

Architettura degli Elaboratori

Corso A

Lab 3

Operandi Allocati in Memoria (load/store)

Esercizio 0 - Media interi

Si scriva un programma in linguaggio RISC-V che carichi 4 numeri interi presente nella memoria in word contigue e calcoli il valore intero della loro media aritmetica (arrotondamento per difetto). Il valore calcolato va salvato in un'ulteriore posizione della memoria contigua a quelle usate per il calcolo.

- **In questo esercizio, utilizzare soltanto il set delle istruzioni "intere di base rv64i".**

Istruzioni per Prendere Decisioni

Obiettivi

- Imparare come un codice C/Java viene tradotto in RISC-V
- Tradurre le istruzioni per prendere decisioni
 - if () then else
 - while ()
 - for()
 - do...while()
- Utilizzare il simulatore per misurare il numero di istruzioni RISC-V eseguite per completare ogni esercizio

Useremo solo i nomi da adesso in poi

RISC-V - I registri

Registro	Nome	Utilizzo
x0	zero	La costante 0
x1	ra	Indirizzo di ritorno
x2	sp	Puntatore a stack
x3	gp	Puntatore globale
x4	tp	Puntatore a thread
x5-x7	t0-t2	Temporanei
x8	s0/_fp	Salvato/puntatore a frame
x9	s1	Salvato
x10-x11	a0-a1	Argomenti di funzione/valori restituiti
x12-x17	a2-a7	Argomenti di funzione
x18-x27	s2-s11	Registri salvati
x28-x31	t3-t6	Temporanei

Tipo delle istruzioni

Istruzione (R)	funz7	rs2	rs1	funz3	rd	codop	Esempio
add	0000000	00011	00010	000	00001	0110011	add x1, x2, x3
sub (sottrazione)	0100000	00011	00010	000	00001	0110011	sub x1, x2, x3
Istruzione (I)	immediato		rs1	funz3	rd	codop	Esempio
addi (addizione immediata)	001111101000		00010	000	00001	0010011	addi x1,x2,1000
ld (caricamento di parola doppia)	001111101000		00010	011	00001	0000011	ld x1, 1000 (x2)
Istruzione (S)	Immediato	rs2	rs1	funz3	immediato	codop	Esempio
sd (memorizzazione di parola doppia)	0011111	00001	00010	011	01000	0100011	sd x1, 1000 (x2)

Tipo delle istruzioni

Tipo S	sb	0100011	000	n.a.
	sh	0100011	001	n.a.
	sw	0100011	010	n.a.
	sd	0100011	111	n.a.
Tipo SB	beq	1100111	000	n.a.
	bne	1100111	001	n.a.
	blt	1100111	100	n.a.
	bge	1100111	101	n.a.
	bltu	1100111	110	n.a.
	bgeu	1100111	111	n.a.

Esercizio 1 - if ... then

Scrivere le sequenze di istruzioni RISC-V corrispondente ai seguenti frammenti di pseudocodice. Si supponga che le variabili **x**, **y** siano contenute rispettivamente nei registri **t0**, **t1**.

Frammento 1

```
x = x - y
if (x < 0)
    x = 0
y = y - 1
```

Frammento 2

```
x = (x - 2) + y
if (x < y)
    x = x + 1
else
    y = y + 1
```

Esercizio 2 - Max

Si scriva un programma in linguaggio RISC-V che carichi tre numeri interi su **t0**, **t1** e **t2**, e poi inserisca il valore massimo tra i tre nel registro **t3**.

Esercizio 3 - Fibonacci Iterativo

Considerando il seguente frammento di codice che ritorna l'N-esimo numero della sequenza di Fibonacci - Fib(n) - scrivere l'equivalente in RISC-V. Assumere che la variabile N sia memorizzata nel registro t0. Il risultato finale (variabile R) va lasciato nel registro t1. Si utilizzino altri registri temporanei per le variabili A e B, e il minor numero possibile di istruzioni.

```
int N = 8;
int R = 1;
int A = 0; int B = 1;
while (N > 0) {
    R = A + B;
    A = B;
    B = R;
    N = N - 1;
}
```

- Quante istruzioni RISC-V sono necessarie per realizzare il frammento di codice C?
- Quante istruzioni RISC-V verranno eseguite per completare il ciclo quando N=8?

Esercizio 4 - Quadrati perfetti

Si scriva un programma RISC-V che calcoli la somma dei primi **N quadrati perfetti**. Il programma deve assumere che N sia nel registro **t1** e stampare a schermo la somma ottenuta.

- Quante istruzioni RISC-V sono necessarie?
- Quante istruzioni RISC-V verranno eseguite quando $N=10$?

Esercizio 4 - Quadrati perfetti

Soluzione possible in C

```
int N=10;  
int S=0;  
int i;  
for (i=1; i<=N; ++i) {  
    S = S + i*i;  
}
```

Esercizio 5 - contauno

Scrivere il codice RISC-V che restituisce il numero di bit uguali a 1 contenuti nel valore binario presente nel registro **t0**. Per esempio, se **t0** ha il valore binario equivalente al numero intero 37, il risultato atteso è 3.

Suggerimento: usare opportunamente le istruzioni logiche **and**, **srl** ...

- Quante istruzioni RISC-V sono necessarie?
- Quante istruzioni RISC-V verranno eseguite quando $t0=37$?

Esercizio 6 - Cicli FOR Annidati

Tradurre il seguente frammento di codice C in codice assembly RISC-V. Si utilizzi il minor numero possibile di istruzioni. Si supponga che le variabili a, b e R siano contenute rispettivamente nei registri t0, t1, t2

```
for (i=0; i<a; i++) {  
    for (j=0; j<b; j++) {  
        R = 2*R + i + j;  
    }  
}
```

- Quante istruzioni RISC-V sono necessarie per realizzare il frammento di codice?
- Supponendo che le variabili a e b vengono inizializzate a 10 e 5, quante istruzioni RISC-V verranno eseguite per completare il ciclo?