

Questão 1 Dê exemplos de aparelhos que utilizam ISA RISC:

A arquitetura RISC já é utilizada desde 2015 por multinacionais de grande porte como Google, Oracle, Mellanox e Apple.

Processadores baseados no RISC já dominam o mercado no âmbito mobile superando em larga escala os processadores Intel.

A Nintendo, na indústria de videogames, adotou o RISC na produção do Game Boy Advance e manteve o uso da tecnologia nos portáteis mais recentes Nintendo DS e 3DS.

A Sony adotou o RISC na produção de todos os seus videogames da família Playstation abandonando a tecnologia com o Playstation 4.

A IBM adotou a tecnologia RISC na fabricação de supercomputadores.

Questão 2 Explique o que é o projeto RISC-V:

O projeto RISC-V é uma Instruction Set Architecture open-source desenvolvida por profissionais interessados, professores e voluntários entusiastas idealizada como um conjunto de instruções para computador de propósito geral moderno e de alta qualidade.

Limpas e modulares, com bases inteiras em 32, 64 e 128 bits além de várias opções de extensão de instruções, cada empresa ou usuário consegue ter plenas opções de personalização de forma a adaptar o processador exatamente para a necessidade objetivada.

Questão 3 RISC-V é utilizado só para uso acadêmico?

O RISC teve início como um projeto acadêmico em 2010 mas foi liberado para uso no mercado em 2015 sendo adotado em seguida por grandes empresas como Google, Mellanox e Oracle. A partir disso diversos projetos funcionais apareceram fazendo uso da licença de código aberto. Um grande exemplo é o escalonador de 5 estágios, RISC-V Rocket.

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

O RISC-V dá suporte a diversos tipos de extensões que complementam o funcionamento de acordo com a necessidade do usuário e objetivo empregado.

Multiplicação: Adiciona 4 instruções de multiplicação, duas de divisão e duas de manipulação de restos.

Sincronização: Adiciona 11 instruções de sincronização divididas em dois grupos, Load reserved e Store Conditional, ambos visando consistência e atomicidade da operação.

Ponto Flutuante: Existem três extensões diferentes de ponto flutuante variando sua precisão em Precisão simples, Precisão dupla e Precisão Quadrupla. Elas adicionam 32 registradores de ponto flutuante de 32 bits além de registradores de 5 bits para exceções.

Compressão de Tamanho decodigo: Essa extensão não adiciona nenhuma função nova mas codifica as instruções inteiras para salvar espaço. É disponível para bases inteiras, bem como load e store para pontos flutuantes. As instruções são comprimidas em 16 bits.

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

Porções do espaço de codificação de instruções já foram idealizadas e reservadas para uso futuro. Isso faz com que seja possível adicionar novas extensões personalizadas como a “extensaoUFOP”.

Questão 6 Quantos registradores o RISC-V possui e quais são eles? Esse número pode ser alterado?

O RISC-V possui 32 Registradores, além 32 Registradores opcionais vindos da extensão de uso de Ponto flutuante.

Uma variante do projeto possui um número menor de registradores, 16 no lugar dos 32.

O Registrador r0 é reservado para manuseio de algumas operações. Fora ele existem registradores de controle e status que só podem acessados por níveis de privilégio mais altos.

Questão 7 Cite duas otimizações realizadas no projeto do RISC-V:

Diferente de outros RISCs, o conjunto de instruções do projeto RISC-V foi desenvolvido com foco na praticidade e otimizado de várias formas, dentre elas as citadas abaixo:

- Colocar os bits mais significantes numa posição fixa.
- Disposição de bits para reduzir o número de multiplexadores no chip em sua implementação.

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

Não, o RISC-V trabalha com a ideia Load/Store, somente essas duas instruções possuem acesso à memória principal.

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

Ela codifica as instruções inteiras para salvar espaço e reduzir o tamanho do footprint. É possível utilizar em bases inteiras ou em loads/store com ponto flutuante.

Basicamente, a extensão em C reduz o tamanho do código binário, energia e custo para pequenos computadores sendo muito útil em sistemas embarcados.

Questão 10 Existe alguma diferença entre os formatos de operações do RV32I, RV64I e a Extensão C?

RV32I e RV64I são variantes do RISC enquanto que a Extensão C, como o nome indica, é uma extensão que pode ser imbutida junto as instruções pre definidas.

RV32I tem o mesmo objetivo que a extensão C, permitir a implementação do RISC em aparelhos de pequeno porte com baixa potencia. Ao contrario da extensão, o RV32I tambem reduz o tamanho fisico do chip tornando totalmente viavel sua utilização em sistemas embarcados. Seu funcionamento se dá de forma diferente da extensão principalmente devido ao fato da redução do numero de registradores para somente o Program Counter e os registradores x0 até x15.

Já o RV64I são somente extensões do original com suporte para 64bits aumentando assim o espaço de endereço e dos registradores de 32 para o tamanho apropriado alem de introduzir novas instruções para operar com dados menores de 32bits.

Questão 11 RISC-V possui o condicional if? Como ele realiza suas instruções de condição?

Ele não possui o condicional if mas existem metodos equivalentes que se utilizam dos jumps condicionais. Utilizando varios jumps em sequencia é possivel simular condicionais if, else e até mesmo um switch case completo.

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

Rs2, rs1, rd e o opcode estão sempre na mesma posição independentemente do formato das instruções de forma a facilitar na hora da decodificação que precisa lidar com menos variações na organização dos registradores.

Questão 13 Qual a diferença das instruções jump e jump and link?

Jump e Jump and Link tem funções parecidas mas funcionam de formas diferentes.

JUMP executa um salto partindo da instrução atual e terminando na instrução indicada finalizando ai o processo.

JUMP AND LINK executa o mesmo salto com o detalhe de armazenar em um registrador fixo o endereço original da instrução responsavel pelo JUMP permitindo que seja possivel retornar até ela em algum momento posterior para continuar as instruções abaixo dela.

Questão 14 O que é o Rocket-Chip e quantos estágios ele possui?

O Rocket-Chip é um processo escalar de cinco estagios para sintetização de chips. Ele permite gerar diferentes configurações de sistema no chip que são especificadas por meio de parametros em linguagem Chisel que podem ser alterados livremente.

Questão 15 Explique sucintamente como Rocket-Chip é utilizado:

O Rocket Chip é formado por diferentes módulos, cada um responsável por uma função diferente. Manipulando esse módulo é possível configurar de diferentes formas o chip. Essas configurações são feitas com base nas diferentes performances e objetivos destinados, isto é, diferentes arquiteturas para diferentes propósitos finais.

Blocos, cores, ou como são chamados, Rockets, são gerados junto de suas caches privadas que interagem com outros itens externos ao rocket e itens compartilhados.

Tendo programado configurado o chip com os parâmetros desejados, é possível gerar o programa utilizando C++ que pode ser executado no submódulo Chisel onde itens podem ser construídos com base na configuração prévia feita.

Tendo feita a configuração do chip e sua contra parte em software no emulador, é possível sintetizar a placa e então adaptar o código ao uso real.