

Revisão de implementações abertas da arquitetura RISC-V

João P. M. Roberto¹

¹Facom – Universidade Federal do Mato Grosso do Sul (UFMS)

joao.melo@ufms.br

1. Introdução ao RISC-V e às Arquiteturas Abertas

A arquitetura RISC-V surgiu em 2010 na Universidade da Califórnia, Berkeley, como uma proposta revolucionária no mundo da computação, fundamentada em um conceito essencial: ser uma arquitetura de conjunto de instruções (ISA) completamente aberta e livre de royalties. Diferente das arquiteturas proprietárias tradicionais, como ARM e x86, o RISC-V foi projetado com simplicidade, escalabilidade e modularidade em mente, permitindo que desenvolvedores e empresas adaptassem a arquitetura às suas necessidades específicas, desde sistemas embarcados até supercomputadores.



Figure 1. RISC-V

O diferencial do RISC-V reside não apenas na liberdade para sua implementação, mas também na comunidade ativa que se desenvolveu em torno da arquitetura. Essa abordagem colaborativa tem fomentado a inovação, permitindo que pesquisadores, engenheiros e educadores explorem e criem novas soluções tecnológicas. Entre suas principais qualidades, destacam-se a simplicidade no design, que reduz custos de implementação; a escalabilidade, que atende a diferentes níveis de aplicação; e a personalização, que permite adicionar extensões específicas sem comprometer a interoperabilidade.

Na era da tecnologia aberta e acessível, compreender as implementações do RISC-V é crucial para aqueles que desejam navegar pelo futuro da computação. No contexto acadêmico e educacional, estudar implementações abertas baseadas nessa arquitetura promove uma aprendizagem prática, desenvolvendo habilidades essenciais para a criação e avaliação de soluções tecnológicas.

Este estudo se dedica a explorar algumas das principais implementações abertas baseadas na arquitetura RISC-V, com ênfase em cinco projetos: PicoRV32, NEORV32, PULPino e, em um capítulo conjunto, Rocket Chip e BOOM. Cada uma dessas implementações reflete diferentes facetas da flexibilidade do RISC-V, cobrindo desde projetos minimalistas até arquiteturas avançadas de alto desempenho.

Ao longo desta revisão, serão analisadas as características de cada implementação, suas principais funcionalidades e contribuições para o cenário computacional. Este estudo não apenas busca consolidar o conhecimento teórico sobre a arquitetura RISC-V, mas também enfatiza sua aplicabilidade e impacto, destacando a importância de compreender essas tecnologias em um ambiente de aprendizado dinâmico e integrado.

2. Implementação Minimalista – PicoRV32

2.1. Introdução ao PicoRV32

O PicoRV32 é uma implementação minimalista de um núcleo RISC-V de 32 bits, projetada e desenvolvida por Clifford Wolf. Caracteriza-se por sua simplicidade e eficiência em termos de utilização de área, sendo especialmente útil para sistemas embarcados e projetos em FPGA que demandam baixo consumo de recursos. Essa implementação se destaca por suportar o conjunto de instruções RV32IMC (instruções inteiras, multiplicação/divisão e compactação), tornando-a uma solução prática para aplicações de controle, processamento básico e integração em sistemas maiores.

Por ser um núcleo de código aberto, o PicoRV32 pode ser facilmente integrado em projetos personalizados, promovendo a flexibilidade no desenvolvimento de hardware. Seu design é altamente otimizado para minimizar a quantidade de recursos utilizados, o que o torna uma excelente opção para dispositivos que exigem um custo computacional reduzido, como sensores inteligentes e microcontroladores embarcados.

2.2. Visão Geral do PicoRV32

O núcleo PicoRV32 é uma implementação in-order que executa instruções de forma sequencial, simplificando o design e reduzindo a complexidade. Algumas das principais características incluem:

- **Compactação:** O suporte às instruções compactadas (C-extension) reduz o tamanho do código, economizando memória.
- **Portabilidade:** Projetado em Verilog, o PicoRV32 pode ser facilmente implementado em diversas plataformas FPGA ou integrado em projetos ASIC.
- **Baixo uso de recursos:** Em síntese, o PicoRV32 ocupa uma área mínima no FPGA, sendo uma das menores implementações de núcleo RISC-V disponíveis.
- **Confiabilidade:** Apesar de seu design minimalista, o PicoRV32 é robusto e atende aos padrões da especificação RISC-V, garantindo conformidade com a arquitetura.

2.3. Aplicações Típicas

O PicoRV32 é ideal para aplicações que requerem núcleos simples, como:

1. **Sistemas embarcados:** Pode atuar como um controlador básico para dispositivos IoT, sensores ou interfaces periféricas.
2. **Dispositivos de baixo custo:** Seu design compacto é adequado para dispositivos que têm restrições de área, consumo de energia ou custo.
3. **FPGAs pequenos:** É frequentemente utilizado como um núcleo em projetos de FPGA que exigem o uso eficiente de recursos.

Um exemplo comum é a utilização do PicoRV32 como controlador de lógica em sensores industriais, onde ele gerencia a comunicação e o processamento básico, mantendo o consumo de energia em níveis mínimos.

2.4. Análise da Arquitetura e Desempenho

O design do PicoRV32 adota uma abordagem modular e eficiente, especialmente projetada para atender às necessidades de sistemas minimalistas. Seu pipeline é um dos elementos que refletem essa simplicidade: diferentemente de núcleos mais complexos,

que empregam pipelines com várias etapas para aumentar o desempenho, o PicoRV32 opera com um pipeline básico que executa instruções em ciclos consecutivos. Essa característica elimina a necessidade de mecanismos avançados de controle de fluxo, como reordenação de instruções ou previsão de saltos, simplificando o hardware e contribuindo para uma redução significativa no uso de recursos.

Outra característica fundamental do PicoRV32 é sua interação com a memória. O núcleo foi projetado para operar com memórias externas, o que reduz a complexidade de seu design interno e facilita a integração em sistemas com diferentes configurações de memória. Isso também permite maior flexibilidade em sua utilização, uma vez que ele pode ser combinado com diferentes tipos de memórias dependendo das restrições do projeto, como área, consumo energético e custo.

Além disso, o suporte às instruções compactadas da arquitetura RISC-V, conhecidas como extensão "C", é um diferencial que aumenta a eficiência do PicoRV32. Essa extensão permite que o núcleo execute instruções menores, otimizando o uso da memória de programa e reduzindo o espaço necessário para armazenar o código. Essa economia de espaço é especialmente valiosa em aplicações embarcadas, onde os recursos de memória são frequentemente limitados.

A eficiência energética também é uma marca registrada do PicoRV32. Graças ao seu design simplificado, o consumo de energia é significativamente menor em comparação com núcleos mais complexos. Essa característica o torna particularmente adequado para dispositivos de baixo consumo, como sensores IoT e sistemas embarcados em ambientes com restrições energéticas.

Embora o PicoRV32 seja limitado em termos de capacidade de processamento devido à sua abordagem minimalista, ele compensa essas limitações com uma excelente eficiência de área e baixo custo de implementação. Esse equilíbrio entre simplicidade e funcionalidade faz do PicoRV32 uma solução ideal para aplicações que exigem núcleos compactos, acessíveis e de fácil integração em projetos mais amplos.

2.5. Conclusão

O PicoRV32 é uma implementação que exemplifica a simplicidade e a eficiência que a arquitetura RISC-V pode oferecer. Seu design minimalista o torna uma excelente escolha para iniciantes em desenvolvimento de hardware, bem como para aplicações práticas em dispositivos com restrições de área e energia. Com sua natureza aberta e fácil acessibilidade, o PicoRV32 promove a aprendizagem prática e incentiva a inovação no campo das arquiteturas de hardware.

3. Implementação Versátil – NEORV32

3.1. Introdução ao NEORV32

O NEORV32 é um núcleo de processador baseado na arquitetura RISC-V projetado para ser altamente configurável e versátil, ideal para aplicações em sistemas embarcados. Desenvolvido por Stephan Nolting, esse núcleo de código aberto é voltado para aqueles que desejam uma solução compacta, mas com capacidade de expansão por meio de periféricos e extensões personalizáveis. Ele suporta o conjunto de instruções RV32I (instruções inteiras) como padrão e oferece suporte opcional a extensões como multiplicação/divisão

(M), operações atômicas (A), ponto flutuante (F) e compactação (C). Essa flexibilidade o torna aplicável tanto em dispositivos simples quanto em sistemas mais complexos que demandam maior desempenho.

Além de sua adaptabilidade, o NEORV32 é conhecido por incluir uma vasta gama de periféricos integrados e interfaces de comunicação que eliminam a necessidade de hardware adicional em muitos casos. A documentação completa e o suporte ativo pela comunidade tornam esse núcleo uma excelente opção para aprendizado, prototipagem e até mesmo produção comercial.

3.2. Visão Geral do NEORV32

O NEORV32 foi projetado para funcionar como um sistema em chip (SoC) ou como um controlador dedicado dentro de sistemas maiores. Seu foco na modularidade permite que desenvolvedores escolham exatamente quais recursos são necessários, otimizando o uso de área e consumo energético. A flexibilidade do núcleo é refletida em seu design configurável, que permite aos usuários habilitar ou desabilitar extensões e periféricos conforme as exigências do projeto. Seu código-fonte, disponibilizado em VHDL, facilita a

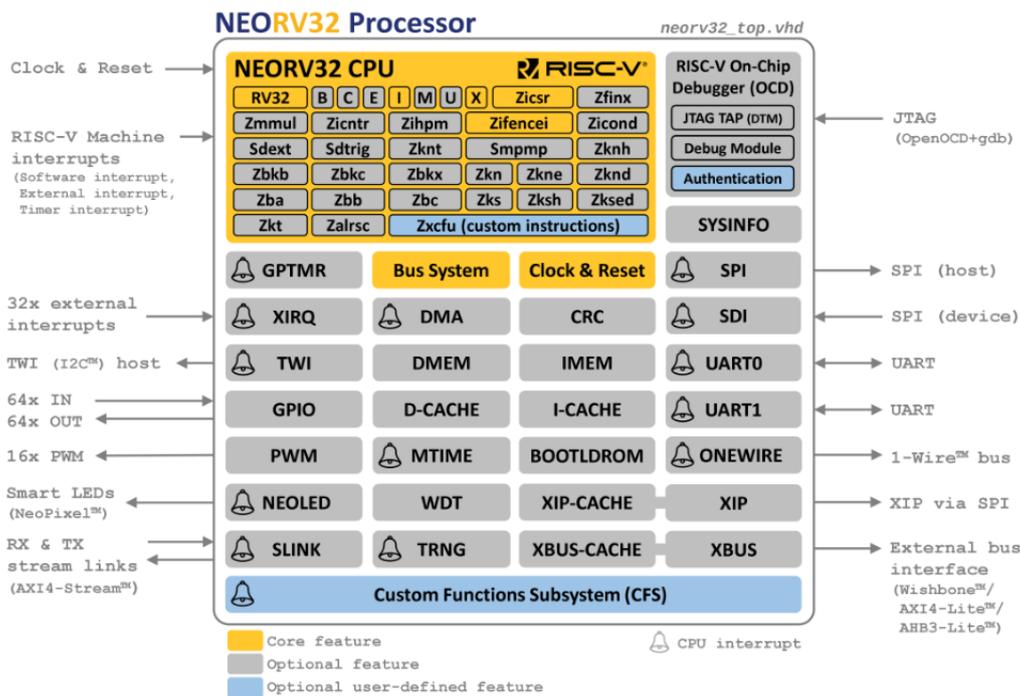


Figure 2. Processador NEORV32 (Diagrama de Bloco)

síntese em diversas plataformas FPGA e ASIC. Isso o torna ideal para ambientes educacionais e projetos que precisam de soluções rápidas e adaptáveis.

3.3. Funcionalidades Principais e Diferenciais

O NEORV32 se destaca por sua ampla gama de funcionalidades que o tornam uma opção versátil para diversas aplicações. Entre essas funcionalidades estão:

- Conjunto de Instruções Modular: O núcleo suporta o conjunto básico de instruções RV32I e pode ser expandido com extensões como M (multiplicação

e divisão), C (compactação) e F (ponto flutuante). Isso permite que ele seja ajustado para aplicações específicas, desde dispositivos de baixo custo até sistemas que exigem maior desempenho.

- **Periféricos Integrados:** O NEORV32 vem com uma série de periféricos integrados, incluindo UART, SPI, I²C, PWM e temporizadores, o que reduz a necessidade de componentes externos e facilita o desenvolvimento de sistemas completos em hardware limitado.
- **Alta Personalização:** A modularidade do núcleo permite personalizações detalhadas, como ajustar o tamanho da memória ou habilitar/desabilitar módulos periféricos. Isso torna o NEORV32 uma ferramenta poderosa para aprendizado, permitindo que estudantes explorem diferentes configurações e compreendam os trade-offs de design.
- **Compatibilidade com FPGA e ASIC:** O núcleo pode ser facilmente implementado em FPGAs para prototipagem ou em ASICs para produção, ampliando suas possibilidades de aplicação.
- **Baixo Consumo de Recursos:** Mesmo com suas opções de personalização, o NEORV32 é eficiente em termos de área e consumo energético, sendo uma solução viável para sistemas embarcados com restrições de hardware

3.4. Aplicações em Sistemas Embarcados e Prototipagem

A flexibilidade e os recursos integrados do NEORV32 o tornam particularmente adequado para sistemas embarcados e projetos de prototipagem. Ele pode ser utilizado em uma ampla gama de aplicações, incluindo:

- **Automação industrial:** Como controlador para máquinas e sistemas que requerem interfaces de comunicação e processamento básico.
- **Dispositivos IoT:** Em sensores e atuadores inteligentes, aproveitando sua capacidade de integrar periféricos e operar com eficiência energética.
- **Prototipagem rápida:** Ideal para estudantes, pesquisadores e engenheiros que precisam de uma solução pronta para validar conceitos ou desenvolver sistemas completos.

A capacidade de ajustar o núcleo às necessidades específicas de um projeto, combinada com a disponibilidade de periféricos integrados, reduz significativamente o tempo e o custo de desenvolvimento.

3.5. Análise da Arquitetura e Desempenho

O design arquitetural do NEORV32 reflete seu foco na versatilidade e eficiência. O núcleo adota uma abordagem modular, onde cada componente pode ser habilitado ou desabilitado dependendo das exigências do projeto. Isso resulta em um equilíbrio eficaz entre funcionalidade e consumo de recursos.

A arquitetura do NEORV32 inclui um pipeline básico que executa instruções de forma sequencial, garantindo simplicidade e confiabilidade. Embora não seja projetado para desempenho extremo, ele oferece extensões opcionais, como multiplicação/divisão e ponto flutuante, que aumentam sua capacidade de lidar com tarefas computacionalmente mais exigentes. Além disso, o suporte à extensão compactada (C) reduz o tamanho do código, economizando memória e melhorando a eficiência geral.

A integração de periféricos diretamente no núcleo é outro aspecto que contribui para seu desempenho. A presença de UART, SPI, I²C e outros módulos periféricos elimina a necessidade de controladores externos, reduzindo a latência e otimizando a comunicação entre componentes. Essa integração também facilita a construção de sistemas completos em plataformas FPGA, especialmente em ambientes com restrições de área e energia.

3.6. Conclusão

O NEORV32 se posiciona como uma das implementações mais versáteis da arquitetura RISC-V, combinando um design modular com uma ampla gama de funcionalidades. Sua capacidade de personalização e suporte a periféricos integrados o tornam uma ferramenta poderosa para sistemas embarcados e prototipagem. Com uma documentação abrangente e uma comunidade ativa, o NEORV32 oferece um ambiente acessível para estudantes, pesquisadores e desenvolvedores explorarem os potenciais da arquitetura RISC-V de forma prática e eficiente.

4. Computação de Baixo Consumo – PULPino

4.1. Introdução ao PULPino

O PULPino é um microcontrolador de código aberto baseado na arquitetura RISC-V, desenvolvido como parte do projeto PULP (Parallel Ultra Low Power), uma colaboração entre a ETH Zurich e a Universidade de Bolonha. O objetivo principal do PULPino é atender a demandas de computação de baixo consumo energético, sendo ideal para aplicações em dispositivos IoT (Internet das Coisas), sensores inteligentes e sistemas embarcados que necessitam de alta eficiência energética.

Projetado com foco na simplicidade e na eficiência, o PULPino utiliza o conjunto de instruções RV32IMC (instruções inteiras, multiplicação/divisão e compactação) com extensões personalizadas para processamento digital de sinais (DSP). Sua arquitetura oferece flexibilidade e desempenho, mantendo o consumo de energia e os custos de implementação reduzidos. Além disso, o PULPino é disponibilizado como um design em Verilog, facilitando sua integração em plataformas FPGA ou produção de ASICs.

4.2. Visão Geral do PULPino

O PULPino foi projetado para ser um sistema completo em chip (SoC) autônomo, capaz de operar sem a necessidade de hardware adicional. Ele inclui não apenas o núcleo de processamento, mas também uma série de periféricos integrados e módulos de memória, proporcionando uma solução pronta para uso em aplicações embarcadas. O núcleo do PULPino, chamado de RI5CY (atualmente renomeado como CV32E40P), é um processador RISC-V de 32 bits otimizado para computação de baixo consumo. Ele oferece suporte a instruções compactadas, o que reduz o tamanho do código e economiza memória, além de extensões específicas para DSP, que aumentam sua capacidade de lidar com tarefas de processamento intensivo, como filtragem e análise de sinais.

4.3. Detalhes do Conjunto de Instruções e Extensões

O núcleo RI5CY no PULPino suporta o conjunto de instruções RV32IMC padrão do RISC-V, mas também introduz extensões específicas para melhorar o desempenho em aplicações específicas:

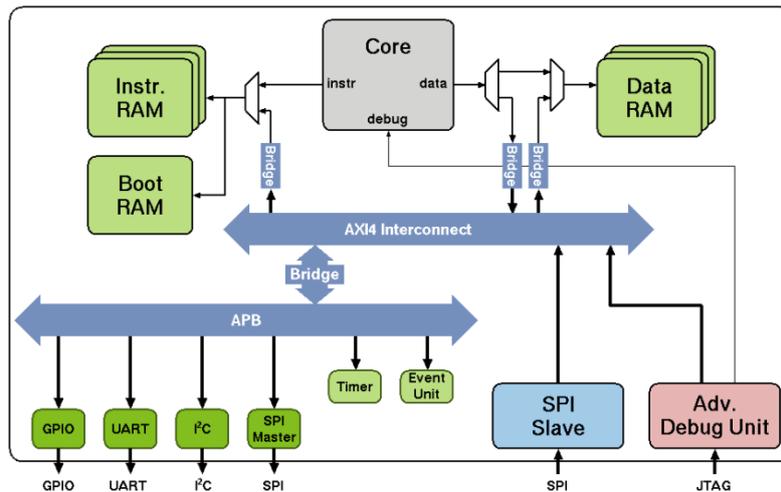


Figure 3. PULPino (Diagrama de Bloco)

1. **Instruções DSP:** O suporte a extensões para processamento digital de sinais permite que o PULPino execute operações como multiplicação acumulada (MAC) de forma mais eficiente, atendendo a requisitos comuns em aplicações IoT e de sensores.
2. **Compactação de Instruções:** A extensão compactada (C-extension) reduz o tamanho do código executado, o que economiza memória e melhora a eficiência, especialmente em dispositivos com recursos limitados.
3. **Personalização:** O design aberto do PULPino permite a inclusão de extensões personalizadas, adaptando o núcleo às necessidades de aplicações específicas sem comprometer o consumo de energia.

4.4. Aplicações Típicas

O PULPino é amplamente utilizado em aplicações que exigem eficiência energética e recursos computacionais moderados. Algumas das áreas em que ele é aplicado incluem:

- Dispositivos IoT: Como um microcontrolador eficiente, o PULPino é ideal para sensores inteligentes e atuadores conectados que precisam processar dados localmente antes de transmiti-los pela rede.
- Sensores Industriais: A capacidade de realizar processamento digital de sinais diretamente no dispositivo permite que o PULPino atue em sistemas industriais, como monitoramento de máquinas e análise de vibrações.
- Sistemas de Saúde: Dispositivos médicos portáteis, como monitores de frequência cardíaca ou oxímetros, podem usar o PULPino para processar sinais biométricos de forma eficiente.

4.5. Análise da Arquitetura e Desempenho

O PULPino adota uma arquitetura projetada para maximizar a eficiência energética sem comprometer o desempenho básico. O núcleo RI5CY opera com um pipeline otimizado que permite a execução de instruções em ciclos curtos, reduzindo o consumo de energia sem sacrificar a capacidade de processamento. Além disso, a integração de extensões DSP aumenta a versatilidade do PULPino, tornando-o capaz de lidar com tarefas computacionalmente mais intensivas, como algoritmos de filtragem e transformadas matemáticas.

Apesar de ser um núcleo pequeno, ele entrega um desempenho competitivo em relação a outros microcontroladores da mesma classe.

Outro aspecto importante da arquitetura é a presença de periféricos integrados, como controladores de UART, SPI e GPIO, além de um módulo de memória SRAM. Essa integração permite que o PULPino funcione de forma autônoma em diversas aplicações, eliminando a necessidade de hardware adicional e simplificando o design de sistemas embarcados. Em termos de consumo de energia, o PULPino é otimizado para operar em dispositivos com baterias de longa duração ou em ambientes com restrições de energia, como sensores alimentados por colheita de energia (energy harvesting). Essa eficiência é alcançada por meio de técnicas de otimização no design do núcleo e no gerenciamento de periféricos, garantindo que o consumo energético permaneça baixo mesmo em tarefas contínuas.

4.6. Conclusão

O PULPino é uma implementação exemplar da arquitetura RISC-V voltada para computação de baixo consumo. Com seu núcleo otimizado para eficiência energética e extensões personalizáveis para DSP, ele atende a uma ampla gama de aplicações embarcadas, especialmente em dispositivos IoT e sensores inteligentes. Sua flexibilidade, combinada com a disponibilidade de periféricos integrados, torna o PULPino uma solução robusta e acessível, promovendo a inovação em sistemas de computação de baixo custo e alto impacto.

5. Implementações Avançadas – Rocket Chip e BOOM

5.1. Introdução ao Rocket Chip e BOOM

O Rocket Chip e o Berkeley Out-of-Order Machine (BOOM) são dois projetos de arquitetura de microprocessadores baseados na especificação RISC-V, desenvolvidos pelo grupo de pesquisa ASPIRE na Universidade da Califórnia, Berkeley. Ambas as implementações são conhecidas por sua capacidade de configurabilidade e desempenho avançado, representando algumas das soluções mais sofisticadas dentro da comunidade RISC-V.

O Rocket Chip serve como um gerador de SoC (System-on-Chip) parametrizável que inclui, como padrão, o núcleo Rocket, um núcleo escalável in-order. Já o BOOM é um núcleo out-of-order de alto desempenho, também desenvolvido sobre a infraestrutura do Rocket Chip, ilustrando a flexibilidade deste gerador de SoC. Ambos compartilham a mesma infraestrutura de desenvolvimento e podem ser personalizados com uma variedade de extensões e configurações, adaptando-se às necessidades específicas de pesquisa ou desenvolvimento de produtos.

5.2. Rocket Chip: Design Parametrizável e Suporte a SoCs

O Rocket Chip é uma plataforma de design de SoC altamente configurável que permite aos desenvolvedores construir microprocessadores customizados. Ele foi projetado utilizando Chisel, uma linguagem de descrição de hardware baseada em Scala que facilita a escrita de geradores de hardware parametrizáveis e reutilizáveis. A principal vantagem do Rocket Chip é sua capacidade de gerar uma variedade de configurações de núcleos, permitindo desde simples designs in-order até complexos sistemas com múltiplos núcleos e cache coletivo. Além disso, o Rocket Chip incorpora um ambiente de tiles (ladrilhos),

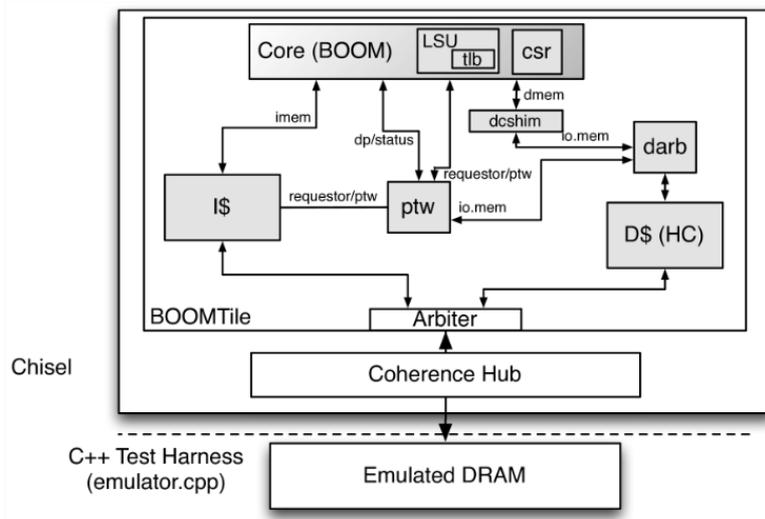


Figure 4. Rocket Chip(Diagrama de Bloco)

onde cada tile pode conter um ou mais núcleos e seus respectivos caches L1. Esse design modular facilita a adição de novos componentes, como aceleradores customizados ou unidades de gerenciamento de memória, tornando-o uma escolha robusta para aplicações que demandam processamento específico e otimizado.

5.3. BOOM: Execução Fora de Ordem e Desempenho em Aplicações Computacionais Complexas

O Berkeley Out-of-Order Machine (BOOM) é um exemplo de como a infraestrutura do Rocket Chip pode ser utilizada para implementar designs de microprocessador mais avançados. Diferente do núcleo Rocket padrão, o BOOM opera com uma arquitetura out-of-order, que permite a execução de instruções fora da sequência programada, aumentando o throughput e a utilização eficiente dos recursos do processador. O BOOM utiliza técnicas avançadas como especulação de branch, renomeação de registros e execução fora de ordem para acelerar a execução de cargas de trabalho complexas, como aquelas encontradas em servidores e aplicações de alta performance. Esse núcleo é ideal para pesquisa e desenvolvimento em arquiteturas de computadores, pois oferece insights sobre otimização de pipeline, gestão de cache e previsão de desvios, todos críticos para melhorar o desempenho de sistemas modernos.

5.4. Aplicações do Rocket Chip e BOOM

As capacidades avançadas e a flexibilidade do Rocket Chip e do BOOM permitem que ambos sejam aplicados em uma ampla gama de cenários, desde sistemas embarcados até servidores de alto desempenho. Cada um é adaptado para atender a demandas específicas de desempenho e eficiência energética, tornando-os opções robustas para diferentes indústrias e aplicações acadêmicas.

- **Aplicações do Rocket Chip:**

O Rocket Chip, com sua arquitetura in-order e foco em eficiência, é amplamente utilizado em sistemas embarcados e dispositivos IoT. Ele é ideal para aplicações que demandam processamento confiável, consumo de energia reduzido e escalabilidade. Exemplos incluem:

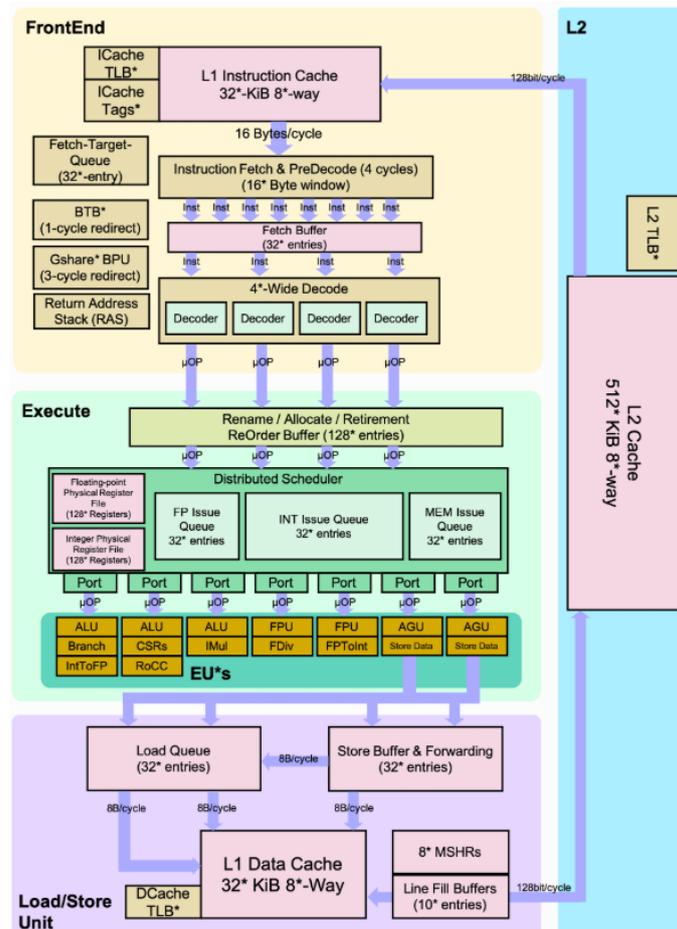


Figure 5. BOOM(Diagrama de Bloco)

1. Microcontroladores avançados: Implementações eficientes para dispositivos com baixa necessidade de processamento paralelo.
 2. Sistemas embarcados de médio porte: Controladores em dispositivos médicos, automação residencial e sensores inteligentes.
 3. Sistemas educacionais e de pesquisa: Utilizado em simulações e aprendizado sobre design de processadores devido à sua simplicidade de modificação e integração.
- **Aplicações do BOOM:** O BOOM, com sua arquitetura fora de ordem, é projetado para ambientes que exigem alto desempenho e capacidade de paralelismo. Ele é usado em:
 1. Servidores e Data Centers: Aplicações que necessitam de alto throughput para lidar com cargas de trabalho como processamento em nuvem e análise de big data.
 2. Pesquisa de Arquitetura de Processadores: Devido à sua infraestrutura personalizável, o BOOM é usado para explorar novas técnicas de execução fora de ordem e extensões de ISA.
 3. Simulação de desempenho avançado: Teste de algoritmos complexos que requerem reordenação de instruções e otimização de uso de recursos computacionais.

5.5. Comparação entre Rocket Chip e BOOM

Embora compartilhem a mesma base de desenvolvimento e algumas tecnologias subjacentes, o Rocket Chip e o BOOM atendem a requisitos distintos. O Rocket Chip é mais focado em oferecer um sistema flexível e escalável para uma ampla gama de aplicações, desde sistemas embarcados até configurações de servidores, com ênfase na customização de SoCs completos. Por outro lado, o BOOM é direcionado especificamente para cenários que beneficiam de execução fora de ordem, sendo mais adequado para tarefas que requerem alto desempenho de processamento.

O Rocket Chip adota uma abordagem in-order, onde as instruções são executadas na ordem em que chegam. Isso simplifica o design do pipeline e reduz os custos de implementação, tornando-o eficiente em termos de área e consumo de energia. Ele utiliza cache configurável e uma estrutura de interconexão escalável, permitindo que seja adaptado para diferentes configurações de sistemas multicore. No entanto, sua execução sequencial pode limitar o desempenho em cargas de trabalho que exigem paralelismo intensivo.

O BOOM, por outro lado, implementa execução fora de ordem, permitindo que instruções sejam reorganizadas dinamicamente para maximizar o uso de recursos do processador. Ele possui unidades de reordenação, execução especulativa e múltiplos pipelines paralelos, que permitem alcançar alto desempenho em cargas de trabalho complexas. Essa complexidade adicional exige maior uso de área e energia, mas proporciona um throughput significativamente maior, especialmente em sistemas com alta concorrência de tarefas.

Desempenho Comparativo

- O Rocket Chip se destaca em aplicações onde eficiência energética e simplicidade são prioritárias. Ele oferece desempenho consistente em cenários de baixa complexidade, mas pode apresentar gargalos em ambientes de processamento intensivo.
- O BOOM é projetado para lidar com alta demanda de processamento, superando o Rocket Chip em throughput e capacidade de execução paralela. No entanto, seu design mais avançado resulta em maior consumo de energia e complexidade de implementação.

Ambos os projetos são exemplares da capacidade inovadora da arquitetura RISC-V e fornecem plataformas valiosas para a experimentação e desenvolvimento de novas tecnologias em arquiteturas de microprocessadores.

5.6. Conclusão

O Rocket Chip e o BOOM representam os avanços na implementação de arquiteturas de processadores baseados em RISC-V, oferecendo soluções desde núcleos simples até complexos sistemas out-of-order. Através de suas capacidades parametrizáveis e a utilização da linguagem Chisel, ambos permitem uma customização extensiva, tornando-os ideais para pesquisa e desenvolvimento em diversas áreas da computação. Como ferramentas de inovação, eles continuam a impulsionar o desenvolvimento de sistemas computacionais mais eficientes e adaptados às necessidades futuras.

6. Comparação Geral entre as Implementações

A análise comparativa entre as implementações discutidas nesta revisão — PicoRV32, NEORV32, PULPino, Rocket Chip e BOOM — destaca a diversidade e a flexibilidade da arquitetura RISC-V. Cada uma dessas implementações foi projetada para atender a diferentes necessidades, desde simplicidade e eficiência energética até alto desempenho e escalabilidade. A seguir, são apresentados os aspectos mais relevantes que diferenciam e conectam essas implementações, com ênfase em suas características, aplicações e impacto no design de sistemas computacionais.

Comparação Focada

1. Foco de cada implementação

- **PicoRV32:** Voltado para simplicidade e baixo consumo de recursos, ideal para dispositivos embarcados pequenos e sistemas FPGA compactos. Seu design minimalista o torna adequado para aplicações onde espaço e eficiência são prioridades.
- **NEORV32:** Oferece um núcleo mais versátil e configurável, capaz de integrar periféricos e suportar múltiplas extensões, permitindo sua aplicação em sistemas embarcados e prototipagem que necessitam de maior funcionalidade.
- **PULPino:** Projetado para computação de baixo consumo, atende bem ao ecossistema IoT e sistemas embarcados que exigem maior eficiência energética e suporte a processamento digital de sinais (DSP).
- **Rocket Chip:** Adota uma abordagem escalável e parametrizável, ideal para sistemas multicore e aplicações onde simplicidade e escalabilidade coexistem.
- **BOOM:** Focado em alto desempenho, utiliza execução fora de ordem para maximizar throughput e eficiência em cenários de alta complexidade computacional.

2. Características e Configurações

- O PicoRV32 se destaca por ser extremamente leve, com um pipeline básico e execução in-order. Sua simplicidade reflete em um design eficiente para dispositivos com restrições severas de hardware.
- O NEORV32 introduz maior modularidade, permitindo a inclusão de periféricos integrados e extensões como multiplicação/divisão (M), operações atômicas (A) e compactação (C). Isso o torna mais versátil em comparação ao PicoRV32, mas também mais complexo.
- O PULPino combina suporte ao conjunto de instruções RV32IMC com extensões de DSP, proporcionando maior capacidade de processamento em aplicações IoT e de sensores inteligentes. Sua ênfase está na eficiência energética.
- O Rocket Chip apresenta uma arquitetura configurável e escalável, suportando núcleos multicore, caches ajustáveis e periféricos variados. Ele é altamente adaptável a diferentes níveis de complexidade.
- O BOOM, sendo uma implementação fora de ordem, oferece tecnologia avançada como execução especulativa, múltiplos pipelines e buffers de reordenação. Esses elementos aumentam sua complexidade, mas proporcionam ganhos significativos de desempenho.

3. Desempenho e Eficiência

- O PicoRV32 é o mais eficiente em termos de área e consumo de energia, mas seu desempenho é limitado a tarefas simples devido à sua execução in-order e falta de extensões avançadas.
- O NEORV32, com suas configurações modulares, oferece um equilíbrio entre desempenho e eficiência, sendo capaz de lidar com tarefas moderadamente complexas em sistemas embarcados.
- O PULPino se destaca na eficiência energética, especialmente em aplicações que exigem processamento contínuo, como sensores industriais e dispositivos IoT.
- O Rocket Chip equilibra desempenho e eficiência, sendo adequado para aplicações escaláveis. Sua configuração multicore e cache ajustável o tornam competitivo em uma ampla gama de cenários.
- O BOOM, devido à sua execução fora de ordem, alcança os melhores resultados em termos de throughput e desempenho, mas a um custo mais elevado de consumo de energia e área.

Tabela Comparativa

Implementação	Foco	Conjunto de Instruções	Aplicações Típicas	Desempenho	Consumo de Recursos
PicoRV32 https://github.com/YosysHQ/picorv32	Simplicidade	RV32IMC	Sistemas embarcados simples	Baixo	Muito baixo
NEORV32 https://github.com/stnolting/neorv32	Versatilidade	RV32I + extensões opcionais	Prototipagem e embarcados	Médio	Moderado
PULPino https://github.com/pulp-platform/pulpino	Eficiência energética	RV32IMC + extensões DSP	IoT e sensores inteligentes	Médio	Baixo
Rocket Chip https://github.com/chipsalliance/rocket-chip	Escalabilidade	Configurável	Multicore e SoCs	Alto	Moderado
BOOM https://boom-core.org	Alto desempenho	Configurável	Servidores e HPC	Muito alto	Alto

Table 1. Comparação Geral entre Implementações RISC-V

7. Conclusão e Perspectivas Futuras

O estudo das implementações abertas da arquitetura RISC-V demonstrou sua versatilidade, desde soluções minimalistas como o PicoRV32 até núcleos de alto desempenho como o BOOM. Essas implementações destacam a capacidade do RISC-V de atender a diferentes cenários, promovendo aprendizado e inovação. Além disso, sua natureza aberta democratiza o acesso à tecnologia, permitindo que educadores e desenvolvedores integrem conceitos teóricos com aplicações práticas em projetos reais. O RISC-V não é apenas uma alternativa viável às arquiteturas proprietárias, mas também um catalisador para o avanço tecnológico e educacional.

O futuro do RISC-V promete maior adoção comercial, com empresas integrando a arquitetura em dispositivos de consumo e servidores, ampliando sua presença no mercado. Ferramentas específicas para desenvolvimento e depuração continuarão a crescer, reduzindo barreiras para novos usuários e promovendo a integração em currículos educacionais. Em sistemas de baixo consumo, como IoT e sensores, o RISC-V fortalecerá seu impacto, enquanto no segmento de alto desempenho, otimizações para inteligência artificial e HPC devem consolidar sua posição competitiva.

Apesar dos desafios de padronização devido à modularidade, a comunidade global se beneficiará ao equilibrar flexibilidade com compatibilidade. O RISC-V tem o potencial de se tornar um padrão universal, transformando a forma como hardware e software são projetados, enquanto impulsiona uma era de inovação colaborativa e acessível.