

## SOLUÇÃO DA LISTA

ALuno: Yago José Araújo dos Santos. Matrícula: 14.2.4526.

1 - Smartphones, tablets, smartwatch

2 - RISC-V (pronunciado "risco-cinco") é uma arquitetura de conjunto de instruções abertas (ISA) baseado em princípios estabelecidos de computação de conjunto de instruções reduzidas (RISC). Em contraste com a maioria dos ISAs, o RISC-V ISA pode ser usado livremente para todos os tipos de uso, permitindo que qualquer pessoa para projetar, fabricar e vender chips RISC-V e software. Embora não seja o primeiro ISA aberto, é significativo porque é projetado para ser útil em modernos dispositivos computadorizados, como computadores em nuvem em escala de armazém, telefones celulares high-end e os menores sistemas embarcados. Esses usos exigem que os projetistas considerem o desempenho ea eficiência de energia. O conjunto de instruções também tem um corpo substancial de software de suporte, que corrige uma fraqueza usual de novos conjuntos de instruções.

3 - Projeto iniciado em 2010, foi disponibilizado em 2015 e já é utilizado em indústrias de grande porte como Google, Mellanox e Oracle além de grandes centros acadêmicos. Com isso vários projetos funcionais online já estão disponíveis, usufruindo da permissão de licença BSD. Um exemplo é o escalar de 5 estágios chamado RISC-V Rocket.

4 - RV32I, RV64I e Extensão C(ompacta). Extensão: Multiplicação, Extensão: Sincronização, Extensão: Ponto Flutuante; Extensão: Compressão De Tamanho De Código.

5 - Sim, a arquitetura utiliza conjuntos de instruções abertas.

6 - RISC-V possui 32 registradores inteiros, e 32 registradores opcionais para ponto flutuante. Também existe uma variante RISC-V com 16 registradores inteiro e sua memória possui endereçamento de 1 byte.

7 - 1) Colocar os bits mais significantes numa posição fixa; 2) Disposição de bits para reduzir o número de multiplexadores no CPU;

8 - Soma rx com 0: add r0 to rx;

Salva o resultado em ry: store in ry.

Somente estas duas instruções que acessam a memória principal e todas as operações lógico-aritméticas ocorrem entre registradores.

9 - Codifica as instruções inteiras para salvar espaço e com isso reduzir o tamanho do footprint. 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.

10 - 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. A 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.

11 - 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. Construíram

operandos de comparação dentro dos jumps condicionais, não possui carry de operações aritméticas complicadas nem tem detector ou flag para erros aritméticos, incluindo overflow, underflow e divisão por 0 (zero).

12 - Foi desenvolvido para suportar sistema de memória e instruções inteira e ponto flutuante de 32, 64 e 128 bits. Suas funções load e store pode realizar operações com 16 e 8 bits, mas não operações aritméticas. Já as instruções de 64 bits, incluem aritméticas de 32-bits.

13 - Jump coloca um determinado endereço no contador de programa e continua a execução a partir daí, no entanto, JAL (jump and link) faz o mesmo, mas salva o endereço de retorno em ra para que você possa continuar a execução após a sua sub-rotina termina.

14 - Um exemplo é o escalar de 5 estágios chamado RISC-V Rocket em Scala e para sintetização em chips.

15 - O Rocket é um pipeline em andamento de 6 estágios em questão que executa o escalar RISC-V ISA de 64 bits. O Rocket implementa uma MMU que suporta memória virtual baseada em página e é capaz de inicializar sistemas operacionais modernos, como o Linux. O Rocket também possui uma FPU opcional compatível com IEEE 754-2008, que implementa operações de ponto flutuante de precisão simples e dupla, incluindo multiplicação fundida.

Sites consultados:

- <http://www.decom.ufop.br/imobilis/o-risc-v/>
- <https://riscv.org/>
- <https://pt.wikipedia.org/wiki/RISC>
- <http://www.hardware.com.br/dicas/arquitetura-hardware-smartphones.html>
- <http://stackoverflow.com/questions/9936605/different-between-jump-and-jump-and-link>