

# Calcolatori Elettronici

## Esercitazione 0

M. Sonza Reorda – M. Monetti

M. Rebaudengo – R. Ferrero

L. Sterpone – M. Grosso

Politecnico di Torino

Dipartimento di Automatica e Informatica

# Obiettivi

- Acquisire familiarità con il simulatore QtSpim
- Esempi mirati a :
  - Costruire un programma a partire dal template
  - Dichiarare Variabili
  - Manipolare dati da Memoria a Registro
  - Manipolare dati da Registro a Memoria
  - I/O basico da Console

# Esercizio 1

- Scrittura di un valore in un registro e sua verifica su QtSpim.
- Vengono memorizzati
  - `$t0` valore 10 decimale
  - `$s0` valore DC esadecimale

# Soluzione-1

```
.data
```

```
.text
```

```
.globl main
```

```
.ent main
```

```
main:
```

```
li $t0, 10
```

```
li $s0, 0xdc
```

```
li $v0, 10
```

```
syscall
```

```
.end main
```

## Esercizio 2

- Scrittura di un valore in una cella di memoria
- Dichiarazione della variabile inizializzata con il valore 3 decimale
  - `wVar: .word 3`

# Soluzione-2

```
        .data
wVar:   .word    3

        .text
        .globl main
        .ent main
main:

        li  $t0, 10
        sw  $t0, wVar

        li  $v0, 10
        syscall

        .end main
```

# Esercizio 3

- Somma di 2 valori contenuti in due variabili e memorizzazione risultato in una variabile Risultato
- I due operandi
  - wOpd1: .word 10
  - wOpd2: .word 24
- La variabile Risultato
  - wResult: .space 4

# Soluzione-3

```
        .data
wOpd1:  .word   10
wOpd2:  .word   24
wResult: .space  4
        .text
        .globl main
        .ent main
main:
        lw  $t0, wOpd1
        lw  $t1, wOpd2
        add $t2, $t1, $t0
        sw  $t2, wResult
        li  $v0, 10
        syscall
        .end main
```



# Esercizio 4

- Somma degli elementi di un Vettore (I)
  - `wVett: .word 5, 7, 3, 4,`
- Risultato in una variabile Risultato
  - `wResult: .space 4`
- Tecnica molto semplice fatta di somme successive con utilizzo di un Registro come accumulatore.
  - `$t1`     ACCUMULATORE
  - `$t0`     INDIRIZZO Vettore
  - `$t2`     Secondo OPERANDO

# Soluzione-4

```
                .data

wVett:          .word  5, 7, 3, 4, 3
wResult:        .space 4

                .text
                .globl main
                .ent main

main:
    li    $t1, 0
    la    $t0, wVett
    lw    $t2, ($t0)      # 0
    add   $t1, $t1, $t2
    add   $t0, $t0, 4      # 1
```

# Soluzione-4 [cont.]

```
lw    $t2, ($t0)
add   $t1, $t1, $t2
add   $t0, $t0, 4      # 2
lw    $t2, ($t0)
add   $t1, $t1, $t2
add   $t0, $t0, 4      # 3
lw    $t2, ($t0)
add   $t1, $t1, $t2
add   $t0, $t0, 4      # 4
lw    $t2, ($t0)
add   $t1, $t1, $t2
sw    $t1, wResult
```

```
li $v0, 10
syscall
.end main
```

# Esercizio 5

- Somma degli elementi di un Vettore (II) con un loop
- Registri
  - \$t0. Indirizzo Vettore
  - \$t1. Accumulatore
  - \$t2. Temporaneo
  - \$t3. Contatore

# Soluzione-5

```
.data
DIM=15
wVett:    .word  2, 5, 16, 12, 34, 7, 20, 11, 31, 44, 70, 69, 2, 4, 23
wResult:  .space  4

.text
.globl main
.ent main

main:

    li    $t1, 0
    li    $t3, DIM
    la    $t0, wVett
```

# Soluzione-5 [cont.]

```
ciclo:    lw    $t2, ($t0)
          add   $t1, $t1, $t2
          add   $t0, $t0, 4      #  ++
          sub   $t3, $t3, 1
          beq   $t3, 0, fine
          j     ciclo
```

```
fine:     sw    $t1, wResult

          li    $v0, 10
          syscall
          .end main
```

# Esercizio 6

- Lettura da tastiera e visualizzazione a video di un vettore di 5 caratteri
- Registri
  - \$t0.      Indirizzo Vettore
  - \$t1.      Contatore
  - Syscall (\$v0=**1**).Print integer(\$a0 = integer)
  - Syscall (\$v0=**4**).Print string (\$a0 = string)
  - Syscall (\$v0=**5**).Get integer (\$v0 = integer)

# Esercizio 6

- Tipo **asciiz** - NULL terminated ASCII string
  - `.asciiz "Inserire numeri\n"`
- Il NULL è un carattere ASCII non stampabile e viene utilizzato per contrassegnare la fine della stringa. La terminazione NULL è standard ed è richiesta dal servizio di sistema della stringa di stampa (per funzionare correttamente).



# Esercizio 6

- Get Integer, si chiude con un ENTER
- Non viene eseguito nessun controllo su tipo di carattere inserito ( ovvero se corrisponde ad una cifra numerica)

- `li $v0, 5`

- `syscall #(result in $v0)`

- `sw $v0, ($t0)`

# Soluzione-6

```
                .data
                DIM=4
wRes:           .space    20
message_in :    .asciiz   "Inserire numeri\n"
message_out :  .asciiz   "Numeri inseriti\n"
space :        .ascii    " ; "

                .text
                .globl main
                .ent main

main:
                la $a0, message_in
                li $v0, 4          # call code, Print string
                syscall

                la $t0, wRes
                li $t1, 0
```

# Soluzione-6 [cont.]

```
uno:      li $v0, 5          # call code, Read integer
          syscall          # system call (result in $v0)
          sw $v0, ($t0)
          beq $t1, DIM, print_num
          add $t1, $t1, 1
          add $t0, $t0, 4
          j uno

print_num: la $a0, message_out
          li $v0, 4          # call code, Print string
          syscall
          la $t0, wRes
          li $t1, 0          # contatore
ciclo_print: li $v0, 1       # call code, Print int
          lw $a0, ($t0)      # value for int to print
          syscall           # system call
```

# Soluzione-6 [cont.]

```
beq    $t1, DIM, fine
la     $a0, space
li     $v0, 4                      # call code, Print string
syscall
```

```
add    $t1, $t1, 1
add    $t0, $t0, 4
j      ciclo_print
```

```
fine:   nop
```

```
li     $v0, 10
syscall
```

```
.end main
```

# Esercizio 7

- Ricerca del carattere minimo, vengono inseriti da tastiera DIM valori, si calcola il minimo e si visualizza
- Registri
  - \$t0.    Indirizzo Vettore
  - \$t1.    Contatore
  - \$t2.    Valore Minimo
  - \$t3.    Temporaneo

# Soluzione-7

```
.data
    DIM=4
wVet:    .space    20
wRes:    .space    4
message_in : .asciiiz  "Inserire numeri\n"
message_out: .ascii    "Valore Minimo : "

    .text
    .globl main
    .ent main

main:

    la $t0, wVet
    li $t1, 0                # contatore

    la $a0, message_in
    li $v0, 4
    syscall
```

# Soluzione-7 [cont.]

```
uno:      li  $v0, 5                # call code, Read integer
          syscall                  # system call (result in $v0)
          sw  $v0, ($t0)
          beq $t1, DIM, calc
          add $t1, $t1, 1
          add $t0, $t0, 4
          j   uno
```

```
calc:     la  $t0, wVet
          li  $t1, 0                # contatore
          lw  $t2, ($t0)            # $t2 memorizzo MIN
```

```
loop_min: beq $t1, DIM, print_num
          lw  $t3, ($t0)
          blt $t3, $t2, change_min
          add $t1, $t1, 1
          add $t0, $t0, 4
          j   loop_min
```

# Soluzione-7 [cont.]

```
change_min:  lw  $t2, ($t0)
              j   loop_min

print_num:   la  $a0, message_out    # addr of NULL
              li  $v0, 4              # call code, print
              syscall                # system call

              li  $v0, 1              # call code, print int
              move $a0, $t2           # value for int to print
              syscall                # system call

fine:        nop

              li  $v0, 10
              syscall
              .end main
```