
ARQUITETURA DE COMPUTADORES

Prática 4

Resolva os seguintes exercícios usando RISC-V Assembly. Para auxiliar a resolução dos exercícios utilize os guias de consulta rápida disponibilizados na página da UC.

Exercício 1

Considere o seguinte trecho de um programa, colocado em memória a partir do endereço 0x0000 0000. Assuma também que as variáveis declaradas são alocadas na memória a partir do endereço 0x1000 0000.

```
.data
vector1:    .word  2, 10, -27, 4, 13, -7, 5, 11      # address 0x1000 0000
vector2:    .word  3, -5, 14, 0, -10, -2, 21, 15
minimum:    .zero  32

.text
la x12, vector1      # address 0x0000 0000
li x13, 8

proc:
    lw x10, 0(x12)      # (*)
    lw x11, 32(x12)     # (*)
    jal calc_min        # (*)
    sw x10, 64(x12)     # (*)
    addi x12, x12, 4    # (*)
    addi x13, x13, -1   # (*)
    bgt x13, zero, proc # (*)

    li x17, 10
    ecall

calc_min:
    blt x10, x11, cm_end # (*)
    mv x10, x11          # (*)
cm_end:
    ret                  # (*)
```

Indique a codificação das instruções assinaladas com (*).

Exercício 2

Para evitar o aquecimento excessivo de um determinado componente do computador foi decidido instalar um sensor de temperatura e uma ventoinha junto desse componente, que serão utilizados por um programa para realizar o controlo da temperatura.

Para o efeito, existe um temporizador que envia um sinal de interrupção a cada 10 segundos, sendo essa interrupção identificada com o `scause=3`. A temperatura, medida em graus Celsius, com a resolução de 16-bits (com sinal), é obtida através de uma leitura do endereço `0xFFFF 000C`. O funcionamento da ventoinha é definido pelo endereço `0xFFFF 000A`, sendo ativada quando se escreve o valor 1 para este endereço, e desativada quando se escreve o valor 0.

O funcionamento desejado deste controlador é o seguinte:

- se a temperatura for superior a 50°C e a ventoinha estiver desligada, o controlador liga a ventoinha
 - se a temperatura for inferior a 40°C e a ventoinha estiver ligada, o controlador desliga a ventoinha
- a) Escreva o código Assembly da rotina de tratamento de interrupções que deve chamar a rotina `checkTemp` quando `scause=3`.
- b) Escreva o código da rotina `checkTemp`.
- c) Considere o seguinte código em Assembly, armazenado nos endereços de memória indicados:

```
0x50:          ori x11, zero, 3
0x54:          ori x12, zero, 5
0x58:          blt x11, x12, skip_mul15 # <-- Interrupção 3
0x5C:          lsl x13, x14, 4
0x60:          sub x13, x13, x14
0x64:  skip_mul15  add x13, x13, x11
```

Assuma que ocorre uma interrupção, desencadeada pelo temporizador, no instante em que o processador está a executar a instrução `blt`. Indique todas as alterações que ocorrem nos diferentes registos (nomeadamente `scause`, `sepc` e `pc`) de modo a garantir o correto tratamento desta interrupção. Explique também o que acontece aquando do término da rotina de interrupção.