

# Laboratorio 4



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

- L'Instruction set del Cortex-M4 contiene la seguente istruzione:

UADD8 <Rd>, <Rn>, <Rm>

- UADD8 somma i byte corrispondenti di Rn e Rm e salva il risultato in Rd.
- Esempio:  
Rn = 0x 7A 30 45 8D  
Rm = 0x C3 15 9E AA  
Rd = 0x 3D 45 E3 37
- Si noti l'assenza del riporto fra i byte in Rd.

# Esercizio 1 (cont.)

- UADD8 non è presente nell'Instruction Set del Cortex-M3.
- Si scrivano le istruzioni per il Cortex-M3 equivalenti a `UADD8 r4, r0, r1`.

## Esercizio 2

- L'Instruction set del Cortex-M4 contiene la seguente istruzione:

USAD8 <Rd>, <Rn>, <Rm>

- ciascun byte di  $R_n$  e  $R_m$  è inteso come numero in binario puro.
- USAD8 calcola il valore assoluto della differenza fra ciascun byte in  $R_n$  e  $R_m$ .
- successivamente, USAD8 somma i quattro valori assoluti e salva il risultato in  $R_d$ .

## Esercizio 2 (cont.)

- Esempio:  $R_n = 0x\ 7A\ 30\ 45\ 8D$   
 $R_m = 0x\ C3\ 15\ 9E\ AA$

1.  $|0x8D - 0xAA| = 0x1D$
2.  $|0x45 - 0x9E| = 0x59$
3.  $|0x30 - 0x15| = 0x1B$
4.  $|0x7A - 0xC3| = 0x49$

$$R_d = 0x1D + 0x59 + 0x1B + 0x49 = 0xDA$$

- Nota: il valore in  $R_d$  può essere su più di 8 bit

## Esercizio 2 (cont.)

- USAD8 non è presente nell'Instruction Set del Cortex-M3.
- Si scrivano le istruzioni per il Cortex-M3 equivalenti a `USAD8 r5, r0, r1`.

# Esercizio 3

- L'Instruction set del Cortex-M4 contiene le seguenti istruzioni:

SMUAD <Rd>, <Rn>, <Rm>

SMUSD <Rd>, <Rn>, <Rm>

- Entrambe le istruzioni moltiplicano la halfword bassa di  $R_n$  per la halfword bassa di  $R_m$ , e la halfword alta di  $R_n$  per la halfword alta di  $R_m$ .

## Esercizio 3 (cont.)

- Le halfword sono considerate in complemento a due.
- SMUAD somma i due prodotti e salva il risultato in  $R_d$ .
- SMUSD sottrae il prodotto delle halfword alte dal prodotto della halfword basse, e salva il risultato in  $R_d$ .



## Esercizio 3 (cont.)

- Esempio:  $R_n = 0x7A30\ 458D$   
 $R_m = 0xC315\ 9EAA$
- $0x458D * 0x9EAA = 0xE58E35A2$
- $0x7A30 * 0xC315 = 0xE2EC95F0$
- Con SMUAD,  $R_d = 0xC87ACB92$
- Con SMUSD,  $R_d = 0x02A19FB2$

## Esercizio 3 (cont.)

- SMUAD e SMUSD non sono presenti nell'Instruction Set del Cortex-M3.
- Si scrivano le istruzioni per il Cortex-M3 equivalenti a

SMUAD r6, r0, r1

SMUSD r7, r0, r1

## Esercizio 3 (cont.)

- Bisogna estendere il segno delle halfword prima della moltiplicazione.
- Esempio in binario puro:
  - $0x458D = 17805$
  - $0x9EAA = 40618$
  - $0x458D * 0x9EAA = 0x2B1B35A2 = 723.203.490$
- In complemento a due:
  - $0x9EAA = -24918$
  - $0x458D * 0x9EAA = 0xE58235A2 = -443.664.990$

## Esercizio 4 (facoltativo)

- Creare un nuovo progetto selezionando una scheda con il core Cortex-M4, ad esempio NXP LPC4072. Scrivere le istruzioni:

UADD8 r4, r0, r1

USAD8 r5, r0, r1

SMUAD r6, r0, r1

SMUSD r7, r0, r1

- Verificare che i risultati siano coerenti con quanto ottenuto negli esercizi precedenti.