From Signal to Substrate

Architecture, Irreversibility, and the Capital Economics of Custom Silicon

Dana Xiadani
Founder & Chief Architect, Living Cipher
Living Cipher Analysis 001 | August 2026 | Version 1.0

The Hidden Economics of What Becomes Hardware

DOI: 10.5281/zenodo.21958390
Architecture determines where capital becomes irreversible.

Abstract

Hardware partitioning is commonly treated as an engineering decision: identify a computational bottleneck, estimate the performance or energy benefit of specialization, and determine whether the function belongs in software, programmable logic, or fixed silicon.

That framing is incomplete. Moving computation toward dedicated hardware converts flexibility into commitment. Algorithms, dataflows, security assumptions, workload forecasts, interfaces, manufacturing dependencies, and supplier choices begin accumulating non-recurring engineering cost, verification effort, package constraints, capacity reservations, inventory exposure, and lifecycle obligations.

Architecture is therefore also a capital-allocation discipline. This paper proposes a framework for deciding what deserves to become hardware by examining four questions: whether specialization creates measured technical advantage; whether the underlying computation is sufficiently persistent; whether risk-adjusted lifetime economics justify greater irreversibility; and whether the resulting implementation is manufacturable, supplyable, trustworthy, and deployable.

The resulting question is not simply what can become silicon. It is what deserves to become irreversible.

1. The Hardware Question We Keep Asking Too Late

Most hardware programs are described as engineering problems until the moment somebody has to fund one.

Then a decision that sounded technical, such as moving a function into silicon, retaining another in software, splitting an accelerator into chiplets, or selecting a process node, becomes something else entirely. It becomes a capital-allocation decision.

Moving computation toward dedicated hardware exchanges flexibility for commitment. An algorithm, dataflow, security assumption, workload forecast, and architectural boundary that could still change relatively cheaply in software begin becoming attached to RTL, verification, physical design, IP licenses, masks, packaging, test, qualified suppliers, inventory, and lifecycle obligations.

That conversion can create enormous value. It can also manufacture the wrong assumption remarkably efficiently.

The usual question is therefore incomplete. The question is not simply whether a computation can be accelerated in hardware. It is whether that computation has earned the right to become hardware.

A workload may demonstrate excellent acceleration and still be a poor candidate for fixed silicon if its algorithmic life is shorter than the intended hardware lifecycle. A technically elegant accelerator may become economically unattractive once memory movement, package complexity, or infrastructure power is included. A process-node transition may improve density while worsening the business case because verification, yield, package, capacity, or useful lifetime do not support the additional commitment.

The same distinction applies to modularity. A system does not become economically superior merely because it has been divided into chiplets. Partitioning creates additional decisions around die-to-die communication, testing, packaging, power delivery, thermal behavior, and assembly. Cost-aware chiplet research shows that the preferred partition can change with yield, technology assignment, interconnect, test, assembly, substrate, and packaging assumptions.[3][4]

What matters is not the elegance of an implementation category but the relationship between a computation and the consequences of making it physical.

Software leaves many assumptions mutable. Programmable hardware reduces some forms of flexibility while creating deterministic and parallel execution. Fixed silicon creates greater specialization and can deliver substantial efficiency, performance, and scale, but it also raises the cost of being wrong.

Architecture should therefore begin one step earlier than implementation: What is sufficiently valuable, sufficiently persistent, and sufficiently evidenced to justify becoming expensive to change?

That is a different question from whether something fits. It is the question that connects computation to capital.

2. Architecture Is Where Capital Becomes Irreversible

Semiconductor architecture is often discussed as a sequence of technical partitions: hardware versus software, compute versus memory, host versus accelerator, monolithic versus disaggregated. Those boundaries are also economic boundaries.

Each partition determines where engineering effort, tooling, verification, qualification, manufacturing, inventory, and infrastructure begin accumulating around a technical assumption.

At advanced nodes, the magnitude of that commitment can be extraordinary. Semiconductor Industry Association material reports that the estimated cost of designing a leading-edge chip rose from about $30 million at 65 nm to more than $540 million at 5 nm.[1] These figures are estimates rather than universal program budgets, but they illustrate the scale of commitment that can follow an architectural decision.

The implication is straightforward: The evidentiary burden should rise with the irreversibility of the implementation.

An organization should demand more evidence before committing a volatile computation to fixed silicon than before deploying the same computation in software. It should demand more evidence before introducing a new package dependency than before retaining an established board-level interface. It should demand more evidence before reserving long-lived manufacturing capacity than before running an experiment on programmable hardware.

The economic test should therefore compare the fixed implementation with the best less-irreversible alternative over the life of the program, not only at the component level.

$$NPV_{fixed} - NPV_{flexible} = \Sigma[(Q_t \times \Delta CM_t - \Delta OPEX_t) / (1 + r)^t] - \Delta NRE - E[L] \geq H$$

where Q_t is economically capturable volume in period t, Delta CM_t is the contribution-margin advantage per unit of the fixed implementation, Delta $OPEX_t$ captures operating-cost differences such as power, cooling, infrastructure, support, or maintenance, r is the required discount rate, Delta NRE is incremental

development and implementation cost, E[L] is expected loss from redesign, stranded inventory, capacity commitments, requalification, or other failures of the underlying assumptions, and H is the organization's required economic hurdle for accepting the additional irreversible commitment.

Break-even occurs when the NPV advantage equals zero. The economic decision threshold is higher. It is reached only when the risk-adjusted NPV advantage is large enough to satisfy the organization's required return and downside-risk tolerance.

This distinction matters because a rational organization should not commit to fixed silicon simply because expected NPV is one dollar above the flexible alternative. The decision must compensate for uncertainty, capital exposure, schedule risk, and the option value surrendered when the design becomes harder to change.

Replacement radius enters this model through expected loss. A change that can be localized to one die, one programmable region, or one software component creates a different downside profile from a change that strands a package, board, qualification program, capacity reservation, and inventory simultaneously.

The central architectural problem is therefore not simply efficiency. It is where to locate irreversibility.

Exhibit 1. The Commitment Ladder

Implementation	Relative specialization	Relative reversibility	Typical reason to use it
Software	Low	High	Requirements or algorithms remain volatile
FPGA / adaptive hardware	Moderate	Moderate-high	Hardware advantage matters but implementation still needs to evolve
Modular fixed silicon	High	Moderate-low	Stable functions can be hardened while other regions evolve independently
Monolithic fixed silicon	Very high	Low	Workload and architecture are sufficiently durable to justify maximum commitment

The best implementation is the one whose degree of commitment matches the evidence supporting the computation.

Architecture determines not only how computation executes. It determines where capital becomes irreversible.

3. The Feasible Architecture Is Larger Than PPA

Performance, power, and area remain essential architectural quantities. They are no longer a complete description of the feasible design space.

For many advanced systems, process technology, package architecture, supply availability, energy infrastructure, and public policy feed backward into architecture rather than appearing only after the architecture has been completed.

Process and foundry. Different functions benefit differently from process scaling. Digital logic, memory structures, analog circuitry, I/O, high-voltage functions, and specialty devices do not necessarily share the same optimum process. UCIe materials describe systems that mix process technologies and explain that I/O circuits do not scale well to advanced nodes, creating an economic and technical rationale for heterogeneous integration.[2]

That means process selection can alter partitioning itself. Foundry choice is therefore not always a downstream procurement exercise. In sufficiently constrained systems, the capabilities and availability of the manufacturing path can determine which architecture is realizable.

Packaging. A partition that appears attractive at the logical level may become unattractive when die-to-die signaling, assembly, test, interposer area, substrate choice, power delivery, thermal density, and package yield are included. CATCH and ChipletPart both show that packaging, test, I/O, substrate, and technology-assignment assumptions can change the economically preferred partition.[3][4]

Packaging is therefore not merely the cost of realizing a partition. In heterogeneous systems, packaging can change which partition is economically rational.

The scale of current industry investment reinforces the point. TSMC says its Arizona expansion plans include six semiconductor wafer fabs, two advanced-packaging facilities, and an R&D center. Its first Arizona facility began 4 nm volume production in the fourth quarter of 2024.[5] Fabrication and advanced packaging increasingly belong to the same capacity conversation.

Supply. If changing a memory technology, substrate, package, foundry, interface component, or supplier would require redesign, re-verification, or requalification, supply is no longer merely a purchasing concern. The architecture has inherited the dependency.

GAO reported in December 2025 that about three-quarters of semiconductor manufacturing and packaging occurred in Asia as of 2022 and that Commerce-funded projects are intended to address vulnerabilities across manufacturing, materials, and packaging. Commerce estimated that funded projects could move the U.S. share of global leading-edge logic manufacturing from 0 percent in 2022 to 20 percent by 2030.[6] These are dated baselines and program estimates, not permanent descriptions of capability.

Geographic concentration, supplier concentration, and capacity availability can therefore alter which architectures are commercially realizable.

Policy and market access. Industrial policy does not become an architectural component. It can, however, alter the feasible design space. In January 2026, the U.S. Bureau of Industry and Security revised licensing treatment for Nvidia H200-, AMD MI325X-, and similar-class accelerators exported to China, providing case-by-case review under specified conditions.[7] Nothing about the underlying transistor changed. The economically accessible market changed because the external constraint changed.

A technically optimal architecture whose critical manufacturing path or target market becomes unavailable may cease to be economically optimal without a transistor changing.

Power. Power economics no longer terminate at the chip boundary. Berkeley Lab's June 2026 update estimates that data centers could account for 11.8 percent of total U.S. electricity consumption by 2030 in its reference case, with modeled scenarios from 9.5 to 15.3 percent.[8] At those scales, architectural efficiency propagates into rack density, cooling, power delivery, facility utilization, and infrastructure capital.

A semiconductor therefore cannot always be evaluated economically by looking at the semiconductor alone. The feasible architecture may include the building around it.

4. Heterogeneity Without Romance

Chiplets have become one of the industry's important responses to scaling, specialization, and heterogeneous integration. They deserve neither dismissal nor romance.

Disaggregation can enable reuse, process heterogeneity, modular product families, systems that exceed monolithic reticle limits, and independent evolution of some functions. UCIe is designed to support standardized die-to-die integration and mix-and-match chiplet construction.[2]

But disaggregation also introduces costs that monolithic systems do not carry in the same way: additional interfaces, die-to-die PHYs, package routing, assembly, known-good-die requirements, test complexity, power-delivery challenges, latency, and thermal interactions.[2]

The correct question is not whether chiplets are cheaper. It is which functions benefit from being independently implemented, and whether that independence justifies the cost of the boundaries between them.

One way to make that question more concrete is replacement radius.

Replacement radius is the portion of an implemented system that must be redesigned, requalified, or discarded when a material assumption changes.

Consider a system containing several functions. Some may be persistent: deterministic data movement, protocol framing, memory control, or stable signal-processing operations. Others may remain volatile: a

rapidly changing accelerator, model-specific datapath, customer-specific interface, or policy-dependent function.

If all functions are committed to a single fixed implementation, a change in one volatile assumption can force a disproportionately large redesign. A modular architecture may isolate that uncertainty.

The potential economic value of modularity is therefore not solely that individual dies may be cheaper to fabricate. Modularity can reduce the replacement radius of future change.

This is not an argument that chiplets always win. A smaller replacement radius may be economically worthless if achieving it introduces unacceptable latency, package complexity, thermal problems, test cost, power overhead, or development burden.

Replacement radius is instead another quantity to expose during architecture review: If this assumption changes, how much of the physical system changes with it?

Exhibit 2. Matching Implementation to Uncertainty

Implementation	Attractive when	Principal risk
Software	Algorithms and requirements are changing rapidly	Continuing performance or energy cost
FPGA / adaptive hardware	Hardware acceleration is valuable but requirements remain mutable	Power, area, and recurring device cost
Monolithic ASIC	Workload is persistent and scale strongly rewards specialization	Large replacement radius if an assumption fails
Chiplet / modular fixed silicon	Different functions have different rates of change or process needs	Package, test, thermal, and die-to-die integration overhead

Heterogeneity is therefore not an end state. It is a way of deciding which commitments should share the same lifetime.

5. Trust Determines Admissibility

Trust is often treated as a security feature layered onto a completed system. For consequential hardware, that is too late.

In this framework, trust is the evidence required to admit a signal, component, computational resource, or result into a consequential decision path.

That definition deliberately separates trust from cryptography. NIST defines digital signatures as mechanisms that provide services including origin authentication, data integrity, and signatory non-repudiation.[9] Those are powerful properties. They do not, by themselves, establish that the physical assertion represented by the signed information was true.

A measurement can be faithfully preserved after the wrong thing was measured. A spoofed physical source can generate data that is later transmitted without alteration. A compromised component can participate in an otherwise cryptographically protected system if the trust boundary admits it before its integrity is established.

The inference is important: Cryptographic integrity can establish that information was not altered and can provide assurance about its asserted source. It does not, by itself, establish that the physical assertion represented by that information was true.

For hardware architecture, this matters because trust requirements can change admissibility. If a deployment requires evidence of component provenance, device state, authentic firmware, authorized

execution, or supply-chain integrity, those requirements can affect device choice, boot architecture, partitioning, component sourcing, lifecycle management, and qualification.

NIST's work on validating computing-device integrity explicitly addresses cyber-supply-chain integrity and provenance by demonstrating verification that acquired device components are genuine and have not been tampered with.[10]

Trust therefore belongs on the architectural decision plane when the market or mission requires evidence before a component or computation can participate. An architecture that cannot satisfy the evidence requirements of its intended deployment may be economically inadmissible regardless of its technical performance.

6. Evidence Before Fabrication

A decision framework for hardware should not produce a mystical score announcing that a computation is 83 percent worthy of becoming an ASIC. The relevant quantities are too dependent on workload, lifetime, volume, manufacturing path, and deployment context for a universal weighted score to be credible.

A more useful framework is sequential. A candidate computation must clear four gates.

Gate 1: Measured advantage. First ask whether specialization produces a material improvement on the actual workload. The relevant metric may be energy, latency, throughput, determinism, memory traffic, bandwidth efficiency, reliability, size, or another mission-specific constraint. A hardware class should not be justified by folklore about how much faster an ASIC is than an FPGA, or how much more efficient an FPGA is than a CPU. Those relationships depend on workload, implementation, precision, memory behavior, process, clocking, and utilization.

The correct question is: What advantage did specialization demonstrate on the workload this system must actually execute? If the benefit cannot be measured, there is not yet sufficient evidence to increase irreversibility.

Gate 2: Persistence. Technical advantage is not enough. The computation must also be sufficiently persistent. How often has the algorithm changed? Is the model architecture still evolving? Will precision requirements move? Are protocols stable? Will the sensors, interfaces, or data formats remain compatible with the proposed hardware life?

A workload can be technically ready for hardware before it is economically ready for irreversibility. If the benefit of specialization is real but the computation remains volatile, adaptive hardware may preserve valuable optionality while continuing to produce architectural evidence.

Gate 3: Risk-adjusted lifetime economics. The relevant test is whether specialization generates enough discounted lifetime value to exceed both break-even and the organization's required return on the additional irreversible capital. The analysis should include expected volume by period, contribution-margin advantage, power and infrastructure economics, development cost, schedule, package, test, yield, qualification, inventory, expected useful life, commitment exposure, and E[L], the expected economic loss when the assumptions supporting the architecture fail.

The economic threshold is project-specific. It should be derived from the economics of the best available alternative, not assumed from an industry rule of thumb. Volume alone does not set the threshold. Timing, margin, workload durability, capital commitments, downside scenarios, and replacement radius all matter.

Gate 4: Implementation admissibility. Finally, ask whether the implementation can actually be fabricated, packaged, supplied, qualified, trusted, and sold. Are there adequate manufacturing paths? Does the package exist at the required cost and capacity? Are memory and substrates available? Does the architecture depend on a single constrained supplier? Will the deployment accept the component provenance and security model? Can the product be built in the regions and markets assumed by the business case?

An architecture that cannot clear this gate is not rescued by excellent PPA.

The resulting decision logic is simple. If a computation fails the measured-advantage gate, leave it flexible. If it clears technical advantage but fails persistence, retain programmability or reconfigurability. If it clears advantage, persistence, risk-adjusted lifetime economics, and implementation admissibility, fixed silicon becomes defensible. If different portions of the system clear these gates differently, evaluate modular or heterogeneous partitioning, while measuring both integration overhead and replacement radius.

This is not a formula for selecting a substrate. It is a discipline for determining how much evidence should exist before increasing commitment.

7. What Must Remain True?

Every custom-silicon investment embeds a forecast.

It forecasts demand. It also forecasts the persistence of a workload, the availability of manufacturing capacity, the durability of customer requirements, the adequacy of packaging and memory, the economics of power, and the continued accessibility of suppliers and markets.

Technical diligence should therefore ask more than whether the chip is plausible. It should ask: What must remain true for the capital committed to this architecture to earn its return?

What technical assumption is being capitalized? "AI is growing" is not an architectural assumption. A useful diligence question is more specific: Which model behavior, protocol, precision, memory pattern, interface, latency requirement, or customer workflow must remain sufficiently stable for the proposed specialization to retain its advantage? And for how long? A company that cannot identify the assumption it is hardening cannot assess the risk of hardening the wrong one.

What durable volume actually underwrites the design? Total addressable market is not production volume. The relevant number is the durable, economically capturable demand available while this implementation remains useful. That means examining customer commitments, design wins, attach rates, expected product life, ramp timing, customer concentration, and downside scenarios.

The correct question is not simply how large the market is. It is how much durable demand this particular architecture can capture before its assumptions expire.

What becomes committed before demand is proven? This may be one of the most consequential questions in semiconductor investing. NVIDIA's fiscal 2026 filing states that it has experienced supply lead times exceeding twelve months and has used premiums, deposits, long-term supply agreements, and capacity commitments to secure future supply. It also warns that inaccurate demand estimates can create mismatches that cannot always be unwound quickly.[11]

AMD reported approximately $25.7 billion of unconditional commitments as of March 28, 2026, primarily relating to wafers, substrates, components, cloud arrangements, and software and technology licenses.[12] Marvell reports wafer obligations and capacity-reservation arrangements with foundries and test-and-assembly suppliers, including agreements with terms ranging from four to ten years.[13]

The lesson is not that such commitments are imprudent. They may be essential to securing supply. The diligence question is: How much money becomes difficult to reverse before management knows whether its demand forecast was correct?

Forecast risk can become balance-sheet risk.

Where is the actual bottleneck? An investor should never accept "we have foundry capacity" as a complete manufacturing answer. The constrained resource may be wafer capacity, HBM, substrates, package assembly, test, power, qualification, or export access. The architecture-to-revenue chain is only as strong as its narrowest dependency.

What would cause management not to build the chip? This may be the most revealing diligence question of all. A credible architecture thesis should contain its own kill conditions.

If expected lifetime demand cannot support the required return after development cost, operating economics, time-to-market, commitment exposure, and downside scenarios are included, the architecture has not crossed the economic decision threshold for fixed silicon. If workload changes invalidate the specialized datapath before design freeze, retain programmability. If package cost or availability destroys system economics, repartition. If power exceeds the facility economics of the intended deployment, reconsider the architecture.

The exact thresholds differ by company. The discipline is requiring them to exist.

A semiconductor investment thesis is incomplete until management can describe not only why custom silicon should win, but what evidence would cause it not to build the chip.

Exhibit 3. Attack Every Arrow

Workload assumption -> Architecture -> Foundry / package / memory dependency -> Capacity commitment -> System economics -> Customer economics -> Revenue and margin thesis

Technical diligence should interrogate every transition.

A technically plausible chip does not automatically produce an economically plausible investment.

Technical diligence tests whether the technical assumptions underlying projected revenue, margins, capacity commitments, and capital requirements are sufficiently durable to support the investment thesis.

Conclusion: What Deserves to Become Irreversible?

The semiconductor industry is extraordinarily good at optimizing designs after the important decisions have been made. The harder problem occurs earlier.

Before place and route, before masks, before packaging, and frequently before the final workload has stabilized, somebody must decide which assumptions are sufficiently durable to manufacture.

That decision cannot be reduced to performance per watt. It requires a view across computation, architecture, data movement, energy, manufacturing, packaging, supply, trust, policy, lifecycle, and capital.

The objective is not to resist specialization. Specialization is one of the principal reasons hardware can create extraordinary performance, energy efficiency, and economic value.

The discipline is to match the permanence of the implementation to the strength of the evidence supporting it.

Software preserves optionality. Adaptive hardware allows some commitments to be tested before they become permanent. Fixed silicon can harvest the value of durable computation. Modularity can

sometimes isolate uncertainty so that one changing assumption does not invalidate an entire physical system.

But every movement toward specialization creates a corresponding obligation to ask what must remain true.

The most expensive hardware mistake is therefore not necessarily a design that fails.

It is a design that works exactly as intended after the assumptions that justified building it have stopped being true.

The discipline is not merely to accelerate computation.

It is to decide what deserves to become irreversible.

References

1. Semiconductor Industry Association, "Chip Design and R&D." https://www.semiconductors.org/policies/chip_design/

2. Stefan Rusu, "How UCIe Will Enable Broad Chiplet Adoption," UCIe Consortium, March 31, 2023. https://www.uciexpress.org/post/how-ucie-will-enable-broad-chiplet-adoption

3. Alexander Graening, Jonti Talukdar, Saptadeep Pal, Krishnendu Chakrabarty, and Puneet Gupta, "CATCH: a Cost Analysis Tool for Co-optimization of chiplet-based Heterogeneous systems." https://arxiv.org/abs/2503.15753

4. Alexander Graening, Puneet Gupta, Andrew B. Kahng, Bodhisatta Pramanik, and Zhiang Wang, "ChipletPart: Scalable Cost-Aware Partitioning for 2.5D Systems." https://arxiv.org/abs/2507.19819

5. Taiwan Semiconductor Manufacturing Company Limited, 2025 Annual Report. https://investor.tsmc.com/static/annualReports/2025/english/index.html

6. U.S. Government Accountability Office, GAO-26-107882, "Semiconductors: Information on Projects Funded to Strengthen U.S. Supply Chain." https://files.gao.gov/reports/GAO-26-107882/index.html

7. U.S. Bureau of Industry and Security, "Department of Commerce Revises License Review Policy for Semiconductors Exported to China," January 13, 2026. https://media.bis.gov/press-release/department-commerce-revises-license-review-policy-semiconductors-exported-china

8. Lawrence Berkeley National Laboratory, "United States Data Center Energy Usage Report: 2025 Update," June 2026. https://eta-publications.lbl.gov/publications/united-states-data-center-energy-2025

9. NIST CSRC Glossary, "Digital Signature." https://csrc.nist.gov/glossary/term/digital_signature

10. NIST SP 1800-34, "Validating the Integrity of Computing Devices," December 2022. https://csrc.nist.gov/pubs/sp/1800/34/final

11. NVIDIA Corporation, fiscal 2026 Form 10-K. https://www.sec.gov/Archives/edgar/data/1045810/000104581026000021/nvda-20260125.htm

12. Advanced Micro Devices, Inc., Form 10-Q for quarter ended March 28, 2026. https://www.sec.gov/Archives/edgar/data/2488/000000248826000076/amd-20260328.htm

13. Marvell Technology, Inc., fiscal 2026 Form 10-K. https://www.sec.gov/Archives/edgar/data/1835632/000183563226000011/mrvl-20260131.htm

Author Note

Dana Xiadani is Founder & Chief Architect of Living Cipher, where she works at the boundary of computation, architecture, trust, and physical realization. This paper presents a public analytical framework. It does not disclose Living Cipher proprietary scoring methods, implementation mechanisms, or protected architecture.

Suggested citation: Xiadani, Dana. 2026. "From Signal to Substrate: Architecture, Irreversibility, and the Capital Economics of Custom Silicon." Living Cipher Analysis 001, Version 1.0. DOI: 10.5281/zenodo.21958390.