Advantages and disadvantages of the RISC-V ISA (Instruction Set Architecture) in comparison to the ARMv8 ISA

1st Michael Schneider
Faculity of Computer Science and Mathematics
OTH Regensburg
Regensburg, Germany
michael4.schneider@st.oth-regensburg.de

2nd Florian Henneke
Faculity of Computer Science and Mathematics
OTH Regensburg
Regensburg, Germany
florian.henneke@st.oth-regensburg.de

3rd Alexander Schmid
Faculity of Computer Science and Mathematics
OTH Regensburg
Regensburg, Germany
alexander2.schmid@st.oth-regensburg.de

***Abstract*—text**
***Index Terms*—keyword1, keyowrd2**

I. Introduction

A. Topic

B. Motivation

C. Goal

D. Overview of paper

II. Background

A. Instruction Set Architectures

B. RISC

C. ARM

text

D. RISC-V

text

III. Concept and Methods

In order to determine whether RISC-V will gain a significant market share in the following years, it is useful to compare the two Instruction Set Architectures (ISAs) across a set of criteria that are relevant to semiconductor companies when evaluating which ISA to use with a new CPU design.

The first of these criteria is the ISA's business model. ISAs are often protected by patents that prohibit anyone not licensed by the patent owner from distributing Central Processing Units (CPUs) that implement that ISA. [1] Whether these patents exist and the licensing terms are an important factor when deciding which ISA to use.

The ISA's complexity refers to the amount of effort required to implement the ISA. The more complex an ISA is, the more developer time is spent on implementing and verifying a CPU's compatibility to the ISA, instead of optimizing the CPU for performance and efficiency, increasing a CPU's development cost.

A CPU's performance can be evaluated under multiple aspects, including the rate of instructions, the rate of floating point operations or the speed at which the CPU executes a given program. Efficiency considerations, such as the code size of a given program when compiled to the CPU's instruction set or the amount of power the CPU consumes when executing a given program are closely related to performance and shall, for the purposes of this paper, be grouped under performance.

A program's code size is greatly influenced by the ISA, since the instruction set and the compiler used are the only two factors that influence code size. As such, the code size is an important factor to consider when choosing which ISA to implement for a new processor design, especially for microcontrollers that are usually very constrained in the size of their program memory. For the other performance aspects, it is debatable to which extent they are influenced by the CPU's instruction set as opposed to the concrete implementation of the CPU. [2] [3]

ISAs often allow for a number of instruction set extensions that may or may not be implemented by a given CPU. These usually allow faster and more efficient processing of programs for a given use case, such as Single Instruction, Multiple Data (SIMD) extensions that optimize signal processing and media applications, Advanced Encryption Standard (AES) extensions that optimize cryptography or ISA-extensions with shorter instructions for applications that are constrained in program memory. Having a small base instruction set with many fine grained extensions improves the flexibility of the ISA, allowing CPUs to be optimized for specific use cases, increasing performance and efficiency for those use cases. ISA extensions do however pose a disadvantage when distributing

precompiled software to end users, as a piece of software that uses a certain ISA extension can't be executed on CPUs that don't implement that extension, potentially increasing the number of different versions of that software that need to be distributed.

An ISA's ecosystem refers to the software that supports that ISA, especially compilers that compile to that ISA, operating systems and libraries. When developing a new CPU it is preferrable to use an ISA with a large ecosystem, in order to maximize the amount of software that can run of that CPU. This is especially important in consumer desktop and mobile devices where a large variety of software is to be executed, but less important in embedded applications, where the software running on a microcontroller is specifically developed for that application and microcontroller only.

A. Business Models

Who develops the CPU cores, how can you get access to them? Who supports chip manufacturers in designing a chip with that CPU core?

B. Structure and Complexity

How many instructions are there? How complex does that make the implementation of a core?

C. Performance

What are the differences in code size? Can we accurately compare the execution speed of both ISAs?

D. Extensibility

What instruction set extensions are there for both ISAs? Who can develop new extensions?

E. Ecosystem

Which compilers support ARM and RISC-V? Which operating systems and libraries?

IV. Discussion

A. ARM

What are the advantages of ARM compared to RISC-V?

B. RISC-V

What are the advantages of RISC-V compared to ARM?

C. Future directions and challenges

How can we more accurately measure performance differences between ARM and RISC-V and how do ISA extensions affect performance?

V. Conclusion and Outlook

A. Summary of results

B. Interpretation of results

C. Future directions

VI. Overview of Literature

Alexander Schmid [3] [4] [5] [6] [7] [8] [9] [10] [11]
Florian Henneke [10] [12] [5] [13] [14] [15] [16] [14]
Michael Schneider [17] [18] [19] [20] [21] [22] [23]

References

[1] G. Tang and I. W. Brown, "Intel and the x86 Architecture: A Legal Perspective," Harvard Journal of Law & Technology Digest, 2011, accessed on 2020-11-02. [Online]. Available: https://jolt.law.harvard.edu/digest/intel-and-the-x86-architecture-a-legal-perspective

[2] E. Blem, J. Menon, and K. Sankaralingam, "Power struggles: Revisiting the risc vs. cisc debate on contemporary arm and x86 architectures," in *2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*, 2013, pp. 1–12.

[3] A. Akram, "A Study on the Impact of Instruction Set Architectures on Processor's Performance," Ph.D. dissertation, Western Michigan University, 08 2017.

[4] *Arm® Architecture Reference Manual. Armv8, for Armv8-A architecture profile*, Issue F.c ed., ARM, Jul. 2020.

[5] K. Asanović and D. A. Patterson, "Instruction Sets Should Be Free: The Case For RISC-V," University of California, Berkeley, Tech. Rep. UCB/EECS-2014-146, 2014.

[6] Heui Lee, P. Beckett, and B. Appelbe, "High-performance extendable instruction set computing," in *Proceedings 6th Australasian Computer Systems Architecture Conference. ACSAC 2001*, 2001, pp. 89–94.

[7] D. Patterson, J. Bennett, P. Dabbelt, C. Garlati, and O. Shinaar, "Initial Evaluation of Multiple RISC ISAs using the Embench™ Benchmark Suite," Dec. 2019, accessed on 2020-10-24. [Online]. Available: https://riscv.org//wp-content/uploads/2019/12/12.10-12.50a-Code-Size-of-RISC-V-versus-ARM-using-the-Embench%E2%84%A2-0.5-Benchmark-Suite-What-is-the-Cost-of-ISA-Simplicity.pdf

[8] M. Perotti, P. D. Schiavone, G. Tagliavini, D. Rossi, T. Kurd, M. Hill, L. Yingying, and L. Benini, "HW/SW Approaches for RISC-V Code Size Reduction," in *Workshop on Computer Architecture Research with RISC-V. CARRV 2020*, 2020.

[9] C. Shore, *ARMv8-A Architecture Overview*, ARM Limited, Sep. 2015.

[10] A. S. Waterman, "Design of the RISC-V Instruction Set Architecture," Ph.D. dissertation, University of California, Berkeley, 2016.

[11] X. H. Xu, S. R. Jones, and C. T. Clarke, "ARM/THUMB code compression for embedded systems," in *Proceedings of the 12th IEEE International Conference on Fuzzy Systems (Cat. No.03CH37442)*, 2003, pp. 32–35.

[12] L. Ryzhyk, "The ARM Architecture," Tech. Rep., 2006.

[13] S. Furber, *ARM System-on-Chip Architecture*, 2000, no. a.

[14] Microsoft, "Windows 10 on ARM," 2020. [Online]. Available: https://docs.microsoft.com/en-us/windows/uwp/porting/apps-on-arm

[15] Greenwaves Technologies, "Arm® Mbed™ OS Porting Manual for GAP8 ," accessed on 2020-10-28. [Online]. Available: https://greenwaves-technologies.com/manuals/BUILD/MBED-OS/html/index.html

[16] Amazon Web Services, "Using FreeRTOS on RISC-V Microcontrollers," accessed on 2020-10-28. [Online]. Available: https://www.freertos.org/Using-FreeRTOS-on-RISC-V.html

[17] D. Patterson, "50 Years of computer architecture: From the mainframe CPU to the domain-specific tpu and the open RISC-V instruction set," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, 2018, pp. 27–31.

[18] J. Hennessy, *Computer architecture : a quantitative approach*. Waltham, MA: Morgan Kaufmann, 2012.

[19] R. D. Vladimir Herdt, Daniel Große, *Enhanced virtual prototyping featuring risc-v case studies*. S.l: SPRINGER NATURE, 2020.

[20] M. D. H. of Wisconsin-Madison; Dave Christie; David Patterson; Joshua J. Yi; Derek Chiou; Resit Sendag, "Proprietary versus open instruction sets," *IEEE Micro*, 2016.

[21] S. Higginbotham, "The Rise of RISC," *IEEE Spectrum*, 2018.

[22] R. Dirvin, "The arm ecosystem: More than just an ecosystem, it's oxygen for soc design teams," April 2019. [Online]. Available: https://www.arm.com/company/news/2019/04/the-arm-ecosystem-more-than-just-an-ecosystem

[23] Z. Bandic, IEEE Computing Society, March 2019. [Online]. Available: http://www.hsafoundation.com/the-inherent-freedom-of-heterogeneous-systems/