

Resolução dos exercícios

**Questão 1** – Iphone, Ipod, Ipad, Nintendo DS, Playstation, Playstation 2 são alguns dos aparelhos que utilizam RISC.

**Questão 2** – RISC-V é um ISA *open-source*. O RISC-V foi desenvolvido por profissionais, professores e entusiastas no ramo de Arquitetura de computadores com a intenção de criar um conjunto de instruções “universal”, que é livre e aberto para todos usuários, provendo tudo que é necessário para suportar perfeitamente qualquer projeto comercial.

**Questão 3** – Não. Apesar de ser disponibilizado recentemente, em 2015, o RISC-V já é utilizado em indústrias de grande porte como Google, Mellanox e Oracle.

**Questão 4** – As extensões oficiais do RISC-V são: M, A, F, D, Q, C.  
Funcionalidades:

M – Adiciona 4 instruções de multiplicação, duas de divisão e duas de manipulação de restos.

A – Adiciona 11 instruções de sincronização visando consistência e atomicidade da operação.

F – Precisão simples.

D – Precisão dupla.

Q – Quádrupla precisão.

C – Codifica as instruções inteiras para salvar espaço e com isso reduzir o tamanho do *footprint*.

**Questão 5** – Sim. A extensibilidade do RISC-V é parte essencial de sua universalidade, logo pode-se adicionar sua própria extensão.

**Questão 6** – RISC-V possui 32 registradores inteiros e 32 registradores opcionais para ponto flutuante. Sim, existe uma variante do RISC-V contendo apenas 16 registradores inteiros.

**Questão 7** – Duas das várias otimizações realizadas no RISC-V são: todas instruções têm 32 bits e os bits mais significantes foram colocados em uma posição fixa.

**Questão 8** – Não. O RISC-V não possui *carry* de operações aritméticas complicadas, como multiplicação e divisão.

**Questão 9** – A extensão C codifica as instruções inteiras para salvar espaço e com isso reduzir o tamanho do *footprint*.

**Questão 10** – O RV32I possui o seguinte formato de operações: ADDI, SLTI[U], ANDI, ORI, XORI; já o RV64I e a Extensão C possuem o ANDIW;

**Questão 11** – RISC-V não possui a condicional *if*. A instrução *jump* é utilizada para realizar operações condicionais no RISC-V.

**Questão 12** – Deixar os registradores rd, rs1 e rs2 na mesma posição em todos os formatos é um método utilizado para facilitação da decodificação. Neste método os imediatos estão sempre no final da instrução

**Questão 13** – A diferença entre as instruções *jump* e *jump and link* é que a instrução *jump and link* tem um endereço de retorno e a instrução *jump* não.

**Questão 14** – Rocket-Chip permite gerar diferentes configurações de *System-on-a-chip*. Ele contém 5 estágios.

**Questão 15** – As configurações geradas pela ferramenta são especificadas por meio de parâmetros em linguagem Chisel que podem ser alterados livremente e com isso, pode-se selecionar o que se quer gerar como chip em nosso projeto almejado.