

Prova scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri

10 Giugno 2005

1. Si progetti il grafo degli stati di una macchina di Mealy minima con tre stati di ingresso (0, 1 e 2) e quattro stati di uscita: *A*, *B*, *C*, *D*. Chiamiamo *S* la somma di tutti gli stati di ingresso ricevuti fino ad un certo istante. In ogni istante, la macchina produce:
 - *A* se *S* è pari e non è multiplo di 3 (ad es. 2, 4, ...);
 - *B* se *S* è pari e multiplo di 3 (ad es. 6. Si consideri "0" come pari e multiplo di 3);
 - *C* se *S* è dispari, ma non è multiplo di 3 (ad es. 1, 5, 7, ...);
 - *D* se *S* è dispari e multiplo di 3 (ad es. 3, 9, ...).
2. Supponiamo di avere un programma scritto in parte in Assembler e in parte in C++ (*v1* e *v2* sono vettori di 100 interi):

```
.data
m:          c0:      movl $0, m
            .long    cpl $100, %ecx
            .long    jl c1
v1:         .long    jmp fine
            .long    c1:      pushl (%esi, %ecx, 4)
            ...      pushl (%edi, %ecx, 4)
v2:         .long    call f1
            .long    addl $8, %esp
            ...      addl %eax, m
.text
main:      movl $0, %ecx
            movl $v1, %esi
            movl $v2, %edi
            fine:    jmp c0
            ret
```

```
int f1(int a, int b)
{
    int c = 0; int i;
    for (i = 0; i < b; i++)
        c += a;
    return c;
}
```

- (a) Dire quale sarà il contenuto della locazione di memoria *m* quando l'esecuzione raggiunge l'istruzione con etichetta *fine*;
- (b) Tradurre la funzione *f1* in Assembler.

3. Consideriamo il seguente script per la shell `sh` di Unix:

```
mkdir 'seq $1'
for D in *[13579]
do
    touch $D/f
    cd 'expr $D + 1'
    ln -s ../$D/f f
    cd ..
done
for F in */f
do
    ls -l $F
done | nl | sort -n
```

Dire quale sarà l'output dello script, eseguito dall'utente "utente1", con gruppo principale "gruppo1", con parametro "6", il 10 Giugno alle 13.00, in una directory precedentemente vuota.

4. Supponiamo di avere una cache a indirizzamento diretto, con 4 linee di 8 byte ciascuna. La DRAM del sistema è indirizzabile al byte. Il programma in esecuzione nel sistema produce la sequenza di accessi ai byte di indirizzo:

9, 20, 49, 17, 30 (ripetuta 1000 volte)

Supponiamo che tutto il tempo di esecuzione del programma sia la somma dei tempi di accesso ai byte e che, in caso di *hit*, l'accesso duri 10 ns (10 nanosecondi), mentre in caso di *miss* l'accesso duri 200 ns.

- (a) Calcolare il tempo di esecuzione del programma.
- (b) Per ridurre il tempo di esecuzione di questo programma, è meglio, a parità di costo, sostituire la cache a indirizzamento diretto con una associativa (a 2 vie e di pari capacità) oppure sostituire la DRAM in modo che il tempo di accesso in caso di *miss* diventi 100 ns? perché?