing accelerators today seem likely to suffer the same fate. Since the speed of

general-purpose computation on the GPU is a moving target, a promise of a 10× speedup today may be no speedup at all once shipping in volume.

3 GPU Designs Are Suitable for Real-Time Systems

Beyond their adoption, flexibility, and density, GPUs are suitable for real-time systems in both their architectural design and native scheduling features. These features are not unique to GPUs, but are important aids to their timing predictability.

Architecturally, GPUs have highly parallel memory subsystems. This means that a memory request from any GPU core is likely to have an exclusive path through the memory hierarchy (relative to any other GPU core). This reduces the chance of memory requests contending for on-chip resources—a highly beneficial property for a real-time system. Contention can slow computations unpredictably, making it hard to provide response-time bounds. Quantitatively, this parallelism comes across as memory bandwidth—3.4 TB/s on a GPU versus 614 GB/s on a CPU (for the representative models in Table 1). We speculate that this strength stems from the architectural underpinnings necessary to support the original use of GPUs in rendering, where it is essential to transfer a large amount of data (*e.g.*, textures, frames, models) very frequently.

The architectural benefits of the GPU extend into other areas. For example, GPUs are capable of fine-grained preemption (since 2016)—a feature motivated to lower latency for head-mounted displays (NVIDIA 2016). Or consider the GPU's ability to concurrently run multiple applications on different sets of cores, and its ability to directly interface with peripherals (both since 2013)—both features motivated to increase GPU utilization in high-performance computing clusters (NVIDIA 2025b,a). NVIDIA has not always been prompt to document or provide software APIs for these hardware features, but the capability exists (even if it is yet to be unlocked).

These strengths seem unlikely to vanish, as NVIDIA has stated that the general architecture of their GPUs has not changed in many years (Luebke 2022a), and other vendors appear to be designing their architectures by following NVIDIA's footsteps. While NVIDIA is notoriously reticent to share architectural and scheduling details broadly with the real-time systems community (and we strongly urge them to stop impairing their own platform in this way), we note that there have been successful collaborations where NVIDIA has privately shared details for papers (Capodieci et al. 2018) and scheduling research done under contract.

4 Looking Forward

To enable real-time AI task-systems, we need more broadly applicable schedulers and tighter response-time analysis. Without such work, AI designers bear an additional burden: they must simplify their tasks to fit into the mold of primitive response-time analysis, *e.g.*, by merging all AI tasks into one which exclusively uses the accelerator (Loquercio et al. 2021), or by developing AI tasks that jointly optimize response-times and accuracy (Lee et al. 2023). This burden could be lifted, and AI task research kept separate from scheduling research, by developing better schedulers and response-time

analyses. This is partially predicated on accurate, less-conservative assumptions about the accelerator to be used.

Whatever the choice, upcoming problems loom for large compute accelerators of any sort. Ever-larger dies are making the presence of manufacturing defects a near-certainty, and such defects can make different cores in an accelerator effectively operate at different speeds. This complicates scheduling and response-time analysis. To confront such problems, we need foundational response-time analyses to build on. For these analyses to be practical and useful, they must be built on accurate assumptions about accelerators; we argue that GPU behavior should be the basis of these refined assumptions. No other accelerator can provide the same combination of ubiquity, density, flexibility, and architectural suitability for real-time systems.

References

AMD: 5th Gen AMD EPYC Processor Architecture. Whitepaper (2025). https://www.amd.com/content/dam/amd/en/documents/epyc-business-docs/white-papers/5th-gen-amd-epyc-processor-architecture-white-paper.pdf Accessed 2025-07-04

Baidu: Apollo Open Source Autonomous Driving Platform README. GitHub (2024). https://github.com/ApolloAuto/apollo/blob/c48541b4/README.md Accessed 2025-02-27

Capodieci, N., Cavicchioli, R., Bertogna, M., Paramakuru, A.: Deadline-based scheduling for GPU with preemption support. In: Proceedings of the 39th IEEE Real-Time Systems Symposium, pp. 119–130 (2018)

green: @greentheonly Post on X (2023). https://x.com/greentheonly/status/1691905611452084635 Accessed 2025-07-04

Herger, M.: Waymo's $5.6 Billion Round and Details of the AI Used (2024). https://thelastdriverlicenseholder.com/2024/10/27/waymos-5-6-billion-round-and-details-of-the-ai-used/ Accessed 2025-02-27

Lie, S.: Cerebras architecture deep dive: First look inside the hardware/software co-design for deep learning. IEEE Micro **43**(3), 18–30 (2023) https://doi.org/10.1109/MM.2023.3256384

Loquercio, A., Kaufmann, E., Ranftl, R., Müller, M., Koltun, V., Scaramuzza, D.: Learning high-speed flight in the wild. Science Robotics **6**(59), 5810 (2021)

Luebke, D.: The Evolution of the GPU. Presentation at the Pixel-Planes@40 Colloquium, UNC Chapel Hill, 29 August (2022). https://youtu.be/iBTbUF7zVQw?t=8943 Accessed 2025-03-09

Luebke, D.: The Story of Ray Tracing at NVIDIA. Presentation at the Pixel-Planes@40 Colloquium, UNC Chapel Hill, 29 August (2022). https://youtu.be/

iBTbUF7zVQw?t=20438 Accessed 2025-02-27

Lee, J., Wang, P., Xu, R., Jain, S., Dasari, V., Weston, N., Li, Y., Bagchi, S., Chaterji, S.: Virtuoso: Energy- and latency-aware streamlining of streaming videos on systems-on-chips. ACM Transactions on Design Automation of Electronic Systems **28**(3) (2023)

NVIDIA: VRWorks - Context Priority (2016). https://developer.nvidia.com/vrworks/headset/contextpriority Accessed 2025-05-04

NVIDIA: AutoX Unveils Full Self-Driving System Powered by NVIDIA DRIVE. NVIDIA Blog (2021). https://blogs.nvidia.com/blog/autox-full-self-driving-nvidia-drive/ Accessed 2025-02-27

NVIDIA: Electric Dreams Charged Up at Auto Shanghai with NVIDIA DRIVE. NVIDIA Blog (2023). https://blogs.nvidia.com/blog/auto-shanghai-nvidia-drive/ Accessed 2025-02-27

NVIDIA: Nuro to License Its Autonomous Driving System. NVIDIA Blog (2024). https://blogs.nvidia.com/blog/nuro-driver/ Accessed 2025-02-27

NVIDIA: NVIDIA and Zoox Pave the Way for Autonomous Ride-Hailing. NVIDIA Blog (2024). https://blogs.nvidia.com/blog/nvidia-zoox-autonomous-ride-hailing/ Accessed 2025-02-27

NVIDIA: GPUDirect RDMA. Release 12.8 (2025). https://docs.nvidia.com/cuda/pdf/GPUDirect_RDMA.pdf

NVIDIA: Multi-Process Service. Release R570 (2025). https://docs.nvidia.com/deploy/pdf/CUDA_Multi_Process_Service_Overview.pdf

Patel, D., Kostovic, A.: Tesla AI Capacity Expansion – H100, Dojo D1, D2, HW 4.0, X.AI, Cloud Service Provider. SemiAnalysis (2023). https://semianalysis.com/2023/06/27/tesla-ai-capacity-expansion-h100/ Accessed 2025-07-04

Vengineer: Google TPU v5p was also a chiplet (2024). https://vengineer.hatenablog.com/entry/2024/04/18/080000 Accessed 2025-07-04

WeRide.ai: WeRide builds its next-gen autonomous driving solutions with NVIDIA DRIVE Orin-Powered Hyperion compute platform. Medium (2022). https://werideai.medium.com/weride-builds-its-next-gen-autonomous-driving-solutions-with-nvidia-drive-orin-powered-hyperion-35a9e843ec4 Accessed 2025-02-27

Yee, A.J.: Zen5's AVX512 Teardown + More... (2024). http://www.numberworld.org/blogs/2024_8_7_zen5_avx512_teardown/ Accessed 2025-07-04