OpenAI Jalapeño: Better Than Nvidia Blackwell

OpenAI's self-designed ASIC compared with Rubin, Jalapeño's TCO, throughput per MW, and spicy deets

BRYAN SHAN, MYRON XIE, JORDAN NANOS, AND 3 OTHERS

AUG 25, 2026 • PAID

64 5 Share

OpenAI has spent the past couple years quietly building "Jalapeño," an inference chip just announced at Hot Chips. Rumors of a successful tapeout had been swirling for a while. But now we have details. OpenAI invited us to look at their chip, go to their labs to check out how real it is, and benchmark it with our InferenceX suite.

In June, OpenAI unveiled the chip program in partnership with Broadcom, built from a blank slate exclusively for LLM inference. Design work began in the middle of 2024, going from initial team hiring to manufacturing tape-out in ~16 months, an extremely fast ASIC development cycle.

In general first generation chips are not competitive, but OpenAI bucks the trend by being industry leading and beating every Nvidia, AMD, and Google chip we have been able to test on multiple top open source models. OpenAI does this with extreme hardware software codesign. Surprisingly, OpenAI is not over specialization on any specific part of model inference, but instead by focusing on being a general chip that delivers high performance in all scenarios.

In this article, we will go into architectural details, software details and performance results for Jalapeño on InferenceX.

Source: OpenAI

A generalized inference chip

Everyone says that OpenAI's chip is specialized for OpenAI models, but that's wrong, OpenAI made a generalized chip for AI inference.

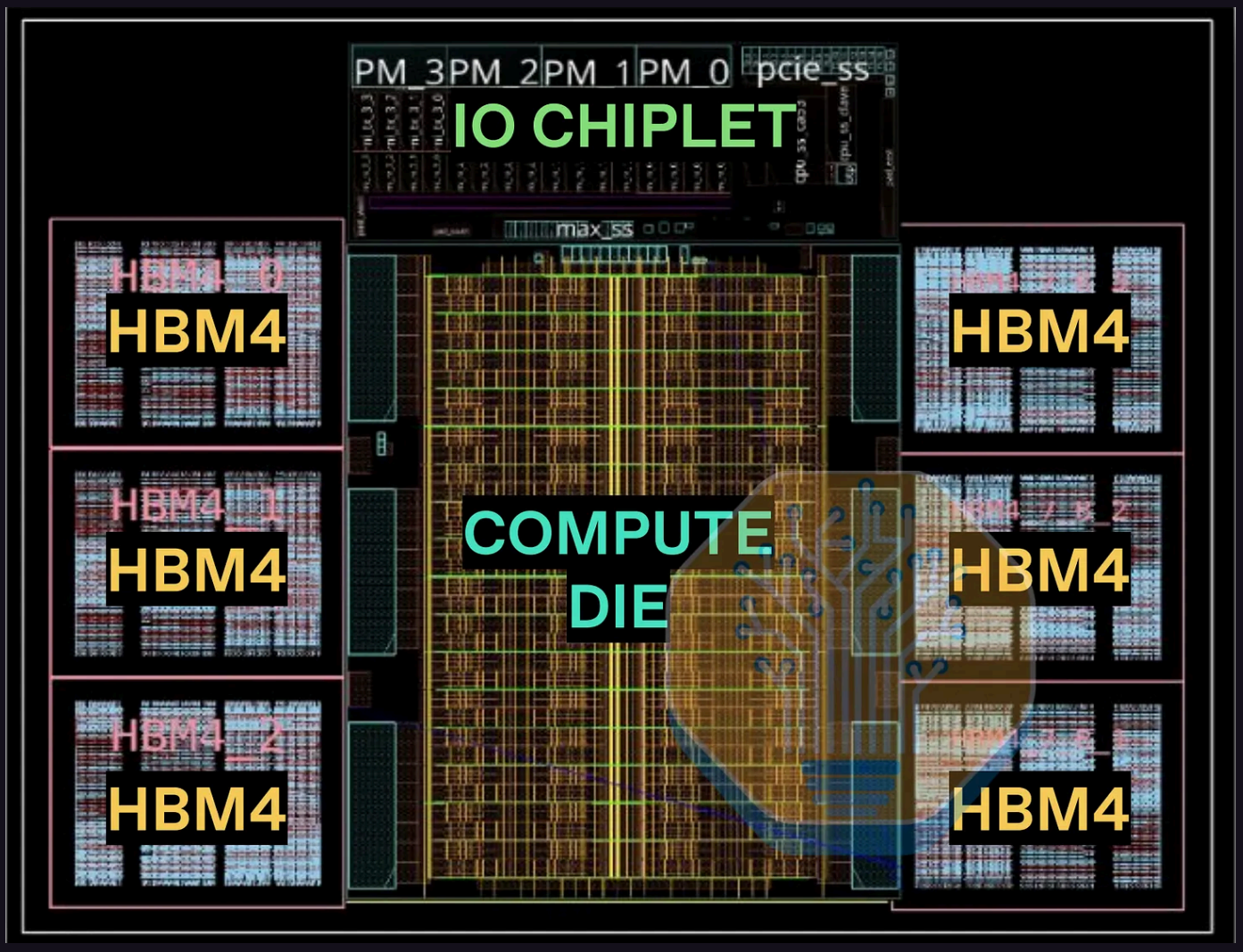
The timelines are insane. It shows that claims that use of AI is being used to accelerate chip design are real. Regardless of the quick timelines,Open AI spent a bunch of money, made pragmatic design decisions and their team is cracked, so this comes as no surprise.

Just looking at the specs, it is an immediate contender:

OpenAI Jalapeno Chip Specs						
	FP16 perf	FP8 perf	FP4 perf	HBM capacity	HBM bandwidth	TDP
H100	0.989 PFLOPS	1.979 PFLOPS	-	80 GB	3.35 TB/s	700W
H200	0.989 PFLOPS	1.979 PFLOPS	-	141 GB	4.80 TB/s	700W
MI355X	2.3 PFLOPS	4.6 PFLOPS	9.2 PFLOPS	288 GB	7.987 TB/s	1400W
GB200	2.5 PFLOPS	5 PFLOPS	10 PFLOPS	192 GB	8 TB/s	1200W
GB300	2.5 PFLOPS	5 PFLOPS	15 PFLOPS	288 GB	8 TB/s	1400W
Jalapeno	-	3.4 PFLOPS	13.4 PFLOPS	216 GiB	15.4 TB/s	700W
Rubin	4 PFLOPS	17.5 PFLOPS	35 PFLOPS	288 GB	20 TB/s	
MI450X	10 PFLOPS	20 PFLOPS	40 PFLOPS	432 GB	23.347 TB/s	

Source: SemiAnalysis

人在草木间



Source: OpenAI

A lot of the media coverage of this chip has followed a few throwaway comments from OpenAI that claim the chip will be optimized for their models in a way that other chips are not. This is wrong. Jalapeño is a generalized inference chip capable of running all sorts of models, and all sorts of workloads, including our benchmark InferenceX, where we ran the benchmark with OpenAI engineers in the lab. As a joke, OpenAI even showed us it running Doom, which was ported to their chip with just Codex prompts.

The following is our headline perf/W result, looking at token throughput per All-in utility MW. **Jalapeño smokes every other chip**. All this is done without Multi Token Prediction (MTP), while the other chips on the chart are the best performing configs of each respective SKU, all with MTP.

Source: SemiAnalysis

Jalapeño beats Blackwell on perf/W across almost all scenarios without being tuned for any specific point in the curve. It excels not only in low-latency scenarios but also in high-throughput scenarios. A more apples to apples comparison is against Single Token Prediction results, it knocks every competitor out of the water. At low concurrency scenarios, Jalapeño demonstrates remarkable interactivity, hitting over 700 tokens per sec per user at concurrency 1 on the DeepSeek R1 model.

Incredibly, this is all achieved with single-token prediction (STP), no speculative decoding and no prefill-decode disaggregation. In addition to DeepSeek R1, we also got to see some other models, including Kimi-K2.5 and GPT-OSS which ran at approximately 1,400 tok/sec/user. For all models, we confirmed that Jalapeño's GSM8k evals attained results on par with Nvidia chips.

Some caveats on this. First, all numbers are provided to us by OpenAI. We verified the InferenceX runs in person in the lab, but we did not run the full suite of InferenceX benchmarks nor have we seen AgentX results. AgentX is our preferred suite for comparing chip performance due to the datasets' long context and multi-turn characteristics that reflect the cache behavior of realistic production workflows. Frameworks that perform well on 8k1k may perform worse on AgentX as real production loads stress components like

AgentX - InferenceXv3: Does CUDA Moat Hold up in Agentic Inferencing?

CAM QUILICI, BRYAN SHAN, AND 5 OTHERS • 8 24

Read full story →

Second, we believe that comparison to Blackwell is somewhat incomplete and unfair. Jalapeño is really competing against chips like Rubin that also use HBM4. Vera Rubin systems are starting to ship to customers right now, while it will still be some time before OpenAI has anything beyond engineering samples of Jalapeño.

Thus, performance should really be compared against Rubin, not Blackwell, and in some sense we expect a custom chip like Jalapeño to outperform Blackwell. Vera Rubin NVL72 delivers 5.4x the perf/MW of GB200 NVL72 as we described in our article analyzing the NVIDIA performance claims in their launch with CoreWeave last month. We will compare Jalapeño to Vera Rubin's July performance figures later below.

Vera Rubin NVL72 vs GB200 NVL72? Inference TCO & Architecture Analysis

ALEC IBARRA, BRYAN SHAN, AND 6 OTHERS • 7 23

Read full story →

Third, the models being tested are not on the open frontier. NVIDIA and AMD have published results on larger models such as DeepSeek V4 Pro and Kimi K3, using AgentX. The larger the model and the more recent the release, the more complicated it is to bring up on a new chip. With that said the models OpenAI has working on Jalapeno aren't exactly small either.

Performance Analysis

OpenAI designs for perf/W. The reason is simple: OpenAI is currently limited by datacenter power, not by budget or floorspace, and thus tokens per MW is paramount. At Computex 2026, Jensen said that perf/W, reliability and long lifetime are the core features of future GPUs. To quote: "If you have 1 gigawatt of power, then throughput per watt is revenue". He also mentioned that choosing the wrong architecture just because the chips are cheaper doesn't make sense.

Source: Computex 2026 keynote

This was emphasized by Nvidia during the Vera talk at Hot Chips 2026 while showing the same revenue graph: “The data center is power limited today.” Power matters and drives revenue.

Operators cannot simply obtain more MW because adding GPUs and adding grid capacity happen on very different timescales. Datacenter power envelopes have constraints such as their utility interconnection, infrastructure, cooling capacity, and UPS/backup-generation design. Grid delays repeatedly outpace hardware and construction timelines, driving the need for BtM (behind-the-meter) power capacity: gas turbines and on-site generators built and located at the data center itself. This capacity sits behind the utility’s meter rather than being drawn from the public grid. It lets an operator power a facility without waiting on grid interconnection and utility upgrades, which is exactly why xAI’s Colossus 2 relies so heavily on BtM while its actual grid connection lags far behind. Find out more in our Energy model.

As we wrote in an X post, tok/s/MW reduces to tokens per joule since a watt is a joule per second. This makes tok/s/MW representative of a system’s efficiency and ability to convert energy into tokens.

Source: SemiAnalysis

On this front, even when compared with Rubin, Jalapeño wins. OpenAI's Jalapeño has STP output token throughput per MW surpassing Vera Rubin's MTP results that NVIDIA and CoreWeave published in July. It also far exceeds GB200's 2025 MTP results. As mentioned in our Vera Rubin article, VR was compared to 2025 GB200 results because that was a similar stage of early bring-up, and comparing to GB200 in 2025 holds software maturity

public Rubin numbers, and OpenAI taped out their chip after Rubin. Both OpenAI and Rubin are still immature thus performance will continue to rise.

Source: OpenAI, SemiAnalysis

On perf/TCO, Vera Rubin and Jalapeño are head-to-head, producing almost the same number of output tokens per $. However, as previously mentioned, **Jalapeño's results are obtained without speculative decoding** and Vera Rubin's results use speculative decoding. Speculative decoding adds a 3x+ improvement is cost per token. When speculative decoding is implemented on Jalapeño, this will mean Jalapeño will serve much lower cost tokens. Of course, part of this TCO advantage comes from trading Nvidia's high margins for Broadcom's lower (though still high) margins. But this is not all of it. For example, Meta and Microsoft's AI ASIC programs not getting off the ground despite being at it for much longer shows that cost is only one part of the equation. For Jalapeño's full TCO breakdown, see the SemiAnalysis AI Cloud TCO model.

Source: OpenAI, SemiAnalysis

Architecturally, OpenAI chose not to disaggregate prefill and decode (PD) across separate chip pools. The draft model and main model share the same chips and fabric, a design philosophy that trades some theoretical efficiency for practical operations. The motivation is that the workload mix changes over time, for example the ratio of input to cache write to cache read to output tokens has changed significantly as we have moved through the three eras of models (knowledge, reasoning, and agentic, as discussed in our recent article). Therefore, picking a fixed amount of heterogenous prefill silicon and decode silicon up front can lead to inefficiencies over time. OpenAI chooses a homogenous pool in this architecture and tries to make the chip perform well on everything.

And it does. On Kimi K2.5 (which Cursor Composer 2.5 is based on), Jalapeño reaches nearly 700tok/s/user and more than 9x the next best performing chip at 100tok/s/user.

Source: OpenAI, SemiAnalysis

On GPT-OSS, it's another bloodbath. Jalapeño's iso-interactivity throughput per MW is nearly double GB200's highest throughput point and more than 50x GB200's concurrency 1 point. The higher concurrency Jalapeño points use EP8.

Source: OpenAI, SemiAnalysis

These results are impressive! However, we have to nitpick: they're just 8k1k, a much easier workload to tune for, and there are no AgentX runs yet. As mentioned in our AgentX article, multiturn, long context workloads stress much more aspects of the serving stack, such as routers and prefix cache. Many more optimizations are needed to excel in agentic workloads. Read more about this in the AgentX article.

AgentX - InferenceXv3: Does CUDA Moat Hold up in Agentic Inferencing?

CAM QUILICI, BRYAN SHAN, AND 5 OTHERS · 8 24

Read full story →

Digging into the specs and architecture

All these results were gathered on the A0 stepping of Jalapeño, just 9 months into the program. But there is already a B0 stepping that is currently in the fab! B0 has optimizations that deliver roughly a 25% perf-per-watt improvement over the earlier A0 silicon. Specifically, the B0 stepping delivers 13.4 PFLOPs of MXFP4 on a single reticle-sized compute die that is manufactured on TSMC's N3P. This compares to 17.5 PFLOPs of

This is more respectable considering Jalapeño's TDP is only 700W compared to Rubin's at 900-1,150W per compute die. As Jalapeño is geared towards inference rather than training, it is understandable that OpenAI doesn't need to push TDPs higher to maximize FLOPs, but regardless the above shows that Jalapeño delivers respectable peak theoretical FLOPs.

When compared directly to other accelerators, Jalapeño has the highest HBM bandwidth per watt, and the highest FLOPs per watt, comparable to the 1,800W Rubin Max-Q configuration:

Source: SemiAnalysis

Off-package I/O is provided by an N3E I/O chiplet with 32 lanes of 800G SerDes, for the compute fabric, with 24 lanes (600GB/s) being used for local scale-up within the rack, and 8 lanes (200GB/s) for global scale-up which is the 2,048 XPU multi-rack domain. PCIe Gen 5 is used for system I/O to connect to the x86 host CPU.

Jalapeño will ship with HBM4, making this chip one of the relatively early adopters after Nvidia and AMD, even beating the established TPU and Trainium programs. As one of the key architectural principles behind Jalapeño is getting the most out of HBM bandwidth, settling for anything but the best HBM would run counter to that goal. This results in 15.4TB/s of memory bandwidth per package which bests all the other accelerators shipping that are using HBM3E. The 15.4TB/s bandwidth shows its HBM4 can hit 10Gbps pin speeds, which would give it a slight edge over the 9.6Gbps Nvidia is getting out of its HBM4 in Rubin. The HBM is likely provided by Samsung.

Source: OpenAI

OpenAI taped out Jalapeño in November 2025, or more specifically, this was a tape out of the CoWoS design, not just the top die silicon. Within 9 months of that Nov 2025 tapeout, and with only 3 months of bring-up on actual silicon, OpenAI has already delivered very good results with Jalapeño. This is all the more impressive as the team is starting from zero on the software stack.

Meanwhile, Rubin's CoWoS tape out was completed in October 2025, a month earlier, and yet the only early results we have seen are from CoreWeave's engineering samples. Nvidia has not let us test and release benchmarks in the same way that OpenAI has, indicating their chip software is still immature. The CUDA moat is potentially dead given how fast OpenAI can bring up new models on their silicon.

They are still far from optimized and we can see that generally Jalapeño has delivered better numbers. We don't think that Nvidia hardware is inferior, but more so that Jalapeño's software bring-up has progressed more quickly than Nvidia's. This speaks to the power of hardware/software co-design, which is the main area where a cracked frontier lab ASIC team can excel over more established merchant silicon players. Counterintuitively, starting from scratch may also have benefited OpenAI as it could make clean-sheet architectural decisions without worrying about backwards compatibility or older software versions.

While OpenAI has engineering samples of Jalapeño, production is currently scheduled to gradually ramp over 2027 with most of the output currently scheduled for the end of next year. For more details of unit volumes and ASPs, see the SemiAnalysis Accelerator Model.

Suffice to say, OpenAI Jalapeno is a real high volume ASIC.

When compared against Rubin's timeline, Jalapeño's is shockingly quick. As shown earlier, Jalapeño's results beat Rubin's despite Rubin's head start.

Source: SemiAnalysis

Jalapeno Architecture

Digging into the architecture now, the chip's matrix engine uses MXFP numerical formats and a weight stationary systolic array, similar to TPU. But when compared directly to TPU, it has support for smaller shapes / dimensions, meaning that it doesn't have weird performance cliffs that get exposed by awkwardly shaped matmuls on bigger systolics.

It also has 64-bit scalar cores and FP32/INT32 vector cores. OpenAI has also invested in redundancy at the tray level and has yield harvesting built in at the core and channel level. They claim that AI assistance in chip design delivered an 8% reduction in SIMD area and a 10% reduction in matrix-engine area during design. While they did not clarify the exact process/voltage/temperature (PVT) conditions, they also mentioned the AI-assisted blocks improved timing and power over the initial blocks.

The Jalapeño architecture design focuses on eliminating memory movement of KVCache and weights as well as fixed latencies and overheads in order to make it possible to get closer to the raw peak flops/bandwidth even for small batches or shapes as compared to other accelerators.

The cores and the HBM are divided into slices, where each core slice has a low-latency local view on its own slice of HBM. Synchronization between slices occurs on a high-bandwidth dedicated collective network. This minimal memory hierarchy already gives Jalapeño a big potential advantage over GPUs, where memory accesses must traverse a complicated memory system, resulting in large latencies that must be amortized or hidden over larger shapes.

This choice is feasible because with careful placement of weights and KVs, synchronization between cores can be restricted to limited, known high-bandwidth comms such as tensor-parallel communication that can be overlapped with compute.

Source: OpenAI

There is also an additional general NoC which is used for general comms and to access the scale-up network. In general OpenAI saves huge power and gets big performance gains with a simplified NOC and memory subsystem vs Nvidia and Google.

Source: OpenAI

At the core level, OpenAI describes an out-of-order (OoO) core with an L1 cache. This is a large divergence from the pattern we have seen in other accelerators, all of which instead use software-managed scratchpad commonly paired with some async DMA support. Again, the argument being made here is that this allows Jalapeño to avoid fixed overheads such as barrier latencies, which on other accelerators (such as GPUs) need to be hidden or

The tradeoff is that Jalapeño therefore relies on good prefetching to ensure timely arrivals of memory requests, which is less predictable and more difficult to reason about. However, with Codex in a good harness with access to detailed tracing, it is likely that finding the optimal kernel with the best prefetching for a given shape requires little human intervention. We think that is exactly what OpenAI has done to bring up DeepSeek R1, Kimi K2.5, and GPT-OSS so quickly.

The cores also have support for "small" matrix dimensions, which (depending on how small) should make it more general across different model and batch dimensions less sensitive to matrix dimension alignment, padding overhead, and tiling inefficiency. For instance, TPUs, Trainium, and Etched chips have very large systolic arrays which can require large batches or exactly-divisible model dimensions to avoid tiling inefficiencies.

With Jalapeño, OpenAI has focused on eliminating fixed latencies in the system to allow for as-close-to-roofline performance as possible across all areas of the pareto curve. In theory, this could give them advantages over the GPU at multiple operating points:

- Much better upper-bound performance on low-latency/small-batch inference, which on GPUs is limited by many fixed overheads such as launch latencies, barrier latencies, memory system latency
- Some potential to achieve closer to the hardware roofline even for large-batch or long-context

This comes with the caveat that even if the upper-bound performance is available in theory, it may be more difficult to realize that performance for real kernels. So it seems the approach is:

1. Design for the highest upper-bound performance across all workload shapes
2. Let Codex do the tedious work of finding the kernels that achieve that upper bound

Judging by the extremely fast turnaround for the OpenAI team to bring up InferenceX workloads on Jalapeño, we are optimistic about this approach.

If Jalapeño is a success, it will be a strong signal that the industry's obsession over programming models and perfect, universal compilers are invalidated by frontier AI models.

OpenAI writes Jalapeño kernels like assembly. Each kernel gets hand-tuned code, some running to ~3,000 lines, backed by correctness checks and a custom sanitizer. Early kernel work was human-in-the-loop rather than fully automated, but this shifted with a more scaled-up, internal version of Codex, one which OpenAI plans to pitch to enterprise customers. The internal serving engine is called "Teacup". Interestingly, **OpenAI had no internal implementation of MLA kernels until they benchmarked DeepSeek with InferenceX**. The ability for Codex to write functional and efficient kernels so quickly (without any of OpenAI's kernel engineering team intervening) shows the software pipeline's developmental ability.

OpenAI programs Jalapeño with Gluon. Gluon is OpenAI's kernel programming language. Built on top of Triton, Gluon preserves Triton's SPMD (Single Program Multiple Data) programming model, but it exposes **low-level programming abstractions**. For example, for NVIDIA GPUs, it offers APIs that map to PTX instructions, including MMA instructions, TMA instructions, mbarrier mechanisms, and many more. The most unique abstraction Gluon provides is **the layout**. Generally speaking, a layout defines a mapping between a hardware resource (e.g. 5th register of warp 9) and a tensor element (e.g. tensor element on row 6 column 7). Gluon's layout abstraction is based on Linear Layouts, a type of layout algebra OpenAI invented. Linear Layouts mathematically formalizes what a layout is and provides tools to operate on layouts. This enables many features, such as provably correct layout conversions and optimal memory swizzling.

In terms of Jalapeño's programming model, each Gluon program maps to a persistent thread. We believe this hints that Jalapeño suits the **persistent kernel programming pattern**, where each program executes on multiple tiles, and the programmer, rather than the hardware scheduler, assigns the work. OpenAI mentioned TensorInfo, an abstraction that explicitly encodes layouts. This is likely the set of layouts designed for Jalapeño, which will be powered by Linear Layouts. Finally, each core offers data prefetching and decoupled out-of-order units. We believe this indicates a data flow-like programming model, where programmers explicitly manage semaphores. For example, a user might program a wait on a prefetched data, which is locked behind a semaphore.

In a weird twist of fate, OpenAI models like GPT 5.6 Sol, which currently run on NVIDIA GPUs, have been used to design a chip that poses a real threat to the CUDA moat - NVIDIA's own GPUs are helping usher in their potential successor in real time.

tarball we get from the Jalapeño team has a world of wonders inside.

Source: OpenAI, SemiAnalysis

Not only did kernel performance improve, in the span of 8 days, the Jalapeño team enabled TP32, building on the previous TP8 configs and expanding beyond a single system to get a full rack-scale config running on a large model. This is a really impressive pace of development.

Source: OpenAI, SemiAnalysis

To validate performance before committing to real hardware runs, OpenAI also has a simulator “chilisim” accurate to within 5% of measured hardware, using a fixed-width trace bus. Tracing on A0 was limited but has improved substantially on B0, likely with inputs from actual runs on A0 silicon. Engineers have demoed the Codex CLI running an internal model, nicknamed “Raiku” or “5.3 Codex Spark”, at 1.2ms TPOT.

The team also showed off Codex-written demos running directly on the chip: Doom at 36 FPS, an FP32 fluid-dynamics simulation, and a “Liquid Light” mouse-drag visualization.

Source: SemiAnalysis

On the model side, OpenAI's internal megakernel approach, nicknamed "gigakernel", is built around a single megakernel that loops on-device to reduce CPU overhead and launch time. The team is also leaning further into test-time compute strategies, with internal interest specifically in how to coherently use 1 million rollouts.

To Disagg or Not to Disagg, 'tis the Question

We mentioned earlier that OpenAI is not using prefill decode disaggregation on these chips. This came as a surprise to us, as NVIDIA and AMD GPU performance benefits significantly from PDD, even on homogenous hardware. Let's dig into why the Jalapeño team went this way.

can improve efficiency at one chosen input/output ratio. Production traffic, however, does not stay at that ratio. Input and output sequence lengths, concurrency, cache-hit rates, speculative-acceptance rates, and latency targets all move throughout the day.

Once devices are divided into prefill and decode pools, too much prefill demand leaves decode chips idle while requests queue. But too much decode demand does the opposite. The operator must continuously predict the right split, provision spare capacity on both sides, and rebalance a system whose ideal ratio is always moving.

In a unified system, some resources may be underused during a particular phase, but every device remains available to serve the next request. In a disaggregated system, an entire chip can sit idle simply because it belongs to the wrong pool. Local utilization looks better, but global utilization can be bad.

Source: SemiAnalysis

Disaggregation also breaks locality. The prefill worker produces a large KV cache that the decode worker immediately needs, so the system must transfer that state across the

sequence length because KV cache grows. However, avoiding the movement of KVs is largely a power and latency optimization; being willing to move some KVs around can allow for increased hardware utilization at the expense of some power and per-request latency.

A fungible fleet shifts capacity between latency-sensitive requests and throughput-oriented batches, while a fixed split strands hardware whenever the traffic mix changes. Moreover, context length changes the balance between attention and FFN work, making any fixed hardware ratio efficient only near its design point.

Source: SemiAnalysis

The same constraint applies to speculative decoding. A draft model has to feed candidate tokens to the verifier with extremely low latency. Separating the two across specialized pools turns a tightly coupled decoding loop into a distributed protocol. The extra communication and coordination can consume the latency saved by drafting. Keeping both models on the same devices and low-latency fabric preserves the locality that makes speculation worthwhile in the first place.

Source: SemiAnalysis

However, disaggregation can still win where demand is sufficiently large, stable, and predictable, particularly when conventional GPUs need large phase-specific batches to reach good throughput. But it is not free lunch.

From Japanese to Indian (Mild to Spicy): Katsu, Vindaloo, and Chana - How are the curry dishes put together into a rack system?

The Jalapeño System at the rack unit level consists of a CPU host rack and an ASIC rack. The host rack houses 16 host CPU trays named “Katsu,” each corresponding to one of the 16 ASIC trays, named “Vindaloo,” to the right of the Katsu. Each host houses two Turin-class AMD EPYC CPUs with 1.5TB of DRAM, 2x E1.S, and 2x M.2 SSDs per rack. Each tray is also specced with 400G (2x200G) frontend networking. Each Katsu tray connects to each Vindaloo tray via 8 external PCIe DAC cables that run horizontally across the rack at the front. The system level design is done in partnership with Celestica.

of 128 Jalapeño ASICs per rack. The ASICs are connected to each of the Chiana switch trays via a copper cable backplane, just like that of Nvidia's Oberon. The scale up topology is split into a local domain of 128 ASICs within the rack and a global domain connecting up to 16 racks or 2,048 ASICs. We will explain the bandwidth and the topology in more detail below.

Power provisioning to a sidecar host rack draws roughly 50kW provisioned (31kW in production), and the ASIC rack draws 130kW, making the total two rack system roughly 160kW. That's basically a double-wide GB300 rack in terms of power draw.

Source: SemiAnalysis, OpenAI

OpenAI can connect up to 2,048 Jalapeño XPUs within a single scale-up network. The scale-up network consists of two domains, a local domain connecting all 128 XPUs over backplane within the rack, as well as a global domain connecting 2,048 XPUs over 16 racks using a hybrid of copper and optical interconnect. Each rack consists of 8 Chana switch trays. Six Chana switches in the middle are for the local domain, which come with one 102.4T Tomahawk 6 switch ASIC each. Two Chana switches at the top and bottom of the local switches are for the global domain, which we think could consist of 2x 102.4T Tomahawk 6 switches making up to 204.8T per switch tray.

ASICs. This would amount to 48-differential pair (DP) male and female connector pairs per XPU translating to a total of 6,144DPs worth of passive copper cables per rack used for local scale-up.

For the global domain, 16 racks totaling 2,048 XPUs are connected together via a combination of copper backplane, electrical 204.8T TH6 switch, 1.6T transceivers, and optical circuit switch. Each XPU has a uni-directional bandwidth of 1.6Tb/s for the global link, which is 16-differential pair (DP) male and female connector pairs per XPU for the backplane between the XPU and the global switch. Bandwidth exiting each global switch tray of 2 ASICs each is split between the backplane and front panel optics.

Between local domain and global domain, backplane connector count per rack comes up to 64 DPs per XPU and a total of 8,192 DPs worth of passive copper cables per rack.

The global domain adopts a rail-only architecture consisting of 8-rails across the global domain. We think OpenAI routes optical links in the global domain via Optical Circuit Switches (OCS) installed in every rack. For every XPU, 1.6Tb/s of global bandwidth will travel to the global switch tray over the copper backplane. This then exits the switch through the front panel via 1.6T transceivers, which go to the passive optical switch before exiting the rack. This expands the scale-up world size to 2,048 XPUs combining 16 racks of 128 XPUs each.

Source: SemiAnalysis

Because scale-up networking is only about 10% of total system cost, that flexibility buys valuable optionality for future 10–20 trillion parameter models or 2–4 million token context windows. On deployment, OpenAI is partnering with neoclouds and is gathering reliability data with datacenter partners through January while optimizing dock-to-rack rollout time.

What's Next

Next, we talk about the future of Jalapeño, whose first production token is coming soon. The next goal is 100MW, and the hurdles will mostly be hardware: How much can they produce, how well can they deploy and operate datacenters, how do they handle monitoring, and resiliency, etc. The software is already proven, and with internal models, every software headstart is easily caught up to. Behind the paywall we will discuss implications for NVIDIA, AMD, Cerebras, and other chip companies who have signed deals with OpenAI in the coming years.

We also cover production volumes, units, and timelines for the next generation chip in our Accelerator model here.

This post is for paid subscribers

Subscribe

Already a paid subscriber? Sign in

← Previous

© 2026 Dylan Patel · Privacy · Terms · Collection notice

Start your Substack

Get the app

Substack is the home for great culture