

1- Smartphones como iphone..

2- Risc-V é um ISA open-souce desenvolvido por profissionais, professores e voluntários entusiastas, no caso, um moderno e de alta qualidade conjuntos de instruções para Computador de Propósito Geral. 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.

3- Não, pois as empresas também esperam um balanceamento em performance e uso energético ao projetar um novo processador, fazendo um trade-off e mesmo sendo um projeto novo suas instruções são bem mais simples se comparado a outras alternativas existentes no mercado.

4- Extensão: Multiplicação

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: Compressão De Tamanho De Código

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.

5- Sim. Suas extensões são o que constitui sua universalidade e por isso pudesse ter uma arquitetura que atenda as necessidades do hardware.

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. Sim, pode ser alterado.

7- Primeira: Colocar os bits mais significantes numa posição fixa;

Segunda: Disposição de bits para reduzir o número de multiplexadores no CPU

8- Por ser uma máquina de load-store ela acessa memória principal e todas as operações lógico-aritméticas ocorrem entre registradores.

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

10- RV32I, RV64I Aumentam o espaço de endereço e estende os registradores de 32 para o tamanho apropriado. Já Extensão C salva espaço para diminuir o tamanho.

11- 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. Construam operandos de comparação dentro dos *jumps* condicionais para funcionar igual a um if.

12- Deixar os registradores rd, r1 e rs2 na mesma posição em todos os formatos é um método utilizado para facilitar a decodificação.

13- **Jump** coloca um determinado endereço no contador de programa e continua a execução a partir daí, no entanto, **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- 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. Possui 5 (Cinco) estágios.

15- As configurações geradas pela ferramenta são especificadas por meio de parâmetros e podem ser alterados livremente e com isso pode selecionar o que quer gerar como chip.