

Laboratório 2 EXERCÍCIOS EXTRAS (não é necessário entregar)

1. Uma das inovações da arquitetura ARM foi criar dois conjuntos de instrução para o mesmo processador: o ARM (32 bits) e o Thumb (16 bits), de forma que o programa pode chavear entre eles conforme a necessidade. Qual a razão desta inovação? Imagine vantagens e desvantagens trazidas por isso.

Note-se que algumas versões modernas do MIPS também adotam isso (MIPS16e).

2. Tanto os processadores x86 quanto os ARM ou MIPS possuem um set básico de instruções, codificadas de forma sempre igual e definindo uma família de processadores, na qual cada componente pode possuir extensões particulares.

Já no caso do microcontrolador PIC, a fabricante Microchip decidiu que alguns processadores da mesma família não teriam código binário compatível entre si (ou seja: um código muito simples no processador mais básico não funciona no mais poderoso).

Discorra sobre as causas e efeitos destes dois comportamentos contrastantes

3. ★ Dê os valores em hexadecimal para o seguinte programa em assembly MIPS.

```
addi $s0,$zero,1234
add $s0,$s0,$s0
sw $s0,0($t0)
```

4. ★ Dados os seguintes valores de memória, dê as instruções assembly MIPS equivalentes.

```
02538820h
02538822h
8E510100h
0C000E1Ah
```

Observação: o opcode da instrução lw é 0x23, e não 0/0x23 como está no livro nacional.

5. Converta a seguinte equação booleana para um programa assembly MIPS:

$$s0 = (s1 + s2.s3).s4 + / (s2./s3 + s2./s4 + /s4./s3)$$

6. A arquitetura do conjunto de instruções (ISA) Intel, desde o princípio (o 8086), possui instruções de tamanhos diferentes entre si, com múltiplos formatos possíveis. O tamanho de uma instrução pode variar de 1 a 17 bytes (!)

Cite vantagens e desvantagens em relação a uma ISA mais regular como a do MIPS, onde todas as instruções têm 4 bytes e há apenas três formatos.

7. Arquiteturas antigas de microprocessadores (ditas CISC) tipicamente possuíam instruções que agiam especificamente sobre determinado registrador, usualmente o chamado acumulador. Isso contrasta com as arquiteturas modernas (RISC), que possuem um conjunto ortogonal de instruções, ou seja, uma instrução pode ser executada com qualquer registrador disponível.

Assim, instruções CISC comuns são NEG \$reg (inverte o sinal do número inteiro), INC \$reg (incrementa, o ++ do C) e DEC \$reg (decrementa, o -- do C). Essas instruções seriam úteis no MIPS? Elas eram úteis nos CISC? Argumente.

Saiba que uma das razões de haverem os operadores especiais ++ e -- era justamente para indicar ao compilador que INC e DEC deveriam ser usados em lugar de ADDI.

8. Outra característica das arquiteturas RISC modernas é que elas tipicamente são denominadas *arquiteturas load/store*, porque o acesso à memória é feito exclusivamente por instruções exclusivas, que carregam ou gravam dados nela, como lw e sw.

Nas arquiteturas CISC, usualmente o acesso pode ser feito por diversos tipos diferentes de instrução. Por exemplo, no 80386 podemos ter uma instrução ADD AX,ES:[BX+DI+0x4000] que vai ler o dado do endereço ES*16+BX+DI+0x4000, somá-lo ao valor do acumulador AX e armazenar o resultado no próprio AX.

Faça uma comparação entre as consequências de se escolher um ou outro tipo de arquitetura.