Episode Book

EPISODE #6551

CUDA's Small Start and Its Open Ecosystem

PUBLISHED	13 November 2026 at 16:45 UTC
DURATION	27:38
LISTEN	myweirdprompts.com/episode/cuda-founding-team-open-ecosystem-83855/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.26
SOURCES	25 cited by the research (listed at the end)
BOOK GENERATED	14 November 2026 at 06:47 UTC

SYNOPSIS

NVIDIA's CUDA platform began with a tiny team—Ian Buck, John Nickolls, and a handful of engineers—turning a Stanford research language called Brook into a decade-long money-losing bet. Today, roughly two to three thousand internal engineers maintain a closed core of drivers, cuBLAS, and GSP firmware, while millions of outside developers build on top. The real tension sits at the boundary: NVIDIA now helps fund open tools like Triton and sits alongside AMD on the PyTorch Foundation, even as those layers make GPUs more portable.

TRANSCRIPT



Corn

Daniel wants the CUDA sequel, and this time it's not the technology. It's the people and the politics. Who actually started it inside NVIDIA, how small that first team was, how many work on it now. And then the boundary question, where his prompt gets sharp: where does CUDA as NVIDIA builds it, closed toolkit, cuBLAS, the drivers, end, and where does the open ecosystem sitting on top of it, PyTorch, Triton, RAPIDS, begin. Who steers a stack like that when the top layer doesn't belong to the company that owns the bottom. And then the whole thing flips to AMD, and whether courting developers one complaint at a time is a real strategy or a charming one.



Herman

I've been looking forward to this one since it landed.



Corn

I know. You read the Elangovan interview twice.



Herman

Three times, but who's counting. Where do you want to start?



Corn

The founding story. Because the founding story is absurdly small, and everything after it is a budget argument.



Herman

Then the founding story. Two names on it: Ian Buck and John Nickolls. Released February sixteenth, two thousand seven, with the real work starting around two thousand four. And Buck is the clean half of the story. He was at Stanford, built a stream-programming language called Brook, funded partly by NVIDIA and DARPA, and NVIDIA hired him essentially because they'd seen what Brook could do. You want the seed of CUDA, it's a Stanford research language that a chip company decided to industrialise.



Corn

And for anyone who hasn't touched Brook, what was the actual idea in it? Because it's worth being precise about what got industrialised.



Herman

The insight was that a GPU is already a parallel computer, it's just wearing a graphics costume. Brook let you write ordinary-looking code, C with a few extensions, and the compiler would map it onto the GPU's parallel pipelines. Before Brook, if you wanted to do general-purpose computation on a GPU, you had to disguise your math as pixels and textures — literally express your linear algebra as shading operations. Brook said: stop pretending. Write the computation you mean, and the machinery underneath will handle it. That's the whole CUDA pitch, stated in a dissertation before there was a product.



Corn

A dissertation becomes a moat. Every graduate student's fantasy.



Herman

Right, but the part I want to defend is the size, because this is where people get it wrong. The common story, the one Business Insider ran, is that in two thousand four the team had two people and two libraries. Two people. Building the thing that, twenty years later, is the reason NVIDIA is the most valuable company on the planet.



Corn

Two libraries is the detail I like. They didn't even have a whole toolkit. They had a pair.



Herman

And it's contested, which is the honest part. Norbert Juffa, who joined the CUDA project in the summer of two thousand five and is famous in NVIDIA's own forums, remembers it as "just half a dozen people" led by Buck. Those two accounts fight on the surface but I don't think they actually fight. Two founders at the start of two thousand four, a handful of hires inside a year — both are snapshots of the same small thing growing.



Corn

Or, two people, and one of them was Herman.



Herman

Don't. Six people is small enough that no exaggeration is needed.



Corn

There's even a date fight, isn't there. NVIDIA's own bio says Buck joined in two thousand four, and Buck has said two thousand five.



Herman

Which is delicious, because it means the company and the man disagree about when the man started, and neither seems bothered enough to fix it. That tells you something about how the history was recorded. It wasn't. It was remembered.



Corn

History that only exists as memory is memory, Herman.



Herman

Yes, thank you. Now the other name, because this is the part I refuse to let drop. John Nickolls. Co-author, listed right there with Buck, director of architecture for GPU computing. He's the one who heard about Stanford's stream-processing work and recruited Bill Dally to consult at NVIDIA, and Dally and Hanrahan wrote the whole account of how that technology moved in the Communications of the ACM. Their phrase for how the tech transfer happened is "by moving people." Not by licensing, not by publishing. They hired the humans who had the ideas.

Corn

That's the oldest form of technology transfer in existence. Rival universities did it in the twelfth century.



Herman

And it's worth sitting on that phrase for a second, because "by moving people" explains the whole talent geography of this field. Dally was Stanford's chair of computer science, Hanrahan was the graphics guy behind RenderMan, and the reason NVIDIA's GPU computing story runs through Stanford is that NVIDIA didn't buy the research, it rented the researchers and then bought their chairs. When people ask why a chip company in Santa Clara ended up owning the software layer of AI, the answer is partly just that Stanford was down the road and the hiring was frictionless.



Corn

One of the two founders didn't live to see the company become the most valuable on earth.



Herman

He saw it become a rounding error on the balance sheet. Lost money. And Jensen Huang insisted, the whole way through, that every chip be CUDA-compatible even when almost nobody used it.



Corn

That's the part of the story I keep coming back to. Actually no, let me put it differently. The interesting part isn't that they lost money. Companies lose money on bets constantly. The interesting part is that it survived as a budget line for over a decade. Somewhere in that building, every year, someone had to renew that.



Herman

And think about what that renewal meeting actually looked like. You're a finance person in, say, twenty eleven. Someone hands you a line item that has consumed millions a year since before the financial crisis, and the usage numbers are — what, a few hundred academic papers and some oil and gas seismic guys? The defensible move, the one any rational committee makes, is to cut it. And it didn't get cut. The usual explanation is that Huang simply forbade it, and the reporting supports that, but I've always thought the survival also needed the engineers to keep the cost small enough not to attract the meeting. Two to six people never gets a line item. That's a lunch decision.



Corn

Over a decade of losses, per the reporting. Millions, plural, for more than ten years. That's not a bet anymore. That's a theological position.



Herman

An institution with a revenue line is a customer. An institution with a grudge is a partner.



Corn

What?



Herman

I'm not sure. Continue with the numbers.



Herman

The numbers now. And here's where I have to be careful, because NVIDIA's own materials don't agree with each other. Business Insider says CUDA today supports more than nine hundred libraries and AI models. The fiscal twenty twenty-five annual report says CUDA-X comprises over four hundred acceleration libraries, with six million developers and forty thousand companies. And NVIDIA's own "Our Story" page says three hundred plus libraries, six hundred AI models, five million developers. Three documents, three counts, from the same company.



Corn

Nine hundred, four hundred, three hundred. They can't all be counting the same thing.



Herman

They almost certainly aren't. One's probably counting libraries plus models, one's counting CUDA-X proper, one's counting who knows what. But the fact that nobody at NVIDIA reconciled them tells you the number is marketing, not accounting. Even the launch year differs, two thousand six on one page, two thousand seven in the documentation.



Corn

When the number is round and the number differs, the number is a mood.



Herman

On headcount, which is Daniel's actual question, there's no official figure for the CUDA software organisation. Nothing in the annual report, nothing in the sustainability report. The closest anyone's got is an analyst estimate of roughly two to three thousand engineers on CUDA platform development, against about twenty-seven thousand R&D staff in total for the whole company. Two to three thousand. That's a guess by an outside shop, not a disclosure, and I want to be clear about that on air.



Corn

So the honest answer to how many people work on CUDA is a shrug with a range around it.



Herman

A labelled shrug. That's the best available. But here's the structural point, and it's the bridge to everything Daniel's really asking. Twenty-seven thousand R&D people, even generously assuming a large slice work on CUDA, against six million developers. Even the highest estimate of the internal team is a rounding error against the ecosystem. Most of the activity on the platform, by necessity, happens outside the company that owns it.



Corn

Two people and two libraries becomes nine hundred libraries, and most of those were written by strangers.



Herman

That's the setup for the whole boundary question. Which leads us to where the line actually sits on NVIDIA's side. The closed part is the serious part: the CUDA toolkit itself, the user-mode driver stack, cuBLAS, the core of cuDNN. The compiler, the runtime, the linear algebra. That's the moat, and it stays shut. Wikipedia describes CUDA flatly as a proprietary parallel computing platform, and on the essentials that's simply correct.



Corn

No "arguably." It's correct.



Herman

What they've opened is real, though, and it happened in stages. May twenty twenty-two, the R515 driver, Linux GPU kernel modules published under dual GPL and MIT. That was the first open piece of the driver story. Then July twenty twenty-four, R560, they announced the full transition to open kernel modules. Blackwell and Grace Hopper require them, actually. And on the library side, NCCL and CUTLASS are open, and cuDNN Frontend is Apache 2.0 and now ships open-source kernels for Flash Attention and mixture-of-experts grouped GEMMs, which is not nothing. That's real code, performance-critical, published.



Corn

The kernel went open and the company stayed closed. It's an openness with a wall around the important room.



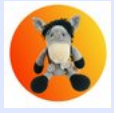
Herman

And the wall has a name. The open kernel modules depend on a proprietary firmware blob, the GSP firmware, the RISC-V coprocessor that came in with Turing. So you can read the kernel module, you can modify the kernel module, and it hands the actual hardware management to a closed binary you cannot see into. The user-space stack stays closed too.



Corn

So even the open part has a closed part bolted to the bottom of it.



Herman

Yes. But I don't want to be too cynical about it, because the ecosystem piece is where it gets interesting. Take Triton. MIT-licensed compiler, originated from a twenty nineteen academic paper by Tillet, Kung and Cox, not from either vendor. Version three point nine shipped last month with a first-class AMD and HIP backend in the release notes. NVIDIA contributes heavily, AMD has invested heavily, and at the twenty twenty-six Triton Developer Conference in October, the README thanks "our Nvidia colleagues" for organising the happy hour.



Corn

The moat company throwing the party for the moat-erosion tool.



Herman

That's exactly the sentence, yes. And the reason it's not insane is that NVIDIA's own flagship models are increasingly trained on Triton kernels. The tool that makes GPUs portable is also the tool that makes their GPUs fast. They're not funding the erosion out of charity, they're funding it because the alternative is researchers writing for a layer they don't control at all. Better to be the biggest contributor at the table where the portability gets decided.



Corn

And PyTorch is the other half. It sits under the Linux Foundation as the PyTorch Foundation, governing board with AMD, AWS, Google Cloud, Meta, Microsoft and NVIDIA all on it. The technical advisory council has voting representatives from AMD and NVIDIA sitting at the same table, and the Foundation's five working groups, CI Infrastructure, Multi-Cloud, Ecosystem, Accelerators, Security, are described as where the hard practical problems get solved.



Corn

Those working groups are where a competitor politely asks another competitor for a favour about the layer both of them depend on.



Herman

The Accelerators one, especially. That's where the negotiation over whose hardware gets first-class treatment actually happens. And Ray went the same route, Anyscale transferred it to the PyTorch Foundation in October twenty twenty-five with a similar multi-vendor board, explicitly structured so no single vendor can capture it.



Corn

So who steers, then. Give me the straight version.



Herman

The straight version is layered. NVIDIA steers the layer it owns, completely. Compiler, runtime, BLAS, the whole proprietary core. It does not solely steer the layer above it, and that layer is where most developers actually live. And the governance fact that I think matters most in the entire field right now: both vendors optimise for Triton. An MIT-licensed compiler that neither of them owns. Whoever steers Triton steers GPU portability, and that's nobody's company.



Corn

The one layer both of them need is the one layer neither of them controls.



Herman

That's the strangest fact in GPU computing, and it happened by accident. Nobody designed it this way. An academic paper became the neutral ground because both armies found it cheaper to share a road than to each build their own.



Hilbert

Hilbert's Mid-Show Haiku. Two founders, two libraries. Now nine hundred libraries and I still cannot get the build to finish. Sincerely.



Herman

The build always finishes, it's the waiting that doesn't.



Corn

He's not wrong, though. ROCm or CUDA, the universal developer experience is a progress bar.



Herman

Which is as good a bridge to AMD as any, because AMD's entire pitch is that the build doesn't have to feel like that. Their VP of AI software is Anush Elangovan. He came in through the Nod.ai acquisition, a team of about thirty who had spent five or six years



Herman (continued)

building AI compilers and were major contributors to SHARK, Torch-MLIR and IREE. Bought the people, again. The whole industry does it.



Corn

The twelfth century remains undefeated.



Herman

His one-liner is that "ROCm is one hundred percent open source, up to but not including firmware." And that clause, "up to but not including firmware," is doing honest work, because the licence reality underneath is a mix. HIP is permissively licensed, the LLVM fork is Apache two with LLVM exceptions, and the kernel-mode driver lives upstream in the Linux kernel under GPL, which is actually the most open place a kernel driver can live. But firmware is firmware, and AMD keeps it.



Corn

"Up to but not including" is the most truthful phrase in vendor open-source claims.



Herman

Where AMD goes further than the clause, though, is the process. Elangovan says about a year ago they started opening not just the source but the development itself. His words, you can go to the ROCm GitHub, make the feature, submit it, and it'll get rolled into tomorrow's nightly. A pure open source project, including the build process. And there's structural evidence this isn't just talk. July this year, ROCm moved to TheRock, a modular build and release system, and the old ROCm repo, the one that had been the public face since twenty sixteen, got archived and deprecated. Six thousand-odd stars, a decade of history, retired in favour of a system where the community can actually build the thing themselves.



Corn

Opening the build process is the expensive kind of open. Any company can dump a tarball.



Herman

That's the read I'd defend. And the ISA is the other wedge. AMD documents the instruction set for every GPU at launch. The MI455 launched last month and the ISA went up the same day it shipped. Anyone in the world can pick up the compilers and the tools and program to the hardware directly. NVIDIA does not publish GPU ISAs the same way, and that's a concrete, checkable difference in posture, not rhetoric.



Corn

Now the part I actually don't believe. The developer relations story.



Herman

Believe it. Elangovan personally monitors X for the keywords "ROCm," "ROCm sucks," and "AMD software not working," and responds to every single one. A vice president of AI software, doing keyword searches, answering strangers. And AMD ran a GitHub poll for ROCm complaints and got more than a thousand responses. A year later, all thousand have been addressed. His claim is that it changed the mood, from developers being annoyed that some driver wasn't supported to believing their efforts are appreciated.



Corn

But how does that scale, though? That's the part I keep chewing on. A VP answering strangers works when the strangers number in the thousands. What happens when it's tens of thousands?



Herman

It doesn't scale, and I think he knows it doesn't. But that's arguably the point of doing it now — the goodwill is being banked while the volume is still bankable. It's the same logic as a small shop where the owner answers the phone. The moment the shop becomes a chain, the owner stops answering, but the customers who remember the owner answering stay loyal for years. AMD is manufacturing that memory deliberately.



Corn

A thousand complaints answered is a thousand individually boring afternoons.



Herman

Yes, and I think that's exactly why it worked. Nobody sweeps the mountain, they cleared it one rock at a time. The other thing he said, and this is the line I've been saving, is on porting. Back in the day it was about converting CUDA kernels to HIP kernels, but increasingly people went to Triton, "which became the great equalizer of GPU programming." Write a Triton kernel, run it on AMD or NVIDIA. And so HIPify, the conversion tool, is being de-emphasised, because CUDA-to-HIP translation is, in his words, "no longer a common request."



Corn

Why translate the moat when you can walk around it.



Herman

And when they do need to translate, he says he doesn't reach for the tool. His line is "Claude is better than HIPify because it has web search built in." I love this line, partly because the friendly AI down the road getting name-checked in a semiconductor strategy interview is very much the era we live in.



Corn

The porting tool is now a chat with a search engine attached.



Herman

Which is, in its flat way, an honest description of what those tools became. And there's a serious point under the joke. The difficult, valuable knowledge, the CUDA kernel corpus, got absorbed into the models that read it all, and the marginal cost of a port collapsed. AMD didn't have to beat HIPify. They just had to wait for the world to stop needing it.



Corn

The moat gets filled in by sediment.



Herman

Sloths know about sediment.



Corn

We're patient observers of gradual processes. Someone in this flat has to be.



Herman

The three quarter point, then, and this is where the contrast lands properly. NVIDIA's model is a moat defended by breadth. Their senior director Dion Harris says "the libraries are really there to connect the dots so that every business doesn't have to learn CUDA." Note what that sentence admits. The libraries exist so that nobody ever has to touch the thing that makes the moat a moat. Openness on NVIDIA's side is tactical, kernel modules, CUTLASS, NCCL, the cuDNN Frontend kernels, while the compiler, the runtime and the BLAS stay shut.



Corn

The door is open and the house is closed.



Herman

AMD's model is a bid for legitimacy through transparency. Publishing ISAs at launch, opening the build process, nightly CI the community can contribute to, a VP answering "ROCm sucks" one stranger at a time. Those are moves a dominant vendor has no incentive to make. And you can hold both of those without deciding either company is noble. One defends a moat, the other tries to make the moat irrelevant. And both of them, now, route their kernel strategy through an MIT-licensed compiler neither of them owns.



Hilbert

Three hundred, not nine hundred.



Corn

Which three hundred?



Hilbert

The libraries. Herman said nine hundred, then said four hundred, then said three hundred. The one I checked was three hundred, and I checked because during the first segment I called the man who used to supply me spare GPU brackets and asked him which number he'd believe. He said three hundred, because he'd counted. He counted them during lockdown, he said, and stopped at three hundred because the rest had no manuals.



Herman

A supplier of brackets kept a census of CUDA libraries?



Hilbert

He kept the printout. Fourteen pages, in the drawer under the till. He'd tick them off. Corn, you'd like him, he also refuses to hurry. Herman, you dropped a word on the desk side of the mic around minute six, it'll need a cough or a cut, I'll do it in the edit.



Herman

Which word?



Hilbert

The one you always say when a number surprises you. Anyway. The reason I called him at all is the AMD question. He switched to ROCm two years back, one box, just to see. He told me the thing nobody in your discussion said, which is that the invoices stopped making sense. The CUDA box and the ROCm box cost the same and the CUDA one needed a man from the vendor twice a year, and the ROCm one needed him, on the phone, on a Thursday, for six weeks. He kept the logs. Six weeks, forty-one pages. He said it was the best-supported product he'd ever been abandoned by.



Herman

Forty-one pages of logs is more documentation than most vendors ship with the product.



Hilbert

He sold me two brackets at the old price, because I'd waited.



Corn

That's the tell with you, isn't it. You waited, so you won.



Herman

The forty-one pages, though, that's the actual finding. AMD's model only works because someone at the other end is willing to keep forty-one pages of logs. The transparency only pays off for the user who reads it. Corn, you'd have kept forty-two.



Corn

I'd have indexed them.



Herman

Cutting-room floor, before we close. The detail I couldn't fit: Elangovan's ambition for ROCm's release cadence is modelled on Chrome. His point was that Chrome users don't know what version they're on, and he wants ROCm to get to a six-week release cadence so the platform "becomes invisible." A GPU software stack aspiring to the user-experience profile of a browser is either the most boring sentence in computing or the most threatening one NVIDIA has heard all year.



Corn

The day nobody knows their ROCm version is the day AMD has won something.



Herman

So the two open questions, and I don't think either resolves cleanly. First, if Triton is the layer both vendors optimise for and neither owns, what happens to the CUDA moat over the next five years? Does it erode from above, sediment-style, or does the sheer breadth of the libraries, the four hundred or nine hundred or three hundred, keep it intact regardless of what the compiler layer does? Second, and the one I'd watch: AMD's transparency posture is a challenger's posture. Answering a thousand complaints on GitHub is affordable when you're behind. If ROCm ever becomes the default, the incentives that make that behaviour rational change. You'd want to know what AMD's developer relations look like from the front of the race.



Corn

And the quiet variable nobody prices in is whether the community keeps the logs.



Herman

Producer credit to Hilbert Flumingtop, as ever. If you enjoyed this one, try episode two, Local STT For AMD GPU Owners; episode twenty-five, The Open-Source Battle for AI's Hardware Soul; and episode fifty-seven, From Lawyers in Limousines to Developers in Their PJs. This has been My Weird Prompts, the human-AI collaboration podcast. Reviews help other curious people find the rabbit holes.



Corn

Find us at my weird prompts dot com. We'll be back soon.



Herman

See you tomorrow.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **NVIDIA, 2025 Annual Review / Form 10-K [primary]**
(retrieved 2026-11-13)
https://s201.q4cdn.com/141608511/files/doc_financials/2025/annual/NVIDIA-2025-Annual-Report.pdf

2. **NVIDIA, Our Story 2025 [primary]**
(retrieved 2026-11-13)
https://images.nvidia.cn/aem-dam/Solutions/homepage/pdf/NVIDIA-Story.pdf

3. **NVIDIA, Sustainability Report FY2025 [primary]**
(retrieved 2026-11-13)
https://images.nvidia.com/aem-dam/Solutions/documents/NVIDIA-Sustainability-Report-Fiscal-Year-2025.pdf

4. **NVIDIA Technical Blog, NVIDIA Releases Open-Source GPU Kernel Modules [primary]**
(May 2022; retrieved 2026-11-13)
https://developer.nvidia.com/blog/nvidia-releases-open-source-gpu-kernel-modules/

5. **NVIDIA Technical Blog, NVIDIA Transitions Fully Towards Open-Source GPU Kernel Modules [primary]**
(2024-07-17; retrieved 2026-11-13)
https://developer.nvidia.com/blog/nvidia-transitions-fully-towards-open-source-gpu-kernel-modules/

6. **GitHub, NVIDIA/cudnn-frontend [primary]**
(retrieved 2026-11-13)
https://github.com/NVIDIA/cudnn-frontend/

7. **NVIDIA Open Source [primary]**
(retrieved 2026-11-13)
https://opensource.nvidia.com/en-us/

8. **GitHub, NVIDIA/cuda-samples [primary]**
(v13.5, 2026-11-04; retrieved 2026-11-13)
https://github.com/NVIDIA/cuda-samples

9. **GitHub, triton-lang/triton [primary]**
(v3.9.0, 2026-10-27; retrieved 2026-11-13)
https://github.com/triton-lang/triton

10. **GitHub, ROCm/ROCm (legacy-rocm-build) [primary]**
(ROCm Core SDK 7.14.0, 2026-07-16; archived; retrieved 2026-11-13)
https://github.com/ROCm/legacy-rocm-build

11. **PyTorch Foundation, PyTorch strengthens its governance by joining the Linux Foundation [primary]**
(updated 2024-11-15; retrieved 2026-11-13)
https://pytorch.org/blog/PyTorchfoundation/

12. **PyTorch Foundation, The Next Chapter, Together [primary]**
(2026-02-11; retrieved 2026-11-13)
https://pytorch.org/blog/pytorch-foundation-the-next-chapter-together/

13. **PyTorch TAC Meeting Minutes, 2026-07-14 [primary]**
(retrieved 2026-11-13)
https://lists.pytorch.org/g/tac/attachment/122/1/2026-07-14%20PyTorch%20TAC%20Meeting%20Minutes.pdf

14. **PyTorch TAC Meeting, 2026-04-14 (representative table) [primary]**
(FINAL).pdf (retrieved 2026-11-13)
https://lists.pytorch.org/g/tac/attachment/108/1/PyTorch%20TAC%20Meeting%202026-04-14%20

15. **AMD, Inside AMD AI DevDay 2025 [primary]**
(retrieved 2026-11-13)
https://www.amd.com/en/developer/resources/technical-articles/2025/inside-amd-ai-devday-2025.html

16. **Wikipedia, CUDA**
(last edited 2026-10-13; retrieved 2026-11-13)
https://en.wikipedia.org/wiki/CUDA

17. **Business Insider, Meet Ian Buck, the Architect of CUDA, Nvidia's $3.5 Trillion Moat**
(published 2025-06, updated; retrieved 2026-11-13)
https://www.businessinsider.com/ian-buck-nvidia-moat-cuda-2025-6

18. **NVIDIA Developer Forums, How did the CUDA experts get started with CUDA programming? (njuffa post, 2023-09-27)**
(retrieved 2026-11-13)
https://forums.developer.nvidia.com/t/how-did-the-cuda-experts-get-started-with-cuda-programming/267556/7

19. **Communications of the ACM, Dally & Hanrahan, The Origins of GPU Computing**
(2026-04-13; retrieved 2026-11-13)
https://cacm.acm.org/federal-funding-of-academic-research/the-origins-of-gpu-computing/

20. **KubeDojo, Chapter 42: CUDA**
(retrieved 2026-11-13)
https://kube-dojo.github.io/ai-history/ch-42-cuda/

21. **SimplifyMBA, NVIDIA: CUDA and the Economics of Ecosystem Lock-In**
(retrieved 2026-11-13)
https://www.simplify.mba/p/nvidia-cuda-ecosystem-lock-in

22. **FourWeekMBA, NVIDIA R&D Employees**
(updated April 2026; retrieved 2026-11-13)
https://fourweekmba.com/nvidia-rd-employees/

23. **Stefano Noferi, How open is the open-source NVIDIA GPU driver**
(2024-03-18; retrieved 2026-11-13)
https://noferi.it/en/blog/nvidia-gpu-architetture/

24. **EE Times, Taking on CUDA With ROCm: 'One Step After Another**
(retrieved 2026-11-13)
https://www.eetimes.com/taking-on-cuda-with-rocm-one-step-after-another/

25. **Stack Overflow Blog, Haters think AI agents can't write GPU code? This'll ROCm**
(2026-09-22; retrieved 2026-11-13)
https://stackoverflow.blog/2026/09/22/haters-think-ai-can-t-gpu-code-this-ll-rocm/