# Arm Ecosystem Reduces SoC Design Cost and Time to Market

By Linley Gwennap
Principal Analyst

April 2019

The Linley Group

www.linleygroup.com

# Arm Ecosystem Reduces SoC Design Cost and Time to Market

By Linley Gwennap, Principal Analyst, The Linley Group

*This paper discusses three critical facets of the Arm ecosystem: design verification, physical design, and software development. Each one directly affects the time required to develop a complete SoC and software stack. Whereas Arm and its partners offer a wide set of tools and support, some other CPU ISAs fall short in these areas. Arm sponsored this white paper, but the opinions and analysis are those of the author. Trademark names are used in an editorial fashion and are the property of their respective owners.*

## *Introduction*

Having shipped in more than 130 billion chips, the Arm architecture has become as familiar as breathing to many SoC designers—so familiar that they may forget its benefits when evaluating new CPU competition. Arm is much more than an instruction set (ISA): it connects designers to a massive ecosystem of compatible CPU cores, tools, middleware, and application software. In addition, the company delivers a wide range of cores that offer quality, security, support, patent protection, and a roadmap for future growth. SoC designers often focus on traditional factors such as performance, power, and area (PPA); the ecosystem is more difficult to quantify but can greatly affect design cost and time to market. Some other CPU ISAs fall short in this regard.

A typical SoC design today has dozens of cores and other IP blocks, often including application CPUs, microcontrollers, graphics, DSP, AI acceleration, crypto engines, memory controllers, and various I/O interfaces. Companies spend tens of millions of dollars (or more) to develop these complex SoCs, including their rich software stacks. These products pay back this investment by generating even larger revenue streams. In a high-profile project, even small schedule delays can cost millions of dollars in missed revenue opportunities. Thus, the design team focuses on time to market and reducing schedule risk.

An ISA ecosystem involves numerous components that can affect the design cycle—too many to discuss in a single white paper. This paper analyzes three critical facets. The first is design verification, including hardware and software tools to assist in logic design and testing. Second, the physical-design phase, including design flows and physical IP to help convert RTL into a manufacturable chip. Third, the software ecosystem, comprising development tools, operating systems, drivers, and applications guaranteed to run on compatible CPUs. Each of these aspects directly affects the time required to develop a complete SoC and software stack.

## *Design Verification*

SoCs often mix licensed and internally designed cores. Unless an internal design is directly leveraged from a previous product without changes, the engineering team typically spends considerable time verifying that the logic (RTL) design for that block is fully functional and meets performance targets. But after paying a license fee for a core, the team expects it to work properly right out of the box. If so, it can focus on the in-house cores, which often add the most value and differentiation to the SoC. Any problems with the licensed cores hinder this effort and create schedule risk.

Arm fully verifies the RTL for its cores, but that's just the start. Most Arm cores are production proven, having shipped in millions or even billions of chips. Even the newest cores are generally based on previous proven designs, and the company often builds test chips to ensure quality. New designs are verified using test suites that have evolved over decades to ensure functional correctness. Any paying licensee receives full support, so even if errata appear after the initial release, Arm communicates them to customers. Many licensed CPU cores don't offer this level of testing and support. New ISAs typically require several years to build reliable validation suites and deliver proven silicon. These challenges are particularly difficult for open-source architectures that rely on community-based verification.

The next step for the SoC design team is connecting these cores and optimizing system-level performance. Combining cores from different sources can cause subtle (or not-so-subtle) problems when they interact, forcing the team to debug the interaction, make necessary design changes, and then revalidate those changes. Arm provides a broad range of IP blocks that are prevalidated to work together, simplifying the design process. They employ standard SoC interfaces such as AMBA and ACE to enable plug-and-play chip development. Together, these proven IP blocks reduce risk in the nondifferentiated portion of the SoC and allow designers to focus on custom or value-added portions of the chip.

Reference designs are another important tool for SoC designers, providing complete RTL for an SoC subsystem or even an entire chip. Instead of creating a chip from scratch, the designer can use portions of the reference design while modifying and extending the platform as needed. These designs demonstrate how to configure and connect a set of cores, including subtleties such as sizing the caches and buses to avoid performance bottlenecks. Even if the SoC designer doesn't use the reference code directly, it provides a template for configuring and connecting the cores. Without a reference design, the development process often takes longer, as the SoC team requires extra iterations before settling on a final design.

Arm supplies reference designs and complete platforms for various end applications. For example, its Juno development boards include an SoC that integrates Cortex-A72 and Cortex-A53 CPUs along with a Mali GPU, CoreLink interconnect, and interface IP, as Figure 1 shows. The platform supports LogicTile Express FPGA plug-in cards, which enable customers to extend the prototyping system with their own logic designs. The boards can also be used for software development.

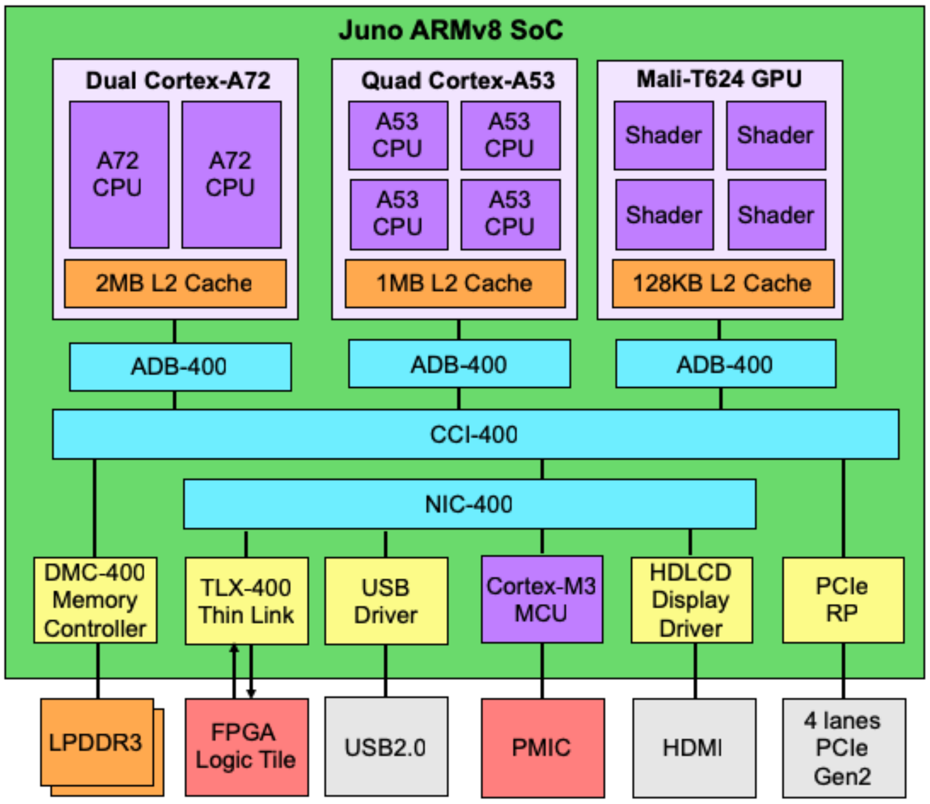

**Figure 1. Juno reference design.** This complete SoC design demonstrates how to integrate the various components and provides a software-development platform.

Even after everything is working together, the SoC may have bottlenecks or mismatches that degrade system performance. Once these problems are identified, the team can address them by adding bandwidth, compute power, memory, or other resources to the design, often in an iterative process. To accelerate this process, Arm provides holistic system debug tools and performance monitoring capability to accelerate SoC-level verification. It also supplies system-analysis tools to enable rapid exploration of different design options for performance and cost (die-area) tuning.

## *Safety and Security*

For IoT and other applications, a significant challenge is security. Some CPU cores provide few, if any, provisions to prevent malicious software from taking control of the system. Even when a CPU offers security features, software may not use them or a firmware bug could undermine them. Creating a truly robust system requires an end-to-end design process based on thorough knowledge of potential threats and how to block them.

Arm's Platform Security Architecture (PSA) establishes a common secure hardware-software framework for devices that use Arm-based processor cores, including those with TrustZone capability. It includes hardware- and firmware-architecture specifications, an open-source firmware reference implementation, security analyses, and threat models. Processor vendors and IoT-device makers can submit their products to an independent testing and PSA-certification program. Arm worked with leading test laboratories, including Brightsight and Underwriters Laboratories (UL), to develop the program for certifying chips, software, and devices that employ the PSA architecture and root of trust (RoT). This approach allows SoC and system designers to deliver secure products without becoming security experts.

To further ease development of secure IoT processors, the company offers the CoreLink SSE-200 subsystem. As Figure 2 shows, the SSE-200 comprises a complete PSA-compliant subsystem integrating Cortex-M33 CPUs along with TrustZone protection of all processor I/Os and IP blocks.

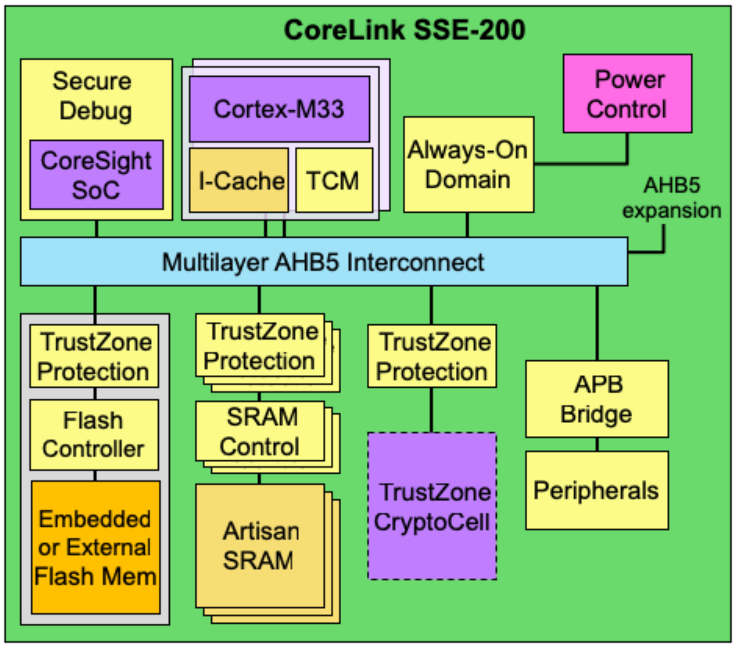

**Figure 2. CoreLink SSE-200.** This reference design demonstrates how to use TrustZone to secure on-chip memory and other IP blocks.

Safety is a crucial concern for automotive SoCs. As vehicles add more driver-assistance (ADAS) features and move toward autonomous driving, automakers place stringent safety requirements on the semiconductors they employ. These products must meet safety standards such as ISO 26262 and ASIL D. Many chipmakers would like to compete for these lucrative automotive designs but are unfamiliar with these complex standards. Building the necessary expertise can delay an SoC project, but so can reworking the chip design because it fails the certification process.

To meet these safety standards, chip designers must identify the provenance (source) of each IP block in the chip and certify it against the standards. Doing so is much simpler with an IP vendor such as Arm, which provides functional-safety packages that assist with safety-critical design. The company works with EDA vendors to develop ISO 26262–compliant verification methods, which ensure the IP and design tools meet functional-safety requirements. It has also worked with automotive vendors over many years, as its cores appear in numerous automotive microcontrollers, in-vehicle information systems, and advanced driver-assistance systems (ADASs).

To meet ASIL D safety requirements, processors must protect against transient faults as well as permanent failures. Arm's automotive-enhanced cores, such as the Cortex-A65AE and Cortex-A76AE, include the necessary fault-detection and protection mechanisms. For example, they support split-lock operation so that two cores can work in lockstep, checking each other for faults. If one core experiences a permanent fault, the software can reboot with the cores in split mode, enabling the system to continue operating with the good core.

## *Physical Design*

Once the RTL is complete and validated, the SoC team must develop the physical layout. Although in theory a synthesis tool can create the physical design at the push of a button, it's never that easy for a complex SoC. Cores from multiple sources can require different design flows to compile properly and satisfy the fab's design rules. The initial layout often fails to meet timing across all process variations. (For example, the slow-slow corner indicates that both P transistors and N transistors operate at their minimum speed, whereas the fast-slow corner indicates fast P transistors but slow N transistors.) The team must iterate the layout until it meets all design rules and timing requirements. Because of these multiple iterations, physical design can be one of the most time-consuming phases of the development process.

To facilitate synthesis, Arm provides reference flows for all major EDA vendors; these scripts aid designers in generating layouts more quickly while meeting all design rules. The flows ensure that power distribution, timing, and noise fall within acceptable limits for all standard process corners, helping designers achieve sign-off.

Layouts should also meet their performance, power, and area targets. Arm doesn't stop at designing RTL; the company tests its reference flows to ensure that its cores meet their PPA targets when used with multiple EDA tool chains. Furthermore, Arm checks that its cores achieve these targets while delivering industry-standard levels of fault testing for a smooth manufacturing ramp. In many cases, the company takes the final step of validating its PPA measurements on real silicon by fabbing test chips. This experience with physical layout and manufacturing enables it to better support its customers.

Customers that want an even simpler process can work with one of several design-services vendors that Arm has approved. These vendors have extensive experience with physical design of Arm IP cores. Alternatively, customers can license hard cores from Arm's Physical IP group. These cores provide a complete physical layout for popular foundry processes, such as TSMC 28HPC or 16FFC, and they include both standard-cell logic and memory. A hard core is guaranteed to meet specific PPA levels in that process and can be inserted directly into a larger SoC layout.

## *Software Ecosystem*

SoC design teams often have as many software engineers as hardware engineers—or more. Chip companies such as Intel and Qualcomm employ thousands of programmers and provide a tremendous amount of firmware and even application code to their customers. Choosing a CPU with a broad software ecosystem can greatly simplify this software development. On the other hand, a poorly supported CPU can delay the software schedule, preventing an SoC from reaching the market, or it can necessitate additional software engineers, putting the project over budget.

Any software ecosystem starts with the basics: compilers, debuggers, and other development tools. Arm is one of the most popular ISAs for developers, so software vendors (ISVs) provide a wide range of code-development tools. To simplify software development, Arm CPUs implement debug hardware called CoreSight that allows access to internal registers and other processor state. Many third-party debuggers work with CoreSight, showing software developers how their code runs on the CPU cycle by cycle. Because this debug approach is common across most Arm CPUs, ISVs can implement CoreSight once and support many different cores. Most embedded-systems programmers are familiar with these Arm-based tools, so they can immediately be productive.

For operating systems, Arm covers the alphabet from Android to Zephyr. Other CPUs support a more limited range of operating systems and thus might be inapplicable to certain types of end products. But an OS is only part of the software stack required to build a complete application. For example, Arm's Mbed OS includes middleware for security and connectivity—important capabilities for IoT and other embedded products. The CMSIS library provides literally thousands of software "packs" from Arm and third parties. Without these components, an SoC vendor might have to develop much of this code itself, adding time and expense to the project.

Software-development tools can emulate an ISA, but programs execute much faster when running natively. Dozens of inexpensive development boards feature Arm-based SoCs, including the popular Raspberry Pi and Arduino boards. Microcontroller vendors such as Microchip, NXP, Renesas, STMicroelectronics, and Texas Instruments offer development boards containing their Arm-based processors. Cortex-M CPUs are also easy to instantiate in FPGA-based development boards, including Arm's MPS2 and MPS3. These development boards allow programmers to quickly test and debug their code even before first silicon is available, speeding SoC bringup and getting the product to market sooner.

Building a flourishing ecosystem requires not just time but instruction-set compatibility. Arm places strict requirements on its ISA licensees as well as its own cores to ensure this compatibility. Although the company has evolved its architecture several times over the past few decades to add modern features, today's cores and chips remain binary compatible with older (sometimes much older) software. This compatibility attracts software developers, as they can write once and deploy many times. Instruction sets that lack these requirements encourage innovation, but this innovation can ultimately lead to fragmentation as different vendors implement different instructions for similar tasks. Fragmentation hinders ecosystem development, as software vendors must decide which version of the ISA to support; targeting the least common denominator ensures compatibility but will degrade performance on at least some processors.

## Conclusion

After nearly three decades and many billions of units shipped, the Arm ISA has achieved a virtuous circle. Because it supports so many design starts and software developers, third parties are eager to target Arm with their software, tools, and development boards. The wide availability of this infrastructure then encourages more design starts and software developers. Any company designing an Arm-based product can easily acquire trained and experienced Arm developers, reducing hiring and development time.

Arm's position as the highest-volume processor ISA is no matter of luck. The company works hard to make life easier for its customers. Proven RTL designs, complementary IP blocks, verification models, and reference designs help customers efficiently complete their SoC-level logic design. Validated EDA tool flows, proven PPA characteristics, and complete physical designs help customers successfully reach tapeout. At the same time, the customer's software team can take advantage of commercial-quality integrated design environments (IDEs), a wide range of operating systems, and extensive library code to build the necessary software stack. Arm and its commercial partners provide full support for all of these components. All of these capabilities help customers get their designs to market as quickly and smoothly as possible.

Open source has become popular as a way to eliminate license fees and royalties. But what works for software applies less well to hardware. Open-source CPU cores are often unproven in high-volume manufacturing and typically lack verification models and EDA tool flows. PPA can vary widely from one implementation to another. Support is generally limited to bare-bones documentation. For software developers, open-source ISAs may offer basic IDEs and operating systems, but they lack the extensive code libraries and commercial OS and tool support that Arm offers. Having no central control, these ISAs by their nature begin to fragment.

A low-cost unconstrained environment is good for research and some small chip developers. But for most commercial SoC projects, CPU license fees are a small portion of the design budget. Choosing a CPU with low or no license fees can create hidden costs—such as creating verification models, tool flows, and library code—that more than offset the reduction in up-front fees. Furthermore, any schedule delays caused by buggy RTL, physical-level redesign, or software problems could cost millions of dollars in missed market opportunities. Before putting an important SoC product at risk, consider the potential cost of a free ISA.

*Linley Gwennap is principal analyst at The Linley Group and editor-in-chief of* Microprocessor Report. *The Linley Group offers the most-comprehensive analysis of microprocessors and SoC design. We analyze not only the business strategy but also the internal technology. Our in-depth articles cover topics including embedded processors, mobile processors, server processors, AI accelerators, IoT processors, processor-IP cores, and Ethernet chips. For more information, see our website at www.linleygroup.com.*