¡¡¡¡¡¡¡ HEAD ======= ¿¿¿¿¿¿¿
7662f47fb83a044f6c911d0137cc5090a4fc36ed

# A comparison of the ARMv8 and RISC-V Instruction Set Architectures

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 ¡¡¡¡¡¡¡ HEAD by the patent owner from distributing Central Processing Units (CPUs) that implement that ISA. [**?**]

Whether these patents exist and the licensing terms ======= by the patent owner from distributing CPUs that implement that ISA. [**?**] Whether these patents exist and the licensing terms ¿¿¿¿¿¿¿ 7662f47fb83a044f6c911d0137cc5090a4fc36ed 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, ¡¡¡¡¡¡¡ HEAD increasing a CPU's development cost. [**?**] ======= increasing a CPU's development cost. ¿¿¿¿¿¿¿ 7662f47fb83a044f6c911d0137cc5090a4fc36ed

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. [**?**] [**?**]

ISAs often allow for a number of instruction set extensions that may or may not be implemented by a given CPU. These usually allow ¡¡¡¡¡¡¡ HEAD 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 [**?**, page 52], Advanced Encryption Standard (AES) extensions that optimize cryptography [**?**] or ISA-extensions with shorter instructions for applications that are constrained in program memory. [**?**] ======= faster and more efficient processing of programs for a given use case, such as SIMD extensions that optimize signal processing and media applications, AES extensions that optimize cryptography or ISA-extensions with shorter instruc-

tions for applications that are constrained in program memory. ¿¿¿¿¿¿¿ 7662f47fb83a044f6c911d0137cc5090a4fc36ed 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

The code size of a given program is the performance factor that is most influenced by a CPU's ISA, since it is independent of the CPU's microarchitecture and only influenced by the ISA and compiler. Given that code size is most critical in embedded applications, the Embench benchmark suite is a good benchmark with which to compare code sizes. The Embench benchmark suite consists of a number of programs frequently used in embedded applications, such as CRC, signal filtering, AES or QR code reading. [**?**] When compiling the Embench suite for both RV32IMC as well as 32-bit ARM with the Thumb-2 extension with GCC version 7, the code for RISC-V is approximately 11% larger than the code for ARM. [**?**] Part of this gap in code size can be explained by the relative immaturity of the RISC-V support of GCC. RISC-V has only been added as a target for GCC in 2017 and the code size of the Embench suite compiled for RISC-V is lower with subsequent versions of GCC, however still larger than ARM as of 2019. [**?**]

In [**?**] an extension for RISC-V is introduced, called HCC, that contains a number of instructions 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. [**?**]

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

### 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 [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**]
Florian Henneke [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**]
Michael Schneider [**?**] [**?**] [**?**] [**?**] [**?**] [**?**] [**?**]