`rvcodec.js`: An Educational Converter for RISC-V Instructions

Joël Porquet-Lupine
jporquet@ucdavis.edu
Department of Computer Science
University of California, Davis
Davis, California, USA

Hikari N. Sakai
Department of Computer Science
University of California, Davis
Davis, California, USA

Abhi Sohal
Department of Computer Science
University of California, Davis
Davis, California, USA

ABSTRACT

`rvcodec.js` is an open-source, online RISC-V instruction encoder-decoder, specifically geared towards students learning about instruction formats. For improved usability, `rvcodec.js` uses bright colors to visually map the relationships between the instruction's assembly tokens and binary fields. Students can, for example, quickly identify which bits in the binary representation correspond to a certain immediate value in the assembly instruction.

`rvcodec.js` is available at https://luplab.gitlab.io/rvcodecjs/.

1 BACKGROUND

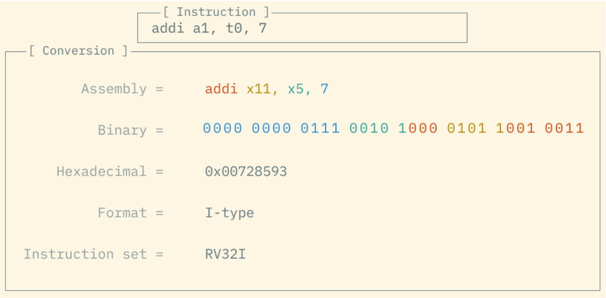
Undergraduate students taking a "Computer Organization and Assembly Language" course typically learn how assembly instructions are encoded in hexadecimal or binary. They learn about the different instruction types (e.g., R-type, I-type, etc. in RISC-V), and how instructions of each type are structured. Eventually, they are able to answer questions such as "Assuming RISC-V instruction `0x00728593` in hexadecimal, what is the corresponding assembly instruction?", by manually decoding each field of the instruction, navigating the various encoding tables (e.g., the opcode map), etc.

In recent years, the RISC-V instruction set architecture (ISA) seems to have surpassed the MIPS ISA as the most education-friendly ISA. Unsurprisingly, a plethora of RISC-V centric educational tools have been created in response, especially online ones [1, 3, 4]. However, to the best of our knowledge, the RISC-V still lacked its counterpart to an existing and very useful online MIPS instruction encoder/decoder [2].

2 TOOL OVERVIEW

Our tool, `rvcodec.js`, is an online RISC-V instruction encoder-decoder, which currently supports the RISC-V instruction sets RV32I, Zifencei, and Zicsr [5]. `rvcodec.js` is implemented as a single-page web application in plain HTML/CSS and client-side Javascript, and has no external dependencies. After the client receives the page, our tool does not make any further server requests, meaning it can be used completely offline.

Figure 1 shows a screenshot of `rvcodec.js`'s interface. The user can enter a RISC-V instruction, either in its assembly format (as shown in the figure) or in its digital format (binary or hexadecimal), and the tool will display all of its possible representations, as well as some additional information about the instruction (i.e., its format and the instruction set it belongs to).



```
[ Instruction ]
addi a1, t0, 7
[ Conversion ]
Assembly =           addi x11, x5, 7
Binary =             0000 0000 0111 0010 1000 0101 1001 0011
Hexadecimal =        0x00728593
Format =             I-type
Instruction set =    RV32I
```

Figure 1: Example of an instruction conversion

Most notably, `rvcodec.js` uses bright colors to visually map the relationships between the instruction's assembly tokens and binary fields. As demonstrated in Figure 1, it shows that the immediate value 7 in the assembly instruction corresponds to the first twelve bits (`000000000111`) of the binary representation using the color blue; or that the next five bits (`00101`) of the binary representation correspond to the source register `x5` (also known by its ABI name `t0`) of the assembly instruction using the color cyan; and so on. Students learning how to convert instructions between different formats can therefore quickly identify the various fields of a particular instruction. Our tool also comes with additional features, such as decoding registers if they are designated with their ABI names.

3 FUTURE WORK

We are currently working to improve `rvcodec.js`'s conversion engine by adding support for more RISC-V instruction sets (such as RV64I and the MAFDC extensions), and improve its user interface by displaying additional information in a tooltip when the cursor points on each token/field of a converted instruction.

REFERENCES

[1] Rashmi Agrawal, Sahan Bandara, Alan Ehret, Mihailo Isakov, Miguel Mark, and Michel A. Kinsy. 2019. The BRISC-V Platform: A Practical Teaching Approach for Computer Architecture. In *Proceedings of the Workshop on Computer Architecture Education.*

[2] Tiago Bozzetti, Ellie Easse, and Chau Tieu. 2014. MIPS Converter. https://www.eg.bucknell.edu/~csci320/mips_web/

[3] Roberto Giorgi and Gianfranco Mariotti. 2019. WebRISC-V: A Web-Based Education-Oriented RISC-V Pipeline Simulation Environment. In *Proceedings of the Workshop on Computer Architecture Education.*

[4] Jason Lowe-Power and Christopher Nitta. 2019. The Davis In-Order (DINO) CPU: A Teaching-Focused RISC-V CPU Design. In *Proceedings of the Workshop on Computer Architecture Education.*

[5] Andrew Waterman and Krste Asanovic (Eds.). 2019. *The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Document Version 20191213.* RISC-V Foundation.

SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada

ACM ISBN 978-1-4503-9433-8/23/03.
https://doi.org/10.1145/3545947.3576340