

Aluno: Sebastião Gonçalves da Silva Júnior Matrícula: 15.1.5871

Questão 1

Dê exemplos de aparelhos que utilizam ISA RISC.

R= Smartphones, Consoles Nintendo 64 e Playstation.

Questão 2

Explique o que é o projeto RISC-V.

R= Projeto desenvolvido por profissionais, professores, entusiastas no ramo de arquitetura, que teve como resultado o RISC-V, que é um moderno e de alta qualidade conjunto de instruções para Computador de Propósito Geral, o ISA RISC-V. 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 e etc. A meta do projeto RISC-V é criar um conjunto de instruções 'universal', que é livre e aberto para todos os usuários, provendo tudo que é necessário para suportar perfeitamente qualquer projeto comercial.

Questão 3

RISC-V é utilizado só para uso acadêmico?

R= Não o melhor exemplo disso são os smartphones que já estão vindo com seus processadores com essa tecnologia.

Questão 4

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

R= Variantes E Extensões

Variantes: RV32I, RV32E, RV64I E RV128I

Todas as outras variantes do RISC-V são baseadas no RV32I.

RV32E:

Possui a meta para ser satisfatório para implementações de pequeno porte.

Sendo um projeto menor, ocupa-se uma pequena porção de área die.

Aumenta o custo/benefício de produção de chips;

Melhora a dissipação de calor por se tratar de um chip com menos área de circuito;

Isso reduz registradores para somente o Program Counter e x0-x15.

De resto, RV32E e RV32I são idênticos, com exceção de:

Suas instruções continuam com tamanho de 32 bits. Os bits de maior índices são 0 (zero).

RV64I e RV128I:

Simples extensões de 64 e 128 bits.

Aumentam o espaço de endereço e estende os registradores de 32 para o tamanho apropriado.

Essa extensão introduz novas instruções que operam com dados menores como 32 e 64 bits.

Extensão: Multiplicação

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.

Extensão: Sincronização

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.

Extensão: 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.

Extensão: 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.

O propósito de reduzir o tamanho do código binário, energia e custo para pequenos computadores, é visando sistemas embarcados. Pesquisas com C mostram que um código 20% menor que um x86 e MIPS Comprimido e 2% maior que um ARM Thumb-2.

Níveis De Privilégios

Níveis De Privilégios

Existe 4 níveis de privilégios:

User (U);

Supervisor (S);

Hypervisor (H); e

Machine (M).

Para reduzir o custo de uma implementação de pequeno porte, somente o machine é obrigatório. Entretanto, todos os 4 privilégios são suportados e quando suportados, é possível até executar o Linux.

<http://www.decom.ufop.br/imobilis/o-risc-v/>

Questão 5

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

R= Sim pois o código é open source.

Questão 6

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

R= Ele possui 3 tipos básicos de registradores: os de serviço ou especiais, os de uso geral, e eventualmente os de ponto flutuante.

Questão 7

Cite duas otimizações realizadas no projeto do RISC-V.

R= Ausência de microcódigo.

O microcódigo não se presta à execução em ciclos únicos, já que requer que o hardware seja dedicado a sua interpretação dinâmica. A programação em microcódigo não faz com que o software seja mais rápido que o programado com um conjunto de instruções simples. Todas as funções e o controle, nos processadores RISC, estão "cablados" para conseguir uma máxima velocidade e eficiência.

Execução em ciclos únicos (single-cycle).

O resultado direto dos conjuntos de instruções que oferecem os processadores RISC, é que a cada instrução pode ser executada em um único ciclo da CPU. Isto invalida a crença de que as microinstruções em microcódigo, criadas para ser executadas em um só ciclo de processador, são mais rápidas que as instruções da linguagem ensambladora. Já que o cache este construído partindo da mesma tecnologia que o armazenamento de controle do microprograma, uma única instrução pode ser executada à mesma velocidade que uma microinstrucción. A execução em ciclos únicos também simplifica o gerenciamento das interrupções e os condutos (pipelines).

http://www.resumosetrabalhos.com.br/arquitetura-de-computadores_9.html

Questão 8

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

R= Sim, porque as instruções de load e store podem acessar a memória diretamente.

Questão 9

Qual é o propósito de utilizar a extensão C?

R= O propósito de reduzir o tamanho do código binário, energia e custo para pequenos computadores, é visando sistemas embarcados. Pesquisas com C mostram que um código 20\% menor que um x86 e MIPS Comprimido e 2% maior que um ARM Thumb-2.

Questão 10

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

R= As instruções RV32I tem por finalidade ser satisfatório para implementações de pequeno porte e suas instruções continuam no tamanho de 32 bits.

As instruções RV64I aumentam o espaço de endereço e estendem os registradores de 32 para o tamanho apropriado, além disso ela introduz novas instruções que operam com dados menores com 32 e 64 bits.

Já a instrução C tem com objetivo reduzir o tamanho do código binário, energia e custo para pequenos computadores, é visando sistemas embarcados. Pesquisas com C mostram que um código 20\% menor que um x86 e MIPS Comprimido e 2% maior que um ARM Thumb-2.

Questão 11

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

R= Ele possui o condicional if.

Questão 12

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

Questão 13

Qual a diferença das instruções jump e jump and link?

R= Jump: Desvio incondicional para um endereço de memória apontado por um label.

Jump and link: Salta para o endereço especificado, salvando o endereço da próxima instrução em \$ra.

Questão 14

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

R= Rocket Chip Generator Do SoC Risc-V

É um gerador open-source de sistemas com cores em ordem que implementam variações do RISC-V ISA. É um projeto altamente “parameterizável” para gerar vários tipos de designs possíveis a abranger a maior quantidade de projetos possível.

O projeto baseia-se no Rocket, versão de 2014 onde possui 5 estágios em ordem baseando na implementação do RV64G.

<http://www.decom.ufop.br/imobilis/soc-risc-v-e-energia/>

Questão 15

Explique sucintamente como Rocket-Chip é utilizado.

R= Acelerador vetorial implementado por meio de instruções customizadas do ISA Hwacha que é uma extensão não padrão do projeto RISC-V, disponível em <http://hwacha.org/>.

É acoplado ao Rocket via interface RoCC.

Unidade De Gerenciamento De Energia Do SoC Risc-V

Tal unidade chama-se Z-scale. É um processador baseado em um RISC-V de pequeno porte sendo este o RV32IM. Possui pipeline de 3 estágios. Um diagrama simples de seu pipeline é exibido abaixo.

