

Esame **A** di Architetture degli Elaboratori

Soluzione

A.A. 2018-19 — III appello — 2 luglio 2019

N.B.: il punteggio associato ad ogni domanda è solo una misura della difficoltà, e peso, di ogni domanda. Per calcolare il voto complessivo bisogna normalizzare a 32.

1. Si indichi se i seguenti numeri hanno senso sì o no. Se la risposta è no si spieghi brevemente il motivo:

3407.19_8	$AA.\bar{A}_{11}$	$-0.\overline{101}_2$	$\bar{3}.456_7$	0_0	A_B
-------------	-------------------	-----------------------	-----------------	-------	-------

R: (3 pt)

3407.19_8	$AA.\bar{A}_{11}$	$-0.\overline{101}_2$	$\bar{3}.456_7$	0_0	A_B
no	sì	sì	no	no	no

Essendo in base 8, 3407.19_8 dovrebbe contenere cifre tra 0 e 7; $\bar{3}.456_7$ non può avere parte intera periodica in quanto sarebbe infinitamente grande; la base zero se esistesse non dovrebbe possedere alcun simbolo; infine, la base B non è definita in quanto ogni base è specificata da un numero intero in base 10.

2. Qual è l'intervallo di valori interi codificati in complemento a due rappresentabile con 7 bit? Qual è l'intervallo di valori interi codificati in complemento a uno rappresentabile con 7 bit? Si spieghi il perché dell'eventuale differenza.

R: (3 pt) Scegliendo il complemento a due l'intervallo è uguale a $[-2^6, 2^6 - 1]$. Scegliendo il complemento a uno l'intervallo è uguale a $[-2^6 + 1, 2^6 - 1]$. La differenza di una unità ha luogo poichè nella codifica degli interi in complemento a uno esistono due rappresentazioni per il valore zero.

3. [INF] Convertire il numero -2^6 in codifica *floating point* IEEE 754 a 32 bit.

R: (3 pt) Il numero può essere subito messo nella forma -1.0_2E6 . La codifica richiesta avrà dunque bit di segno asserito, esponente uguale a $127 + 6 = 133 = 10000101_2$ e infine mantissa uguale a 0_2 . Sistemando sui 32 bit previsti dallo standard IEEE 754 e convertendo alla base esadecimale:

```
1|1 0 0 0 0 1 0 1|0 0 0 0 0 0 0 0 ...
  C | 2 | 8 | 0 | 0 ...
```

da cui la codifica richiesta: **0xC2800000**.

4. Si supponga che ogni sei anni la densità di memoria in un chip aumenti esponenzialmente di un fattore 10^3 . Se nel 1980 in chip di 1 cm^2 memorizzava 2 kB, in quale anno un chip delle stesse dimensioni memorizzava o memorizzerà 2 GB?

R: (3 pt) Dopo 6 anni il chip memorizzava $2 \cdot 10^3$ kB, e dopo altri 6 anni memorizzava $2 \cdot 10^3 \cdot 10^3$ kB, cioè 2 GB. Quindi, l'anno in cui l'obiettivo viene raggiunto è il 1992.

5. Adoperando le regole di equivalenza booleana, riformulare l'espressione seguente in modo che ammetta una realizzazione adoperando esclusivamente porte NAND a due ingressi: $E = A + B + C$.

R: (3 pt) Ricordando l'equivalenza di De Morgan $X + C = \overline{\overline{X} \overline{C}}$ si ha $E = (A + B) + C = \overline{\overline{A + B} \overline{C}}$. Ora, è immediato negare una singola variabile attraverso una porta NAND a due porte: $\overline{C} = \overline{CC}$. Resta da realizzare il termine $\overline{A + B}$ adoperando la stessa porta: $\overline{A + B} = \overline{\overline{A} \overline{B}}$.

In definitiva: $A + B + C = \overline{\overline{A + B} \overline{C}} = \overline{\overline{\overline{\overline{A} \overline{B}}} \overline{CC}}$.

6. [INF] Richiede meno transistor realizzare l'espressione all'esercizio sopra adoperando solo porte OR a due ingressi oppure solo porte NAND a due ingressi?

R: (3 pt) L'espressione all'esercizio sopra è realizzata adoperando 2 porte OR a 2 ingressi, oppure 6 porte NAND a due ingressi. Ciascuna OR contiene 3 transistor mentre ciascuna NAND contiene 2 transistor, per un totale rispettivamente di 6 oppure 12 transistor. In definitiva, la prima delle due realizzazioni richiede metà dei transistor.

7. Realizzare la stessa espressione $E = A + B + C$ adoperando uno o più *multiplexer* a due ingressi di controllo, di cui non occorre esplicitare il circuito logico.

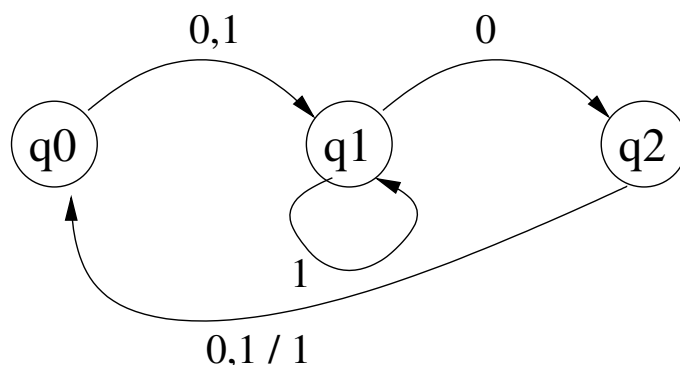
R: (3 pt) È sufficiente scegliere due multiplexer a 2 controlli, entrambi cablati nei 4 ingressi per realizzare l'OR di due variabili: il primo sarà controllato da A e B, realizzando quindi l'espressione $A + B$; il secondo sarà di conseguenza controllato dall'uscita dal primo multiplexer e da C, realizzando quindi l'espressione $(A + B) + C = A + B + C$. L'espressione da realizzare in definitiva è presente all'uscita del secondo multiplexer.

8. [INF] Dare il grafo della macchina di Mealy che riconosce le seguenti sottostringhe:

001
101
000
100

contenute in una sequenza definita sull'alfabeto $\mathcal{A} = \{0, 1\}$, in cui le sottostringhe sono necessariamente disgiunte all'interno della sequenza. La macchina restituisce il simbolo 1 solo quando riconosce la sottostringa, altrimenti restituendo 0 a fronte di ogni nuovo ingresso appartenente alla sequenza. Per esempio, l'uscita della macchina a fronte dell'ingresso 000000 sarà 001001.

R: (3 pt)



9. Si decide di trasmettere una sequenza di caratteri codificati ASCII senza introdurre alcuna codifica a correzione d'errore. Ammesso che la domanda abbia senso, qual è la distanza di Hamming tra due successivi caratteri trasmessi?

R: (3 pt) Due codifiche ASCII differiscono almeno per un bit, quindi la distanza di Hamming in assenza di correzione d'errore varia da uno a sette in base ai caratteri trasmessi. Il che, visto il valore minimo che può assumere la distanza, non garantisce alcuna possibilità di rilevamento di un errore.

10. Un bus parallelo che funziona a una frequenza $f = 37$ MHz è composto da 10 linee seriali, ciascuna delle quali trasmette un bit a ogni istante di clock. Qual è lo *skew* ammissibile tra due linee adiacenti, supponendo che i ritardi si possono accumulare linea dopo linea?

R: (3 pt) Nel caso peggiore i 10 bit sono tutti temporalmente disallineati, e riempiono un periodo di clock $T = 1/f$ senza sovrapporsi ai bit appartenenti al periodo precedente e a quello successivo. In questo caso, ciascun bit è distanziato da quello a lui più vicino di non più di $T/10 = 1/(10 \cdot f) = 3$ ns, che è dunque lo *skew* al di sotto del quale non si può scendere.

11. Si possono ottimizzare i programmi nella piattaforma Arduino in modo da massimizzarne la velocità di esecuzione? Se sì, come?

R: (3 pt) Sì. Esistono istruzioni specifiche che generano un numero inferiore di istruzioni macchina. Inoltre è possibile scrivere direttamente le istruzioni in Assembly per ARM.

12. In un'architettura a 32 bit, qual è lo spazio di memoria più esteso che può essere indirizzato da un'istruzione di tipo B (*branch*) nell'ipotesi che l'ISA preveda di assegnare 8 bit al comando contenuto nell'istruzione? Si motivi la risposta.

R: (3 pt) La soluzione più semplice è assegnare $32 - 8 = 24$ bit all'indirizzo, potendo così indirizzare $2^{24} \cdot 4 = 64$ MB nell'ipotesi di avere word di 32 bit. In realtà nessuno vieterebbe alla stessa istruzione di proseguire la lettura dell'indirizzo nel word successivo, tuttavia un'esecuzione di questo tipo non è comune anche se teoricamente possibile.

13. Una insieme di nodi di calcolo parallelo è completamente interconnesso, nel senso che ogni nodo di calcolo comunica direttamente con tutti gli altri. Se i nodi sono 8, quante linee di comunicazione devono essere allestite? Facoltativo: Se i nodi sono n , quante linee di comunicazione devono essere allestite? Si dia solamente la formula senza produrre un risultato [Suggerimento: aggiungere il nodo n -esimo a una rete completamente interconnessa e contare quante linee devono essere corrispondentemente aggiunte].

R: Rispondiamo prima al quesito facoltativo. L'aggiunta del nodo n -esimo richiede di aggiungere $n - 1$ nodi a una rete completamente interconnessa al fine di preservarne la proprietà. Ne discende immediatamente che una rete completamente interconnessa contenente n nodi possiede $1 + 2 + \dots + (n - 2) + (n - 1)$ connessioni. Di qui è immediato calcolare il caso di 8 nodi, i quali necessitano di allestire $1 + 2 + 3 + 4 + 5 + 6 + 7 = 28$ linee di comunicazione.

14. [INF] Scrivere un programma in assembly per ARM il quale accoda agli elementi di un array presente in memoria il valore del massimo e del minimo elemento. Per semplicità si assuma che le due locazioni di memoria immediatamente successive a quelle occupate dall'array siano libere. La dimensione dell'array è indicata in corrispondenza della locazione immediatamente precedente a quella dove inizia l'array: nel caso in cui in questa locazione compaia il valore 0 allora l'array è privo di elementi e il programma esce senza eseguire alcuna modifica alla memoria.

R: (9 pt)

```
.data
array_dim:
.word 6
array_el:
.word -2, 3, -5, 7, -11, 4
.text
main:
    ldr r0, =array_dim
    ldr r0, [r0]          ; array length in r0
    cmp r0, #0           ; if empty array..
    beq exit             ; ..then exit
    ldr r1, =array_el     ; base in r1
    ldr r2, [r1]          ; assign first element to max
    ldr r3, [r1], #4      ; assign first element to min, go next
loop:
    subs r0, r0, #1      ; decrement length and set (zero) registers
    beq write            ; write results if array visit is finished
    ldr r4, [r1], #4      ; read next element and increment r1
    cmp r4, r2           ; if r4-r2..
    movgt r2, r4          ; >0 then update max
    cmp r4, r3           ; if r4-r3..
    movlt r3, r4          ; <0 then update min
    b loop
write:
    str r2, [r1], #4      ; store maximum element and increment r1
    str r3, [r1]          ; store minimum element
exit:
    swi 0x11             ; exit
.end
```