

Arquitetura de Computadores

Segunda Lista de Exercícios

Assembly MIPS

Prof. Norton Trevisan Roman

18 de outubro de 2019

1. (2.1) Para o seguinte comando em C, qual o código MIPS correspondente? Assuma que as variáveis `f`, `g`, `h` e `i` estão em registradores.

```
f = g + (h - 5);
```

2. (2.4) Para as instruções MIPS abaixo, qual o comando C correspondente? Assuma que as variáveis `f`, `g`, `h`, `i` e `j` foram associadas aos registradores `$s0` a `$s4`, respectivamente. Assuma que o endereço base dos arranjos `A` e `B` estão nos registradores `$s6` e `$s7`, respectivamente

```
sll $t0, $s0, 2    # $t0 = f * 4
add $t0, $s6, $t0  # $t0 = &A[f]
sll $t1, $s1, 2    # $t1 = g * 4
add $t1, $s7, $t1  # $t1 = &B[g]
lw  $s0, 0($t0)    # f = A[f]
addi $t2, $t0, 4
lw  $t0, 0($t2)
add $t0, $t0, $s0
sw  $t0, 0($t1)
```

3. (2.9) Traduza o seguinte código C para MIPS. Assuma que as variáveis `i` e `j` estão associadas aos registradores `$s3` e `$s4`, respectivamente. Assuma que o endereço base dos arranjos `A` e `B` estão nos registradores `$s6` e `$s7`, respectivamente. Assuma que os elementos dos arranjos são palavras de 4 bytes:

```
B[8] = A[i] + A[j];
```

4. (2.10) Traduza o seguinte código MIPS para C. Assuma que as variáveis `f`, `g`, `h`, `i` e `j` foram associadas aos registradores `$s0` a `$s4`, respectivamente. Assuma que o endereço base dos arranjos `A` e `B` estão nos registradores `$s6` e `$s7`, respectivamente

```
addi $t0, $s6, 4
add  $t1, $s6, $0
sw   $t1, 0($t0)
lw   $t0, 0($t0)
add  $s0, $t1, $t0
```

5. (2.20) Encontre a menor sequência de instruções MIPS para extrair os bits 16 a 11 do registrador `$t0` e usar o valor deste campo para substituir os bits 31 a 26 do registrador `$t1`, sem mudar os outros 26 bits desse registrador.
6. (2.23) Assuma que `$t0` tenha o valor `0x00101000`. Qual o valor de `$t2` após as instruções abaixo?

```

        slt  $t2, $0, $t0
        bne  $t2, $0, ELSE
        j    DONE
ELSE:   addi $t2, $t2, 2
DONE:

```

7. (2.26) Considere o seguinte código MIPS:

```

LOOP:  slt  $t2, $0, $t1
        beq  $t2, $0, DONE
        subi $t1, $t1, 1
        addi $s2, $s2, 2
        j    LOOP
DONE:

```

- (a) Assuma que o registrador `$t1` foi inicializado com 10. Qual é o valor no registrador `$s2` assumindo que `$s2` é inicialmente zero?
- (b) Para o código acima, escreva o código C correspondente. Assuma que os registradores `$s1`, `$s2`, `$t1` e `$t2` são inteiros `A`, `B`, `i` e `temp`, respectivamente.
8. (2.27) Traduza o código abaixo para MIPS. Assuma que os valores de `a`, `b`, `i` e `j` estão nos registradores `$s0`, `$s1`, `$t0` e `$t1`, respectivamente. Além disso, assumo que o registrador `$s2` contém o endereço-base do arranjo `D`:

```

for(i=0; i<a; i++)
    for(j=0; j<b; j++)
        D[4*j] = i + j;

```

9. (2.29) Traduza o seguinte laço em C. Assuma que o inteiro `i` é mantido no registrador `$t1`, que `$s2` contém o inteiro `resultado`, e `$s0` contém o endereço base do inteiro `arranjo`:

```

        addi $t1, $0, 0
LOOP:  lw   $s1, 0($s0)
        add  $s2, $s2, $s1
        addi $s0, $s0, 4
        addi $t1, $t1, 1
        slti $t2, $t1, 100
        bne $t2, $0, LOOP

```

10. Como você faria para determinar se um inteiro é par com apenas 1 instrução MIPS? O resultado dessa instrução deve ser que o registrador `$s1` é 1 se `$s0` for ímpar, e 0 se `$s0` for par. (Dica: pense na representação de pares e ímpares em binário, e veja o que os difere)