Instituto Tecnológico de Costa Rica

CE 3201 — Taller de Diseño Digital

Prof. Jason Leitón

Tarea: Palíndromo en ARMv4

Shakime Richards Sparks

2018170667

- Algoritmo palíndromo en ARMv4.

```
1           ;           ROLE OF THE REGISTERS
2           ;           r0  = temp register for assignment
3           ;           r1  = initial pointer address
4           ;           r2  = end pointer address
5           ;           r3  = initial pointer value
6           ;           r4  = end pointer value
7           ;           ...
8           ;           r10 = address and value of string length
9           ;           r11 = address of the beginning of the string
10          ;           r12 = result address
11
12          ;           Assigning addresses and values to registers
13 start    mov         r10, #0x150     ; dir string length
14          ldr         r10,[r10]       ; value string length
15          mov         r11, #0x170     ; dir beginning of the string
16          mov         r12, #0x100     ; dir result
17          mov         r1, r11         ; dir initial pointer
18          mov         r2, r11         ; dir end pointer
19          mov         r0, #1          ; ctr (counter)
20
21          ;           Sum sequence to assign the value of the end pointer
22 mul      add         r0,r0,#1        ; ctr += 1
23          add         r2,r2,#4        ; end += 4
24          cmp         r0, r10         ; ctr == length
25          bne         mul
26
27 loop     ldr         r3, [r1]        ; load initial pointer value
28          ldr         r4, [r2]        ; load end pointer value
29          cmp         r3, r4          ; *ini != *end
30          bne         notpal
31
32          ;           Stop condition
33          cmp         r1,r2           ; ini = end         (odd)
34          beq         ispal
35
36          add         r0,r1,#4
37          cmp         r0,r2           ; ini + 4 == end    (even)
38          beq         ispal
39
40          add         r1,r1,#4        ; ini -= 4
41          sub         r2,r2,#4        ; end -= 4
42          b           loop
43
44 notpal   mov         r0,#0x10
45          b           result
46
47 ispal    mov         r0,#0xFF
48 result   str         r0, [r12]
```

- Código binario de las diez primeras instrucciones del algoritmo.

1. e3a0ae15
2. e59aa000
3. e3a0be17
4. e3a0cc01
5. ela0100b
6. ela0200b
7. e3a00001
8. e2800001
9. e2822004
10. e150000a