



Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM
Disciplina: Arquitetura de Computadores – BCC 236
Aluno: Thiago Oliveira de Santana
Matrícula: 15.1.4313



LISTA DE EXERCÍCIOS- RISC V

1) Dê exemplos de aparelhos que utilizam ISA RISC.

É utilizado nos dispositivos computacionais mais modernos que vão desde projetos embarcados de pequeno porte até servidores. Os celulares high-end, consoles (como o Nintendo 64 e o Playstation), Microcontroladores (que processem imagens e gráficos) e processadores de servidores utilizam ISA RISC.

2) Explique o que é o projeto RISC-V.

RISC-V é um ISA open-source desenvolvido por profissionais, professores e voluntários entusiastas no ramo de Arquiteturas de Computadores (pessoas que possuem experiência em várias especialidades como Eletrônica Digital, Compiladores, Sistemas Operacionais, Microprocessadores, entre outras como simulação e validação dos design do projeto) na qual é basicamente um moderno Conjunto de Instruções para Computador de Propósito Geral de alta qualidade. Foi produzido pela Computer Science Division na Universidade da Califórnia, Berkeley. 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) RISC-V é utilizado só para uso acadêmico?

Não. Seu propósito de mercado é amplo ao ponto de ser suportado para executar em Microcontroladores que processem imagens e gráficos ou até mesmo Processadores de servidores. É utilizado em indústrias de grande porte como Google, Mellanox e Oracle além de grandes centros acadêmicos.

4) Quais são as extensões oficiais do RISC-V ditas em aula e qual o propósito de cada uma?

As extensões oficiais são:

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



Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM
Disciplina: Arquitetura de Computadores – BCC 236
Aluno: Thiago Oliveira de Santana
Matrícula: 15.1.4313



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; e***
- ***Manipulação de bit.***

5) É possível adicionar minha própria extensão, cujo nome é extensaoUFOP ao projeto?

Sim, 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) Quantos registradores o RISC-V possui e quais são eles? Esse número pode ser alterado?

RISC-V possui 32 registradores inteiros, e 32 registradores opcionais para ponto flutuante. Registradores de controle e status também, mas somente o



Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM
Disciplina: Arquitetura de Computadores – BCC 236
Aluno: Thiago Oliveira de Santana
Matrícula: 15.1.4313



nível de privilégio mais alto (user) que poderá acessá-los para medição de performance. Sim, pois existe também uma variante de pequeno porte do RISC-V com só 16 registradores inteiros.

7) Cite duas otimizações realizadas no projeto do RISC-V.

Várias otimizações foram realizadas como:

- Colocar os bits mais significantes numa posição fixa;
- Disposição de bits para reduzir o número de multiplexadores no CPU;

8) RISC-V possui operações de multiplicação com acesso direto à memória principal?

Não, pois o RISC-V é uma máquina load-store. Somente estas duas instruções que acessam a memória principal e todas as operações lógico-aritméticas ocorrem entre registradores.

9) Qual é o propósito de utilizar a extensão C?

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 alguma diferença entre os formatos de operações do RV32I, RV64I e a Extensão C?

Sim, pois 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) RISC-V possui o condicional if? Como ele realiza suas instruções de condição?

O RISC-V, 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.



Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM
Disciplina: Arquitetura de Computadores – BCC 236
Aluno: Thiago Oliveira de Santana
Matrícula: 15.1.4313



12) Observando as posições dos registradores na tabela de formato de instruções, por que eles estão situados na mesma posição em todas os formatos?

Porque isso é um método para a facilitação da decodificação das instruções.

13) Qual a diferença das instruções *jump* e *jump and link*?

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) O que é o Rocket-Chip e quantos estágios ele possui?

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) Explique sucintamente como Rocket-Chip é utilizado.

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).