SIEMENS
Ingenuity for life

Siemens Digital Industries Software

Ensuring the success of your RISC-V product with a commercial-grade software development ecosystem

Executive summary

The headlines continue to grow as more companies move their embedded projects to the RISC-V platform. Before you decide to move your next embedded project to RISC-V, know how to get the most out of this innovative ISA. This white paper explores the history of RISC-V and what it is (and isn't), and explores the current state of open-source RISC-V toolchains, and how Siemens toolchain services can make the most of your application on RISC-V platforms.

Introduction

In the nine years since the initial public release of an open-source microprocessor Instruction Set Architecture, RISC-V has grown from an academic innovation into a compelling option for mainstream products for a wide variety of embedded applications. Storage giant Western Digital is not only moving production hardware to RISC-V, they have released their own RISC-V based SweRV Core™ to the open-source community;[1] and graphics leader NVIDIA has decided to base their next-generation Falcon logic controller on RISC-V.[2] Semico Research estimates that by 2025 there will be over 62 billion RISC-V-based devices shipped globally.[3]

But what exactly is RISC-V, and how would you implement your design when targeting RISC-V? As more hardware implementations and development boards come to market for entry-level designers, Siemens Digital Industries Software, is stepping in to offer toolchain services that will enable your embedded applications to take maximum advantage of RISC-V hardware.

In this white paper we will explore:

- The History of RISC-V
- RISC-V Benefit
- An overview of the RISC-V Architecture
- Toolchain Considerations for RISC-V-based commercial products
- Sample Commercial-grade RISC-V SDKs

RISC-V history

In 2010 computer science experts Krste Asanović and David Patterson, along with a group of graduate students at the University of California at Berkeley ParLab, decided as part of their research into advanced parallel computing, to develop and publish an open-source compute architecture. The reasoning behind their decision was that in general, RISC-based (Reduced Instruction Set Computer) ISAs (Instruction Set Architectures) published in the prior twenty years possessed a majority of commonalities, with few unique differences between them. Their frustration was that the majority of RISC-based ISAs were owned and copyrighted by CPU vendors that, with only a few limited exceptions, charged royalty fees for their use in end products. Further, these vendors might release limited documentation, often only under non-disclosure agreement, and rarely gave insight into the original decisions that drove their architectural design choices.[4] In May of 2011, with the Berkeley group's initial ISA publication, the RISC-V initiative was born.

Following that initial publication, the non-profit RISC-V Foundation formed in 2015, was established to expand upon a growing community of contributors, formalizing their ISA and placing it in the public domain, available via GitHub. At that point, RISC-V became royalty-free, usable by anyone. Not just for research or education, and not just for prototyping, but for production. You never have to pay a fee to use the RISC-V ISA, no matter how many times you download it, and no matter how many devices you use it in. You also don't have to be a member of the RISC-V Foundation to download the ISA. The foundation and membership exist to promote the RISC-V initiative and to vote and approve updates to the ISA via extensions. Use of the architecture is open to all who can access GitHub.

RISC-V benefits

RISC-V provides a new option for microcontroller and microprocessor designs. Embedded designers now have a choice for basing their products either on one of several proprietary commercial processor architectures, or using the RISC-V open architecture with no royalty fees or charges per use. In many markets where the number of processors in a production series can potentially number in the millions, the cost savings from eliminating paying royalty fees can be substantial. Several disk drive vendors and automotive companies are already heavily invested in RISC-V projects to reduce costs.

And the upsides extend beyond cost. Defense and aerospace designers have historically needed to enter into legal negotiations and pay additional fees when they have to consider modifying a proprietary ISA for specialized applications, or alternatively they had to come up with their own architecture from scratch. Now they can download the well-documented open-source RISC-V ISA for free, and modify it for their own specialized use. Per the language of the RISC-V open license, users can modify the ISA at will, with no obligation to upload or publish their modifications, companies can keep their modified RISC-V-based ISA proprietary.

An overview of the RISC-V architecture

RISC-V is a processing architecture, but not a processor implementation. The Berkeley designers were keen to avoid any bias towards any particular microarchitecture, or silicon technology as they developed the ISA. RISC-V can be implemented in an ASIC or FPGA, and on any process node. The first processor implementations created by the Berkeley team were developed in the open-source Chisel hardware description language, and have also been posted publicly for anyone's use. In addition, the list of silicon vendors offering RISC-V based hardware is rapidly growing. At publication of this article, over 70 RISC-V processor implementations are listed on the Foundation website.

The RISC-V architecture is modular, starting from the base specification that supports 32 and 64-bit integer ISAs. A 128-bit integer ISA is also defined, but intentionally left open as more designs become available using these denser memory maps. The base ISA is broken into two volumes, the User-Level Architecture (Volume I), and the Privileged Architecture (Volume II). The Privileged architecture is designed for isolation of software stack components from the user portion of the operating system and application code.

Starting from a base ISA and then adding the necessary extensions like single-precision, double-precision, and quad-precision floating point, packed-SIMD instructions, or virtualization support; RISC-V can address a variety of system models from small footprint needs, to multiprocessor applications, and eventually even DSP applications using SIMD.

Toolchain considerations for RISC-V based commercial products

With embedded RISC-V processor IP beginning to show up in new products, enabling software developers to port application code is a key requirement. The compiler, libraries, linker and debugger tool set used to convert application source code to binary code and get it running on the target hardware, are collectively referred to as a toolchain. Toolchains integrated into an application development suite to support a specific product are known as a Software Development Kit (SDK) which is a key part of the ecosystem needed to support RISV-C commercial product deployment.

There are two open-source compiler framework projects that offer at least a partial solution to the RISC-V toolchain requirement. The GCC (GNU Compiler Collection) project is an open-source initiative that provides compiler support for many of the RISC-V standard ISA variants as shown in table 1. GCC also offers support for both Linux and "Bare-metal" (RTOS and no operating system) targets. GCC versions 7.1 and higher include support for RISC-V. The LLVM project is a newer open-source compiler framework initiative which also provides compiler backend and library support for RISC-V standard ISA variants as shown in table 1. LLVM versions 9.0.0 and higher include support for RISC-V. The LLVM project has traditionally focused on Linux OS target operating system support. However, there is growing interest within the LLVM community to extend support for embedded bare-metal target devices.

ISA variant	Description	ISA status	GCC	LLVM
RV32I	Base 32-Bit integer instruction set	Released	X	X
RV64I	Base 64-Bit integer instruction set	Released	X	X
RV128I	Base 128-Bit integer instruction set	Released		
Zifenci	Instruction-fetch fence	Released		X
RV32E	Base 32-Bit integer instruction set for embedded	Released	X	
M	Integer multiplication and division	Released	X	X
A	Atomic instructions	Released	X	X
Zicsr	Control and status registers	Released		
F	Single-precision floating-point	Released	X	X
D	Double-precision floating-point	Released	X	X
Q	Quad-precision floating-point	Released		
RVWMO	Consistency model	Released		
L	Decimal floating-point	Future		
C	Compressed instructions	Released	X	X
B	Bit manipulation	Future		
J	Dynamically translated languages	Future		
T	Transactional memory	Future		
P	Packed-SIMD	Future		
V	Vector operations	Future		
N	User-level interrupts	Future		
Zam	Misaligned atomics	Released		
Ztso	Total store ordering	Released		X

Using open-source toolchains

While software developers targeting RISC-V devices are certainly free to download GCC or LLVM source code directly from the projects' public open-source repositories, this is not usually an efficient first step in porting application software to a RISC-V processor. Building and validating your own open-source based toolchain is a complex and lengthy process fraught with many potential pitfalls, especially for those new to building their own toolchains. Fortunately, many RISC-V chip and IP vendors provide open-source based toolchain reference distributions to help developers evaluate their IP. While these reference toolchains are useful for evaluating RISC-V IP, they are not productized solutions to enable developers and end users to deploy software applications to commercial RISC-V based products.

There are three key requirements to consider when procuring customer-ready open-source based toolchains to enable developers to use a RISC-V based platform: Commercialization, Customization and Support.

Toolchain commercialization

Toolchain commercialization is the process of transforming the publicly available open-source code into a tested and supportable distribution of executable toolchain components and libraries configured to support the target devices. The two most important toolchain commercialization considerations are selection of the compiler version as well as the target device variants to be supported. Leveraging the open-source compiler project test suites and test frameworks, such as DejaGNU, will help provide toolchain test coverage. The devil is in the details in terms of getting the toolchain executable correctly built and passing the test suites. This can be a daunting task since open-source software is always under construction. Thousands of community members are working on various bug fixes and new features at any given time. As a result, there are typically a number of tests in each test suite which are expected to fail. Differentiating between expected test failures (XFAILS) and real test failures and, of course, actually resolving the real failures is tedious work requiring highly specialized compiler design domain knowledge. The end objective is to set up a reliable, repeatable test harness with appropriate toolchain test coverage which can be re-run whenever needed for ongoing toolchain support. Another key commercialization consideration is remediation of Common Vulnerabilities and Exposures (CVEs). Applying the correct cybersecurity patches to toolchain components (usually libraries) during the toolchain build process is essential to producing high quality toolchain releases for developer SDKs.

Toolchain customization

Toolchain customization provides feature extensions and performance optimizations beyond what is currently available from the open-source community. For RISC-V, toolchain customization can take the form of compiler extensions to support hardware vendor-specific RISC-V ISA or customizations enabled via the RISC-V open ISA. Target-specific application performance optimization is another area of toolchain customization that can help maximize target hardware application performance. In addition, toolchain customization can also include compiler bug fixes and new feature completion not yet publicly available from the open-source community. The latest source code may not always be available via the open-source project's upstream public repositories. Work in progress typically resides on private development branches until release to the mainline. Also, there is no overall open-source project development plan with a fixed delivery schedule for specific new features. Thus toolchain customization may be needed in order to deliver an optimum well-configured toolchain that meets commercial RISC-V based product requirements and release schedules.

Toolchain support

Toolchains regularly have CVEs reported against them, including issues in the compiler runtime libraries. Undiscovered CVEs in compiler runtime libraries get linked into application software and can thus be inadvertently deployed to customers via a product release. This in turn will create the risk of having to provide a hot-fix software update to address newly discovered CVEs at some point in the future. In such cases having a toolchain support contract in place that provides ongoing CVE remediation will ensure your toolchain is ready to build the software update needed to quickly and effectively resolve such a crisis without having to first sort out potentially years of toolchain patches and updates. Toolchain long-term support is intended to keep older version toolchains updated with critical CVE patches and bug fixes while minimizing the total number of overall toolchain changes to reduce the risk of introducing new issues.

Sample commercial-grade RISC-V SDKs

Siemens has recently announced GCC and LLVM based sample commercial-grade RISC-V SDKs targeting the SiFive HiFive Unleashed[5] reference board. The toolchain configuration details for the two sample SDKs are shown in table 2. The RISC-V ISA Variants supported by the sample SDKs are shown in table 3. These sample SDKs are available for free download, without technical support.

	Description	ISA status
Compiler	GCC	llvm
Linker	GNU ld	lld
Debugger	GDB	GDB (future plans for lldb)
Language support	C, C++	C, C++
Host OS support	Windows, Linux	Linux
Target OS support	Bare-metal (RTOS or No OS)	Bare-metal (RTOS or No OS)
Target hardware support	SiFive HiFive Unleashed[5] Board, Ibex Rocket Core[6]	SiFive HiFive Unleashed[5] Board, Ibex Rocket Core[6]

Table 2. Toolchain configuration for the Siemens sample RISC-V SDKs.

Variant	Board/Core	SDK type	ISA status
RV32IMC	SiFive HiFive LowRISC Ibex	GCC	**RV32I** + integer **M**ultiplication and division + **C**ompressed instructions
RV64IMAC	SiFive HiFive	GCC LLVM	**RV64I** + integer **M**ultiplication and division + **A**tomic instructions + **C**ompressed instructions
RV64IMAFDC	SiFive HiFive	GCC LLVM	**RV64IMAC** + integer **M**ultiplication and division + Single-Precision **F**loating-Point + **D**ouble-Precision Floating-Point + Atomic instructions + Compressed instructions
RV32GC	QEMU	GCC	**RV32I** + **G** (integer Multiplication and division + Atomic instructions + Single –Precision Floating-Point + Double-Precision Floating-Point + Instruction-Fetch Fence) + **C**ompressed instructions

Table 3. RISC-V ISA variants supported by the Siemens sample RISC-V SDKs.

Conclusion

As RISC-V enters the mainstream of embedded commercial products, a customer-ready software development ecosystem will be essential for marketplace success. Toolchain commercialization, customization and support are essential ingredients for deploying commercial-grade open-source based toolchains in SDKs supporting RISC-V based product development and post release software updates.

Siemens has a 20+ year proven track record in delivering commercial-grade toolchains and toolchain support services for a wide variety of processor architectures.

For more information on the RISC-V ISA, see the specification published at the RISC-V Foundation website https://riscv.org/specifications/isa-spec-pdf/ or via the RISC-V area on GitHub https://github.com/riscv.

For more information on Siemens Digital Industries Software embedded solutions go to siemens.com/software.

Pricing references

1. Western digital delivers new innovations to drive open standard interfaces and RISC-V processor development. Press Release – December 2018. https://www.westerndigital.com/company/newsroom/press-releases/2018/2018-12-04-western-digital-delivers-new-innovations-to-drive-open-standard-interfaces-and-risc-v-processor-development
2. Xie, Joe. NVIDIA RISC V Evaluation Story. 4th RISC-V Workshop, July 2016. https://riscv.org/wp-content/uploads/2016/07/Tue1100_Nvidia_RISCV_Story_V2.pdf https://www.youtube.com/watch?v=gg1lISJfJI0
3. RISC-V will see 146.2% CAGR 2018-2025, forecasts Semico Research, November 2019 https://www.electronicsweekly.com/news/business/146-cagr-risc-v-2025-2019-11/ https://semico.com/content/risc-v-market-analysis-new-kid-block
4. Asanović, K. and Patterson, D. Instruction sets should be free: the case for RISC-V (EECS-2014-146), August 2014. https://people.eecs.berkeley.edu/~krste/papers/EECS-2014-146.pdf
5. HiFive Unleashed is a trademark of SiFive Inc. https://www.sifive.com/boards/hifive-unleashed
6. Ibex is an open-source RISC-V processor design maintained by LowRISC. https://github.com/lowrisc/ibex

Siemens Digital Industries Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Unit 901-902, 9/F
Tower B, Manulife Financial Centre
223-231 Wai Yip Street, Kwun Tong
Kowloon, Hong Kong
+852 2230 3333

About Siemens Digital Industries Software

Siemens Digital Industries Software is driving transformation to enable a digital enterprise where engineering, manufacturing and electronics design meet tomorrow. Xcelerator, the comprehensive and integrated portfolio of software and services from Siemens Digital Industries Software, helps companies of all sizes create and leverage a comprehensive digital twin that provides organizations with new insights, opportunities and levels of automation to drive innovation. For more information on Siemens Digital Industries Software products and services, visit siemens.com/embedded or follow us on LinkedIn, Twitter, Facebook and Instagram. Siemens Digital Industries Software – Where today meets tomorrow.

siemens.com/embedded

82911-C1 5/20 H