

# Calcolatori Elettronici

## Esercitazione 1

M. Sonza Reorda – M. Monetti

M. Rebaudengo – R. Ferrero

L. Sterpone – E. Vacca

Politecnico di Torino

Dipartimento di Automatica e Informatica

# Esercitazione 1 - Obiettivi

- Assegnazione di valori a registri e in memoria
- Operazioni aritmetiche: ADD e SUB
  - con segno/senza segno
  - tra due registri o tra registro e immediato
- Istruzioni di input/output
  - lettura di un intero inserito da tastiera
  - stampa a video di interi e stringhe

# Esercizio 1

- Siano definite le seguenti variabili di tipo byte già inizializzate in memoria:
  - `n1: .byte 10`
  - `n2: .byte 0x10`
  - `n3: .byte "1"`
- Sia inoltre definita la variabile di tipo byte, non inizializzata, `res`
- Si calcoli la seguente espressione e si verifichi il risultato:  $res = n1 - n2 + n3$

# Soluzione

```
                                .data
n1:                            .byte 10
n2:                            .byte 0x10
n3:                            .string "1"
res:    .zero 1
                                .text
main:
    lb t0,n1
    lb t1,n2
    sub t0,t0,t1
    lb t1,n3
    add t0,t0,t1
    la s11,res
    sb t0,0(s11)
    addi a7,x0,10
    ecall
```

## Esercizio 2

- Siano definite cinque variabili di tipo byte:

```
var1 = "m", var2 = "i", var3 = "p", var4 = "s",  
var5 = 0
```

- Si scriva un programma che converta in maiuscolo le prime 4 variabili.
- Successivamente, stampare una stringa utilizzando la system call 4 e copiando in `$a0` l'indirizzo di `var1`.
- Quali sono i caratteri stampati a video? A cosa serve `var5`?

# Soluzione

```
                .data
var1:   .byte 0x6D    #"m"
var2:   .byte 0x69    #"i"
var3:   .byte 0x70    #"p"
var4:   .byte 0x73    #"s"
var5:   .byte 0x00
```

```
                .text

main:   li t0,0x41     # A MAIUSCOLO
        li t1,0x61     # a minuscolo
        sub t0,t0,t1   # conversione prima variabile

        la s11,var1
        lb t1,0(s11)
        add t1,t1,t0
        sb t1,0(s11)
```

# Soluzione [cont.]

```
la s11,var2
lb  t1,0(s11)          # conversione seconda variabile
add t1,t1,t0
sb  t1,0(s11)
la s11,var3
lb  t1,0(s11)          # conversione terza variabile
add t1,t1,t0
sb  t1,0(s11)
la s11,var4
lb  t1,0(s11)          # conversione quarta variabile
add t1,t1,t0
sb  t1,0(s11)
la  a0,var1            # stampa
addi a7,x0,4
ecall

addi a7,x0,10
ecall
```

# Esercizio 3

- Siano date le seguenti variabili di tipo byte inizializzate in memoria:
  - `op1: .byte 150`
  - `op2: .byte 100`
- Si stampi a video la somma delle due variabili, utilizzando la system call 1, e si verifichi che il risultato sia corretto.



# Soluzione

```
                .data
op1:    .byte 150    # 2^7 = 128 quindi utilizza bit di peso 7
op2:    .byte 100
```

```
                .text
main:
    #lb t0, op1
    la s11, op1
    lbu t0, 0(s11)

    lb t1, op2
    add a0, t0, t1

    addi a7, x0, 1
    ecall

    addi a7, x0, 10
    ecall
```

## Esercizio 4

- Sia data la seguente variabile di tipo word inizializzata in memoria:

```
var:      .word 0x3FFFFFFF0
```

- Si memorizzi nel registro  $t1$  il doppio del valore di `var` e poi lo si stampi a video.
- Aggiungere a  $t1$  il valore immediato 40 (usando un altro registro come destinazione per non modificare  $t1$ ). Cosa accade? E' possibile stampare un risultato numerico?

# Soluzione

```
                .data
var:            .word 0x3FFFFFF0
mes:            .string "\n"

                .text
main:
    lw  t0,var
        add t1,t0,t0
        mv  a0,t1
        addi a7,x0,1
        ecall
```

# Soluzione [cont.]

```
la    a0,mes  
addi  a7,x0,4  
ecall
```

```
addi  a0,t1,40  
addi  a7,x0,1  
ecall
```

```
addi  a7,x0,10  
ecall
```

# Rappresentazione dei numeri

Complemento a 2		Binario puro
-1	0xFFFFFFFF	4.294.967.296
-2	0xFFFFFFFFE	4.294.967.295
	...	
-2.147.483.640	0x80000008	2.147.483.640
2.147.483.616	0x7FFFFFFE0	2.147.483.616
	...	
1	0x00000001	1
0	0x00000000	0

# Verifica dell'overflow in Ca2

- Sommando 40 a 0x7FFFFFFE0, il risultato ottenuto in complemento a 2 su 32 bit (0x80000008) è in *overflow*
- Risultato non significativo

## Esercizio 5

- Utilizzando la ecall 63, leggere un intero introdotto tramite tastiera e salvarlo in  $t1$ .
- Leggere un altro intero e salvarlo in  $t2$ .
- Senza utilizzare altri registri, scambiare il valore di  $t1$  e  $t2$ .
- Suggerimento: utilizzare istruzioni di somma e sottrazione.

# Soluzione

```
                .data
num1:           .zero 4
num2:           .zero 4
message: .string "\nInserisci un numero: "
```

```
                .bss
buffer: .zero 255
```

```
                .text

main:           la  a0, message                #lettura primo numero
                addi a7,x0,4
                ecall
```



# Soluzione

```
li a7, 63
li a0, 0
la a1, buffer
li a2, 255
ecall
lw a0, 0(a1)
andi a0, a0, 255
li s11, 0x30
sub a0, a0, s11
```

```
mv t1, a0
```

```
la a0, message
addi a7, x0, 4
ecall
```

#lettura secondo numero

# Soluzione

```
li a7, 63
li a0,0
la a1, buffer
li a2,255
ecall
lw a0, 0(a1)
andi a0,a0,255
li s11, 0x30
sub a0,a0,s11
    mv t2,a0
    #scambia t1 e t2 senza usare altri registri
    add t1,t1,t2
    sub t2,t1,t2
    sub t1,t1,t2

    addi a7,x0,10
    ecall
```