

### Esercizio 1

Considerare il seguente frammento di codice.

- a) Indicare il contenuto (in base 10) dei registri x20 e x21 dopo l'esecuzione di ogni singola istruzione:

```
mov x20, #2
lsr x21, x20, #1
sub x20, x20, x21
add x20, x20, x21, lsl #2
```

- b) Eseguire il programma "esercizio1.s" riportato in "Lab 04 - esercizi.zip" per verificare se i valori indicati coincidono con quelli ottenuti. Si noti che il programma fa uso delle due seguenti macro per stampare tramite **printf**, il contenuto dei due registri dopo ogni singola istruzione.

```
.section .rodata
fmt: .asciz "%d\n"

.macro print_x20
    adr x0, fmt
    mov x1, x20
    bl printf
.endm

.macro print_x21
    adr x0, fmt
    mov x1, x21
    bl printf
.endm
```

### Esercizio 2

Considerare il seguente frammento di codice.

- a) Indicare i numeri contenuti nei registri x20 e x21. in base 10 e in base 2 (in complemento a 2), dopo l'esecuzione di ogni singola istruzione:

```
mov x20, #0b00000001
mov x21, #0b00000011
and x21, x21, x20
bic x20, x20, x21
orr x20, x20, x21, lsl #1
orn x20, x20, x21
eor x20, x20, x21, lsl #2
eon x20, x20, x21
```

- b) Eseguire il programma "esercizio2.s" riportato in "Lab 04 - esercizi.zip" per verificare se i valori indicati coincidono con quelli ottenuti. Si noti che il programma fa uso due stesse macro riportate per il punto b) dell'esercizio 1 per stampare tramite **printf**, il contenuto dei due registri dopo ogni singola istruzione.

### Esercizio 3

Supponendo che il registro x0 contenga il byte 01001011 e il registro x1 contenga 00001111, indicare il contenuto del registro x2 in base 10 dopo ciascuna delle seguenti istruzioni. Si indichi inoltre come vengono impostati i flag NZCV di PSTATE.

- a) `adds x2, x1, x0`
- b) `subs x2, x1, x0`

Eseguire il programma “esercizio3.s” riportato in “Lab 04 - esercizi.zip” per verificare se i valori indicati coincidono con quelli ottenuti. Si noti che il programma usa la macro **print** per stampare il contenuto dei flag NZCV di PSTATE.

### Esercizio 4

- a) Partendo dal programma riportato nel file “esercizio4.s”, scrivere un programma che calcoli l’or esclusivo (eor) tra tutti gli elementi dell’array A. Sia N il risultato, il programma deve quindi impostare al valore N l’elemento in posizione 0 di A.

**Esempio.** Essendo  $A = [0, 1, 2, 3]$ , al termine del programma l’array A rimane tale.

- b) Cosa cambierebbe se al posto dell’or esclusivo si usasse quello inclusivo?

### Esercizio 5

- a) Partendo dal frammento di codice riportato in “esercizio5.s”, scrivere un programma che letto da input un numero X, incrementi ogni valore nell’array A di X unità.

**Esempio.** Se X fosse 3, l’array A diventerebbe: [10, 3, 6, 4].

- b) Estendere il programma ottenuto al punto precedente, per fare in modo che, letto un secondo numero da input Y, il programma decrementi ogni valore nell’array A di Y unità.

**Esempio.** Riprendendo l’esempio precedente, se Y fosse 3, l’array A diventerebbe: [7, 0, 3, 1], ovvero ritornerebbe ad avere i valori iniziali.

- c) Estendere ulteriormente il programma ottenuto al punto precedente, per fare in modo che, ogni elemento di A sia moltiplicato per 2: si utilizzi a tal fine un appropriato shift.

**Esempio.** Riprendendo l’esempio precedente, l’array A diventerebbe: [14, 0, 6, 2].

### Esercizio 6

Partendo dal frammento di codice riportato in “esercizio6.s” in cui sono presenti due array di interi A e B, scrivere un programma che modifichi l’array A facendo in modo che ogni elemento in posizione  $i$  nell’array A sia il risultato della somma tra l’intero in posizione  $i$  nell’array A e l’intero in posizione  $i$  nell’array B, ovvero:

$$A[i] = A[i] + B[i]$$

### Esercizio 7

Partendo dal frammento di codice riportato in “esercizio7.s” in cui sono presenti due array di byte A e B, scrivere un programma che modifichi l’array A facendo in modo che ogni byte in posizione  $i$  nell’array A sia il risultato dell’and tra il byte in posizione  $i$  nell’array A e il byte in posizione  $i$  nell’array B, ovvero:

$A[i] = A[i] \text{ and } B[i]$

**(Suggerimento:** dal momento che il programma manipola dei byte, occorre utilizzare opportunamente i modificatori b/sb nelle istruzioni ldr/str ;)