

# Dizionario di Assembler

Riccardo Montagnin

## 1. Load Word

Carica in un dato registro \$R1 la parola che ha come indirizzo il valore di \$R2 sommato a quello di uno shift X.

$$LW \$R1, X(\$R2)$$

Es.

$$LW \$1, 250(\$5) \quad \Rightarrow \quad R1 \leftarrow mem(250 + R5)$$

**N.B.** Il registro \$R1 è **scritto** mentre quello \$R2 è **letto**.

## 2. Store Word

Salva nell'indirizzo di memoria ottenuto dalla somma del valore di \$R2 e quello di uno shift X la parola contenuta nel registro \$R1.

$$SW \$R1, X(\$R2)$$

Es.

$$SW \$1, 250(\$5) \quad \Rightarrow \quad mem(250 + R5) \leftarrow R1$$

**N.B.** I due registri vengono **entrambi letti**.

## 3. Load Byte e Store Byte

**Load Byte (LB)** è simile a **Load Word** ma anzichè caricare una parola intera carica un singolo byte (non cambia niente negli esercizi).

**Store Byte (SB)** è simile a **Store Word** ma anzichè caricare una parola intera carica un singolo byte (non cambia niente negli esercizi).

#### 4. **ADD/SUB**

Somma (o sottrae) i registri \$R2 ed \$R3 e salva il risultato nel registro \$R1.

*ADD \$R1, \$R2, \$R3*

*SUB \$R1, \$R2, \$R3*

Es.

*ADD \$1, \$5, \$8*     $\Rightarrow$      $R1 \leftarrow R5 + R8$

*SUB \$1, \$5, \$8*     $\Rightarrow$      $R1 \leftarrow R5 - R8$

**N.B.** I registri \$R2 ed \$R3 vengono **letti**, mentre il registro \$R1 viene **scritto**.

#### 5. **ADDI/SUBI**

Somma (o sottrae) al registro \$R2 un valore immediato X e salva il risultato nel registro \$R1.

*ADDI \$R1, \$R2, X*

*SUBI \$R1, \$R2, X*

Es.

*ADDI \$1, \$5, 4*     $\Rightarrow$      $R1 \leftarrow R5 + 4$

*SUBI \$1, \$5, 4*     $\Rightarrow$      $R1 \leftarrow R5 - 4$

**N.B.** Il registro \$R2 viene **letto**, mentre il registro \$R1 viene **scritto**.