

TRABALHO DE GRADUAÇÃO

RISC-V SiMPLE

Arthur de Matos Beggs

Brasília, Dezembro de 2019



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

RISC-V SIMPLE

Arthur de Matos Beggs

*Relatório submetido como requisito parcial de obtenção
de grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Marcus Vinicius Lamar, CIC/UnB
Orientador

Profa. Carla M. C. C. Koike, CIC/UnB
Examinador Interno

Prof. Marcelo Grandi Mandelli, CIC/UnB
Examinador Interno

Brasília, Dezembro de 2019

FICHA CATALOGRÁFICA

ARTHUR, DE MATOS BEGGS

RISC-V SiMPLE,

[Distrito Federal] 2019.

???, ???p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2019). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.

1. RISC-V

2. Verilog

3. FPGA

I. Mecatrônica/FT/UnB

II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

BEGGS, ARTHUR DE MATOS, (2019). RISC-V SiMPLE. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-*n*º???, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, ???p.

CESSÃO DE DIREITOS

AUTOR: Arthur de Matos Beggs

TÍTULO DO TRABALHO DE GRADUAÇÃO: RISC-V SiMPLE.

GRAU: Engenheiro

ANO: 2019

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Arthur de Matos Beggs

SHCGN 703 Bl G N° 120, Asa Norte

70730-707 Brasília – DF – Brasil.

Dedicatória

Arthur de Matos Beggs

Agradecimentos

Arthur de Matos Beggs

RESUMO

Desenvolvimento e documentação de uma plataforma de ensino de arquitetura de computadores em *Verilog* sintetizável em *FPGA*, com foco em um processador com arquitetura do conjunto de instruções *RISC-V* implementado em três microarquitecturas para ser utilizado como recurso de laboratório na disciplina de Organização e Arquitetura de Computadores da Universidade de Brasília. A plataforma funciona em dois modelos de *FPGA* disponíveis no laboratório da Universidade, possui periféricos de depuração como *display* dos registradores do processador na saída de vídeo, além de outros periféricos como *drivers* de áudio e vídeo para uma experiência mais completa de desenvolvimento, e permite que o processador seja substituído por implementações de diversas arquiteturas de *32 bits* com certa facilidade.

Palavras Chave: RISC-V, Verilog, FPGA

ABSTRACT

Keywords: RISC-V, Verilog, FPGA

SUMÁRIO

1	Introdução	1
1.1	MOTIVAÇÃO	1
1.2	POR QUE RISC-V?	1
1.3	O PROJETO RISC-V SIMPLE	2
2	Revisão Teórica	3
2.1	ARQUITETURA DE COMPUTADORES	4
2.1.1	<i>MIPS</i>	4
2.1.2	<i>RISC-V</i>	4
2.2	MICROARQUITETURAS	4
2.2.1	UNICICLO	4
2.2.2	MULTICICLO	4
2.2.3	<i>Pipeline</i>	4
2.3	REPRESENTAÇÃO DE <i>Hardware</i>	4
2.3.1	<i>Verilog HDL</i>	4
2.4	SÍNTESE LÓGICA	4
2.4.1	ANÁLISE E SÍNTESE	4
2.4.2	<i>Fitting</i>	4
2.4.3	<i>Timing Analyzer</i>	4
2.5	SIMULAÇÃO	4
2.6	UTILIZANDO <i>Hardware</i> REAL	4
2.6.1	<i>Field Programmable Gate Arrays</i>	4
3	Sistema Proposto	5
4	Resultados	6
5	Conclusões	7
5.1	PERSPECTIVAS FUTURAS	7
	Anexos	8
I	Descrição do conteúdo do CD	9

II Programas utilizados.....	10
------------------------------	----

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SÍMBOLOS

Siglas

ASIC	Circuito Integrado de Aplicação Específica — <i>Application Specific Integrated Circuit</i>
BSD	Distribuição de Software de Berkeley — <i>Berkeley Software Distribution</i>
CSR	Registradores de Controle e Estado — <i>Control and Status Registers</i>
DSP	Processamento Digital de Sinais — <i>Digital Signal Processing</i>
FPGA	Arranjo de Portas Programáveis em Campo — <i>Field Programmable Gate Array</i>
hart	<i>hardware thread</i>
ISA	Arquitetura do Conjunto de Instruções — <i>Instruction Set Architecture</i>
MIPS	Microprocessador sem Estágios Intertravados de Pipeline — <i>Microprocessor without Interlocked Pipeline Stages</i>
OAC	Organização e Arquitetura de Computadores
PLL	Malha de Captura de Fase — <i>Phase-Locked Loop</i>
RISC	Computador com Conjunto de Instruções Reduzido — <i>Reduced Instruction Set Computer</i>
SDK	Conjunto de Programas de Desenvolvimento — <i>Software Development Kit</i>
SoC	Sistema em um Chip — <i>System on Chip</i>
SiMPLE	Ambiente de Aprendizado Uniciclo, Multiciclo e Pipeline — <i>Single-cycle Multicycle Pipeline Learning Environment</i>
RAS	Pilha de Endereços de Retorno — <i>Return Address Stack</i>

Capítulo 1

Introdução

1.1 Motivação

O mercado de trabalho está a cada dia mais exigente, sempre buscando profissionais que conheçam as melhores e mais recentes ferramentas disponíveis. Além disso, muitos universitários se sentem desestimulados ao estudarem assuntos desatualizados e com baixa possibilidade de aproveitamento do conteúdo no mercado de trabalho. Isso alimenta o desinteresse pelos temas abordados e, em muitos casos, leva à evasão escolar. Assim, é importante renovar as matérias com novas tecnologias e tendências de mercado sempre que possível, a fim de instigar o interesse dos discentes e formar profissionais mais capacitados e preparados para as demandas da atualidade.

Hoje, a disciplina de Organização e Arquitetura de Computadores da Universidade de Brasília é ministrada utilizando a arquitetura *MIPS32*. Apesar da arquitetura *MIPS32* ainda ter grande força no meio acadêmico (em boa parte devido a sua simplicidade e extensa bibliografia), sua aplicação na indústria tem diminuído consideravelmente na última década.

Embora a curva de aprendizagem de linguagens *Assembly* de alguns processadores *RISC* seja relativamente baixa para quem já conhece o *Assembly MIPS32*, aprender uma arquitetura atual traz o benefício de conhecer o *estado da arte* da organização e arquitetura de computadores.

Para a proposta de modernização da disciplina, foi escolhida a *ISA RISC-V* desenvolvida na Divisão de Ciência da Computação da Universidade da Califórnia, Berkeley como substituta à *ISA MIPS32*.

1.2 Por que RISC-V?

A *ISA RISC-V* (lê-se “*risk-five*”) é uma arquitetura *open source* com licença *BSD*, o que permite o seu livre uso para quaisquer fins, sem distinção de se o trabalho possui código-fonte aberto ou proprietário. Tal característica possibilita que grandes fabricantes utilizem a arquitetura para criar seus produtos, mantendo a proteção de propriedade intelectual sobre seus métodos de implementação e quaisquer subconjuntos de instruções não-*standard* que as empresas venham a

produzir, o que estimula investimentos em pesquisa e desenvolvimento.

Empresas como Google, IBM, AMD, Nvidia, Hewlett Packard, Microsoft, Qualcomm e Western Digital são algumas das fundadoras e investidoras da *RISC-V Foundation*, órgão responsável pela governança da arquitetura. Isso demonstra o interesse das gigantes do mercado no sucesso e disseminação da arquitetura.

A licença também permite que qualquer indivíduo produza, distribua e até mesmo comercialize sua própria implementação da arquitetura sem ter que arcar com *royalties*, sendo ideal para pesquisas acadêmicas, *startups* e até mesmo *hobbyistas*.

O conjunto de instruções foi desenvolvido tendo em mente seu uso em diversas escalas: sistemas embarcados, *smartphones*, computadores pessoais, servidores e supercomputadores, o que permitirá maior reuso de *software* e maior integração de *hardware*.

Outro fator que estimula o uso do *RISC-V* é a modernização dos livros didáticos. A nova versão do livro utilizado em OAC, Organização e Projeto de Computadores, de David Patterson e John Hennessy, utiliza a *ISA RISC-V*.

Além disso, com a promessa de se tornar uma das arquiteturas mais utilizadas nos próximos anos, utilizar o *RISC-V* como arquitetura da disciplina de OAC se mostra a escolha ideal no momento.

1.3 O Projeto RISC-V SiMPLE

O projeto *RISC-V SiMPLE* (*Single-cycle Multicycle Pipeline Learning Environment*) consiste no desenvolvimento de um processador com conjunto de instruções *RISC-V*, sintetizável em *FPGA* e com *hardware* descrito em *Verilog*. A microarquitetura implementada nesse trabalho é uniciclo, escalar, em ordem, com um único *hart* e com caminho de dados de 64 bits. Trabalhos futuros poderão utilizar a estrutura altamente configurável e modularizada do projeto para desenvolver as versões em microarquiteturas multiciclo e *pipeline*.

O processador contém o conjunto de instruções I (para operações com inteiros, sendo o único módulo com implementação mandatória pela arquitetura) e as extensões *standard* M (para multiplicação e divisão de inteiros) e F (para ponto flutuante com precisão simples conforme o padrão IEEE 754 com revisão de 2008). O projeto não implementa as extensões D (ponto flutuante de precisão dupla) e A (operações atômicas de sincronização), e com isso o *soft core* desenvolvido não pode ser definido como de propósito geral, G (que deve conter os módulos I, M, A, F e D). Assim, pela nomenclatura da arquitetura, o processador desenvolvido é um *RV64IMF*.

O projeto contempla *traps*, interrupções, exceções, *CSRs*, chamadas de sistema e outras funcionalidades de nível privilegiado da arquitetura.

O *soft core* possui barramento Avalon para se comunicar com os periféricos das plataformas de desenvolvimento. O projeto foi desenvolvido utilizando a placa DE2-115 com *FPGA Altera Cyclone* e permite a fácil adaptação para outras placas da Altera.

Capítulo 2

Revisão Teórica

2.1 Arquitetura de Computadores

2.1.1 *MIPS*

2.1.2 *RISC-V*

2.2 Microarquiteturas

2.2.1 Uniciclo

2.2.2 Multiciclo

2.2.3 *Pipeline*

2.3 Representação de *Hardware*

2.3.1 *Verilog HDL*

2.4 Síntese Lógica

2.4.1 Análise e Síntese

2.4.2 *Fitting*

2.4.3 *Timing Analyzer*

2.5 Simulação

2.6 Utilizando *Hardware* Real

2.6.1 *Field Programmable Gate Arrays*

Capítulo 3

Sistema Proposto

Capítulo 4

Resultados

Capítulo 5

Conclusões

5.1 Perspectivas Futuras

ANEXOS

I. DESCRIÇÃO DO CONTEÚDO DO CD

Descrever CD.

II. PROGRAMAS UTILIZADOS

Quais programas foram utilizados?