

ARQUITETURA DE COMPUTADORES - BCC 236

Lista: RISC-V

Nome: Mateus Vitor Pereira Lana

Matrícula: 15.1.4340

- 1- A arquitetura ISA RISC é empregada em dispositivos computacionais mais modernos, desde projetos embarcados de pequeno porte até servidores. Os celulares high-end (top de linha), Microcontroladores que processam imagens e gráficos. Esta é também a arquitetura adotada para os processadores dos videogames modernos, que proporcionam um hardware extremamente dedicado somente à execução do jogo, tornando-o muito mais rápido em relação a micro computadores com mais recursos, embora com processador x86.
- 2- O projeto RISC-V refere-se a um ISA open-source desenvolvido por profissionais, professores e voluntários entusiastas da área de Arquitetura de Computadores. Estes possuem experiência e conhecimento em várias especialidades como Sistemas Operacionais, Microprocessadores, Compiladores, Eletrônica Digital, dentre outras. O ISA RISC-V é um moderno e de alta qualidade Conjunto de Instruções para Computador de Propósito Geral e foi desenvolvido com base em uma série de outros projetos acadêmicos de design de computadores. Além disso, ele é um conjunto limpo, modular com inteiros bases em 32, 64 e 128 bits e várias opções de extensão de instruções como ponto-flutuante, multiplicadores, etc. Seu projeto é modelado para ser útil nos dispositivos computacionais mais modernos que vão desde projetos embarcados de pequeno porte até servidores e celulares high-end.
- 3- Esse projeto não é utilizado apenas para fins acadêmicos, ele possui um propósito de mercado bem amplo, tanto é que ele é suportado para executar em Microcontroladores que processam imagens e gráficos ou até mesmo Processadores de servidores. A arquitetura é empregada em indústrias de grande porte como Oracle, Google e Mellanox, além de grandes centros acadêmicos.
- 4- São elas:

Multiplicação M: Adiciona 4 instruções de multiplicação, duas de divisão, e duas de manipulação de restos. A base é indicada pela base do ISA. A divisão por 0, por exemplo, não causa interrupção, mantendo a simplicidade.

Sincronização A: Adiciona 11 instruções de Sincronização visando consistência e atomicidade da operação. São divididos em dois grupos: Load Reserved e Store Conditional para operações atômicas na memória.

Ponto Flutuante: Possui três extensões diferentes: Precisão simples (F), Precisão Dupla (D) e quádrupla precisão (Q). A extensão F é pré-requisito para D, que é pré-requisito para Q. Introduz também 32 registradores de ponto flutuante (f0-f31) de tamanho 32 bits e um registrador de 5 bits para exceções. Exceções não geram interrupções e com isso devem ser verificadas por meio de consultas. As instruções load-store usam o mesmo endereçamento base+offset.

Compressão De Tamanho De Código: A última extensão C não adiciona nenhuma outra função, mas, ao invés disso, codifica as instruções inteiras para salvar espaço e com isso reduzir o tamanho do footprint. É disponível para bases inteiras, bem como load e store para pontos flutuantes e cria-se então instruções comprimidas em 16 bit. Basicamente, cada função compactada é mapeada diretamente à instrução real e possui algumas restrições para a compressão sobre o formato dos operandos.

Extensões potenciais do RISC-V incluem novos tipos como:

- Extensões Cray-style vector;
 - Memória Transacional;
 - Manipulação de bit.
- 5- Sim, é possível. Pois porções do espaço de codificação de instruções já foram reservados para uso futuro, ou seja, é só adicionar a extensãoUFOP e usar. Isso se dá pela sua universalidade.
- 6- Ele possui 32 registradores inteiros e 32 opcionais para ponto flutuante. Possui também registradores de controle e status, mas somente o nível de privilégio mais alto que poderá acessa-los para medição de performance.
- 7- São elas:
- Disposição de bits para reduzir o número de multiplexadores no CPU;
 - Colocar os bits mais significantes numa posição fixa;
- 8- RISC-V não possui operações de multiplicação com acesso direto à memória principal, pois é uma máquina load-store. Somente estas duas instruções que acessam a memória principal e todas as operações lógico-aritméticas ocorrem entre registradores.
- 9- O propósito de utilizar a extensão C é codificar as instruções inteiras para salvar espaço e com isso reduzir o tamanho do footprint. É disponível para bases inteiras, bem como load e store para pontos flutuantes e cria-se então instruções comprimidas em 16 bits.
- 10- Existe diferença entre os formatos de operações do RV32I, RV64I e a Extensão C. O RV32I opera com dados de 32 bits e o RV64I opera com dados de 32 e 64

bits, já a extensão C cria instruções comprimidas em 16 bits, operando com esse formato.

- 11-** Ele, intencionalmente, não possui códigos condicionais, nem mesmo bit de carry e sendo assim, os projetistas afirmam que isso pode simplificar o desenvolvimento do CPU, minimizando interações entre instruções. Os projetistas construíram jumps condicionais com operando de comparação afim de realizar as instruções de condição.
- 12-** Eles estão situados na mesma posição em todas os formatos porque isso é um método para a facilitação da decodificação das instruções.
- 13-** A diferença é que a instrução jump realiza um desvio incondicional para um endereço de memória apontado por um label, já o jump and link salta para o endereço especificado, salvando o endereço da próxima instrução.
- 14-** É uma ferramenta que permite gerar diferentes configurações de RISC-V para diferentes ambientes SoC(System-on-a-chip, que se refere a todos os componentes de um computador/sistema eletrônico, em um circuito integrado). Essas configurações são especificadas por meio de parâmetros em linguagem Chisel (Constructing Hardware In a Scala Embedded Language) que podem ser alterados livremente e com isso, pode-se selecionar o que se quer gerar como chip em nosso projeto almejado. O Rocket-Chip possui 5 estágios.
- 15-** O Rocket-Chip é utilizado de modo geral como um ‘gerador de código para hardware’ onde pode-se alterar seus parâmetros. Esses parâmetros são em linguagem Chisel (Constructing Hardware In a Scala Embedded Language) onde pode-se selecionar o que se quer gerar como chip em nosso projeto almejado. Assim, utiliza-se de um emulador C++ de RLT (Register transfer level).