The Chip Is Not The Product

A Preprint

Sidney Scott

Reckon Research

sidney@reckonresearch.com

Abstract

Capital is funding a Cambrian explosion in silicon on a thesis that is correct: the era of one architecture for every workload is ending. It is buying the wrong object. This paper argues that architectures do not win on performance but on the size of the population that can reach them, a property we call reachability. The historical record is unusually clean. The graphics processor won deep learning across a decade in which it contained no feature designed specifically for deep learning, and the architectures that lost during the same period lost on the software layer rather than on the silicon. We review five of them, including one whose entire design premise was a compiler that never arrived, and one whose defining bet had already failed commercially a decade earlier under partly the same engineers. We then formalize buyer substrate choice as a risk transfer problem in which the cost of a failed deployment falls on an individual while the saving accrues to a firm, and show that the resulting system has two stable equilibria. The prevailing one percent merchant share of challenger silicon is the low one. It is not a verdict on the silicon. We identify reversibility, the property that a substrate choice can be undone, as the variable that moves the system, argue that no party which sells silicon can supply it, and close with four dated and falsifiable predictions.

Keywords heterogeneous compute · hardware software co-design · compilers · technology adoption · computer architecture history

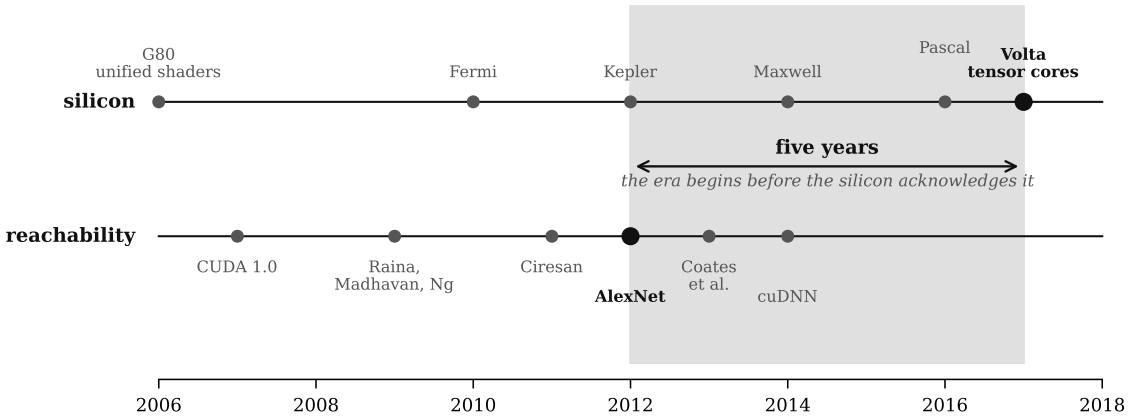


Figure 1: The decoupling. Silicon milestones above, reachability milestones below. The deep learning era begins in 2012 on a graphics part containing no feature designed specifically for deep learning, and the silicon does not acknowledge it until tensor cores arrive with Volta in 2017. Historical record; events are ordinal markers on a time axis and no quantity is plotted.

1 Introduction

Roughly two hundred and fifty distinct AI chip designs are active today, on a defensible range of two hundred and thirty to two hundred and sixty five. More than one hundred and eighty of them are new in the last three years. About one hundred and thirty ship in volume; the remainder are pre tape-out, licensable intellectual property cores, or recently discontinued. Around one hundred and ninety companies now design AI silicon.

Two designs are counted as distinct here when code tuned for one does not run well on the other. That is an optimization surface test rather than a specification test: a distinct design is one that requires its own compiler backend and its own hand tuned kernels to approach its rated peak. The rule matters, because the count is entirely a function of it, and quoting a figure without one is how these numbers become folklore. Provenance differs across the four figures above and is worth stating with them: the stock of active designs and its range are estimated, the designer count is third party, the volume shipping figure is counted under the rule above, and the three year new-design figure follows from the same count applied to announcement dates.

The thesis underneath that wave is correct. Workloads are specializing faster than any single architecture can chase them, and no chip optimizes simultaneously for throughput, control flow, latency, memory bandwidth and power. That is a physics constraint rather than an engineering gap.

Now look at where the revenue landed. One vendor takes roughly seventy eight cents of every dollar of merchant AI silicon revenue. Five firms take about ninety nine. Every startup and every newly public challenger on earth, summed, splits the last cent.

The usual reading is that the challengers built worse silicon. Several of them did not. This paper argues that the distribution above is not a ranking of architectures at all. It is a ranking of software ecosystems, and the ranking was substantially decided once, in 2007, and has not been revisited since.

This paper makes three contributions. It assembles the historical record showing that the decisive variable in architecture adoption has been reachability rather than performance. It formalizes buyer substrate choice as a coordination problem with two stable equilibria and argues that the prevailing challenger share is the low one. And it identifies reversibility as the term in that model with slack, which is a claim about what kind of intervention could move the market rather than about which silicon is best.

We introduce two terms and they do different jobs.

Reachability is the size of the population that can put a workload on a substrate and get a correct, performant result without first becoming a specialist in that substrate. It is a property of the software layer above a chip rather than of the chip.

Reversibility is the property that a substrate choice can be undone at low cost. It is what determines whether a buyer treats a substrate decision as an experiment or as a commitment.

The relationship to adjacent work is worth stating precisely, because three mechanisms are in play and they operate at different points in a workload's life. Hooker's hardware lottery [14] concerns which research ideas survive: an algorithm that maps poorly to available silicon becomes a curiosity regardless of merit. Reachability concerns which architectures survive: a part that no population can reach does not reach a market regardless of merit. The mismatch tax [22] concerns what deployed work costs: an operation executed outside its substrate's design point wastes capital regardless of how well it was implemented. Same physics, three points in the lifecycle. This paper takes the middle one.

2 The GPU did not win on architecture

The strongest available evidence for reachability is the case everyone believes they already understand.

2.1 The hardware was built for something else

The G80 in 2006 introduced a unified shader architecture, adopted so that a single pool of processors could be load balanced across vertex and pixel work in games. CUDA 1.0 followed in 2007 [17], descended from the Brook stream programming work at Stanford, which itself existed because researchers were already expressing general computation as texture operations because no other route to the hardware was available.

Nothing in that sequence was aimed at machine learning.

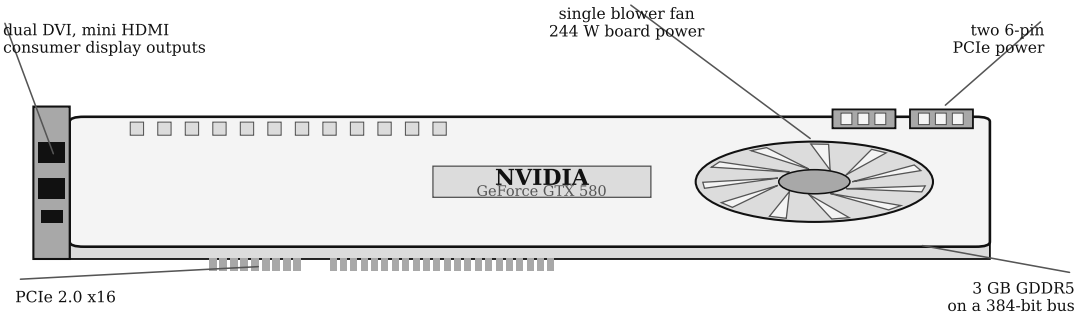
2.2 The chip that won had nothing built for the workload

AlexNet [15] trained on two GeForce GTX 580 cards with three gigabytes of memory each, at roughly five hundred dollars a card. It ran for ninety epochs over five to six days. The released source is on the order of six thousand lines of CUDA. It reported 15.3 percent top five error against 26.2 for the second best entry in ILSVRC 2012.

One thousand dollars of consumer gaming hardware against the assembled resources of computer vision.

Two details deserve more weight than they usually receive.

First, the famous two column structure of AlexNet is not an architectural insight. The network was split across two devices because it did not fit in the memory of one three gigabyte card. The most reproduced architecture diagram in modern machine learning encodes a hardware limitation.



GeForce GTX 580, November 2010. Retail price about 500 dollars. Two of these trained AlexNet.
No unit on this board was designed for machine learning. Drawn to proportion; not to scale with the die.

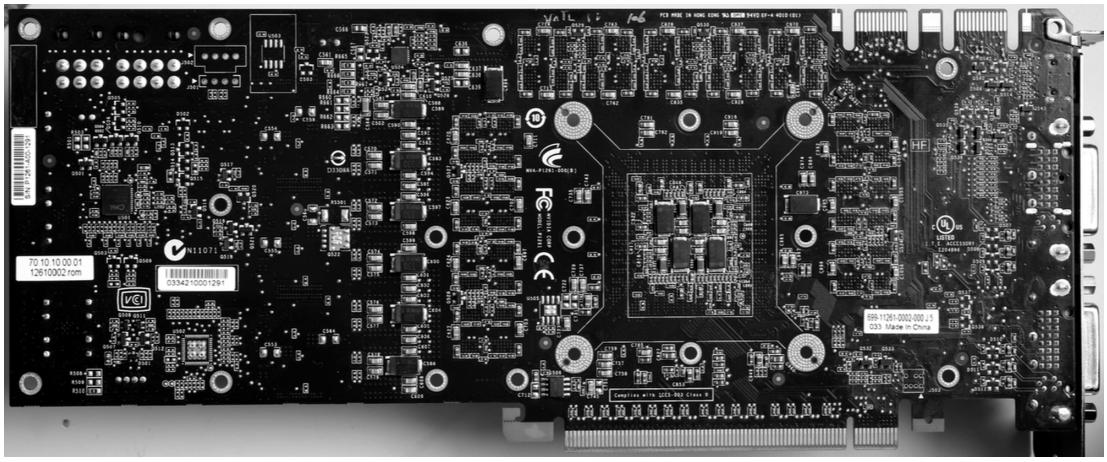


Figure 2: The substrate of the deep learning revolution. Above, the card as a buyer encountered it in 2010: display outputs, a blower fan, a five hundred dollar retail price. Below, the board itself. Nothing on it was designed specifically for machine learning, and two of them trained AlexNet. Drawing and photograph by the author.

Second, and decisively: **tensor cores did not ship until Volta in 2017**, five years after the result that founded the era. Across the entire period in which the graphics processor became the substrate of deep learning, it contained no unit designed specifically for deep learning. Programmable parallel compute reachable from a general purpose language was designed in, for graphics and high performance computing. Contemporaneous technical descriptions of the GTX 580 describe a device arranged for texture processing, shadow map processing and other graphics centric work.

2.3 The population moved, not the silicon

The trajectory from 2009 to 2012 is a sequence of groups discovering that a commodity graphics part could be reached, not a sequence of hardware improvements. Raina, Madhavan and Ng demonstrated large scale unsupervised learning on graphics processors in 2009. Ciresan and colleagues won image classification competitions on the same class of hardware in 2010 and 2011. AlexNet in 2012 is when the wider field noticed. The silicon roadmap over those years, Fermi to Kepler, was driven by graphics and general high performance computing, not by the workload that was quietly taking it over.

2.4 The comparison that settles it

In 2012 the DistBelief system trained networks with over one billion parameters using sixteen thousand CPU cores across one thousand machines [6, 16]. Within a year, Coates and colleagues [5] trained a comparable billion parameter network on three machines, and scaled past eleven billion parameters on sixteen.

One thousand machines to three. The two systems differ in hardware generation and scope, so this shows a direction, not a measured ratio. No new physics appears.

The authors state the mechanism themselves, in the abstract, in 2013. Their infrastructure, they write, is *much more easily marshaled by others*, and the value they claim for it is wider spread research rather than raw speed. That is reachability, named by the people who demonstrated it, thirteen years before this paper.

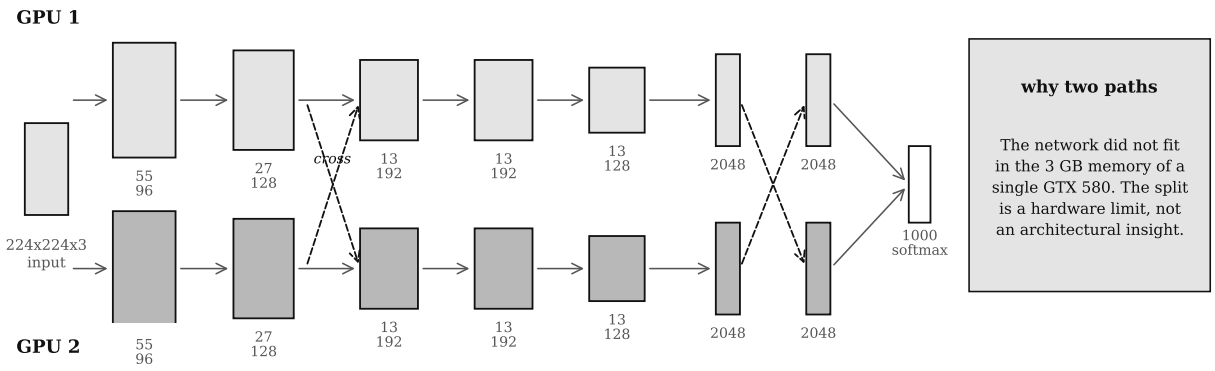


Figure 3: AlexNet, redrawn. Two paths, one per device, joined only at the third convolutional layer and at the dense layers. The structure is a consequence of the memory available on a single card in 2012. Redrawn from the description in Krizhevsky et al. [15].

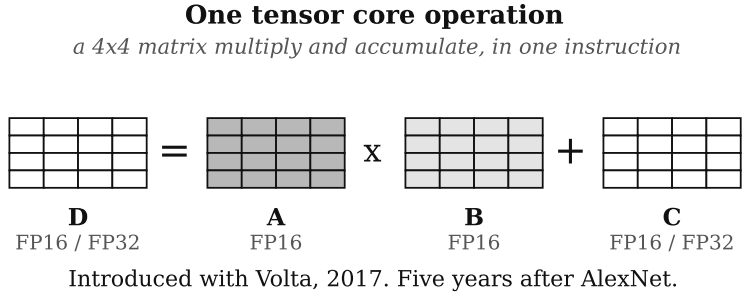


Figure 4: The unit the graphics processor did not have while it was winning deep learning. Redrawn from the published Volta description.

The lesson is not that graphics processors were secretly good at neural networks. It is that in 2012 the number of people who could run a large neural network went from a handful of institutions with cluster budgets to any graduate student with a thousand dollars and a CUDA manual. The die did not change. The reachable population did.

3 The graveyard

If reachability rather than performance selects architectures, the record should contain technically sound designs that died on the software layer. It does, and the cases are not marginal.

3.1 Itanium: the pure case

Explicitly Parallel Instruction Computing moved the complexity of instruction scheduling out of the processor and into the compiler, on the argument that a compiler examining a larger window could schedule better than dynamic hardware, and that the silicon freed by removing scheduling circuitry could be spent on execution resources.

The reasoning was sound. Intel and Hewlett Packard committed a decade and a very large amount of capital to an architecture whose realized performance was contingent on a compiler that did not yet exist. The silicon shipped. The compiler never arrived in the form the design required.

AMD extended x86 to sixty four bits rather than replacing it, shipped it in Opteron in 2003, and gave buyers a path on which software they already owned ran unmodified [1]; Intel shipped a binary compatible extension in 2004 [9]. The substrate that displaced Itanium was the one buyers reached without rewriting.

An architecture whose stated premise was a software layer someone else had to build is this paper's thesis with no interpretation required.

Design	Period	Binding constraint
Itanium	1994-2021	Compiler that was the design premise
Cell	2005-2013	Cost of mapping code to heterogeneity
Larrabee	2008-2010	Parallel programming model, despite x86
OpenCL	2008-	No patron, no free reference stack
i860	1989-1995	Compiler scheduling

Table 1: Five architectures, five distinct technical bets, one failure mode. In each case the part was capable and the route to it was the limit.



Figure 5: Roadrunner at Los Alamos National Laboratory, the first machine to sustain a petaflop. It was decommissioned in 2013 while still among the twenty five fastest systems in the world, in part because of the cost of adapting code to its heterogeneity. Photograph courtesy of the U.S. Department of Energy.

3.2 The same bet, lost twice

The part of the Itanium story that is usually omitted is that the wager had already been settled.

Fisher's 1983 work at Yale on trace scheduling and very long instruction word architectures [10] argued that hardware complexity for dynamic scheduling was wasteful and that a sufficiently capable compiler could do the work once, statically. Two companies commercialized it. Multiflow and Cydrome both failed by the start of the 1990s. Hewlett Packard then hired former Multiflow engineers, and that expertise fed directly into the collaboration with Intel on EPIC.

The same bet, placed twice, twenty years apart, partly by the same people, lost twice for the same reason. That is not one firm misjudging a market. It is a structural blind spot in how the industry prices silicon against the software required to reach it.

3.3 Cell, Larrabee, OpenCL

Roadrunner, built from PowerXCell 8i processors paired with Opterons, sustained 1.026 petaflops in May 2008 to become the first machine past the petaflop barrier. It was decommissioned in March 2013 while still among the twenty five fastest systems in the world. Los Alamos cited, alongside power efficiency, the cost of adapting code to fit the Cell. That is this paper's subject stated by an operator rather than an advocate.

Larrabee had the strongest software argument anyone in the period could make, which was x86 compatibility [23], and it was still insufficient, because binary compatibility is not the same thing as a mature parallel programming model. Schedule and performance, not software alone, are the standard explanations. The first is Intel's own: it withdrew the first product in December 2009, saying silicon and software development were behind where it hoped to be, while calling the SC09 throughput performance extremely promising [12]. Reachability survives both: a part that slips cannot be reached on the date promised, and the programming model x86 compatibility was meant to replace never arrived in the line that continued as Xeon Phi before it was discontinued in turn.

OpenCL is the sharpest case because it was an attempt at the layer itself. Ratified through the Khronos Group at the end of 2008, roughly a year behind CUDA, with support from essentially every relevant silicon vendor [24]. On specification it was credible. It lost on ecosystem, and the reason is structural: it was a consortium document produced by members who each sold silicon, with no disinterested patron funding a free and complete reference implementation. Every vendor had the strongest possible incentive to make its own proprietary stack excellent and its implementation of the neutral standard merely adequate. That is

what each did.

3.4 Instruction sets have never won on merit

The pattern predates accelerators. x86 did not persist because it was elegant; it persisted on accumulated binary compatibility. Intel attempted to escape it twice, with the i860 and with Itanium, and failed both times when the software did not follow. ARM required roughly two decades and workloads that could be recompiled from source before it reached servers at scale. RISC-V is technically clean and its central commercial difficulty is fragmentation, which is a software problem wearing hardware clothing.

4 Relation to the economics of standards

The model in the next section is a network externality model, and it is worth saying so before deriving it, because the economics literature settled the general structure decades ago and this paper contributes the application rather than the mathematics.

Increasing returns to adoption, lock-in to an installed base, and the possibility of a market settling on an inferior standard are established results [2, 7, 13, 8]. What that literature usually treats as the adopter's cost is switching cost: the expense of moving from one standard to another. The substrate case has a feature that is not standard in those models and that changes which intervention works.

The cost of a failed substrate decision does not fall on the same party as the benefit. A firm captures the saving; an individual carries the failure. The relevant friction is therefore not switching cost in the aggregate but the private risk borne by a decision maker inside an organization, which is why lowering the price of challenger silicon has had so little effect. The intervention that moves a switching cost model is a subsidy. The intervention that moves this one is a change in who bears the downside.

5 Reachability, formalized

Why does a market with roughly two hundred and fifty distinct designs allocate ninety nine percent of its revenue to five of them? The usual answer is that the five are better. A more careful answer starts with who bears the risk.

5.1 The buyer's problem

Consider a buyer choosing a substrate for a production workload. Let Δ be the cost saving from the challenger substrate relative to the default, p the probability that deployment fails or overruns badly enough to be judged a failure, and C the cost of that outcome to the individual making the decision.

The saving Δ accrues to the firm. The individual captures some small share α of it, through recognition rather than through cash. The cost C does not have the same structure. A failed migration onto unusual silicon is legible, attributable and personal.

The buyer chooses the challenger when

$$\alpha\Delta > p\,C.$$

With α small and C large, the inequality requires p to be near zero. For any substrate outside the default, it is not.

This is the formal content of the observation that nobody is dismissed for buying the default. The default is not merely safe; it is the only choice whose failure is not attributable to the person who made it.

5.2 Two stable equilibria

The step that matters is that p is not a fixed property of a substrate. It is a decreasing function of the population that has already deployed it. Deployment expertise, documented failure modes, hireable engineers, fallback paths and reference architectures all accumulate with prior adoption. Write $p = p(n)$ with $p' < 0$.

Substituting gives a condition that depends on how many others have gone first:

$$\alpha\Delta > p(n)\,C.$$

At low n the inequality fails, no one switches, and n stays low. At high n it holds comfortably, adoption is self reinforcing, and n stays high. Between them lies an unstable threshold. The system has two stable fixed points and nothing between them is stable.

The one percent is the low equilibrium. It is not a market verdict on challenger silicon. It is where a risk transfer problem settles when nothing removes the risk, and it is stable against any improvement in the silicon alone.

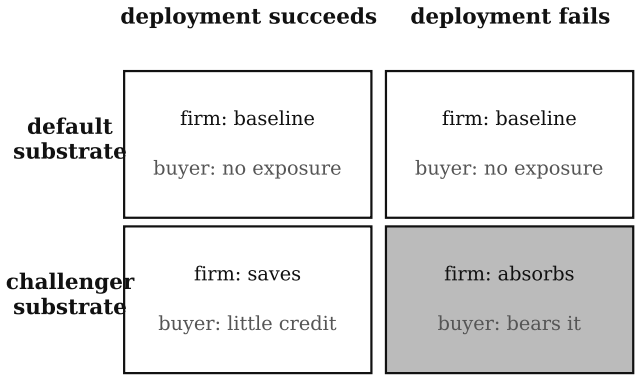


Figure 6: The asymmetry. The firm captures the saving; the individual carries the failure. Only one cell in the matrix has a named owner.

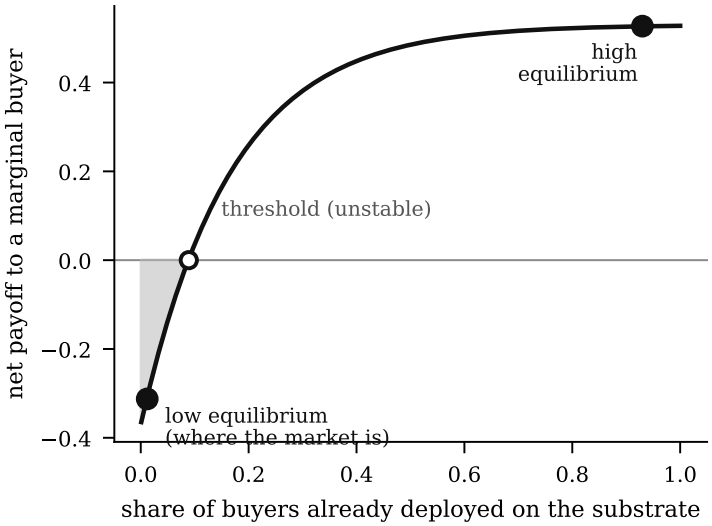


Figure 7: Net payoff to a marginal buyer as a function of prior adoption. Two stable fixed points with an unstable threshold between them. Curve shape is illustrative; the claim is the existence of two equilibria, not the coordinates.

This also re-explains 2007 more precisely than the usual account. CUDA did not make the graphics processor faster. It raised n for a single substrate, past the threshold, until p collapsed for everyone who came after. The advantage compounds from there, which is why eighteen years later it has not been dislodged by parts that are better on paper.

5.3 What is measured and what is not

Three of the four quantities in this model can be anchored to public data. One cannot, and saying which is which is more useful than pretending otherwise.

n is measured, and it is the variable the argument turns on. The vendor that crossed the threshold has disclosed its adoption population repeatedly. It took thirteen years to reach one million registered developers, then under two more to reach two million [18]. Five million was stated in mid 2024 [19], and six million in early 2026 [20]. The definitions behind those counts have varied, so the series should be read as an order of magnitude rather than a measurement, but the shape is not in dispute and no competing substrate has a comparable series to report.

C can be bounded from the acquisition record. Cross vendor routing companies were acquired in the low hundreds of millions against private marks several times higher, in one case with hosted services terminated five weeks after announcement [22]. A buyer who had standardized on such a layer absorbed a total loss of that path on five weeks of notice. That is the shape of the cost the model calls C, observed



Figure 8: Merchant AI silicon revenue concentration. Roughly two hundred and fifty distinct designs are active, more than one hundred and eighty of them new since 2023. The last cent is shared by all of them.

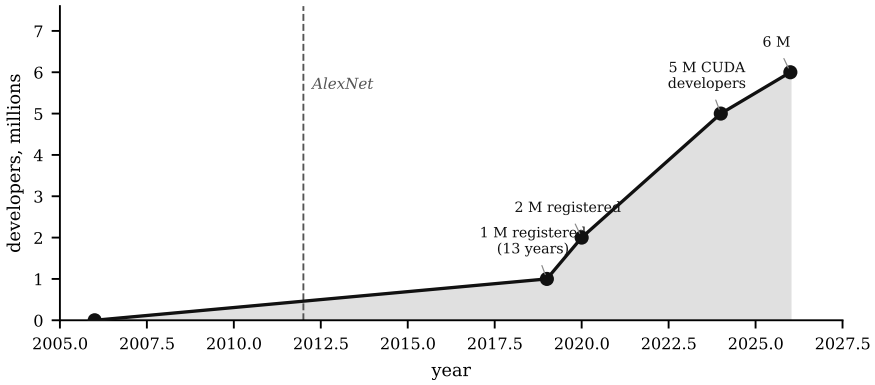


Figure 9: The adoption population for the substrate that crossed, as disclosed by its vendor. Counts are the vendor's own public statements and the underlying definition has varied between registered developers and CUDA developers, so the series is an order of magnitude rather than a measurement. No competing substrate reports a comparable figure.

rather than assumed.

α **is small by construction** and does not need estimating. No common procurement structure returns a material share of an infrastructure saving to the individual who authorized the substrate.

One exploratory measurement bears on the premise without establishing it. Companion work runs one model, Llama-3-8B in bf16 under vLLM, on five cards from two vendors, and reports a median decode efficiency gap against vendor specified bandwidth of 0.17 on L40S, 0.24 on H100 PCIe, 0.27 on A100 80GB, 0.46 on MI300X and 0.55 on H100 SXM [22]. Those ratios are exploratory upper bounds, three of the five cells carry self reported silicon identity, and they support no ranking. What they support is narrow: the same model behaves very differently across chips and configurations, so the chip alone does not determine the outcome.

$p(n)$ **is not measured, and this paper does not measure it.** The claim that deployment risk declines in prior adoption is plausible, is the standard result in the diffusion literature [2, 13], and is consistent with the concentration data above, but it is not established here. The measurement that would settle it is specific: for a fixed workload class, track time from procurement to production traffic against the cumulative count of prior production deployments on that architecture. Any organization operating several substrates already holds the data. None publishes it.

5.4 Two doors that bypass the layer

The model predicts that the only challengers posting real revenue will be those that route around p rather than reducing it, and the record matches. The first door is a small number of customers large enough to absorb the deployment work internally. The second is regulatory substitution, where a government excludes the default and domestic silicon inherits a captive market.

The clearest instance is documented in its own filings. The largest independent challenger disclosed that a single customer accounted for 83 percent of revenue in 2023 and 87 percent in the first half of 2024 [3]. Two years later the concentration had moved but not dissolved: 86 percent of 2025 revenue came from two related entities in one country, at 62.0 and 24.0 percent respectively [4].

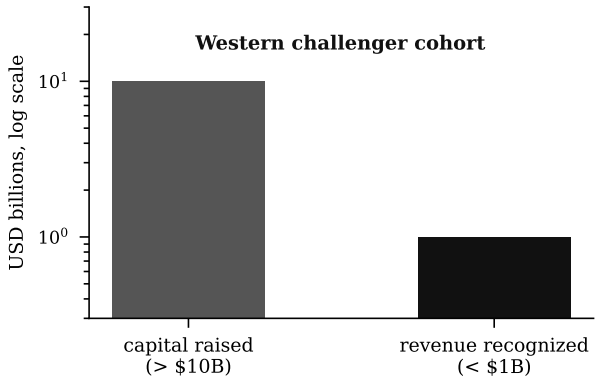


Figure 10: Capital in against revenue out for the Western challenger cohort, excluding China and excluding wafer scale. Silicon was not the bottleneck and capital was not the bottleneck.

The number inside that disclosure that matters most for this paper is not the concentration. It is the direction of the open market line. Revenue billed to United States customers fell 34 percent year over year, from 282.7 to 187.6 million dollars, while total revenue grew [4]. A company with genuinely differentiated silicon, ample capital and a public profile shrank in the market where it had to reduce p, and grew in the market where it did not have to.

Neither door scales, and neither is reachability. The cohort figures make the point without interpretation. On disclosed funding rounds and disclosed or reported revenue, the Western challenger cohort has raised over ten billion dollars and recognizes under one billion between all of them. Both figures are assembled from public announcements rather than audited accounts, and the ratio rather than either number is what carries the argument.

6 Reversibility

If $p(n)C$ is what holds the system at the low equilibrium, the question is which term can be moved.

Δ cannot carry it. Making challenger silicon cheaper or faster raises Δ, but α is small and C is large, so the inequality is insensitive to exactly the variable the entire chip industry is optimizing. This is the formal statement of why better silicon has not worked.

C is the term with slack, and it has an unexamined structure. The cost of a failed deployment is large primarily because the decision is a one way door. Committing a workload to an unusual substrate means porting kernels, rebuilding a serving path, retraining staff and accepting that if it fails there is no cheap route back. What makes the outcome career defining is not the loss itself but its irreversibility.

A layer that makes substrate choice undoable collapses C without touching the silicon. If a workload can move off a part as easily as it moved on, a failed trial stops being a terminal event and becomes an experiment with a known cost. Under that condition the inequality can hold at values of n that are otherwise far below threshold.

Reversibility is therefore not a convenience feature. It is the mechanism by which a market at the low equilibrium can cross to the high one without waiting for an accident.

Three properties are required of any layer that supplies it. The workload must be expressible once in a form not specific to a vendor. The behavior of each substrate must be described in a form the layer can act on. And the layer must be neutral, because a reversibility layer operated by a party that sells silicon is not reversible in the direction that matters. The moment such an operator acquires the cross vendor layer it must continue making competitors' parts run beautifully, which is not a thing a silicon vendor will sustain. The record on this is unambiguous: cross vendor routing companies have been acquired and terminated, while single vendor optimizers have been acquired and integrated [22].

7 Why the gap widens

Two curves are moving in opposite directions.

The cost and time to design a new part is collapsing from two directions. AI driven electronic design automation is compressing design cycles, and dozens of venture backed companies are now applying machine learning to chip design while the incumbent design tool vendors embed the same capability.

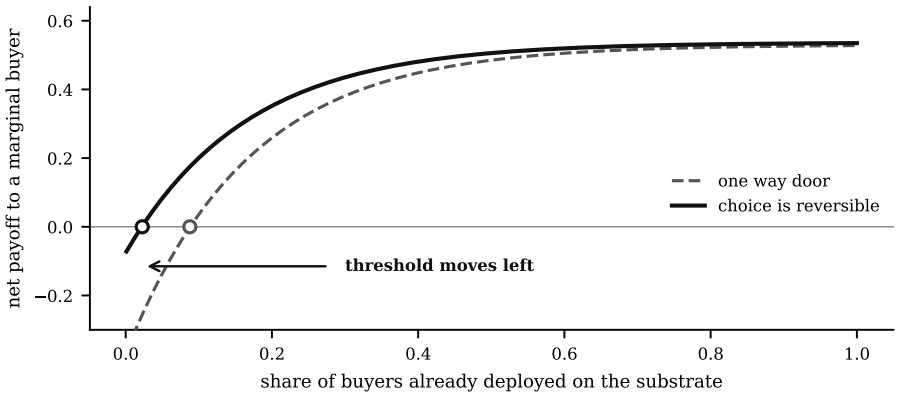


Figure 11: What reversibility does to the model. The silicon, the saving and the risk of failure are unchanged in both curves. Only the cost of being wrong differs, and the adoption threshold moves by roughly a factor of four. Curve shapes are illustrative; the direction and the mechanism are the claim.

Hardwired model silicon pushes from the other side, and the economics are worth stating precisely because they are new. Taalas, acquired by AMD under a definitive agreement announced on 6 August 2026, etches a model's weights directly into a mask ROM fabric rather than loading them from memory at runtime. The company states that only two of the chip's hundred plus layers change between one model and the next, putting tape-out at roughly two months. Those figures are vendor stated and not independently verified. Google is reported to be developing a part that bakes its own frontier model into silicon, and OpenAI has commissioned a model specific inference part with Broadcom. A further report, unconfirmed by either party at the time of writing, has the dominant vendor acquiring the largest open model repository.

The implication needs care, because it is easy to overstate. A family of hardwired parts sharing a base die and a toolchain is one architecture instantiated many times, not many architectures, and it does not multiply compiler backends. What it multiplies is deployment targets: each finished part runs exactly one model and nothing else. A fleet assembled that way poses the hardest placement problem yet, which is the subject of the companion paper rather than of this one [22].

These are not counterarguments to this paper. They are its forcing functions, and they guarantee that the number of distinct compute targets keeps rising. The underlying driver is the one Hennessy and Patterson identified: with general purpose scaling exhausted, specialization is the remaining source of performance [11].

The supply of people who can deploy that silicon does not move. Cross architecture deployment at a high level is done by a population plausibly numbering in the tens of thousands worldwide, and training one takes the better part of a decade. No rigorous estimate of that population appears in the published literature, and producing one would be a useful contribution in itself. No policy lever moves that curve on a relevant timescale.

There is an irony worth stating in the design direction as well. Systems that design chips optimize placement, routing and power against a model of the chip. What they do not begin from is measured behavior of real silicon under real load. The deployment seat and the design seat draw on the same missing ground truth.

8 Crossing

The model implies that markets do not drift from the low equilibrium to the high one. Something has to move them. The record contains exactly three mechanisms.

A patron. The networking case is the canonical instance and it is usually told wrong. OSI was neutral, better specified and government mandated, and it lost. What separated TCP/IP was that a party with no position in the underlying products funded implementations and gave away a complete working stack [21], effectively free, at a moment when the alternative cost thousands of dollars. Neutrality did not decide it. Patronage did. On that criterion OpenCL sits with OSI, and the substrate layer is currently unpatronized.

A forcing application. AlexNet made the switching cost worth paying regardless of risk, because the alternative was not competing at all. Forcing applications cannot be scheduled.

An actor who absorbs the deployment risk. If C falls on the buyer, a party that contractually takes it back moves the inequality directly. This is the least discussed mechanism and the only one available to a commercial entity acting deliberately.

It is worth naming the reflexive difficulty rather than leaving it to a reader. A reversibility layer requires adoption to reduce p, and reduced p is what drives adoption. The argument of this paper describes a cold start problem and any party attempting to build the layer is inside it. The three mechanisms above are the only exits the record contains. The first two arrive on their own schedule. The third is the only one available to a party acting deliberately.

9 Predictions

A position paper without a checkable claim is indistinguishable from advocacy. Probabilities are the author's.

Prediction 1. By 31 December 2028, no major chip vendor or hyperscaler will have shipped a generally available product whose documented default routes production inference to a named competitor's accelerator. Probability 0.8. Falsified by any such general availability release.

Prediction 2. By 31 December 2030, the share of merchant AI silicon revenue held by vendors outside the incumbent five will exceed ten percent, excluding revenue arising from regulatory exclusion of the default vendor in a closed domestic market. Probability 0.45. This is the direct test of whether the system crosses the threshold.

Prediction 3. By 31 December 2030, the median member of a fixed reference set will run three or more architecturally distinct substrates carrying more than five percent each of a single workload class. The reference set is the fifty United States organizations with the largest disclosed AI accelerator capacity as of 31 December 2027, fixed at that date so the sample cannot be reselected in hindsight. Substrates are architecturally distinct under the functional test in Section 5; two generations of one vendor's family count once. Probability 0.5. Falsified if the median is two or fewer, or if fewer than thirty of the fifty disclose enough to score. This is the test of reversibility rather than of share.

Prediction 4. By 31 December 2028, at least one sovereign or national laboratory procurement above one billion dollars will carry an explicit cross substrate portability requirement. Probability 0.4. This is the patronage test, and it requires no cooperation from any commercial party.

Predictions 2 and 3 both lose if the central claim is wrong. Prediction 1 is the one that would most cleanly falsify the neutrality argument.

10 Limitations

Five weaknesses are worth stating by the author rather than leaving to a reader.

The graveyard selects on the dependent variable. Section 3 examines architectures that failed and infers a common cause. It does not survey architectures that succeeded quietly or failed for unrelated reasons, and no such survey exists to draw on. The pattern should be read as a hypothesis generated by five cases rather than a finding established across a population.

The model is a model. Section 4 gives a structure, not a measurement. No value of α, p or C is estimated here, and the two equilibria result follows from the assumption that p decreases in n, which is plausible and unmeasured. A reader who believes p is independent of prior adoption should discard the section.

The simplest rival explanation is not fully excluded. The plainest account of the revenue distribution is that the dominant vendor simply built better silicon, and this paper does not disprove it. What it offers against that account is the 2006 to 2012 record, in which the winning part contained no feature designed specifically for the workload it won, and the graveyard, in which capable parts failed on the route to them rather than on their arithmetic. A reader who believes the performance gap alone explains a ninety nine to one distribution should weigh those two bodies of evidence and decide.

The exploratory measurement does not reach the mechanism. The ratios in Section 5 are upper bounds from one model on five cards, three self identified. They bear on whether the chip determines the outcome, not on $p(n)$.

The counterfactual is unavailable. The claim that particular architectures would have succeeded given a neutral layer cannot be tested, for the same reason Hooker declined to quantify the hardware lottery. What is offered instead is a mechanism, a set of cases in which it operated, and predictions that can settle the disagreement by observation.

11 Conclusion

The current distribution of compute revenue is not the outcome of competition among architectures. It is the outcome of competition among software ecosystems, decided substantially in 2007 and not seriously revisited since.

That is a different claim from saying the winner was undeserving. The winner paid for the layer, across eighteen years, when nobody else would. The point is that what was purchased was reachability, and that reachability is transferable in a way silicon performance is not.

Two hundred and fifty distinct designs are in the field. One product came out of them. The difference between them was never the chip.

References

[1] AMD. Opteron processor launch, April 2003.

[2] W. B. Arthur. Competing technologies, increasing returns, and lock-in by historical events. *The Economic Journal* 99(394), 1989.

[3] Cerebras Systems Inc. Form S-1 registration statement. U.S. Securities and Exchange Commission, September 2024.

[4] Cerebras Systems Inc. Form S-1 registration statement, revised. U.S. Securities and Exchange Commission, 2026.

[5] A. Coates, B. Huval, T. Wang, D. Wu, A. Y. Ng, B. Catanzaro. Deep learning with COTS HPC systems. *ICML*, 2013, pp. 1337-1345.

[6] J. Dean et al. Large scale distributed deep networks. *NIPS*, 2012.

[7] P. A. David. Clio and the economics of QWERTY. *American Economic Review* 75(2), 1985.

[8] J. Farrell, G. Saloner. Standardization, compatibility, and innovation. *RAND Journal of Economics* 16(1), 1985.

[9] Dell. Intel Extended Memory 64 Technology. White paper, August 2004.

[10] J. A. Fisher. Very long instruction word architectures and the ELI-512. Yale Technical Report, 1983.

[11] J. L. Hennessy, D. A. Patterson. A new golden age for computer architecture. *CACM* 62(2), 2019.

[12] Intel. Statement on the first Larrabee product, December 2009.

[13] M. L. Katz, C. Shapiro. Network externalities, competition, and compatibility. *American Economic Review* 75(3), 1985.

[14] S. Hooker. The hardware lottery. arXiv:2009.06489, 2020.

[15] A. Krizhevsky, I. Sutskever, G. E. Hinton. ImageNet classification with deep convolutional neural networks. *NIPS*, 2012.

[16] Q. V. Le et al. Building high-level features using large scale unsupervised learning. *ICML*, 2012.

[17] J. Nickolls, I. Buck, M. Garland, K. Skadron. Scalable parallel programming with CUDA. *ACM Queue* 6(2), 2008.

[18] NVIDIA. Two million registered developers, countless breakthroughs. NVIDIA Blog, 2020.

[19] NVIDIA. Remarks on the CUDA developer population, Computex, Taipei, June 2024.

[20] NVIDIA. Twenty years of CUDA. GPU Technology Conference, March 2026.

[21] A. L. Russell. *Open Standards and the Digital Age*. Cambridge University Press, 2014.

[22] S. Scott. The mismatch tax. Reckon Research, preprint v1.2, 2026.

[23] L. Seiler et al. Larrabee: a many-core x86 architecture for visual computing. *ACM TOG* 27(3), 2008.

[24] J. E. Stone, D. Gohara, G. Shi. OpenCL: a parallel programming standard for heterogeneous computing systems. *CiSE* 12(3), 2010.