

Parte II

1.3 - Exercícios propostos:

⑥ a) add R2, R2, #2
add R0, R1, R2

b) add R0, R1, R2
sub R0, R0, R3
add R0, R0, R4
sub R0, R0, #30

c) ldr R3, [R7, #16]
add R0, R1, R2
add R0, R0, R3

d) ldr R2, [R8, #40]
(mult. por 4) lsl R2, R2, #2
ldr R3, [R7, R2]
sub R0, R1, R3

e) add R3, R3, #9
lsl R3, R3, #2
sub R0, R1, R2
ldr R4, [R7, R3]
add R0, R0, R4

$$f = g + (j + 2)$$

f → R0
g → R1
j → R2

$$K = a + b - f + d - 30$$

K → R0 f → R3
a → R1 b → R2 d → R4

$$f = g + h + A[4] \quad 4 \times 4 = 16$$

f → R0
g → R1 A[] → R7
h → R2

$$f = g - A[B[10]]$$

f → R0 A[] → R7
g → R1 B[] → R8

$$f = K - g + A[h + 9]$$

f → R0 h → R3
K → R1 A[] → R7
g → R2

f) ldr R2, [R7, #8]
 add R2, R2, #4
 lsl R2, R2, #2
 ldr R3, [R8, R2]
 sub R0, R1, R3

$$b = g - A[B[2] + 4] \quad 2 \times 4 = 8$$

$b \rightarrow R0$ $B[] \rightarrow R7$

$g \rightarrow R1$ $A[] \rightarrow R8$

⑦ a) $b = b + g + h + i + j$

$b \rightarrow R0$ $h \rightarrow R2$
 $g \rightarrow R1$ $i \rightarrow R3$ $j \rightarrow R4$
 $b \rightarrow R0$ $A[] \rightarrow R6$

b) $b = A[1]$

c) $b = A[g-3]$

$A[] \rightarrow R6$ $b \rightarrow R0$
 $g \rightarrow R1$

⑧ a) R0: 0101'0101'0101'0101'0101'0101'0101'0101'

lsl R2, R0, #4 R2: 0101'0101'0101'0101'0101'0101'0101'0101'

R1: 0001'0010'0011'0100'0101'0110'0111'1000'

or R2, R2, R1 R2: 0101'0111'0111'0101'0101'0111'0111'1000'

R: R2 = 0x57755778

b) lsl R2, R0, #4 R2: 0101'0101'0101'0101'0101'0101'0101'0101'

and R2, R2, #-1 R2: 0101'0101'0101'0101'0101'0101'0101'0101'

R: R2 = 0x55555550

e) R0: 0101'0101'0101'0101'0101'0101'0101'0101'

ldr R2, R0, #3 R2: 0000'1010'1010'1010'1010'1010'1010'1010'

and R2, R2, #0xEEF 0xEEF = 0000'0000'1110'1111'

R2: 0000'0000'0000'0000'0000'0000'1010'1010'

R: R2 = 0x000000AA

9 a) movs R2, R1
 rshlmi R2, R1, #0

b) mov R1, #0
 cmp R2, R3
 movgt R1, #1

10 A cada iteração do ciclo L1 será guardado na memória (no endereço dado pelo conteúdo do registro R5) o valor do registro R4. Como em cada ciclo o valor do registro R4 sofre um deslocamento de 4 bits para a esquerda, o seu valor será zero após 8 iterações, pelo que a instrução de salto não será reexecutada, terminando o programa.

Endereço	Valor
0x7D0	0x12345678
0x7D4	0x23456780
0x7D8	0x34567800
0x7DC	0x45678000
0x7E0	0x56780000
0x7E4	0x67800000
0x7E8	0x78000000
0x7EC	0x80000000

11
 mov R0, #0
 ciclo ldrl R1, [R4]
 cmp R1, #0
 beq fim
 add R4, R4, #1
 add R0, R0, #1
 b ciclo

fim

12 a) Fragmento 1: $R2 = 20$

Fragmento 2: $R2 = 200$

b) Fragmento 1: $5 \times N + 2$

Fragmento 2: $(1 + 30 + 2) \times N = 33 \times N$

13

mov R6, #100

mov R0, #1

proximo ldr R1, [R4]
ldr R2, [R5]
cmp R1, R2
beq continua
mov R0, #0
b fim

continua sub R6, R6, #1
beq fim
add R4, R4, #4
add R5, R5, #4
b proximo

fim

14

mov R1, #50

mov R7, #0

ciclo cmp R1, #0
beq fim
ldr R2, [R0]
lsl R2, R2, #31
cmp R2, #0
beq continuar
add R7, R7, #1

continuar add R0, R0, #4
add R1, R1, #-1
b ciclo

15 a)

R0: endereço-base da sequência
R1: número de elementos da sequência

soma mov R2, #0
L1 cmp R1, #0
beq L2
ldr R3, [R0]
add R2, R2, R3
add R0, R0, #4
add R1, R1, #-1
b L1

L2 mov R0, R2
mov PC, LR

b)

R0: endereço-base da sequência
R1: número de elementos da sequência

impar mov R7, #0
L21 cmp R1, #0
beq L23
ldr R9, [R0]
ands R10, R9, #1
beq L22
add R7, R7, #1

L22 add R0, R0, #4
sub R1, R1, #1
b L21

L23 mov R0, R7
mov PC, LR

e) R0: endereço-base da sequência
R1: número de elementos da sequência
R2: valor de comparação

comparar mov R7, #0

L31 cmp R1, #0
beq L33
ldr R9, [R0]
cmp R9, R2
bgt L32
add R7, R7, #1

L32 add R0, R0, #4
sub R1, R1, #1
b L31

d)

R0: endereço-base da sequência 1
R1: endereço-base da sequência 2
R2: número de elementos da sequência

iguais mov R7, #1
L41 cmp R0, #0
beq L43
ldr R9, [R0]
ldr R10, [R1]
cmp R9, R10
bne L42
add R0, R0, #4
add R1, R1, #4
sub R2, R2, #1
b L41

L42 mov R7, #0

L43 mov R0, R7
mov PC, LR