

Esercizi

Corso di Architettura degli Elaboratori (A.A. 2021/22)
Tutor: Dall'Occo Francesco (francesco.dallocco@unife.it)

Ripasso: Branch

In assembler spesso bisogna ragionare «al contrario», quando si implementano gli IF.

Se la condizione viene verificata, si effettua il salto **fuori** dall'IF (quindi non si esegue il codice dentro all'IF)

Spesso è necessario definire il controllo al contrario:

Es: if($a \leq 0$): $a = 2 * a$

Dovrò effettuare il salto (e quindi *non eseguire la funzione all'interno*) solo se $a > 0$

If (senza else)

[...]

lw \$t0,a

bgtz \$t0, ifNonVerificato

sll \$t0,\$t0,1

ifNonVerificato:

#il resto del programma

Ripasso: Branch

If-then-else

```
[...]
bne $t0,$t1,else
    #istruzioni da eseguire dentro all'if
    # ovvero se la condizione di salto NON è
    verificata
    j out #esco dall'if
else:
    #istruzioni da eseguire nell'else
out:
    #Il resto del programma
[...]
```

While loop

```
[...]
Startloop:
    #istruzioni da eseguire dentro al ciclo
    bne $t0,$t1, Exitloop # condizione di uscita
    dal ciclo
    #se la condizione precedente non viene
    verificata, salto all'inizio del ciclo
    j Startloop
Exitloop:
    #codice da eseguire usciti dal ciclo
[...]
```

Ripasso: Branch

Il mips implementa solo i comandi di branch riportati in tabella.

Per effettuare altri confronti (es. è \$t0 minore di \$t1?) è necessario usare questi comandi in congiunzione con **slt**

Beq	Branch if equal
Bne	Branch if NOT equal
Bgtz	Branch if greater than zero
Bltz	Branch if less than zero
Bgez	Branch if greater or equal to zero
Bleq	Branch if less or equal to zero

Esercizio: valore assoluto

Vogliamo calcolare il valore assoluto di una variabile intera (32 bit) salvata in memoria, e stamparlo a schermo

- Se il valore è positivo, dobbiamo solo stamparla
- Se il valore è negativo, dobbiamo effettuare il complemento a 1 e poi aggiungere 1

Provate ad eseguire l'esercizio: se non ce la fate, nella prossima slide ci sono dei suggerimenti

Esercizio: valore assoluto

Suggerimento 1:

- Lavorando con gli IF, non è sempre banale decidere la condizione che porta ad effettuare il salto.
- Devo entrare nell'IF se il numero è positivo? O se il numero è negativo?

Suggerimento 2:

- Per ottenere il complemento a 1 di un bit, possiamo fare lo xor con 1.
- Per ottenere il complemento a 1 di una parola di 32 bit, dobbiamo fare lo xor bit a bit con una parola di soli 1.

Esercizio: successione di Fibonacci

Vogliamo calcolare l'n-esimo valore della successione di Fibonacci.

La successione di Fibonacci è definita in questo modo:

- $F_i = F_{i-1} + F_{i-2}$
- $F_0 = 0$
- $F_1 = 1$

Usate un ciclo per calcolare e stampare a schermo F_{15}

Esercizio: successione di Fibonacci

Suggerimento 1:

- Di fatto dovete implementare un ciclo FOR, e stampare a schermo il risultato una volta usciti dal ciclo
- Un FOR altri non è che un WHILE con un incremento subito prima del jump

Suggerimento 2:

- Ad ogni ciclo, dovrete aggiornare i valori dei registri che contengono F_i, F_{i-1}, F_{i-2}
- Per aiutarvi a riconoscerli (e non confonderli) potete usare la direttiva .eqv

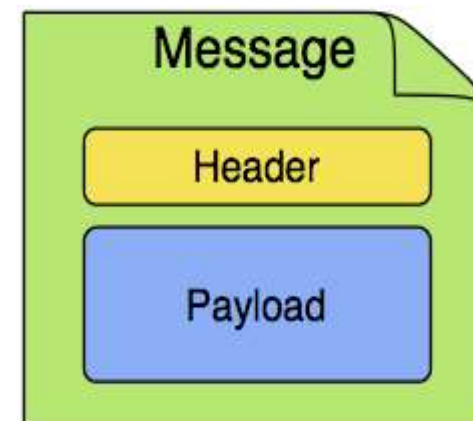
Suggerimento 3:

- Fate molta attenzione all'ordine con cui aggiornate i registri

Esercizio: Il pacchetto da tradurre

La comunicazione fra due entità (es: client con il server, chip con chip, microcontrollore-sensore/trasduttore...) è in genere molto lenta, e spesso porta a dei rallentamenti nelle operazioni.

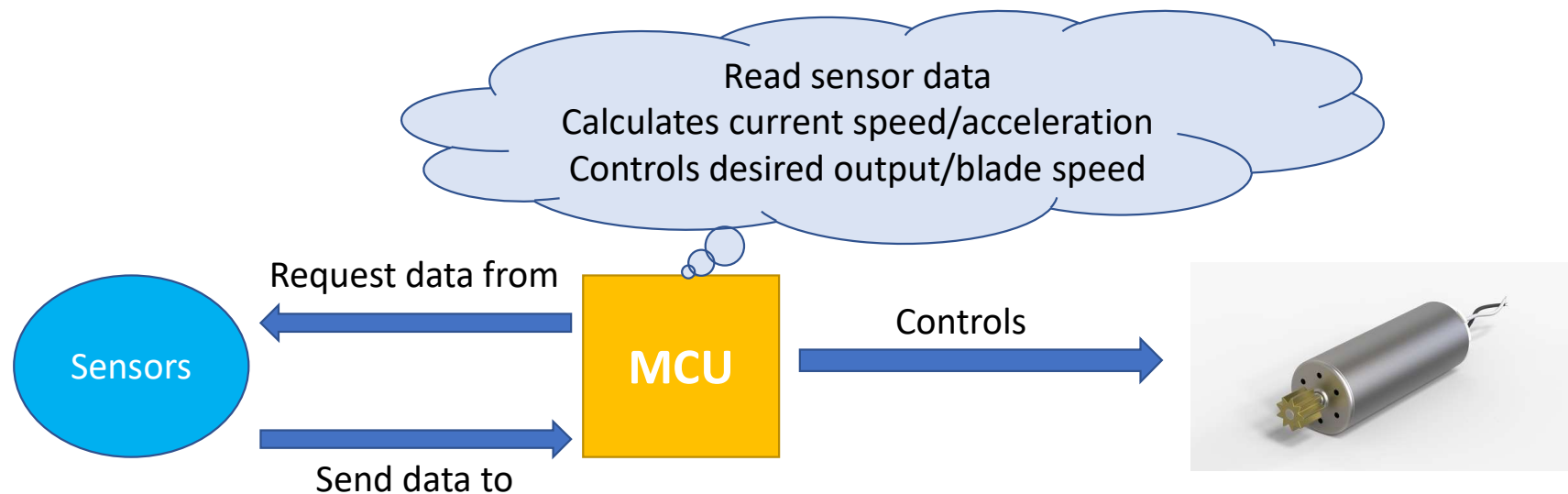
Poiché in genere i pacchetti hanno una *dimensione massima limitata*, per ridurre il numero di comunicazioni da effettuare e risparmiare un po' di tempo, si cerca di **compattare quante più informazioni** nel **minor numero di byte** possibile.



Esercizio: Il pacchetto da tradurre

Supponiamo di avere un drone, controllato da un microcontrollore e dotato di diversi sensori per la navigazione.

Il microcontrollore deve essere in grado di tradurre i pacchetti ricevuti da questi sensori, in modo da poter gestire al meglio il volo



Esercizio: Il pacchetto da tradurre

In particolare, vogliamo far comunicare il microcontrollore (es. arduino) con un accelerometro, montato sul drone:

Dal datasheet del sensore, vediamo che ci invia dei pacchetti a 32 bit in questo formato:

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
r/w	num1	num2	num3	cont	pz	py	px	z-axis								y-axis								x-axis							

Dove:

X-axis: intero con segno a 8 bit, che indica l'accelerazione misurata sull'asse x, in dG (decimi dell'accelerazione di gravità)

Y-axis: accelerazione sull'asse y, in dG

Z-axis: accelerazione sull'asse z, in dG

r/w, num1-3, cont: segnali di controllo per la comunicazione

px: bit di parità per il valore di accelerazione su x
(py e pz: analogo a px, ma per y-axis e z-axis)

(il bit di parità è spesso necessario perché, durante la comunicazione, possono esserci dei disturbi che alterano il valore del payload)

Esercizio: Il pacchetto da tradurre

Note le specifiche della slide precedente, vogliamo effettuare un controllo sull'accelerazione del drone.

Vogliamo quindi:

- Estrarre il valore dell'accelerazione sull'asse z
- Assicurarsi che l'accelerazione sia compresa fra -20dG e +20dG:
 - in tal caso, stampare a schermo il valore 0
- Se così non fosse, il drone rischia di danneggiarsi gravemente in caso di collisione, e bisogna prendere provvedimenti.
 - Scrivere nella variabile «desired_acceleration» lo scarto fra l'accelerazione misurata e il range «safe» di operabilità, e stampare questo valore a schermo.
 - Es, se l'accelerazione misurata è 33, dovremo stampare -13 a schermo ($20-33=-13$)
 - Es, se l'accelerazione misurata è -82, dovremo stampare 62 a schermo ($-20+82=62$)

Esercizio: Il pacchetto da tradurre

Il pacchetto ricevuto sarà una word che dovrete dichiarare nell'area dati del codice

Inizializzate quella word con i seguenti valori, ed eseguite il codice:

- 0x00D4F614
- 0x01562210
- 0x000F4499
- 0x00F6DD11

Esercizio: Il pacchetto da tradurre

Suggerimento 1:

- Questo problema, nel suo complesso, è molto difficile da risolvere. Vi conviene dividere il macroproblema in più diversi sottoproblemi

Esercizio: Il pacchetto da tradurre

Suggerimento 2:

Il problema può essere diviso in questo modo:

1. Estrazione del valore di accelerazione
2. Verifica del valore di accelerazione
 1. Verifica che l'accelerazione sia minore di +20
 - In caso contrario, prendere provvedimenti
 2. Verifica che l'accelerazione sia maggiore di -20
 - In caso contrario, prendere provvedimenti
 3. Saltare alla stampa a schermo
3. Stampa a schermo

Esercizio: Il pacchetto da tradurre

Suggerimento 3:

Riformuliamo le operazioni del suggerimento 2

1. Estrazione del valore di accelerazione
2. Verifica del valore di accelerazione
 1. Verifica che l'accelerazione sia minore di +20
 1. Se è valido, saltare al prossimo controllo
 2. Se non è valido, prendere provvedimenti
 3. Stampare a schermo e uscire
 2. Verifica che l'accelerazione sia maggiore di -20
 1. Se è valido, saltare al caso in cui tutto è corretto
 2. Se non è valido, prendere provvedimenti
 3. Stampare a schermo e uscire
 3. Caso in cui tutto è corretto
 - Stampare 0 e uscire