

# Laboratorio 3



R. Ferrero

Politecnico di Torino

Dipartimento di Automatica e Informatica (DAUIN)

Torino - Italy

This work is licensed under the Creative Commons (CC BY-SA) License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/>



# Esercizio 1

Si scriva un programma in Assembly ARM in grado di svolgere le seguenti operazioni:

- rinominare il registro `r1` in `valore`, `r2` in `doppio`, `r3` in `triplo`, `r4` in `quadruplo` e `r5` in `quintuplo`
- assegnare un valore a piacere a `valore`

# Esercizio 1 (cont.)

- usando solo `MOV` e somme, assegnare i seguenti valori ai registri:
  - `doppio = valore * 2`
  - `triplo = valore * 3`
  - `quadruplo = valore * 4`
  - `quintuplo = valore * 5`

# Esercizio 1: suggerimenti

- sfruttare l'inline barrel shifter con MOV
- per la somma  $Rd = Rn1 + Rn2$  usare:  
`ADD Rd, Rn1, Rn2`

# Esercizio 2

Si scriva un programma in Assembly ARM in grado di svolgere le seguenti operazioni:

- allocare 26 byte in un'area di memoria DATA READWRITE, senza inizializzarli
- inizializzare  $r0$  e  $r1$  ad 1
- assegnare ai registri  $r2$ - $r12$  gli elementi della successione di Fibonacci. Ad esempio:
  - $r2 = r1 + r0$
  - $r3 = r2 + r1$

## Esercizio 2 (cont.)

- assegnare a  $r_{14}$  l'indirizzo del primo byte nell'area di memoria allocata al punto 1
- usando la modalità di indirizzamento pre-indexed, salvare il byte meno significativo dei registri  $r_0$ - $r_{12}$ , incrementando  $r_{14}$  ad ogni assegnazione.
- usando la modalità di indirizzamento post-indexed, salvare il byte meno significativo dei registri  $r_{12}$ - $r_0$  (in ordine inverso), incrementando  $r_{14}$  ad ogni assegnazione.

## Esercizio 2 (cont.)

- al termine, verificare che il contenuto della memoria sia il seguente:

01 01 02 03 05 08 0D 15 22 37 59 90 E9 E9 90  
59 37 22 15 0D 08 05 03 02 01 01

# Esercizio 3

Si scriva un programma in Assembly ARM in grado di svolgere le seguenti operazioni:

- definire le seguenti costanti nell'area di codice

```
myConstants DCW 57721, 56649, 15328,  
60606, 51209, 8240, 24310, 42159
```

- allocare 16 byte (4 word) in un'area dati
- considerando le costanti a coppie, scrivere nelle 4 word la somma delle 4 coppie di costanti.