

# ESTUDO SOBRE A ARQUITETURA RISC-V DE MICROPROCESSADORES

Gustavo Segismundo Giraldo de Melo <sup>1</sup>, Salvador Pinillos Gimenez <sup>2</sup>

<sup>1,2</sup> Departamento de Engenharia Elétrica, Centro Universitário FEI

e-mail: [gustavoseg@outlook.com](mailto:gustavoseg@outlook.com) <sup>1</sup>, [sgimenez@fei.edu.br](mailto:sgimenez@fei.edu.br) <sup>2</sup>;

**Resumo:** Este projeto de pesquisa de iniciação científica está na metade de seu desenvolvimento e tem por objetivo estudar a arquitetura intitulada Computador com Conjunto de Instruções Reduzido (*Reduced Instruction Set Architecture*, RISC) V (RISC-V) de microprocessadores, que é moderna, otimizada e revolucionária. Além disso, o seu funcionamento será observado por simulação e por sua implementação física num *kit* FPGA.

## 1. Introdução

A arquitetura RISC, atualmente em sua versão V (RISC-V), é capaz de suprir a recente demanda da indústria por microprocessadores cada vez menores, mais rápidos e versáteis. Tal capacidade, implementada em conjunto com a arquitetura baseada em processamento paralelo intitulada *Microkernel* de *Hardware Intertravado* (*Interlocked Hardware Microkernel*, IHM), desenvolvida para Sistemas Operacionais de Tempo Real (*Real Time Operational Systems*, RTOS), melhora ainda mais a eficiência de processamento, principalmente dos Circuitos Integrados de Aplicações Específicas (*Application-Specific Integrated Circuit*, ASIC) [1] [2] [3] [4] [5]. Dentro desse contexto, os objetivos deste projeto de pesquisa são, em primeiro lugar, o estudo da arquitetura RISC-V, o seu núcleo e código VHDL, e a análise de seu comportamento elétrico por meio de simulações lógicas e funcionais utilizando o *Quartus Prime* da Intel. Adicionalmente, será implementado um microcontrolador RISC-V num *kit* de FPGA de desenvolvimento de projetos de sistemas embarcados. O funcionamento deste microcontrolador será testado de forma experimental, por meio da criação de programas escritos em linguagem *Assembly* ou C.

## 2. A arquitetura RISC e sua versão RISC-V

A arquitetura RISC-V busca atender a todos os tamanhos de processadores, funcionar bem com diversos *softwares* e linguagens de programação, acomodar todas as tecnologias de implementação, ser eficiente para todos os tipos de microarquitetura e manter-se inalterada ao longo do tempo. Tal arquitetura também é modular, ou seja, é composta primeiramente por um núcleo, chamado RV32I, e por extensões, dentre elas a RV32M para multiplicação e divisão de inteiros, RV32F e RV32D para instruções de ponto flutuante de extensão simples e dupla, RV32A para instruções atômicas, RV32C para instruções compactadas e a RV32V para vetores. Essas extensões podem ser incluídas ou excluídas de acordo com a necessidade de aplicação do microcontrolador contendo a arquitetura, possibilitando maiores velocidades de processamento ao custo de instruções que não serão utilizadas no projeto [6] [7].

A maior vantagem da arquitetura RISC-V é ser disponibilizada de forma gratuita (*open source*), permitindo a sua utilização por qualquer usuário sem pagar *royalties* aos seus criadores [6] [7].

## 3. O núcleo RISC-V RV32I

O núcleo RV32I, de 32 *bits*, é onde a arquitetura RISC-V se destaca por apresentar mais recursos de programação. Contando com 6 tipos de formatos de instruções otimizados, ele é capaz de oferecer uma maior simplicidade para elaboração de programas, de facilitar a inserção de novas instruções pelos desenvolvedores, de otimizar o tamanho de programa e de melhorar o desempenho de sua capacidade de processamento. Seus formatos de instruções são o tipo-R para os seus 32 registradores, tipo-I para valores imediatos e *short loads*, tipo-S para *stores*, tipo-B para desvios condicionais, tipo-U para valores imediatos longos e tipo-J para saltos incondicionais [6] [7].

Dentre os pontos de maior inovação do RV32I, destacam-se, também, a similaridade intencional entre os seus formatos de instruções, a fim de simplificar a lógica e o controle do desenvolvedor ao elaborar códigos. Além disso, o seu registrador “x0”, que contém exclusivamente uma cadeia de zeros, evita, na programação, a instrução que atribui zero a um registrador, a extensão de todos os *bytes* e *halfwords* para 32 *bits*, tornando possíveis as operações com dados de diversos tamanhos, e a sinalização de todos os valores imediatos dentro de seus códigos, com o *bit* mais significativo representando o sinal e sendo tratado de maneira independente, evitando erros nos códigos em operações lógicas e possibilitando que a operação de extensão de sinal seja feita antecipadamente, otimizando o desempenho do processador [6] [7].

## 4. O Kit de desenvolvimento DE1-SoC

O *kit* de desenvolvimento com arquitetura *Cyclone V* DE1-SoC da *Terasic*, fornecido pela Macnica DHW Brasil, contém um processador *dual-core* Cortex-A9 com lógica programável, possibilitando grande flexibilidade para o projeto dos CIs Metal-Óxido Semicondutor Complementar (*Complementary Metal-Oxide-Semiconductor*, CMOS) utilizando-se Transistores de Efeito de Campo do tipo Metal-Óxido Semicondutor (*Metal-Oxide-Semiconductor*, MOSFET) [8].

Contando com uma série de periféricos programáveis, sendo alguns deles dedicados exclusivamente para o microprocessador, o *kit* será configurado no *software Quartus Prime*, onde será implementada a arquitetura RISC-V com base na literatura gratuita disponível no repositório *online*. Na sequência, serão diversas simulações abordando conceitos gerais e direcionados

aos pontos nos quais a arquitetura RISC-V se destaca, comparando os resultados com outras arquiteturas. Por fim, a implementação física dessas simulações será realizada nas FPGAs da placa DE1-SoC.

### 5. Mapa e Formato dos “Opcodes” das Instruções do RV32I

A Tabela 1 ilustra o mapa dos opcodes das instruções do RV32I e seus respectivos formatos binários.

31	25	24	20	19	15	14	12	11	7	6	0	
imm[31:12]						rd		0110111		U lui		
imm[31:12]						rd		0010111		U auip		
imm[20:10:1119:12]						rd		1101111		J jal		
imm[11:0]		rs1	000		rd		1100111		I jalr			
imm[12:10:5]		rs2	rs1	000	imm[4:1:11]		1100011		B beq			
imm[12:10:5]		rs2	rs1	001	imm[4:1:11]		1100011		B bne			
imm[12:10:5]		rs2	rs1	100	imm[4:1:11]		1100011		B bit			
imm[12:10:5]		rs2	rs1	101	imm[4:1:11]		1100011		B bge			
imm[12:10:5]		rs2	rs1	110	imm[4:1:11]		1100011		B bltu			
imm[12:10:5]		rs2	rs1	111	imm[4:1:11]		1100011		B bgeu			
imm[11:0]		rs1	000		rd		0000011		I lb			
imm[11:0]		rs1	001		rd		0000011		I lh			
imm[11:0]		rs1	010		rd		0000011		I lw			
imm[11:0]		rs1	100		rd		0000011		I lbu			
imm[11:0]		rs1	101		rd		0000011		I lbu			
imm[11:5]		rs2	rs1	000	imm[4:0]		0100011		S sb			
imm[11:5]		rs2	rs1	001	imm[4:0]		0100011		S sh			
imm[11:5]		rs2	rs1	010	imm[4:0]		0100011		S sw			
imm[11:0]		rs1	000		rd		0010011		I addi			
imm[11:0]		rs1	010		rd		0010011		I sli			
imm[11:0]		rs1	011		rd		0010011		I sliui			
imm[11:0]		rs1	100		rd		0010011		I xori			
imm[11:0]		rs1	110		rd		0010011		I ori			
imm[11:0]		rs1	111		rd		0010011		I andi			
0000000		shamt	rs1	001	rd		0010011		I slli			
0000000		shamt	rs1	101	rd		0010011		I srl			
0100000		shamt	rs1	101	rd		0010011		I srai			
0000000		rs2	rs1	000	rd		0110011		R add			
0100000		rs2	rs1	000	rd		0110011		R sub			
0000000		rs2	rs1	001	rd		0110011		R sll			
0000000		rs2	rs1	010	rd		0110011		R slt			
0000000		rs2	rs1	011	rd		0110011		R sltu			
0000000		rs2	rs1	100	rd		0110011		R xor			
0000000		rs2	rs1	101	rd		0110011		R srl			
0100000		rs2	rs1	101	rd		0110011		R sra			
0000000		rs2	rs1	110	rd		0110011		R or			
0000000		rs2	rs1	111	rd		0110011		R and			
0000	pred	succ	00000	000	00000	0001111		I fence				
0000	0000	0000	00000	001	00000	0001111		I fence.i				
000000000000			00000	000	00000	1110011		I ecall				
000000000001			00000	000	00000	1110011		I ebreak				
csr		rs1	001		rd		1110011		I csrwr			
csr		rs1	010		rd		1110011		I csrrs			
csr		rs1	011		rd		1110011		I csrrc			
csr		zimm	101		rd		1110011		I csrrwi			
csr		zimm	110		rd		1110011		I csrrsi			
csr		zimm	111		rd		1110011		I csrrci			

Tabela 1 – Mapa de *opcode* do RV32I, indicando o formato de cada uma das suas instruções.

### 6. Conclusões

Dentre os maiores resultados deste projeto de pesquisa de iniciação científica até o momento, destacam-se o aprofundamento do estudo dos conceitos previamente apresentados nas aulas teóricas e práticas da disciplina de Microcontroladores e o estudo de uma arquitetura de microprocessadores moderna, revolucionária e valorizada pela indústria.

As etapas futuras desse projeto de pesquisa serão, primeiramente, a simulação funcional dos blocos construtivos desse microprocessador por meio do *software* de simulações *Quartus Prime* da Intel, e, em segundo lugar, a sua implementação física, que será realizada através do *kit* didático com FPGAs.

### 7. Referências

- [1] EMBEDDED. Trends and highlights from Embedded World 2017. 2017. Disponível em <<https://www.embedded.com/electronics-blogs/say-what-/4458143/Trends-and-highlights-from-Embedded-World-2017>>. Acesso em: 15 mar 2023.
- [2] ARM LTD [GB]. Solutions. 2019. Disponível em <<https://www.arm.com/company>>. Acesso em: 15 mar. 2023.
- [3] EMBEDDED. Upgrading 8- and 16-bit MCU designs: 32-bit MCU architectures. 2017. Disponível em <<https://www.embedded.com/design/mcus-processors-and-socs/4443282/Upgrading-8--and-16-bit-MCU-designs--32-bit-MCU-architectures>>. Acesso em: 15 mar 2023.
- [4] EDN NETWORK. Timing (to market) is everything. 2012. Disponível em: <<https://www.edn.com/electronics-blogs/other/4394155/Timing--to-market--is-everything>>. Acesso em: 15 mar. 2023.
- [5] DANTAS, Leandro Poloni, AZEVEDO, Rodolfo J. de, GIMENEZ, Salvador Pinillos. A Novel Processor Architecture With a Hardware Microkernel to Improve the Performance of Task-Based Systems. IEEE Embedded Systems Letters. 2 jun. 2019, vol. 11, n° 2.
- [6] PATTERSON, David A., HENNESSY, John L. Computer Organization and Design: The Hardware/Software Interface, RISC-V Edition, Morgan Kaufmann Publisher, Elsevier, 2017.
- [7] PATTERSON, David A., WATERMAN, Andrew. Guia prático RISC-V: Atlas de uma Arquitetura Aberta, 1ª Edição, Strawberry Canyon: LLC, 2019.
- [8] TERIC INC. DE1-SoC Board. Disponível em: <<https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=836>>. Acesso em: 04 ago. 2023.

### Agradecimentos

Ao Centro Universitário FEI, por fornecer estrutura física para os estudos e a Macnica DHW Brasil pelo fornecimento do *kit* de FPGA DE1-SoC.

<sup>1</sup> Aluno de Iniciação Científica do Centro Universitário FEI. Projeto com vigência de 03/2023 a 02/2024.