

Architetture dei Sistemi a Elaborazione – a.a. 2010/11

Esercitazione di Laboratorio 8

1. Si scriva un programma in linguaggio ASM dell'ARM, lo si compili e se ne esegua il debug.
Il programma deve:
 - inizializzare i registri R0-R8 dell'ARM con valori a piacimento
 - considerare coppie di registri adiacenti e confrontarli (R0 con R1, R3 con R4, ecc)
 1. in caso siano uguali, ne calcoli la media, memorizzandola nei registri R9-R12
 2. in caso siano diversi, memorizzi sempre nei registri indicati, la differenza dei due valori al quadrato $(R1-R0)^2$
2. Si scriva un programma in linguaggio ASM dell'ARM, lo si compili e se ne esegua il debug.
Il programma deve:
 - inizializzare i registri R0-R1-R2 dell'ARM con valori a piacimento
 - ordinare i valori, facendo in modo che alla fine R0 contenga il valore più piccolo e R2 quello più grande
 - una volta ordinato il vettore, stabilisca se i 2 numeri più grandi sono multipli di quello più piccolo
 1. in caso positivo, memorizzi in R4 il numero di volte per cui R1 contiene R0 e in R5 il corrispondente per R2
3. Si scriva un programma in linguaggio ASM dell'ARM, lo si compili e se ne esegua il debug.
Il programma deve:
 - considerare una sequenza di 8 valori costanti, memorizzati nel literal pool
 - analizzare tale sequenza, stabilendo se si tratta di una sequenza crescente o decrescente, o non monotona.
 1. In caso sia crescente, calcolare la media matematica degli 8 valori
 2. In caso sia decrescente, determinare valore massimo
 3. Se non monotona, allora indichi il valore massimo e minimo assoluto.

```

; Si scriva un programma in linguaggio ASM dell'ARM, lo si compili e se ne
; esegua il debug.
; Il programma deve:
; - inizializzare i registri R0-R7 dell'ARM con valori a piacimento
; - considerare coppie di registri adiacenti e confrontarli (R0 con R1, R2 con
; R3, ecc)
;     1. in caso siano uguali, ne calcoli il quadrato della somma
;     2. in caso siano diversi, ne calcoli la media

```

```

AREA     RESET, CODE, READONLY
ENTRY

```

```

    ldr lr, =loop

```

```

    mov r0, #1
    mov r1, #2
    mov r2, #3
    ldr r3, =3
    mov r4, #4
    mov r5, #5
    mov r6, #5
    mov r7, #6
    mov r8, #0

```

```

loop

```

```

    cmp r0, r1
    add r0, r0, r1
    muleq r9, r0, r0
    movne r9, r0, lsr #1

    cmp r2, r3
    add r2, r2, r3
    muleq r10, r2, r2
    movne r10, r2, lsr #1

    cmp r4, r5
    add r4, r4, r5
    muleq r11, r4, r4
    movne r11, r4, lsr #1

    cmp r6, r7
    add r6, r6, r7
    muleq r12, r6, r6
    movne r12, r6, lsr #1

    bx lr

end

```

```

; Si scriva un programma in linguaggio ASM dell'ARM, lo si compili e se ne
; esegua il debug.
; Il programma deve:
; - inizializzare i registri R0-R1-R2 dell'ARM con valori a piacimento
; - ordinare i valori, facendo in modo che alla fine R0 contenga il valore più
;   piccolo e R2 quello più grande
; - una volta ordinato il vettore, stabilisca se i 2 numeri più grandi sono
;   multipli di quello più piccolo
;   1. in caso positivo, memorizzi in R4 il numero di volte per cui R1
;      contiene R0 e in R5 il corrispondente per R2

```

```

AREA    RESET, CODE, READONLY
ENTRY

```

```

        mov r0, #2
        mov r1, #4
        mov r2, #7

        cmp r0, r1
        ble jump1    ; r1 < r2

        ; r1 > r2
jump4    cmp r1, r2
        ble jump3    ; r2 < r3
        b label6     ; ldr pc, =label6

jump3    cmp r0, r2
        ble label4
        b label5

jump1    cmp r1, r2
        bge jump2
        b label1

jump2    cmp r0, r2
        ble label2
        b label3

label1   b check_1

label2   ; a, c, b
        mov r3, r1
        mov r1, r2
        mov r2, r3
        b fine

label3   ; c, a, b
        mov r3, r0
        mov r4, r1
        mov r5, r2
        mov r0, r5
        mov r1, r3
        mov r2, r4
        b check_1

label4   ; b, a, c
        mov r3, r0
        mov r0, r1
        mov r1, r3
        b check_1

label5   ; b, c, a
        mov r3, r0

```

```

        mov r4, r1
        mov r5, r2
        mov r0, r4
        mov r1, r5
        mov r2, r3
        b check_1

label6   ; c, b, a
        mov r3, r0
        mov r0, r2
        mov r2, r3

check_1  mov r3, r1
loop1_   subs r3, r3, r0
        beq check_2
        bge loop1_
        ldr r12, =0xffffffff
        b fine

check_2  mov r3, r2
loop2_   subs r3, r3, r0
        beq ok
        bge loop2_
        ldr r12, =0xffffffff
        b fine

ok       ldr r12, =0xaaaaaaaa
fine
        END

```

```

; Si scriva un programma in linguaggio ASM dell'ARM, lo si compili e se ne
; esegua il debug.
; Il programma deve:
; - considerare una sequenza di 8 valori costanti, memorizzati nel literal pool
; - analizzare tale sequenza, stabilendo se si tratta di una sequenza crescente
;   o decrescente, o non monotona.
;   1. In caso sia crescente, calcolare la media matematica degli 8 valori
;   2. In caso sia decrescente, determinare valore massimo
;   3. Se non monotona, allora indichi il valore massimo e minimo assoluto.

```

```

        AREA     RESET, CODE, READONLY
        ENTRY

```

```

                ldr      PC, Reset_Addr

```

```

Reset_Addr      DCD      Reset_Handler

```

```

; Literal pool definition
Pool DCD 8, 2, 6, 5, 6, 10, 1, 3

```

```

; Reset Handler
Reset_Handler

```

```

                ldr r0, =Pool ; adr r0, Pool
                mov r11, #6   ; contatore

```

```

                ; vediamo la monotonia iniziale
                ldrb r1, [r0],#4
                ldrb r2, [r0],#4
                cmp r1, r2
                movle r12, #0
                movge r12, #1

```

```

                ; vediamo se e' mantenuta
loop          mov r1, r2
                ldr r2, [r0],#4
                cmp r1, r2
                ble check_up
                bge check_down
cont          subs r11, #1
                beq si_monotonia
                b loop

```

```

check_up      cmp r12, #0
                bne no_monotonia
                b cont

```

```

check_down    cmp r12, #1
                bne no_monotonia
                b cont

```

```

check_same    cmp r12, #2
                bne no_monotonia
                b cont

```

```

no_monotonia  ; determiniamo il minimo e il massimo assoluto
                ; r11 conterra' il minimo
                ; r12 conterra' il massimo
                ldr r0, =Pool
                mov r10, #6   ; contatore
                ; vediamo il minimo e il massimo temporanei
                ldr r1, [r0],#4
                ldr r2, [r0],#4
                cmp r1, r2
                movle r11, r1

```

```

movle r12, r2
movge r11, r2
movge r12, r1

loop2      ldr r1, [r0],#4
           ; vediamo se e' il nuovo minimo
           cmp r1, r11
           movle r11, r1
           cmp r1, r12
           movge r12, r1
           subs r10, #1
           bne loop2
           b fine

si_monotonia      ; la sequenza e' monotona.
o decrescente     ; mi basta un solo confronto per sapere se e' crescente

           ldr r0, =Pool
           cmp r1, r2
           ble is_cresc
           ; monotonia decrecente, il valore massimo e' il primo
           ; nel caso sia tutti gli stessi, metto il primo valore

comunque         ldr r12, [r0]
                 b fine

is_cresc         mov r11, #7
loop3           ldr r1, [r0],#4
                 ldr r2, [r0],#4
                 add r1, r1, r2
                 subs r11, #1
                 bne loop3
                 mov r12, r1, lsr #3
                 b fine

fine            nop

END

```