



Esami A e B di Architetture degli Elaboratori

Soluzione

A.A. 2017-18 — I appello — 30 gennaio 2018

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 30 (circa).

1. Convertire il valore 203.42_5 in base 7.

R: (3 pt) Passiamo attraverso la base decimale:

$$203.42_5 = 2 \cdot 5^2 + 3 \cdot 5^0 + 4 \cdot 5^{-1} + 2 \cdot 5^{-2} = 50 + 3 + 0.8 + 0.08 = 53.88$$

| | |
|--------|---------|
| 53 7 | .88 7 |
| ----- | ----- |
| 7 4 | .16 6 |
| 1 0 | .12 1 |
| 0 1 | .84 0 |
| | .88 5 |
| | |

da cui otteniamo immediatamente il valore nella nuova base: $104.\overline{6105}_7$.

2. Sono dati i seguenti valori n_1 e n_2 codificati in complemento a 2 a 8 bit: $n_1 = 10110111$, $n_2 = 00000011$. Si calcoli il prodotto $n_1 \cdot n_2$ e, se possibile, si esprima il risultato nella stessa codifica.

R: (3 pt) Conviene calcolare il modulo del prodotto e, successivamente, ricordare che n_1 ha segno negativo. Il modulo di n_1 si ottiene per complementazione a 2: $-n_1 = 01001001$, da cui immediatamente $|n_1| \cdot n_2$:

$$\begin{array}{r}
 01001001 * \\
 00000011 = \\
 \hline
 01001001 + \\
 01001001 = \\
 \hline
 11011011
 \end{array}$$

Ora, cambiare di segno di questo risultato richiederebbe almeno 9 bit adoperando per esso la codifica in complemento a 2. Dunque il prodotto appena calcolato non può essere espresso nella codifica richiesta.

3. [INF] Fornire il risultato dell'esercizio precedente in codifica *floating point* IEEE 754 a 32 bit.

R: (3 pt) Il risultato trovato sopra può essere subito messo nella forma $-1.1011011E7$. La codifica richiesta avrà dunque bit di segno asserito, esponente uguale a $127 + 7 = 134 = 10000110_2$ e infine mantissa uguale a 1011011 . Sistemando sui 32 bit previsti dallo standard IEEE 754 e convertendo alla base esadecimale:

$$\begin{array}{cccccccc|cccccccccccc}
 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & \dots \\
 C & & & & & & 3 & & & & 5 & & & & B & & & 0 \dots
 \end{array}$$

da cui la codifica richiesta: $C35B0000_{16}$.

4. La *legge di Moore* afferma che il numero di transistor nell'unità di memoria quadruplica ogni tre anni. Se la densità in un *chip* di memoria è uguale a $3 \cdot 10^7$ transistor/cm², che densità dobbiamo aspettarci nello stesso *chip* dopo 4 anni e mezzo?

R: (3 pt) Per come è definito, il tasso di crescita appena visto è per sua natura *esponenziale*. Se il numero quadruplica ogni tre anni allora il raddoppio avviene ogni anno e mezzo. Quindi dopo quattro anni e mezzo assistiamo a tre raddoppi, dunque la densità attesa sarà $2 \cdot 2 \cdot 2 \cdot 3 \cdot 10^7 = 24 \cdot 10^7$ transistor/cm².

5. Trovare il circuito col minore numero di porte AND, OR, NOT, il quale realizza l'espressione

$$E = \overline{A} B \overline{C} + A \overline{B} \overline{C} + \overline{A} \overline{B} \overline{C}$$

R: (3 pt)

$E = \overline{A} B \overline{C} + \overline{A} \overline{B} \overline{C} + A \overline{B} \overline{C} = \overline{A}(B + \overline{B})\overline{C} + A \overline{B} \overline{C} = \overline{A} \overline{C} + A \overline{B} \overline{C} = \overline{A} \overline{C} + A(\overline{B} + \overline{C}) = \overline{A} \overline{C} + A \overline{C} + A \overline{B} = \overline{C} + A \overline{B}$
da cui discende immediato il circuito contenente 2 porte NOT, una porta AND e una porta OR.

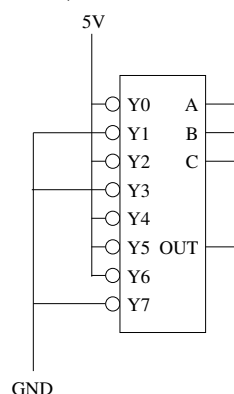
6. [INF] Verificare la minimizzazione ottenuta sopra con una mappa di Karnaugh.

R: (3 pt) Ci sono due regioni connesse, rispettivamente contenenti quattro e due simboli 1. La prima presenta la costanza del solo termine \overline{C} ; la seconda presenta la costanza dei termini A e \overline{B} .

| BC | 00 | 01 | 11 | 10 |
|----|----|----|----|----|
| A | | | | |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 |

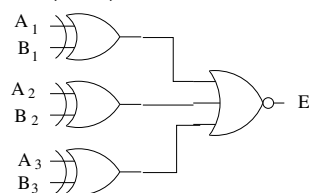
7. [INF] Riprodurre il circuito appena progettato adoperando un *multiplexer*.

R: (3 pt) È sufficiente scegliere un multiplexer a 8 ingressi e 3 controlli collegati rispettivamente ad ABC , e poi connettere a una sorgente in tensione (per esempio 5V) gli ingressi associati ai valori di controllo 000, 010, 100, 101, 110. Viceversa, gli altri 3 ingressi dovranno essere collegati a una tensione nulla (detta anche di massa, o GND).



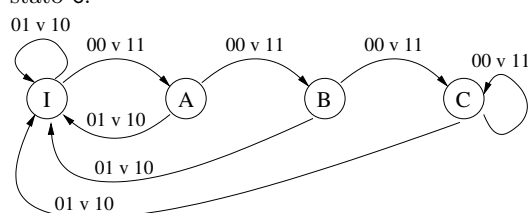
8. Progettare un comparatore di ingressi a 3 bit, cioè una rete combinatoria in grado di decidere se due ingressi $A_1 A_2 A_3$ e $B_1 B_2 B_3$ sono identici.

R: (3 pt)



9. [INF] Progettare la macchina a stati finiti (di Mealy oppure di Moore) che realizza il comparatore di cui sopra, supponendo che essa a ogni istante riceva per un tempo indefinitamente lungo coppie di bit $A_1 B_1, A_2 B_2, A_3 B_3, \dots, A_n B_n, \dots$, e all'istante n -esimo debba stabilire se gli ingressi $A_{n-2} A_{n-1} A_n$ e $B_{n-2} B_{n-1} B_n$ sono identici.

R: (3 pt) La seguente macchina di Moore effettua la comparazione partendo dallo stato I, producendo un'uscita uguale a 0 in corrispondenza degli stati I, A e B; viceversa, produce un'uscita uguale a 1 in corrispondenza dello stato C.



10. Si vogliono codificare le cifre decimali 0,1,...,9 con 4 bit adoperando un codice a *lunghezza variabile* il quale ottimizzi l'efficienza della codifica delle cifre decimali 0 e 1. Si dia un possibile codice che rispetta questo vincolo.

R: (3 pt) Quattro bit sono sufficienti, ma non necessari per codificare 10 simboli. Poichè con 3 bit possiamo codificare 8 simboli, assegniamo il valore 0 del bit più significativo alla codifica delle cifre 0 e 1; altrimenti occorrerà valutare i tre bit meno significativi per decodificare le restanti otto cifre decimali. Quindi, per esempio,

```

0 -> 00
1 -> 01
2 -> 1000
3 -> 1001
4 -> 1010
5 -> 1011
6 -> 1100
7 -> 1101
8 -> 1110
9 -> 1111

```

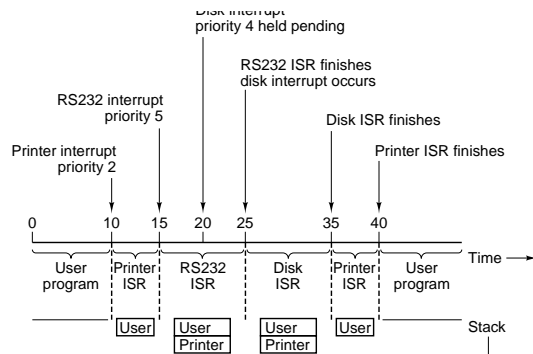
In tal modo, 0 e 1 necessitano di due soli bit per essere codificati mentre tutte le altre cifre necessitano di 4 bit.

11. Una CPU interagisce con 4 dispositivi esterni etichettati A, B, C, D. Ogni richiesta del dispositivo D dev'essere servita prima di quelle provenienti da C, e ogni richiesta del dispositivo B dev'essere servita prima di quelle provenienti da A e D. In quanti modi possono essere disposti i dispositivi in un bus che implementa il protocollo *daisy chain*? Elencarli.

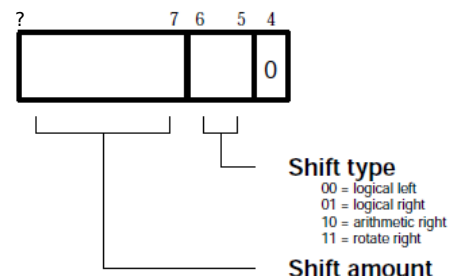
R: (3 pt) Il bus monterà, in ordine di prossimità alla CPU, i seguenti dispositivi: BADC, BDAC oppure BDCA.

12. Si dia schematicamente un esempio di *mascheramento dell'interrupt* durante il servizio a dispositivi esterni da parte della CPU.

R: (3 pt) Qualunque variazione della situazione descritta in figura è accettabile.



13. I campi in figura appartengono a un'istruzione macchina per ARM che prevede la possibilità di traslare o ruotare il contenuto di un registro. Quale valore posizionale va sostituito al '?' in alto a sinistra nella figura? Quali sono le traslazioni o rotazioni minime e massime ammesse in tal caso sul registro?



R: (3 pt) Se parliamo di un'architettura a 32 bit, allora le traslazioni o rotazioni ammesse sul registro variano da un minimo di 0 bit a un massimo di 31 bit. Occorrendo 5 bit per specificare l'entità di queste operazioni, il campo più a sinistra in figura occuperà i bit dal 7 all'11.

14. Una memoria virtuale paginata può contare su una memoria principale di 1 MB. Quanto è lunga ogni riga della *page table*? Se il campo *offset* di ogni indirizzo fisico è lungo 14 bit, qual è la dimensione di ogni pagina?

R: Una memoria di 1 MB necessita di 20 bit per essere indirizzata. Includendo il bit di presenza/assenza della pagina in memoria, la *page table* si comporrà di righe lunghe 21 bit. Infine, noto l'*offset*, ogni pagina avrà un'estensione di $2^{14} = 16$ kB.

15. [INF] Scrivere un programma in assembly per ARM, il quale trova il valore massimo tra n elementi contenuti in un file testuale di nome `inputLista.txt`. Il file contiene il numero n di elementi nella prima riga e n valori interi nelle righe successive. Al termine dell'esecuzione il programma avrà restituito in memoria l'indice dell'elemento di valore massimo, oppure il valore -1 se il file contiene 0 elementi. Nel caso in cui nel file esista più di un elemento di valore massimo il programma restituirà l'indice del primo massimo presente nel file.

R: (9 pt)

```
.data
stringa:
.asciiz "inputLista.txt"
output:
.skip 4
.text
main:
    ldr r0, =stringa        ; string address in r0
    mov r1, #0              ; read mode
    swi 0x66                ; open file in read mode
    mov r2, r0              ; save file handler
    swi 0x6c                ; read number of integers
    mov r3, r0              ; save number of integers in r3
    mov r5, #0x80000000     ; start with minInt in r5
    mov r4, #0              ; current element in r4
    mov r6, #0xFFFFFFFF     ; min element in r6

loop:
    subs r3, r3, #1         ; decrement r3 and set status
    blt exit               ; exit if negative
    add r4, r4, #1         ; increment r4
    mov r0, r2             ; load file handler
    swi 0x6c               ; read integer from file
    cmp r5, r0             ; if r0-r5 < 0..
    movlt r5, r0           ; ..then update minimum..
    movlt r6, r4           ; ..and its index
    b loop                ; next element

exit:
    mov r0, r2             ; load input file handler
    swi 0x68               ; close file
    ldr r0, =output        ; output address in r0
    str r6, [r0]           ; save min element index

;; end
    swi 0x11               ; exit
    .end
```