

1 Domanda I

La funzione `sumV` prende in ingresso l'indirizzo di tre array di interi `u`, `v`, `z`, e la dimensione `size` degli array, il suo compito è quello di effettuare operazioni sugli elementi di `u` e `v` e di salvare il risultato in `z`. Il compilatore fornisce la seguente traduzione in assembly ARM che è incompleta: le righe X1 e X2 sono omesse. (Sono rispettate le convenzioni di chiamata per gli argomenti e per il valore di ritorno specificate dall'ABI vista durante il corso) Quale delle coppie X1 e X2 proposte corrisponde alle righe corrette?

```
void sumV(int * u, int * v, int* z, unsigned int size) {
    for (int i = 0; i < size - 1; i++) {
        *(z+i) = *(v+i) + *(u+i);
    }
    return;
}

X1
beq    .L9
sub    r12, r1, 4
subs   r1, r1, 8
stmdf  sp!, {r11, lr}
subs   r0, r0, #4
add    lr, r1, r3, lsl #2
subs   r2, r2, #4

.L3:
ldr    r3, [r12, #4]!
ldr    r1, [r0, #4]!
cmp    r12, lr
X2
str    r3, [r2, #4]!
bne   .L3
ldmfd sp!, {r11, pc}

.L9:
bx     lr

X1:
cmp   r3, #1
■ X2:
add   r3, r3, r1

X1:
□ X2:
cmn   r3, #1
add   r3, r3, r1

X1:
□ X2:
cmp   r3, #1
add   r3, r3, r0

X1:
□ X2:
cmn   r3, #1
add   r3, r3, r0

□ None of the answers
```

1.1 Risposta

Dobbiamo individuare i pezzi di codice mancanti nella traduzione in assembly della funzione `sumV`. La seconda istruzione è una `beq` che punta a `.L9`, che corrisponde all'uscita dalla funzione. Bisogna quindi uscire dalla funzione se `size` è uguale ad 1. Per questo motivo, la prima istruzione deve essere una `cmp`, quindi possiamo escludere la seconda e la quarta risposta.

Ora dobbiamo individuare l'ultima istruzione mancante. Dalla traduzione incompleta possiamo osservare che i valori caricati dalle variabili in memoria sono contenuti nei registri r1 ed r3, per questo motivo possiamo escludere la terza risposta, che usa r0 come operando.

Di conseguenza, la risposta corretta è la prima.

2 Domanda II

Qual è il risultato della seguente compilazione usando il compilatore gcc
 gcc hello_world.c -o a.out -O2

- Genera il file eseguibile a.out
- Genera il file assembly hello_world.s
- Genera il file oggetto hello_world.o
- Genera il file preprocessato hello_world.i
- Nessuna delle altre risposte.

2.1 Risposta

Il comando eseguito invoca la compilazione del sorgente hello_world.c. Tra gli argomenti viene specificata l'opzione -o a.out, che genera il file eseguibile a.out. Di conseguenza, la risposta corretta è “Genera il file eseguibile a.out”.

3 Domanda III

Quale delle seguenti affermazioni è FALSA:

- Nei circuiti combinatori, l'uscita dipende dallo stato
- Nei circuiti sequenziali, l'uscita dipende dallo stato
- Nei circuiti combinatori, l'uscita dipende dall'ingresso
- Nei circuiti sequenziali, l'uscita dipende dall'ingresso
- Tutte le risposte

3.1 Risposta

Dobbiamo trovare l'affermazione falsa tra quelle proposte. Dato che l'uscita di un circuito combinatorio dipende solo dall'ingresso, mentre in un circuito sequenziale l'uscita dipende sia dall'ingresso che dallo stato, la risposta corretta è “Nei circuiti combinatori, l'uscita dipende dallo stato”

4 Domanda IV

Quale delle seguenti alternative rappresenta un esempio di hazard sui dati?

- and t0, t1, t3
or t2, t4, t5
- or t0, t1, t2
add t2, t4, t5
- add t0, t1, t2
and t3, t1, t5
- sub t0, t1, t2
add t3, t4, t0
- Nessuna delle altre risposte.

4.1 Risposta

Dobbiamo individuare la sequenza di istruzioni che causa un hazard sui dati. Nella prima risposta, non c'è nessun hazard sui dati, poiché nessun registro utilizzato nella prima istruzione compare nella seconda. Nella seconda risposta, t_2 viene usato in entrambe le istruzioni, ma nella prima istruzione è usato come registro sorgente, quindi questo non costituisce un hazard. Lo stesso ragionamento si applica anche alla terza risposta. Nella quarta, invece, osserviamo che t_0 è usato come registro destinazione nella prima istruzione, mentre nella seconda è utilizzato come sorgente. Questa condizione costituisce un hazard sui dati.

Di conseguenza, la risposta corretta è la quarta.

5 Domanda V

Quale delle seguenti risposte rappresenta una gerarchia di memoria ordinata per distanza (crescente) dal processore?

- Registri - Cache - RAM - Hard Disk
- Cache - Registri - RAM - Hard Disk
- Registri - RAM - Cache - Hard Disk
- Registri - Cache - Hard Disk - RAM
- Nessuna delle risposte

5.1 Risposta

Ordinando i vari elementi per distanza dal processore otteniamo: registri (molto veloci, molto costosi); cache (veloce, costosa); RAM (meno veloce, meno costosa); Hard Disk (lento, economico). Quindi, la risposta corretta è “Registri - Cache - RAM - Hard Disk”.

6 Domanda VI

Quale delle seguenti affermazioni riguardanti le traps e le interruzioni è vera?

- sono entrambi causati da eventi esterni e sono asincroni
- sono entrambi causati da eventi esterni e sono sincroni all'esecuzione del programma
- sono entrambi causati da eventi interni e sono asincroni
- sono entrambi causati da eventi interni e sono sincroni all'esecuzione del programma
- Nessuna delle altre risposte

6.1 Risposta

Analizziamo le varie affermazioni.

Nella prima, si afferma che sia le traps che le interruzioni dipendono da eventi esterni. Questa affermazione è falsa perché le traps sono generate da eventi interni, mentre le interruzioni sono generate da eventi esterni. Lo stesso ragionamento si applica per la seconda affermazione. La terza e la quarta affermazione, invece, affermano che siano entrambe generate da eventi interni, il che rende anche queste affermazioni false. Di conseguenza, la risposta corretta è la quinta.