A Guide to Cost-Effective ASIC Design with Open-Source Tools

Business Case for New Start Ups in 2026

Ji Won, Naveed A.A.
PakCrypt NPO

1 Executive Summary

Custom silicon is no longer the exclusive domain of big corporations. Two historic barriers — the capital cost of owning a fab and the six-figure annual price of proprietary design software — have effectively collapsed. A startup with FPGA experience can now design a production-grade chip on a free, open-source toolchain and have it manufactured by a world-class foundry through shared-wafer services. The lessons below distil what founders need to know before taking that step.

- **No fab, no software lock-in:** the fabless model removes fabrication capex, while Multi-Project Wafer (MPW) shuttles let dozens of startups share a single mask set.
- **A complete open-source flow:** tools such as Yosys, OpenROAD, Verilator, and KLayout carry a design from RTL all the way to a manufacturable GDSII layout.
- **Transferable skills:** FPGA front-end expertise (Verilog, verification, timing constraints) transfers directly — only the physical back-end flow is genuinely new.
- **A proven "sweet spot" node:** 28nm planar offers the best convergence of cost, performance, and manufacturability for embedded, AI-edge, automotive, and IoT chips (up to ~1.5–1.8 GHz).
- **Validated at scale:** the open-source Titan-I RISC-V vector processor was designed, verified, and synthesised with this flow down to 7nm.

You can now embark on the journey of proprietary silicon with a budget of $100–200K

Item	Typical cost (28nm)
MPW prototype slot (4 mm^2, ~40–50 dies)	~$52K – $64K (TSMC); SMIC ~30% lower
Broker sign-off verification	1–3 runs often included; extras $3K–$6K each; full package $15K–$35K
Commercial EDA (cloud, startup pricing)	~$20K – $35K for the project lifecycle
Prototype packaging (first ~50 units)	$4.5K – $8K

A Phased Path to Production: Prove the design on an MPW shuttle to validate it in real silicon and gather yield data; refine via a multi-layer mask (MLM) run for low-volume units; and commit to a full production mask set only once functionality and timing are proven. This retires technical risk before serious capital is spent.

The bottom line. The barriers are down and the economics are favourable, but success rewards discipline over improvisation. Founders who pair transferable FPGA expertise with sound financial and supply-chain planning can now build differentiated, workload-optimised silicon that was, until recently, beyond the reach of all but the largest firms. The tools, the foundry access, and the business model are all in place — what remains is the decision to begin.

2 Introduction

The global semiconductor landscape is undergoing a profound structural shift, driven by a confluence of technological advancements and compelling economics that have **lowered the barriers to entry** for creating custom silicon. This whitepaper will provide a detailed guide for startups in navigating this path, from selecting the optimal manufacturing technology to executing a complete design flow using open-source **Electronic Design Automation (EDA)**. We will also look into understanding the "financial equations" for bringing a custom chip to life.

2.1 IC Manufacturing before Fabless

Historically, the domain of custom integrated circuit (IC) design was an exclusive preserve of large technology corporations capable of shouldering the colossal capital expenditures required to build and operate semiconductor fabrication facilities, or "fabs".

Constructing a state-of-the-art fab for leading-edge technologies now requires investments upwards of US$ 15B-20B, encompassing multi-story cleanroom environments, intricate global supply chains for specialized materials, and the acquisition of ultra-expensive lithography machines, such as those used for Extreme Ultraviolet (EUV) processing. This immense upfront capital requirement created a formidable moat, preventing all but the well-funded entities from participating in the creation of integrated circuits (ICs).

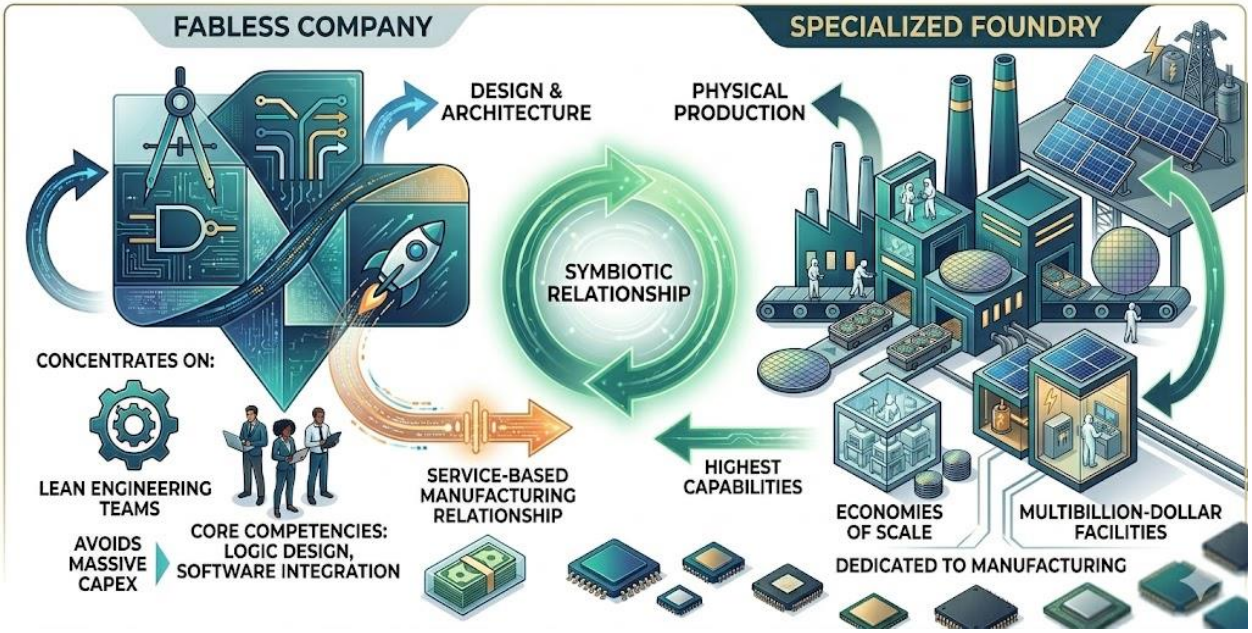
2.2 EDA Tools: Commercial to open source

For decades, EDA software (required to design an ASICs) was the exclusive domain of a few large, publicly-traded corporations, whose tools came with correspondingly **high license fees**. Among these, Synopsys, Cadence Design System, and Siemens EDA are most famous. These proprietary suites, often costing hundreds of thousands of dollars per engineer per year, formed a significant part of the economic barrier that prevented many smaller companies and academic institutions from entering the field of custom silicon design.

However, a quiet revolution has been underway, culminating in the development of a robust, fully-integrated, **open-source EDA ecosystem** capable of supporting a complete **RTL-to-GDSII** design flow for production grade chips . This development has been a critical enabler for the growth of the fabless model, removing the prohibitive software licensing costs.

2.3 RTL to GDSII

RTL (Register Transfer Level) is a hardware design abstraction used to describe a chip's behavior in languages such as Verilog or VHDL, specifying how data moves between registers and how logic operations occur at each clock cycle; it is essentially the human-readable blueprint of a digital circuit before physical implementation.



GDSII (pronounced "G-D-S two") stands for Graphic Data System II, a file format containing the final geometric layout of the chip—every transistor, wire, metal layer, and mask shape required for fabrication. In an RTL-to-GDSII flow, the design starts as an abstract functional description (RTL) and passes through synthesis, placement, routing, timing checks, and physical verification until it becomes a manufacturing-ready layout file that a semiconductor foundry can use to fabricate silicon.

2.4 Fabless ASIC Business Model

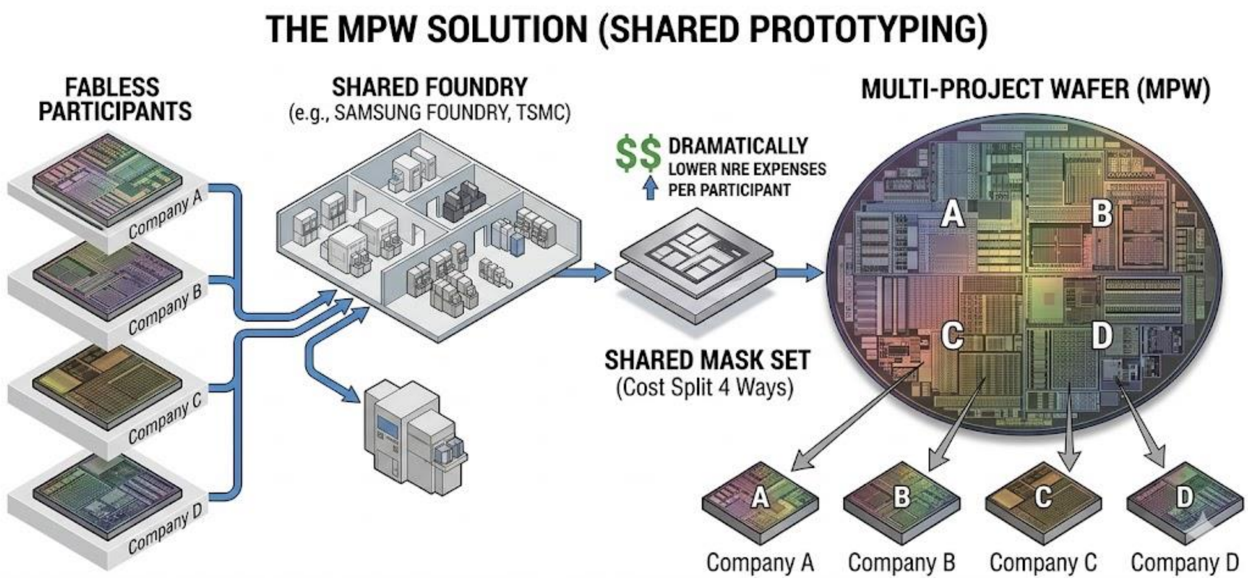
This paradigm has been decisively reshaped by the rise of the fabless semiconductor business model. A fabless company strategically outsources the entire physical production of its chips to specialized, independent foundries, allowing it to concentrate its resources and expertise on the core competencies of "architecture, logic design, and software integration".

The fabless entity avoids the massive capital outlay of owning a fab while gaining access to best-in-class manufacturing capabilities through a **service-based relationship**. The foundry, in turn, achieves economies of scale by dedicating its multibillion-dollar facilities exclusively to manufacturing, serving a diverse clientele ranging from established giants to agile startups.

The emergence of robust, open-source EDA toolchains has effectively eliminated the second major barrier to entry—the prohibitive cost of proprietary software licenses, which could run into hundreds of thousands of dollars per engineer annually. This convergence of the fabless business model and accessible design tools has created a fertile ground for innovation, empowering startups to compete in niche markets.

2.5 From FPGA to ASIC Engineering

For most of computer engineering professionals, who are experienced in Xilinx FPGA-based development, this transition represents a natural and logical evolution in their technological journey.



The skills acquired in defining system constraints, writing Register Transfer Level (RTL) code in HDL like Verilog, and performing functional verification are directly transferable to the world of Application Specific Integrated Circuit (**ASIC**) design.

The primary distinction lies in the back-end physical design and verification processes, which govern how the abstract logic is translated into a physical layout on a silicon die.

2.6 Large Fabs are Supporting Fabless Model

For computer engineers, understanding the economic leverage of large fabs begins with the concept of non-recurring engineering (NRE) costs. In a typical ASIC design flow, mask sets for leading-edge nodes (e.g., 5nm or 7nm) can run into tens of millions of dollars, with over 70 distinct masks required per design. A fabless company—lacking its own fabrication lines—cannot amortize this expense across high-volume internal production. Large fabs like TSMC and Samsung Foundry solve this by aggregating multiple fabless designs onto a single Multi-Project Wafer (MPW). Instead of each customer paying for an entire mask set, the MPW service partitions the reticle area, allowing up to 40–60 different designs to share the same physical masks. This substantially reduces the cost for startups.

The proliferation of fabless model is reflected in the market structure. TSMC, for example, accounted for 34% of the pure-play[1] foundry industry in 2024. Its Open Innovation Platform® (OIP) is a cornerstone of this strategy, providing customers with process design kits (PDKs) and extensive technical support to streamline the design-to-manufacturing handoff and optimize yields.

Similarly, SMIC, China's largest contract chip maker, leverages heavily subsidized domestic capacity expansions to aggressively price mature nodes, making it a highly competitive option for startups

[1] Pure-play refers to a business model within the semiconductor industry where a company functions exclusively as a contract chip maker. These foundries do not design or sell their own branded chips.

focused on strict unit-cost optimization. The existence of these platforms and services ensures that a fabless team, regardless of its size, can tap into world-class manufacturing ecosystems without needing to build one themselves.

2.7 Lessons for Startups

The fabless model has replaced enormous upfront capital investments in fabrication facilities with targeted spending on design, prototyping, software development, and market execution. For startups, this fundamentally changes the economics of innovation. Now startups can direct their energy and capital toward developing unique intellectual property, understanding customer needs, and rapidly refining products. Through specialized design partners, foundry ecosystems, and an expanding suite of capable open-source tools, startups can now access capabilities that were once available only to the largest semiconductor firms.

3 Technology Tour

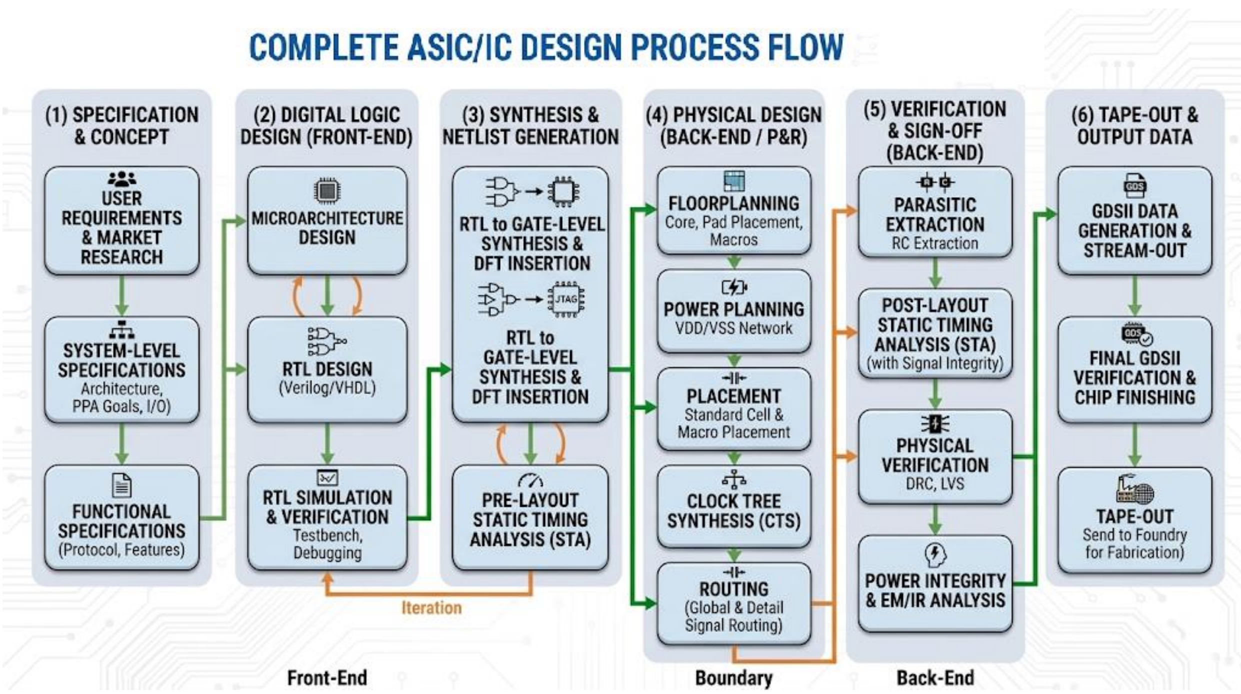
As mentioned earlier, a combination of powerful, freely available tools, the increasing accessibility of Process Design Kits (PDKs), and Multi Project Wafers (MPW) have enabled a cost-effective pathway to ASIC design. For startups and small engineering teams, it is now possible to establish a complete, state-of-the-art design environment with moderate budget of $100-200K.

3.1 Transitioning from Xilinx FPGAs to ASICs

To excel as a Xilinx FPGA developer, an engineer must master far more than just front-end Register-Transfer Level (RTL) coding. True competency requires deep fluency in the vendor ecosystem—specifically the Xilinx Vivado Design Suite—and the specialized art of mapping logic onto a rigid, pre-existing silicon fabric. A skilled developer must expertly optimize resource utilization across finite Lookup Tables (LUTs), Block RAMs (BRAMs), and DSP slices, while driving timing closure via Xilinx Design Constraints (XDC). Hardware-level debugging relies on virtual, in-system instrumentation like the Integrated Logic Analyzer (ILA). Ultimately, the FPGA environment is an exercise in optimization within a structured, forgiving sandbox where design oversights can be resolved with a quick bitstream re-upload—however no such re-uploads possible once the ASIC is in hand.

Now, transitioning from this FPGA landscape to custom ASIC development marks a fundamental shift from a reconfigurable canvas to a fixed, unalterable silicon geometry. While the front-end process of writing and verifying RTL remains identical, the **RTL-to-GDSII transformation** strips away the structural safety nets of the FPGA.

On an FPGA, routing tracks, clock trees, and IO blocks are pre-fabricated; the tool simply routes signals through existing hardware switches. In an ASIC, however, every single wire, buffer, transistor,



and clock distribution network must be synthesized, placed, and routed entirely from scratch onto a blank die.

The modern open-source EDA stack consists of modular, specialized tools orchestrated to automate this complex sequence of tasks required to transform a high-level hardware description into a physical geometric layout.

3.2 The Foundation: Process Design Kits (PDKs)

The viability of an open-source toolchain rests upon the Process Design Kit (PDK). A PDK is a collection of technology files provided by a semiconductor foundry that contains the foundational data required to design manufacturable circuits for a specific process node. This includes transistor models, geometric design rules, parasitic extraction data, and pre-characterized standard cell libraries.

Historically, PDKs were treated as highly confidential intellectual property, accessible only under strict Non-Disclosure Agreements (NDAs). The open-source hardware movement gained momentum when foundries began releasing open PDKs for mature process nodes:

- SkyWater Technology 130nm (SKY130): Released as a free, fully open-source PDK, serving as the foundational proving ground for modern open-source EDA tools.
- GlobalFoundries 180nm (GF180MCU): Made publicly available to target low-cost, mixed-signal, and microcontroller-focused designs.

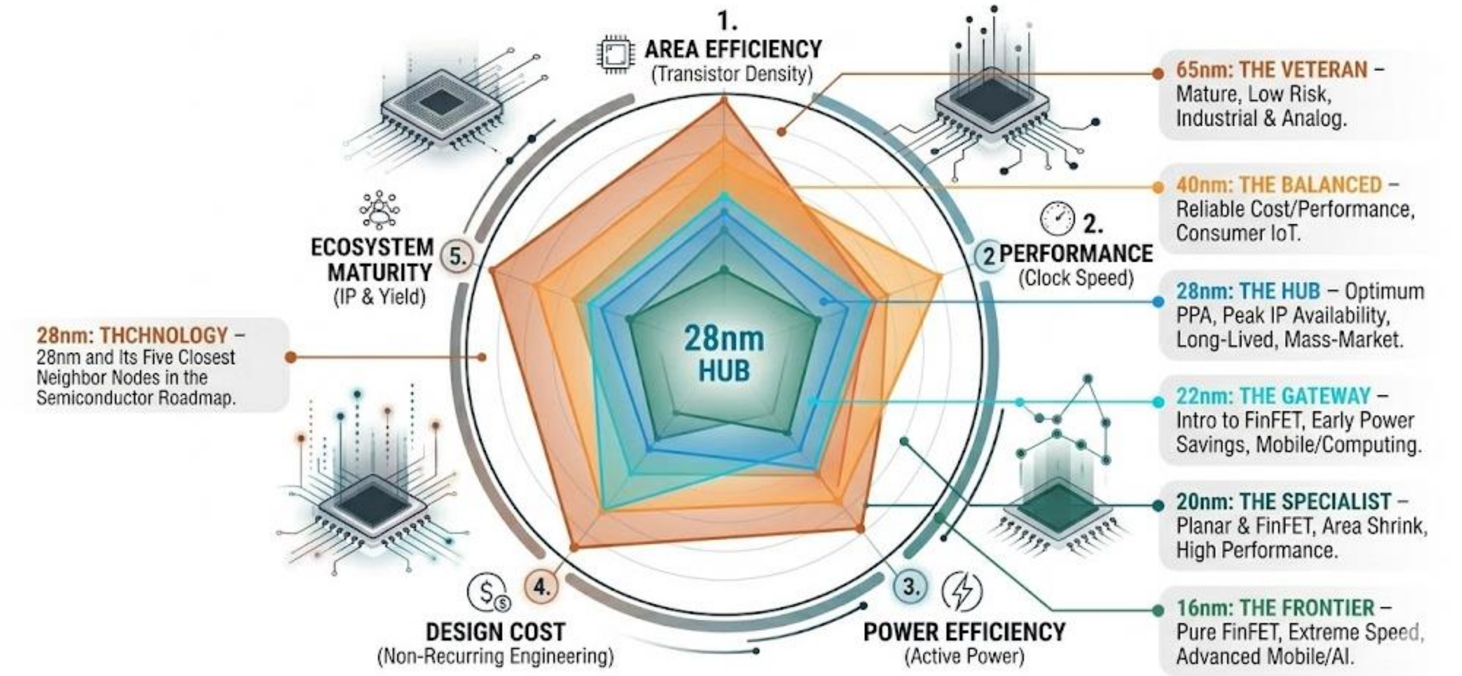
While commercial foundries (such as TSMC or SMIC) rarely release open-source PDKs for advanced nodes (e.g., 28nm and below) due to proprietary business models, access for open-source workflows can be achieved through university partnerships, government programs, targeted research collaborations, or subscription to brokers.

3.3 Selection of Fab Technology

The selection of a semiconductor manufacturing technology node—the physical dimensions of the transistors on a chip, typically measured in nanometers—is arguably the most consequential commercial decision a fabless company makes. This choice dictates the chip's performance potential, power consumption, transistor density, and, most critically, its cost of production. While the industry narrative often pushes towards ever-smaller nodes for higher performance and miniaturization, a pragmatic analysis reveals that for a vast range of applications, including the use case of 1GHz RISC-V embedded processor, the 28nm node represents a "sweet spot" where cost, performance, and manufacturability converge optimally.

3.3.1 Advantage of 28nm for Start Ups

The fundamental reason for the 28nm node's enduring appeal lies in the stark economic divide between mature planar CMOS technology and the more advanced 3D FinFET geometries used in nodes smaller than 28nm . The 28nm process utilizes a planar transistor architecture, where the gate sits flat on top of the channel. This design is a highly evolved and perfected version of CMOS technology.



In contrast, as transistor dimensions shrink below 28nm (e.g., 20nm, 16nm, 12nm, 7nm), planar structures suffer from severe short-channel effects, primarily leakage current, where electrons tunnel through the increasingly thin gate oxide. To combat this, foundries adopted a 3D transistor structure known as FinFET (Fin Field-Effect Transistor), where the gate wraps around the channel on three sides, providing superior electrostatic control. While technologically superior, FinFETs are vastly more complex and expensive to manufacture.

This manufacturing complexity translates directly into astronomical costs. Producing a chip on a FinFET node requires an extensive and intricate photolithography process involving up to 80 to 100 separate photomasks. Many of these layers necessitate complex and expensive double or triple patterning techniques, and the most advanced FinFET nodes rely on prohibitively expensive EUV lithography machines to print the finest features.

Consequently, the Non-Recurring Engineering (NRE) costs for a full mask set at a 5-7nm node can reach US$ 15M, sharp contrast, a 28nm design can typically be processed using a single exposure pass with standard Deep Ultraviolet (DUV) laser equipment, which TSMC and other foundries fully depreciated years ago . Because the capital cost of the equipment has already been paid, foundries can offer 28nm manufacturing at a significantly lower price point, as they are only recovering their operational costs and profit margin, not amortizing billions in new capital expenditure. This economic advantage makes 28nm exceptionally cost effective for volume production.

3.3.2 Clock Frequency and Power

Furthermore, the performance characteristics of the 28nm planar process are perfectly aligned with the needs of embedded processors and other low-power, high-efficiency applications. While shrinking transistor size generally allows for higher clock frequencies, there is a physical limit imposed by heat. Running a 28nm planar chip beyond approximately 1.5 GHz to 1.8 GHz leads to a dramatic increase in power consumption and heat generation due to leakage currents and dynamic power dissipation. This creates a thermal wall that is difficult and inefficient to overcome.

For the target RISC-V embedded processor, which is specified to run at 150MHz, this thermal limitation is irrelevant. Even for high-performance automotive Electronic Control Units (ECUs), which might target clock speeds up to 1.2 GHz, the 28nm planar process offers ample performance. The superior power efficiency of the 28nm planar process, combined with its lower cost, makes it an exceptionally efficient choice for this class of workload.

For applications requiring clock speeds above 2.5 GHz, such as top-tier CPUs or AI training accelerators, the performance benefits of FinFET nodes may justify their premium cost . However, for the majority of IoT controllers, edge devices, and embedded systems, the leap to a FinFET process is premature and economically unjustifiable.

3.4 Designing with 28nm PDK of TSMC

Using open-source Electronic Design Automation (EDA) tools to design a commercial chip on a proprietary node like TSMC 28nm is possible, but it requires a **hybrid workflow**. The reason is purely commercial: Open-source tools like OpenROAD and Yosys are highly capable of handling the math and placement for a 28nm design (even down to 5nm), however, TSMC will not accept a raw design file unless it has been "verified" by industry-standard commercial tools.

3.4.1 Hybrid workflow

Due to verification constraint imposed by large fabs, most fabless startup in this area typically proceed as follows:-

- **Step 1: Legal Foundation**: TSMC and silicon brokers will not sign contracts with individuals or loose collectives. Your startup must be registered as a formal private limited company. You will need a commercial business address (e.g., incubation center works perfectly), a corporate website, and a corporate email domain.

- **Step 2: Legal PDK Access**: Apply for access through a commercial aggregator/broker (such as MOSIS, imec IC-link, or VLSIShuttle). You will sign a corporate NDA with the broker. This grants your startup legal access to the TSMC 28nm PDK and standard cell libraries without talking to TSMC directly.

- **Step 3: Tool Preparation:** TSMC's native PDK is built for Cadence/Synopsys. To use open-source tools, you must extract the standard, non-proprietary files hidden inside the PDK package. You will pull out the .lib (Liberty files for timing), .lef (Library Exchange Format for physical geometries), and .v (Verilog structural models) for the standard cell libraries.

- **Step 4: Open Source Chip Design**: Set up an open-source toolchain (like OpenLane or OpenROAD). Use Verilator or Icarus Verilog for functional simulation. Use Yosys to synthesize your RTL code into a gate-level netlist using your extracted 28nm .lib files. Use OpenROAD to perform floorplanning, placement, Clock Tree Synthesis (CTS), and routing using the 28nm .lef files. This will output a raw GDSII layout file.

- **Step 5: The Bridge:** Open-source physical verification tools (like Magic or Netgen) cannot process TSMC's highly complex 28nm Design Rule Checking (DRC) and Layout Versus Schematic (LVS) decks. TSMC will reject your GDSII if it is not verified by a commercial engine like Siemens Calibre. To avoid buying a very expensive Calibre license, you must pay your silicon broker for "**Sign-Off Verification Services**." You send them your open-source generated GDSII, they run it through Calibre on their servers, send you back the error logs, and you fix the layout in your open-source tools until it passes.

- **Step 6: Manufacturing**: Once your design passes the broker's commercial sign-off check, the broker submits the GDSII file directly to TSMC for the next Multi-Project Wafer (MPW) shuttle run. You should send manufactured dies to packaging services. Once this is done, physical silicon chips will arrive at your doorstep.

3.4.2 Cost of Step 5 – Verification Service

We have multiple scenarios for verification purpose.

- First is the **free** option: when you pay for a TSMC 28nm Multi-Project Wafer (MPW) slot (starting at $16K / mm^2, or US$64K per slot (40-50 chips) of 4mm^2), the broker includes 1 to 3 "pre-shuttle" sign-off runs in the baseline price. They do this to protect the rest of the wafer from being ruined by your design.

- Second, if your open-source toolchain outputs messy GDSII that fails the included verification runs, you exhaust your free limit. The broker will bill you a flat engineering fee for every subsequent time they have to spin up their Calibre cluster to re-verify your fixes, typically **$3,000 – $6,000** per extra run.

- Third, if you want to check if your open-source flow is even capable of hitting 28nm metrics months before you buy a manufacturing MPW slot, you can request a standalone "Dry-Run". The broker runs your GDSII through the golden deck and hands you the raw Calibre error database, which is typically charged at **$5,000 – $10,000** flat fee.

- Finally, the broker's physical design engineers act as your hands. They will actively run the verification, fix advanced 28nm issues (like density rule violations, antenna effects, and dummy metal fill), and bring your layout to 100% compliance. This will typically cost you somewhere between **$15,000 – $35,000** package fee.

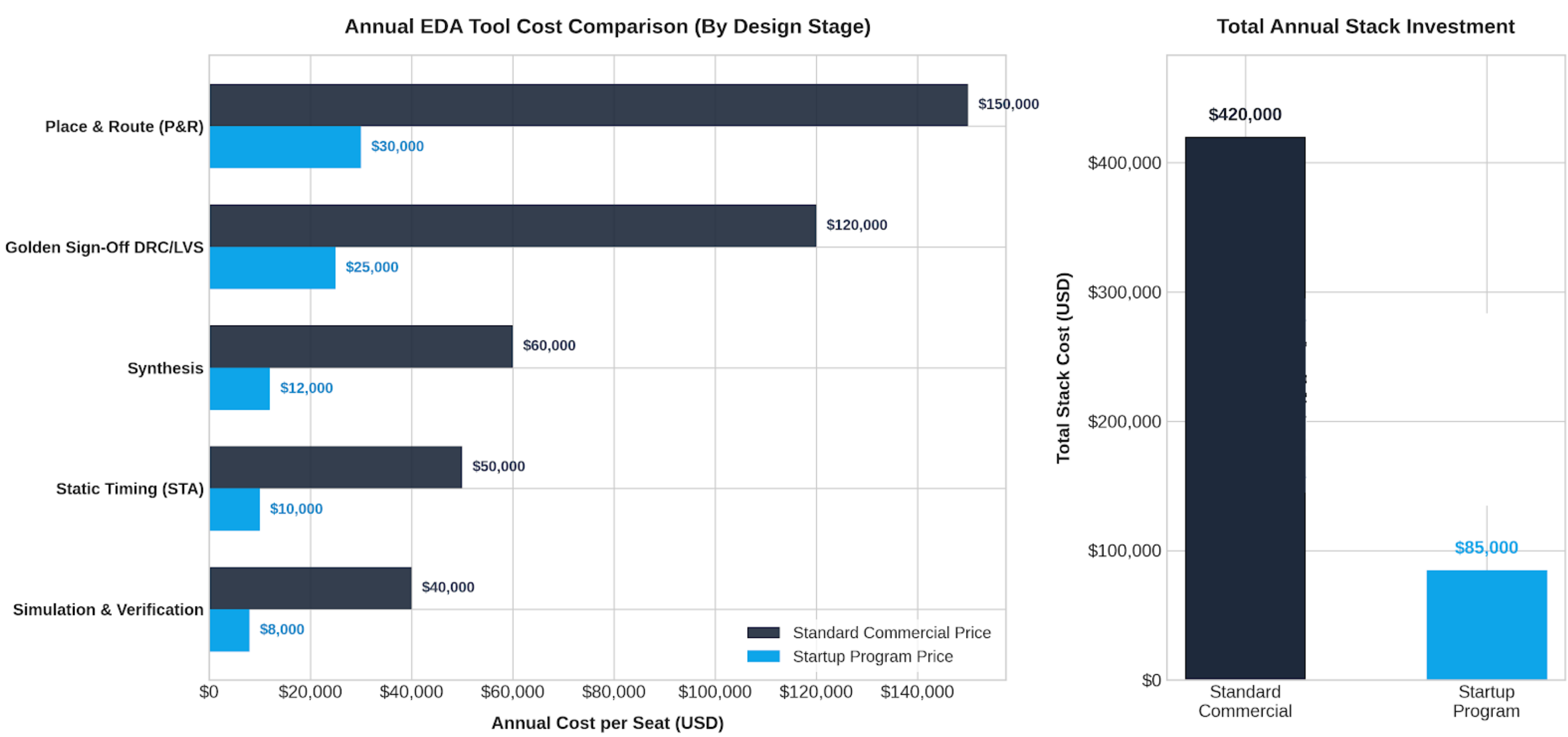
3.4.3 Proprietary Workflow Cost

If your startup wants to completely bypass open-source workflows and use the official, gold-standard commercial Electronic Design Automation (EDA) software that TSMC natively accepts for its 28nm node, you are looking at a significant financial commitment.

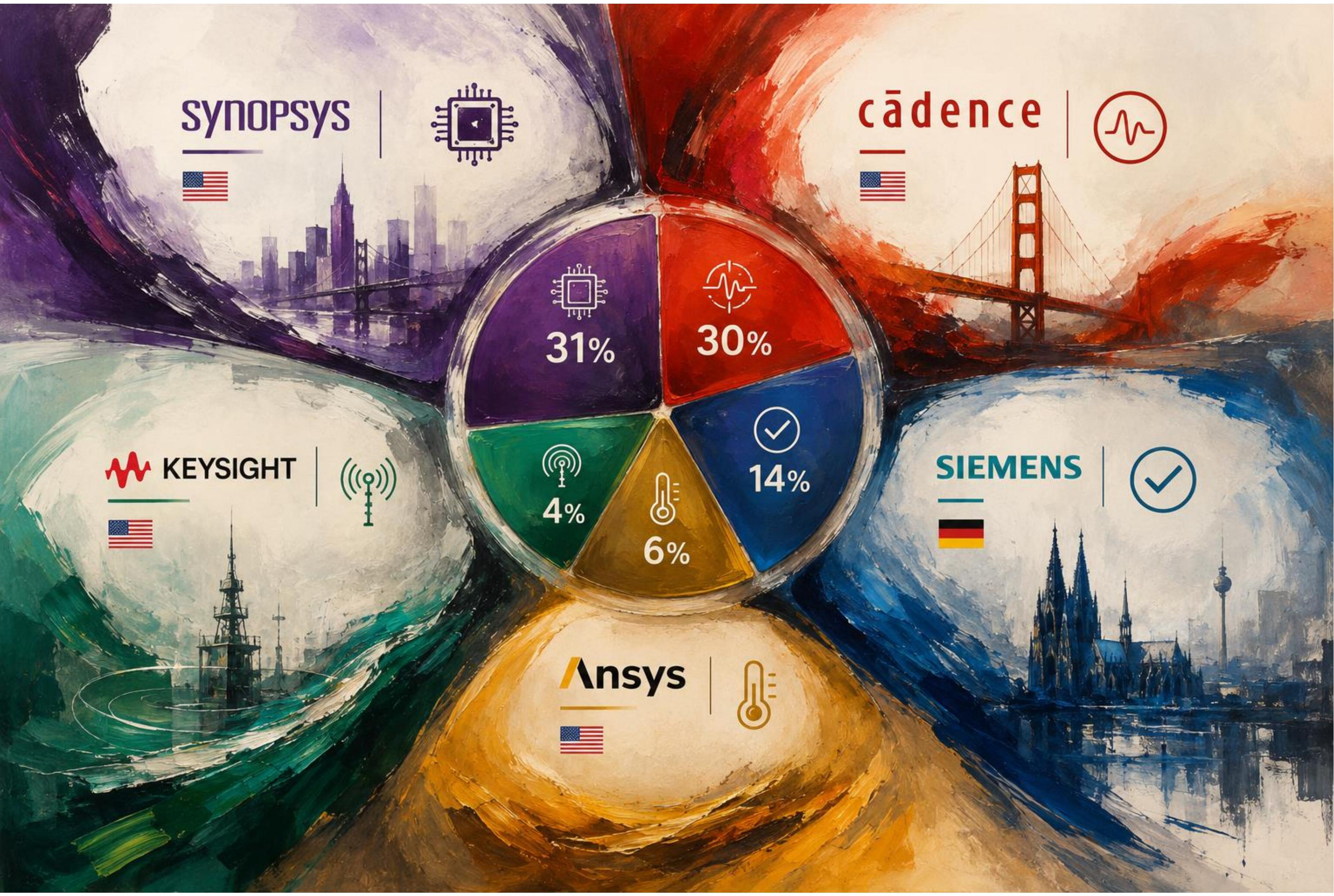
EDA giants like Synopsys, Cadence, and Siemens EDA (Mentor Graphics) do not have a public retail storefront. They use specialized enterprise pricing. However, because they want to capture early-stage companies, they all offer Startup Accelerator Programs that slash standard enterprise costs by 80% to 90%.

On-Premise Installation: If you have local servers in your incubation center and want to host the software locally, you must apply for programs like the Synopsys Startup Program or Cadence Startup Partner Program. Note that to qualify as start-up, your company (not older than 5 years) must be under $10 Million in valuation with annual revenue to be close to zero.

The following graphs breaks down estimated annual subscription costs per single-user seat for a complete TSMC 28nm digital implementation flow under startup discounting compared to standard enterprise retail rates.



Cloud Subscription: Under this model, the vendor gives you unlimited access to their entire 28nm-capable tool stack, and you pay a metered rate strictly per hour, per tool, per core. For an early-stage startup that spends 9 months writing code and only runs synthesis and Place & Route heavily during the final 2 months before tape-out, a Cloud Pay-Per-Use model generally reduces total software expenditures down to $20,000 to $35,000 total for the entire lifecycle of the project.



Big Five EDA Players

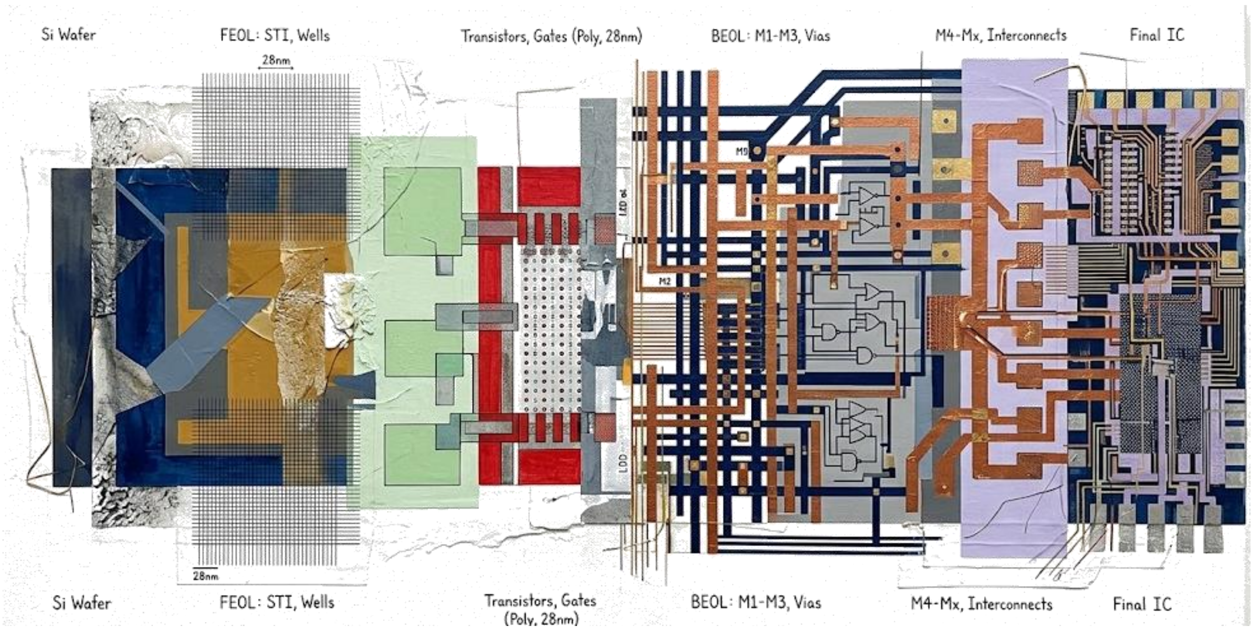
4 Open source EDA Tooling

4.1 Background

In an FPGA development environment, the physical substrate is pre-fabricated, and the vendor's compiler automatically maps logical operations into pre-allocated Lookup Tables (LUTs), block RAMs (BRAMs), and rigid, global clock distribution networks. In contrast, designing an Application-Specific Integrated Circuit (ASIC) on the TSMC 28nm node requires the physical realization of every transistor, standard cell, custom memory macro, power rail, and clock tree wire from the ground up. However, it is not as complex as it sounds for a fabless start up, due to availability of open source tools.

The physical transition from a reconfigurable FPGA architecture to a custom TSMC 28nm silicon die is detailed in the table below:

Architectural Attribute	FPGA Prototyping	ASIC Manufacturing
Logic Realization	Configurable Lookup Tables (LUTs) and flip-flops with fixed physical boundaries.	Standard Cell Libraries containing physical layouts of logic gates (e.g., NAND, NOR, AOI, DFF).
Interconnect Physics	Segmented, pre-routed copper tracks with high intrinsic resistance and programmable switch matrices.	Custom-routed metal layers (typically 9 to 11 copper layers) optimized for minimum wire length and delay.
Clock Distribution	Hardwired, low-skew clock buffers (e.g., BUFG, BUFGCE) distributing signals through dedicated networks.	Synthesized Clock Trees built from clock buffers/inverters, dynamically balanced to minimize skew and insertion delay.
Memory Architecture	Pre-allocated Block RAMs (BRAMs) and UltraRAMs with fixed ports, aspect ratios, and positions.	Custom compiler-generated Static RAM (SRAM) macros tailored to exact word counts, bit widths, and multiplexing factors.
Timing & Physical Sign-Off	Highly pessimistic, vendor-packaged timing models; physical layout failures are impossible.	Multi-Corner Multi-Mode (MCMM) Static Timing Analysis (STA) accounting for extreme physical variations and lithographic defects.
Lithographic Constraints	Abstracted by the FPGA vendor; no manufacturing preparation is exposed to the engineer.	Strict physical rules including double patterning, density limits, antenna effects, and electrostatic discharge protection.



4.2 Tool Selection

This section provides a practical blueprint for fabless ASIC design teams utilizing open-source Electronic Design Automation (EDA) tools to tape out a design on the TSMC 28nm process node via a Multi-Project Wafer (MPW) shuttle. While open-source methodologies have successfully democratized mature nodes, target nodes like TSMC 28nm High-K Metal Gate (HKMG) demand a highly disciplined mixed-toolchain approach to clear the strict sign-off hurdles of foundry brokers (such as Muse Semi, IMEC, or CMP).

Implementing a complex, gigahertz-class design requires an organized toolchain. High-performance, open-source hardware designs like the Titan-I (T1) vector processor rely on a deterministic, fully reproducible development environment to prevent toolchain drift.

4.2.1 Docker and Nix-Based Determinism

To avoid the classic "works on my machine" problem, your team must standardize its environment. Use Nix Flakes or Docker to package your compiler, simulator, and physical design tools. Titan-I projects distribute pre-built Docker containers and Nix environments containing specific, pinned versions of Verilator, Spike, and the OpenROAD toolchain, ensuring that every engineer synthesizes identical netlists and layouts.

4.2.2 The OpenRPDK28 Bridge

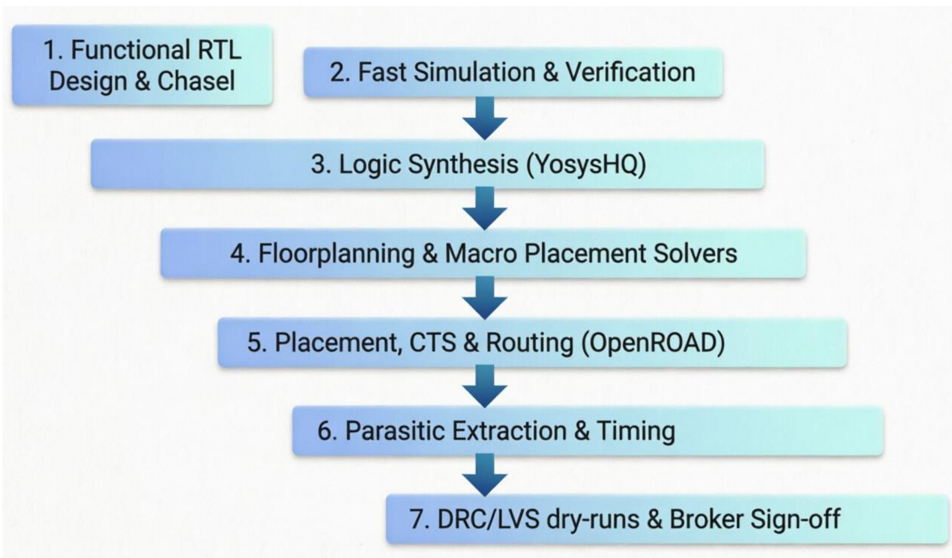
Because TSMC 28nm physical design rules are protected under strict Non-Disclosure Agreements (NDAs), direct integration with open-source tools can be challenging. To bypass this, leverage academic-to-industrial bridges such as the OpenRPDK28 developed by RIOS Lab. OpenRPDK28 acts as an open-source process design kit (PDK) template that mimics industrial 28nm technology, offering Open-source cell libraries, symbols, and SPICE models, Preliminary design rules, layer maps, and electrostatic discharge (ESD) structures, and Runsets for physical verification (DRC/LVS).

Your team can use OpenRPDK28 for initial physical design exploration and layout dry-runs before porting the design into the secure foundry-NDA environment.

4.3 Detailed Physical Design Pipeline: RTL to 28nm GDSII

4.3.1 Step 1: High-Level RTL Design and Parametric Generation

Traditional Verilog and SystemVerilog are static and prone to scaling errors in highly parallel designs. For complex designs like T1, use **Chisel** (Constructing Hardware in a Scala Embedded Language). Chisel allows your hardware engineers to write object-oriented generators that programmatically output highly optimized, synthesizable Verilog while adjusting design parameters (such as the number of execution lanes, cache size, or bus widths).



4.3.2 Step 2: Verification and Co-Simulation

Before starting physical design, achieve functional verification using open-source simulation engines:

- **Verilator:** Compile your synthesizable Verilog directly into highly optimized C++ code to achieve cycle-accurate execution speeds orders of magnitude faster than interpreted simulators.
- **Cocotb:** Drive your test benches with Python co-routines to easily implement modern verification techniques, assertions, and randomized testing.
- **Spike (ISS) Co-Simulation:** For CPU designs, run your Verilated RTL in lockstep with the golden Spike Instruction Set Simulator. Spike checks register states on every instruction commit, flagging functional bugs instantly before synthesis.

4.3.3 Step 3: Logic Synthesis via Yosys

For synthesis, use **Yosys**, the premier open-source synthesis engine. Yosys parses your SystemVerilog files, performs logical optimizations, and maps the design to your TSMC 28nm standard cells using technology mapping scripts.

- **Track Height Selection:** Select 7-track (7T) high-density cells for logic-heavy, area-constrained sub-blocks, or 9-track (9T) standard cells for speed-critical modules.
- **Multi-Threshold Voltage (V_t) Partitioning:** Direct Yosys to map non-critical paths to high-V_t (HVT) cells to reduce static leakage current, and critical timing paths to low V_t (LVT) cells to maximize performance.-
- **DFT Insertion:** Insert test logic (scan chains and memory Built-In Self-Test, or BIST) during or immediately after the synthesis phase to make the fabricated silicon testable.

4.3.4 Step 4: Coarse-Grained Floor-planning and Macro Placement

Memory blocks (SRAM macros generated via TSMC's compiler) and analog IP must be floor-planned carefully. Hand-placing dozens of SRAMs in a complex vector processor often leads to routing congestion and clock distribution issues.

- Use automated **coarse-grained floor-planning solvers** (built into generators like T1) to programmatically calculate the placement coordinates of your memory banks.
- Enforce **placement halos** around the SRAM boundaries. Keep a clearance of at least 2 μm to 3 μm clear of standard cells to ensure that macro pins are fully accessible by the routing engine.
- Design your **Power Delivery Network (PDN)** in OpenROAD by routing a low-impedance mesh of horizontal and vertical power straps on Upper Metals (M7, M8, M9) to combat dynamic IR drop.

4.3.5 Step 5: Placement, Clock Tree Synthesis, and Routing in OpenROAD

Once the floorplan is fixed, use the **OpenROAD** tool suite to execute the core physical design flow autonomously:

- **Global Placement (RePlAce) & Detail Placement (OpenDP):** These engines distribute standard cells uniformly across rows. Well-tap cells must be placed every 30 μm to 40 μm to prevent latch-up.
- **Clock Tree Synthesis (TritonCTS):** Synthesizes a balanced clock tree using symmetric H-tree topologies. Apply **Non-Default Routing (NDR)** rules to clock lines, ensuring double spacing and double width to minimize clock skew and coupling jitter.
- **Detailed Routing (TritonRoute):** TritonRoute assigns nets to specific copper tracks. At 28nm, the router must natively support **double patterning (Litho-Etch-Litho-Etch, or LELE)** rules on the lower, critical metal layers (M1, M2), assigning mask colors to resolve

pitch conflicts. TritonRoute must also perform layer-hopping or insert antenna diodes near standard-cell inputs to prevent dielectric charge damage.

4.3.6 Step 6: Parasitic Extraction and Static Timing Analysis (STA)

Extract actual wire resistances and capacitances to generate standard parasitic exchange format (SPEF) files. Run **OpenSTA** to verify timing across multiple corners. Unlike older nodes, 28nm requires **Parametric On-Chip Variation (POCV)** modeling to dynamically calculate local variations based on physical path depth, preventing overly pessimistic margins. Finally, verify timing closure using the standard setup and hold equations.

4.3.7 Step 7: Physical Verification (DRC/LVS) and DFM on KLayout

Use **KLayout** to run local verification passes as follows:

- **Design Rule Checking (DRC):** Verify minimum spacing, widths, and enclosure rules using rule decks compatible with OpenRPDK28 or the broker's preliminary decks.
- **Layout vs. Schematic (LVS):** Compare the extracted spice netlist from KLayout against the synthesized gate netlist.
- **Design for Manufacturability (DFM):** Run automated Python scripts in KLayout to insert **dummy metal fill** across empty regions to ensure a uniform metal density profile, preventing planarization dishing. Insert redundant vias on all critical routing paths to prevent single-via open failures.

4.4 Verification Requirement for Fabrication

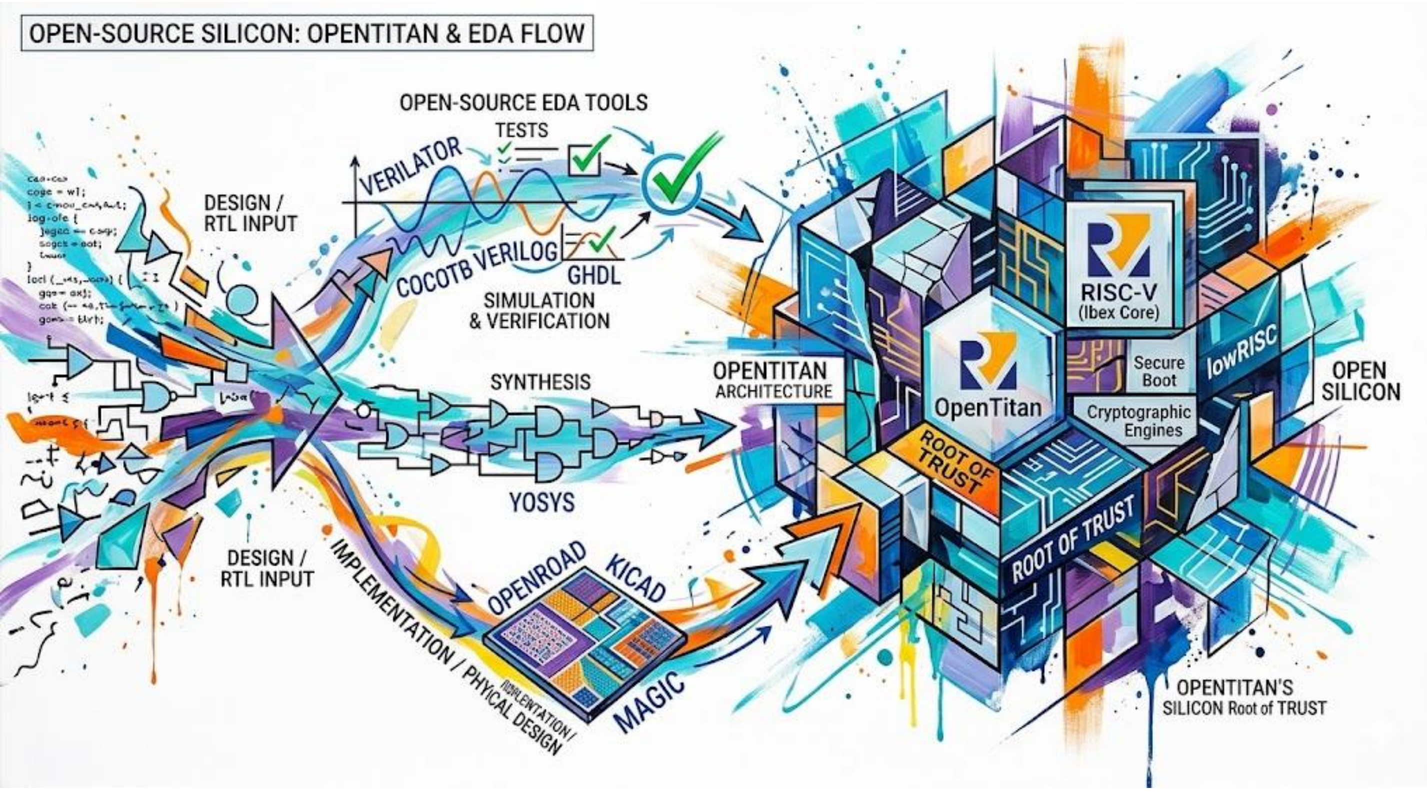
A startup can use open-source tools for for RTL-to-GDSII flow. These tools are highly valuable for rapid iteration, academic research, MPW shuttle participation, and early-stage silicon development.

However, advanced foundries such as TSMC generally require final "signoff" verification to be performed using foundry-certified commercial EDA tools and official rule decks. In practice, this means that the final tapeout database is expected to pass DRC, LVS, extraction, density, reliability, and lithography verification using certified platforms such as Siemens **Calibre**, **Cadence Pegasus**, or **Synopsys IC Validator**.

As a result, most real-world fabless startups adopt a hybrid methodology. Open-source tools are used throughout most of the design cycle to minimize costs and accelerate development, while final signoff verification is outsourced to licensed commercial environments, as described in Section 3.4.

4.5 The Titan-I (T1) Vector Processor

A common concern among system designers is whether open-source hardware tools and architectures can produce high-performance, manufacturing-worthy silicon. Historically, complex high-performance processors were considered the exclusive domain of proprietary EDA toolchains and closed-source IP.



This view has been challenged by the academic and industrial realization of the **Titan-I (T1)** vector processor, which demonstrates that advanced RISC-V vector architectures can be designed, verified, and physically synthesized using fully open-source methodologies.

4.5.1 Titan-I Project Origin and Architecture

Titan-I is an open-source, high-performance out-of-order (OoO) RISC-V Vector (RVV) processor core. It was developed by a collaborative research group including Jiuyang Liu, Qinjun Li, Yunqian Luo, and Mingyu Gao, with contributions spanning Huazhong University of Science and Technology, Tsinghua University, and the Institute of Software at the Chinese Academy of Sciences (ISCAS). The processor was presented at the 58th IEEE/ACM International Symposium on Microarchitecture (MICRO 2025) in Seoul, South Korea.

The architecture of Titan-I is designed to scale both Instruction-Level Parallelism (ILP) and Data-Level Parallelism (DLP) by decoupling scalar instruction execution from a wide, multi-lane parallel vector execution engine. Titan-I fully complies with the official RISC-V Vector Extension (RVV 1.0) specification, enabling efficient parallel computation of multi-precision integers and floating-point datatypes.

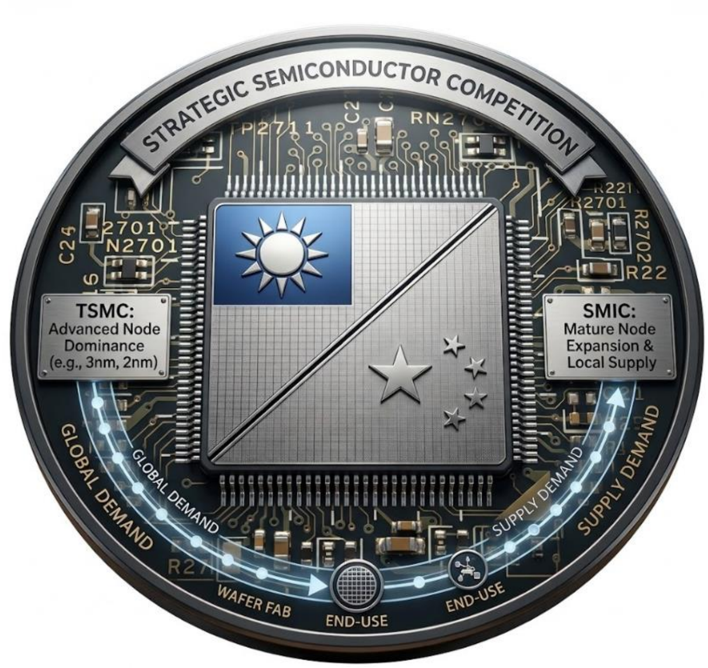
This microarchitecture is optimized for highly parallelized applications, such as deep neural networks (DNNs), high-performance computing (HPC) kernels, and cryptographic workloads.

4.5.2 Design and Physical Execution Environment

Titan-I is implemented in **Chisel** (Constructing Hardware in a Scala Embedded Language), a domain-specific language embedded in Scala. Chisel raises the level of hardware design abstraction by providing object-oriented programming, functional programming, and strongly typed parameterized generators. Instead of manual, error-prone edits to static Verilog code, Chisel allows the Titan-I architecture to be configured programmatically. By adjusting parameters such as vector register file size, vector register length, and the number of physical execution lanes, the Chisel generator dynamically restructures the execution datapath and outputs optimized, synthesizable Verilog code or high-speed, cycle-accurate C++ simulation executables.

The physical design flow of Titan-I is executed using the open-source **OpenROAD** physical synthesis tool suite. This automated, scriptable flow performs physical synthesis, placement, clock tree synthesis, and detailed routing down to advanced nodes (including successful physical synthesis evaluations down to 7nm technology. This demonstrates that open-source tools can successfully resolve the complex lithographic and electrical design rules of modern sub-10nm silicon manufacturing.

5 Foundry Economics: TSMC vs SMIC



Once a design has been completed using an open-source toolchain and a target node has been selected, the next critical step is to engage with a semiconductor foundry for fabrication. The two dominant players in the mature-node space are Taiwan Semiconductor Manufacturing Company (TSMC) and Semiconductor Manufacturing International Corporation (SMIC). While both offer 28nm manufacturing services, their business models, cost structures, and geopolitical contexts create distinctly different value propositions for a startup.

5.1 TSMC, Taiwan

TSMC, based in Taiwan, is the industry's long-standing gold standard and holds a commanding market share, accounting for 34% of the pure-play foundry industry in 2024. Its reputation is built on exceptional manufacturing quality, consistently high initial yields, and a vast ecosystem of third-party Intellectual Property (IP) blocks. This ecosystem includes pre-verified IP cores for common interfaces like PCIe, DDR memory controllers, and high-speed SerDes, which can significantly accelerate a design team's time-to-market by reducing the need to develop and verify these components from scratch.

However, this reliability and convenience come at a premium price. The cost of a 28nm wafer at TSMC ranges from approximately **3,000 to 3,500 USD** per mm. More significantly, the NRE cost for a complete mask set—a prerequisite for a full production run—is estimated to be between 1.5-2.5M USD. This high upfront investment reflects the advanced nature of their fabs, even for a mature node, and the high demand for their capacity.

5.2 SMIC, China

In contrast, SMIC, the largest contract chip maker in mainland China, operates with a different economic model heavily influenced by national policy and industrial strategy. Leveraging substantial government subsidies and capacity expansion plans, SMIC is able to offer more aggressive pricing for mature process technologies like 28nm The company explicitly markets its Multi-Project Wafer (MPW) program as a cost effective service for customer prototyping, highlighting its value proposition for startups and smaller projects.

The per-wafer cost for 28nm at SMIC is reported to be in the range of **2,200 to 2,600 USD** per mm, representing a notable discount compared to TSMC. The NRE/mask costs are also substantially lower, estimated to be between 0.8 and 1.5M USD. This reduced upfront cost makes SMIC an extremely attractive option for startups operating on a tight budget, where minimizing initial capital expenditure is the primary concern.

While the financial advantages of SMIC are clear, choosing this foundry introduces a layer of geopolitical risk that must be carefully considered. As a major Chinese semiconductor company, SMIC operates within a complex international trade environment, particularly concerning technology transfers from the United States. Any future regulatory changes or export controls could potentially impact the availability of certain EDA tools, design methodologies, or other critical components of the supply chain. Similarly, TSMC, while also subject to geopolitical dynamics, but is generally perceived as having a more stable and predictable operating environment in the West.

5.3 Cost Estimates

Taking a design from a 100% complete GDSII to fifty packaged prototypes is a major milestone. Let us first establish a realistic baseline for the technology node and die area.

For a dual-core RISC processor equipped with embedded SRAM, cryptographic accelerators (like AES, SHA, or RSA/ECC engines), and a Root of Trust (RoT), **40nm** and **28nm** are typically the most pragmatic nodes.

Assuming an optimized layout, a design of this scope will likely require a die area between **4 mm^2 and 9 mm^2**.

5.3.1 Silicon Fabrication Costs (MPW)

Foundries and shuttle aggregators (like Europractice, Muse Semiconductor, or IMEC) price MPW runs per square millimeter, but they strictly enforce **minimum area rules**. Even if your chip is 3 mm^2, you will have to pay for the foundry's minimum block size (often 4 mm^2 or 9 mm^2).

An MPW run typically delivers between 40 and 100 bare dies.

Foundry & Node	**Est. Cost per mm^2**	**Typical Min Area**	**Estimated Silicon Cost of (40-100 dies)**
TSMC 28nm (HPC/HPC+)	\$13,000 – \$16,000	1 mm^2 (mini@sic)	\$52,000 – \$64,000+ (4mm^2)
TSMC 40nm (LP/G)	\$6,000 – \$10,000	3 mm^2 (mini@sic)	\$18,000 – \$30,000
SMIC 28nm (PolySiON/HKMG)	\$10,000 – \$13,000	2 – 5 mm^2	\$20,000 – \$65,000
SMIC 40nm (LL/G)	\$4,000 – \$7,000	3 – 4 mm^2	\$12,000 – \$28,000

Note that SMIC is generally ~30% cheaper than TSMC for mature nodes, acting as a highly competitive option if you are price-sensitive at the prototyping stage.

5.3.2 Packaging the First 50 Units

Because you only need 40-50 units, designing a custom organic substrate (like a standard mass-market BGA) is entirely cost-prohibitive due to custom tooling NREs ($10,000 to $30,000 just for the substrate design).

Instead, you will use **prototype packaging services** (e.g., Quik-Pak, Novapack, or aggregator-provided services). You will use standard, off-the-shelf packages that require no custom tooling:

- **Open-Cavity QFNs or Ceramic PGAs:** The die is glued into a pre-made package and wire-bonded based on a custom bonding diagram.
- **Setup Fee:** ~$2,000 - $3,000 for creating the wire-bond profile and machine setup.
- **Per-Unit Cost:** ~$50 - $100 per chip for low-volume manual/semi-automated assembly.
- **Estimated Packaging Total: $4,500 - $8,000**

5.3.3 The "Hidden" Execution Factors

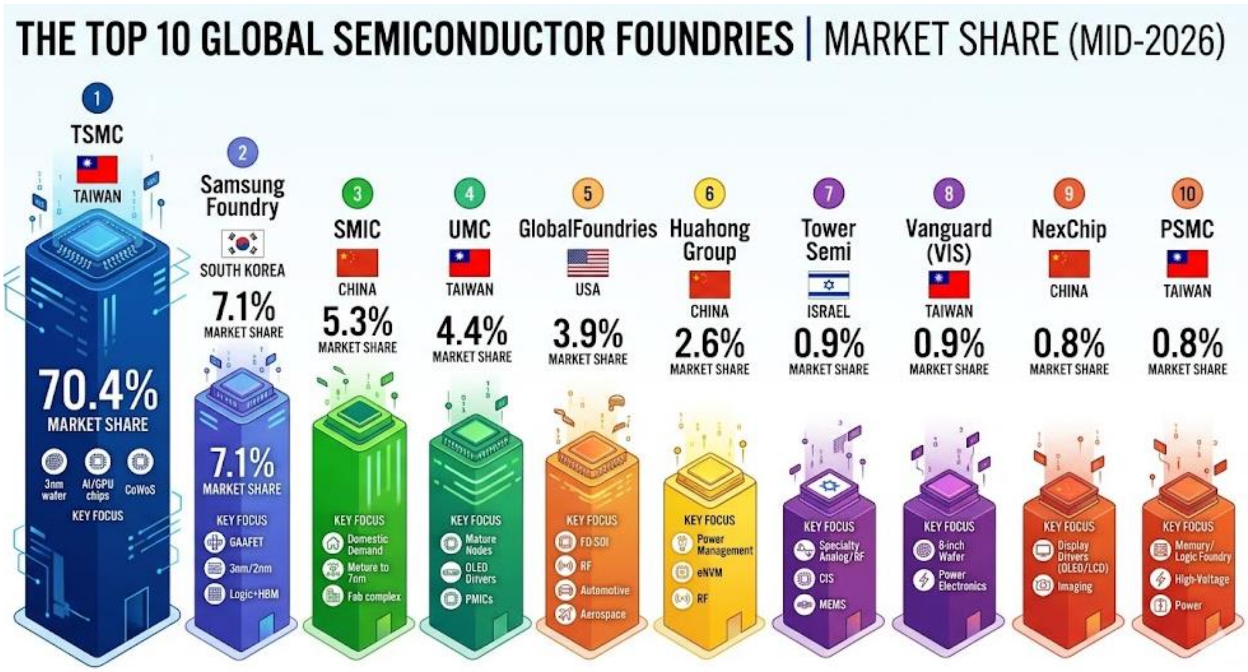
If our design includes cryptographic accelerators and a Root of Trust, the GDSII to silicon process is also about compliance.

- **Export Controls & IP:** Cryptographic hardware falls under strict dual-use export regulations (like the Wassenaar Arrangement or US EAR). If your team sending a tapeout with advanced crypto to SMIC (Mainland China) will require rigorous legal compliance, export licenses, and ECCN (Export Control Classification Number) declarations. Taping out at TSMC (Taiwan) is generally a smoother administrative process for Western entities, though ECCN declarations are still mandatory.
- **Shuttle Access Fees:** If you are not a direct customer of the foundry (which requires massive volume commitments), you will go through a Channel Partner / Aggregator. They may charge a one-time onboarding or IP-handling fee.
- **Lead Times:** GDSII to bare die on a mature node MPW usually takes **3 to 4 months**. Prototype packaging will add an additional **2 to 4 weeks**. You will be bound to the foundry's pre-published shuttle schedule, which typically runs 3 to 4 times a year for these specific nodes. Missing the tapeout window means waiting 3 months for the next one.

5.4 Classic Trade-off

The choice between TSMC and SMIC is a classic trade-off between cost and risk. TSMC offers a premium, reliable, and feature-rich platform that minimizes design risk and accelerates development but at a significantly higher price. SMIC offers a compellingly lower-cost alternative that is ideal for bootstrapped startups prioritizing strict unit-cost optimization, but it comes with inherent geopolitical and supply chain risks that must be managed.

For a startup with strong financial backing and a need for guaranteed high yields and access to a broad IP ecosystem, TSMC remains the safer bet. For a budget-constrained startup where every dollar counts and the technology is well understood, SMIC presents a financially attractive path to production.



6 Roadmap

Armed with an understanding of the economic landscape, the power of open-source tools, and the technical nuances of the ASIC design flow, a clear strategic roadmap emerges. Furthermore, the principles established for the RISC-V embedded processor use case are broadly applicable and can be extended to address a wider array of challenging applications in AI, automotive, and edge computing. The recommended strategy for a startup embarking on its first fabless journey is a phased approach that leverages cost-sharing manufacturing models before committing to a full production run.

6.1 Steps to Final Production

6.1.1 Phase 1: Prototyping with an MPW Shuttle

The very first step should be to fabricate a prototype of your design using a Multi-Project Wafer (MPW) service offered by either TSMC or SMIC . Services like TSMC's CyberShuttle® or Samsung's MPW service allow multiple designs from different customers to be placed on a single wafer. This approach dramatically reduces the upfront NRE costs, as the expense of the mask set is shared among all participants.

The primary goal of this phase is not to produce a final product but to validate the RTL design at the silicon level, verify its functionality in the real world, and, most importantly, collect initial yield data. This yield data is invaluable, as it provides a realistic basis for calculating the per die cost for future production runs and helps identify any unforeseen issues in the physical design.

6.1.2 Phase 2: Iteration and Refinement with MLM

Based on the learnings from the MPW shuttle, the design may require final optimizations before high-volume manufacturing. For this stage, a Multi-Layer Mask (MLM) or Multi-Layer Reticle (MLR) service is often employed. This model allows a company to reduce mask NRE costs by combining up to four different design layers onto a single physical mask plate. While this increases the complexity of the lithography process and the individual wafer cost, it significantly lowers the initial investment compared to a full, dedicated mask set. This provides a strategic middle ground for low-to-medium volume production, allowing for the fabrication of hundreds of prototypes or early-market units on a stable design without the high cost of a high-volume tooling set.

6.1.3 Phase 3: Full Production Run

Once the design has achieved functional sign-off through an MPW shuttle, the company moves toward a full production run. This stage involves the significant capital expenditure of a dedicated mask set (NRE) and the commitment to a minimum wafer start order. While the financial stakes are high, the technical risk is mitigated because the design has been validated in silicon and timing has been closed across all required PVT corners. It is important to note that while MPW runs confirm functionality, accurate per-die cost and yield modeling at this stage rely on foundry-provided baseline data and DFM (Design for Manufacturing) analysis, which are later refined during the initial pilot production lots.

6.2 Example ASIC Types

The principles and workflows outlined in this whitepaper are not limited to RISC-V embedded processors. They can be generalized to address a wide spectrum of modern computing challenges:

6.2.1 AI Accelerators

The demand for efficient AI inference at the edge has created a critical need for specialized hardware acceleration. The 28nm node represents the industry's optimal sweet spot for cost-versus-performance in these applications. While advanced FinFET nodes offer superior absolute power efficiency, their astronomical mask costs are often prohibitive for startups and niche architectures. A 28nm planar or High-K Metal Gate (HKMG) process allows designers to maximize logic density within a viable economic framework. Projects like e-GPU, an open-source and configurable RISC-V platform for TinyAI devices, demonstrate the feasibility of using this flow to create sophisticated compute accelerators. Furthermore, an open-source RTL-to-GDSII toolchain enables rapid architectural exploration, allowing developers to iteratively optimize data paths for specific neural network workloads without incurring compounding software licensing costs.

6.2.2 Automotive ECUs

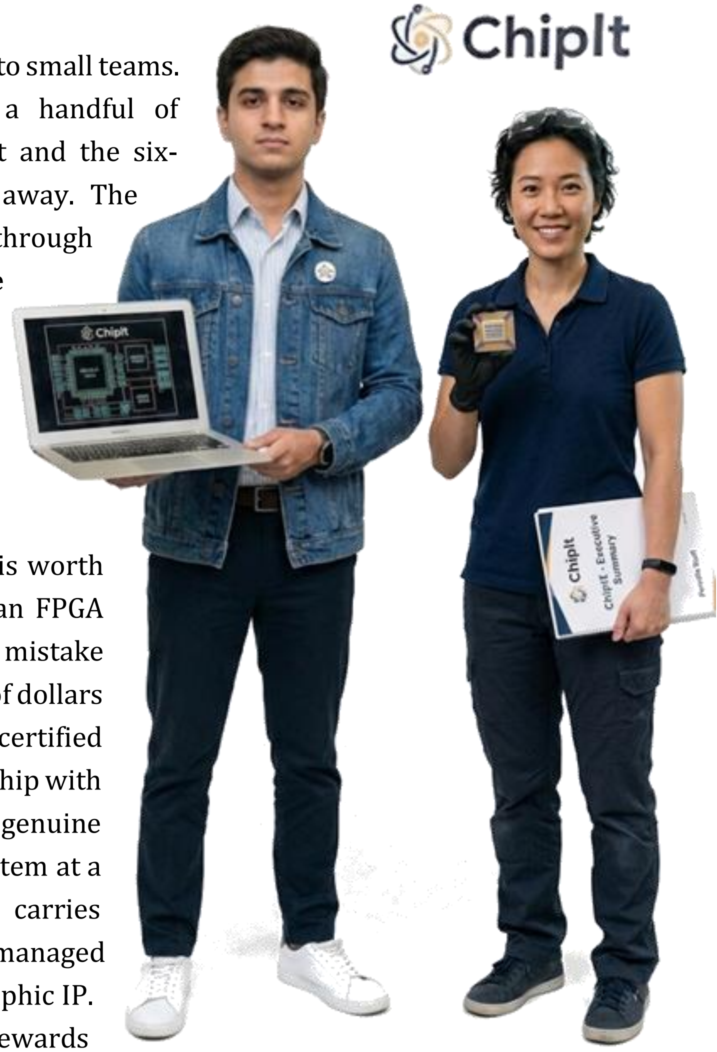
The automotive industry is increasingly moving toward mixed-criticality systems where multiple operational domains run concurrently on a single system-on-chip (SoC)—a challenge perfectly suited to the architectural flexibility of the RISC-V ISA. While safety-critical drive systems require rigorous ISO 26262 toolchain qualifications that open-source tools are still maturing toward, this openflow is highly disruptive for secure telematics, infotainment, and vehicle-to-everything (V2X) communication research. The open-source nature of the entire toolchain—from the processor core to the physical design synthesis—enables unprecedented transparency for security auditing and custom cryptographic extensions.

The Titan-I project, which successfully implemented a chiplet-based RISC-V SoC for secure nano-UAVs, showcases the potential for creating heterogeneous, highly secure, and resilient embedded architectures using this decentralized design methodology.

6.2.3 Edge Computing and IoT

For compute-intensive edge devices and advanced IoT nodes, balancing power efficiency with localized processing power is a primary constraint. The 28nm process excels in this domain by providing the frequency scaling necessary for real-time sensor fusion and local data processing, while maintaining a compact physical footprint. By utilizing an openflow, designers can strip away generic, redundant IP blocks typically found in commercial microcontrollers, engineering a lean, workload-optimized SoC that minimizes active power dissipation. The UET-RVMCU project, which demonstrated the successful transformation of an application-class SoC architecture into a streamlined, feature-rich microcontroller using an open-source physical design flow, serves as a prime validation of this cost-effective edge-computing approach.

7 Final Words



The path to fabless ASIC design has never been more open to small teams. Two barriers that once confined custom silicon to a handful of corporations — the multibillion-dollar fabrication plant and the six-figure-per-seat EDA license — have effectively fallen away. The fabless model lets you rent world-class manufacturing through Multi-Project Wafer shuttles, while a mature open-source toolchain now carries a design all the way from RTL to a manufacturable GDSII layout. For an engineer already fluent in FPGA development, the front-end skills transfer almost directly; what you are really learning is the discipline of the back-end physical flow.

That openness, however, is not the same as ease, and it is worth being honest about what the journey demands. Unlike an FPGA bitstream, a fabricated die cannot be re-uploaded — a mistake caught after tape-out costs months and tens of thousands of dollars to correct. Foundries still require final sign-off on certified commercial tools, so a hybrid flow and a working relationship with a silicon broker are unavoidable. Foundry selection is a genuine trade-off: TSMC offers premium yield and a rich IP ecosystem at a higher price, while SMIC is markedly cheaper but carries geopolitical and export-control exposure that must be managed deliberately, particularly for designs containing cryptographic IP. None of these obstacles is disqualifying, but each one rewards planning over improvisation.

Taken as a whole, it is the economics that make the opportunity compelling. A complete, state-of-the-art design environment is now within reach of a modest budget, and the phased strategy recommended in this paper — proving the design on an MPW shuttle, refining it through a multi-layer mask run, and only then committing to a full production mask set — allows you to retire technical risk before you spend serious capital. The 28nm planar node sits at the center of this strategy, offering the rare convergence of low cost, ample performance, and proven manufacturability for the embedded controllers, AI accelerators, automotive subsystems, and edge devices where most startups will compete.

For founders willing to pair that engineering discipline with sound commercial judgment, the reward is substantial: the ability to build differentiated, workload-optimized silicon that was, until very recently, the exclusive territory of the largest semiconductor firms. The tools, the foundry access, and the business model are all in place. What remains is the decision to begin — and the teams that move thoughtfully now will be the ones that define the next generation of custom hardware.