

# Arquitectura de Computadores I

## Exercícios

Licenciatura em Engenharia Informática

### Aula Prática #4 (Instruções aritméticas e lógicas RISC-V)

- Usando o simulador RARS, escreva um troço de código que calcule  $y = 3x + 1$ . Assuma que os valores das variáveis  $x$  e  $y$  estão nos registos  $t0$  e  $t1$ .
- Sem usar o simulador RARS, indique qual o resultado de cada uma das instruções seguintes. Suponha que  $t0 = 0x00107fff$  e  $t1 = 0x80000000$ .
  - `srli t2, t0, 4`
  - `srli t2, t1, 4`
  - `srai t2, t0, 4`
  - `srai t2, t1, 4`
  - `srli t2, t0, 5`
  - `srai t2, t1, 5`
- Sem usar o simulador RARS, determine o código máquina das instruções seguintes. (consulte o código máquina nos slides)
  - `sub t2, t1, t0`
  - `slli t2, t0, 29`
- Considere o mapa de memória indicado abaixo. Supondo que se tratam de instruções RISC-V, determine o que faz esse pedaço de código. Confirme a sua resposta no simulador RARS.

Address	Byte
	:
0x00010007	0xff
0x00010006	0xf5
0x00010005	0x05
0x00010004	0x13
0x00010003	0x00
0x00010002	0xa5
0x00010001	0x05
0x00010000	0x33
	:

- Escreva uma instrução que coloque  $t2$  com o valor 1 se  $t0$  for zero, e o valor 0 se  $t0$  for diferente de zero, isto é

$$t2 = \begin{cases} 1 & \Leftarrow t0 = 0, \\ 0 & \Leftarrow t0 \neq 0. \end{cases}$$

- Na internet, os endereços IPv4 são habitualmente escritos como 4 números decimais no intervalo 0 a 255, separados por pontos, como por exemplo 192.168.1.17. Na realidade, cada um destes números representa 8 bits e o endereço completo é simplesmente um número de 32 bits.

Dois endereços pertencem à mesma rede local se todos os  $n$  bits da esquerda forem iguais. Numa rede de classe C (como as que usamos nas nossas casas), todos os endereços dessa rede têm os 24 bits da esquerda iguais.

Supondo que os registos **t0** e **t1** contêm dois endereços IPv4, escreva um troço de código que determine se os endereços pertencem à mesma rede classe C. O registo **t2** deverá ficar a 1 caso sejam da mesma rede e a 0 se forem de redes diferentes.

7. Escreva um pedaço de código que permita verificar se o valor do registo **t0** é uma potência de 2. Se for uma potência de 2, então colocamos **t2** = 1, caso contrário **t2** = 0.

(Ajuda: Repare que as potências  $2^n$  têm exactamente um bit a 1 na posição  $n$  e todos os restantes bits a 0, por exemplo **0b0...01000** é uma potência de 2. Se subtrairmos 1 obtemos **0b0...00111** onde todos os bits da direita até a posição  $n$  ficaram invertidos. Verifique que isto só acontece com as potências de 2. Concluímos que para testar se  $x$  é uma potência de 2 podemos simplesmente testar a condição  $x \& (x-1) = 0$ .)

8. Usando apenas as instruções **add** e **sub**, escreva um pedaço de código que troque os valores de dois registos **t0** e **t1**, sem usar nenhum registo auxiliar.