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 (Initial section written by Alexander Schmid)

Given that the goal of this paper is 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, these criteria shall be defined in the following section.

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 the implementation of the ISA, instead of optimizing the CPU for performance and efficiency, increasing a CPU's development cost. [2]

A CPU's performance usually refers to the speed at which the CPU executes a given program. Efficiency considerations, such as the code size of a given program 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.

ISAs often allow for a number of instruction set extensions that may or may not be implemented by a given CPU. The choice of extensions greatly influence the flexibility of an ISA. Thus they are an important factor to consider which ISA to implement for a new CPU, because these have a large impact on performance and development cost of a CPU.

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.

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 (written by Alexander Schmid)

Code size is the performance parameter most influenced by the ISA and not by the CPU's implementation, given that the ISA and the compiler are the only two factors that influence code size. For the other performance aspects, the influence of the ISA is debatable [3] [4], so code size shall be the main focus of the performance comparison. Given that code size is most critical in embedded applications, the Embench benchmark suite is a good benchmark with which to compare code sizes. It consists of a number of programs frequently used in embedded applications, such as CRC, signal filtering, AES and QR code reading. [5] When compiling the Embench suite for both RV32IMC as well as 32-bit ARM with the Thumb-2 extension using GCC 7, the code for RISC-V is approximately 11% larger than the code for ARM. [6] Part of this gap in code size can be explained by the relative immaturity of the RISC-V implementation of GCC. RISC-V was introduced in 2017 and the code size of the Embench suite compiled for RISC-V is lower with newer versions of GCC, however it is still larger than the code generated for ARM as of 2019. [5]

In [6] an extension for RISC-V is introduced, called HCC, that is aimed at reducing the code size of RISC-V. This extension brings the code size gap down to 2.2% for the Embench suite and makes the RISC-V code smaller than ARM by 1.75% in a proprietary IoT benchmark developed by Huawei. [6]

TODO: In subsequent submissions, mention RV64 being significantly smaller than AArch64 as described in [7, page 62], and possibilities of comparing execution speed and energy efficiency as described in [3] and [4].

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. Advantages of ARM

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

B. Advantages of 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 [4] [8] [9] [10] [5] [6] [11] [7] [12]
Florian Henneke [7] [13] [9] [14] [15] [16] [17] [15]
Michael Schneider [18] [19] [20] [21] [22] [23] [24]

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] D. A. Patterson and D. R. Ditzel, "The case for the reduced instruction set computer," *SIGARCH Comput. Archit. News*, vol. 8, no. 6, p. 25–33, Oct. 1980. [Online]. Available: https://doi.org/10.1145/641914.641917

[3] 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.

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

[5] 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

[6] 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.

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

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

[9] 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.

[10] 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.

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

[12] 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.

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

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

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

[16] 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

[17] 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

[18] 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.

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

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

[21] 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.
[22] S. Higginbotham, "The Rise of RISC," *IEEE Spectrum*, 2018.
[23] 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
[24] Z. Bandic, IEEE Computing Society, March 2019. [Online]. Available: http://www.hsafoundation.com/the-inherent-freedom-of-heterogeneous-systems/