



Esame di Architetture degli Elaboratori

Soluzione

A.A. 2019-20 — II appello — 13 febbraio 2020

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 converta il numero 73.74_8 nella base 5.

R: (3 pt) Adoperiamo il metodo standard passando attraverso la base 10:

$73.74_8 = 7 \cdot 8 + 3 + 7/8 + 4/8^2 = 59.9375$. Ora,

59 5	.9375 5
-----	-----
11 4	.6875 4
2 1	.4375 3
0 2	.1875 2
	.9375 0
	...

da cui emerge la periodicità della parte decimale. In conclusione quindi

$$73.74_8 = 214.4320_5.$$

2. Sottrarre il valore 83 dal valore 7 svolgendo i calcoli in aritmetica in complemento a 2 a 8 bit, e se possibile dare il risultato nella stessa codifica.

R: (3 pt) Dopo avere codificato i rispettivi valori in complemento a 2 a 8 bit, cambiamo di segno il numero 83 in modo da eseguire una somma in luogo della sottrazione:

00000111		00000111
01010011	- ->	10101101
-----		-----
		10110100 -> -01001100 = -76

che è il risultato senza overflow della nostra operazione svolta in aritmetica complemento a due a 8 bit.

3. [INF] Convertire il numero $2^9 + 1$ in codifica *floating point* IEEE 754 a 32 bit.

R: (3 pt)

Il numero può essere immediatamente convertito in base 2: $2^9 + 1 = 1000000001_2 = 1.000000001_2 E9$. La codifica richiesta avrà dunque bit di segno non asserito, esponente uguale a $127 + 9 = 136 = 10001000_2$ e infine mantissa uguale a 000000001_2 . Sistemando sui 32 bit previsti dallo standard IEEE 754 e convertendo alla base esadecimale:

0 1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	...
4		4		0		0		4		...						

da cui la codifica richiesta: $0x44004000$.

4. Se ogni due anni la densità di transistor in un chip di memoria triplica, quale sarà la capacità di memoria di un chip di 4 GB dopo 5 anni, a parità di dimensioni?

R: (3 pt) Poiché nelle ipotesi fatte emerge ogni anno un aumento della densità di un fattore $\sqrt{3}$, allora dopo 5 anni la capacità sarà aumentata di un fattore $(\sqrt{3})^5 = 9 \cdot \sqrt{3} \approx 16$, da cui la capacità richiesta: $4 \cdot 16 = 64$ GB.

5. É data la seguente tabella di verità:

A	B	C	E
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Senza eseguire minimizzazioni attraverso l'uso di operazioni algebriche, si dia prima l'espressione booleana che realizza la rete come *somma di prodotti* (forma normale), e successivamente l'espressione booleana che realizza la stessa rete come *prodotto di somme* (forma normale "dualizzata").

R: (3 pt) Forma normale: $E = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}BC + A\overline{B}C + ABC$.

Forma normale "dualizzata": $E = (A + \overline{B} + C)(\overline{A} + B + C)(\overline{A} + \overline{B} + C)$.

6. [INF] Si realizzi la tabella di Karnaugh che minimizza l'espressione booleana relativa alla tabella di verità dell'esercizio 5.

R: (3 pt)

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

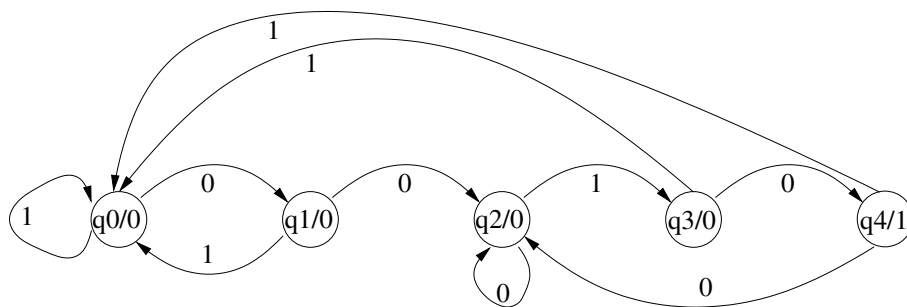
7. Si minimizzi l'espressione booleana relativa alla tabella di verità dell'esercizio 5, partendo indifferentemente dalla forma normale o dalla forma normale "dualizzata".

R: (3 pt) Dalla forma normale: $E = \overline{A}\overline{B}\overline{C} + (\overline{A}\overline{B}C + \overline{A}BC) + (A\overline{B}C + ABC) = \overline{A}\overline{B}\overline{C} + \overline{A}C + AC = \overline{A}\overline{B}\overline{C} + C = \overline{A}\overline{B}\overline{C} + (\overline{A}\overline{B}C + C) = (\overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C) + C = \overline{A}\overline{B} + C$.

Dalla forma normale "dualizzata": $E = (A + \overline{B} + C)(\overline{A} + B + C)(\overline{A} + \overline{B} + C) = (AB + AC + \overline{A}\overline{B} + \overline{B}C + \overline{A}C + BC + C)(\overline{A} + \overline{B} + C) = (AB + \overline{A}\overline{B} + C)(\overline{A} + \overline{B} + C) = ABC + \overline{A}\overline{B} + \overline{A}C + \overline{B}C + C = \overline{A}\overline{B} + C$.

8. [INF] Si disegni, in testa alla pagina successiva, il diagramma di stato della macchina di Moore definita sull'alfabeto $\mathcal{A} = \{0, 1\}$, la quale produce in uscita il valore uno ogni volta che riconosce la sequenza 0010 dai più recenti quattro simboli ricevuti, altrimenti producendo il valore zero a ogni simbolo ricevuto.

R: (3 pt)



9. Quante parole possono al massimo essere generate da un codificatore di *Hamming* a 15 bit?

R: (3 pt) Come ogni codice, anche un codice di *Hamming* a 15 bit genera un numero di codifiche identico al numero di ingressi che può effettivamente ricevere. Ricordando che i bit di controllo in un codice di *Hamming* a 15 bit sono in numero uguale a 4, le codifiche possibili sono $2^{15-4} = 2^{11} = 2048$.

10. Un bus trasmette bit su un canale con una frequenza costante di trasmissione di bit $f_7 = 21.875$ MHz. Il ricevitore legge ogni bit e poi considera gruppi di 7 bit in successione i quali, opportunamente decodificati, formano infine le parole ricevute. Per rinforzare l'invio delle parole si decide di trasmettere un bit di

parità aggiuntivo ogni 7 bit; contemporaneamente si vuole conservare la costanza del numero di parole inviate nell'unità di tempo. Quale sarà il periodo di trasmissione di bit T_8 nel bus, necessario a mantenere costante il numero di parole inviate nell'unità di tempo?

R: (3 pt) Il numero di parole inviate viene mantenuto costante nell'unità di tempo se la frequenza di clock viene aumentata secondo la relazione $f_8 = (8/7)f_7 = (8/7) \cdot 21.875 = 25$ MHz, in modo da trasmettere gruppi di 8 bit in luogo di 7 a parità di tempo. A questo punto è immediato calcolare il periodo di trasmissione di bit $T_8 = 1/f_8 = (1/25) \cdot 10^{-6} = (1000/25) \cdot 10^{-9}$ s, ovvero 40 ns.

11. Che cos'è un'uscita analogica rispetto a un'uscita digitale in un microcontrollore? Si può generare un'uscita analogica con il microcontrollore AVR di Arduino? Se sì, in che cosa consiste quest'uscita?

R: (3 pt) Un'uscita analogica codifica un valore a ogni istante, mentre un'uscita digitale codifica un valore a ogni periodo di *clock*. Il microcontrollore AVR di Arduino può generare in uscita una funzione di tipo *pulse width modulation* (PWM). Detta funzione presenta un valore di tensione alto (3.3 Volt oppure 5 Volt) per una frazione del periodo di *clock* detta *duty cycle*, e un valore basso (0 Volt) per la restante frazione del periodo di clock. Poiché il *duty cycle* è programmabile e, quindi, il valore della funzione può variare a ogni istante durante un periodo di clock, la funzione PWM prodotta in uscita dal microcontrollore è appunto di tipo analogico.

12. Che cosa accade in una CPU quando questa esegue un'istruzione di salto incondizionato?

R: (3 pt) Accade che l'indirizzo di memoria contenuto nell'istruzione di salto incondizionato viene copiato nel registro denominato *program counter*, rimpiazzando l'indirizzo dell'istruzione che in memoria segue quella in esecuzione.

13. Una memoria principale di 4 GB è accelerata da una *cache* ad accesso diretto a 512 *entry*, la quale serve tutta la memoria principale. Ciascuna *entry* contiene 64 byte dedicati a memorizzare i dati. Prescindendo da ogni conoscenza circa la collocazione dei dati in memoria principale e il loro utilizzo da parte del programma, qual è la probabilità di trovare il contenuto di una locazione di memoria principale (a puro titolo di esempio: la locazione di indirizzo 0x00000100) nella *cache*?

R: In mancanza di ogni informazione sui dati e sul programma che li utilizza, essendo la capacità della *cache* uguale a $512 \cdot 64 = 2^9 \cdot 2^6 = 2^{15}$ byte allora la probabilità di trovare una locazione di memoria principale nella *cache* è uguale al rapporto tra la capacità della *cache* e quella della memoria principale: $2^{15}/2^{32} = 2^{-17} \approx 8 \cdot 10^{-6}$.

14. [INF] Scrivere un programma in assembly per ARM il quale, lette una stringa etichettata **target** e una stringa etichettata **search** dalla memoria, genera una terza stringa che viene salvata in una regione della memoria etichettata **result**. Quest'ultima stringa contiene tutti e soli i caratteri di **search** presenti nella **target**. Per semplicità si assuma che **search** non contenga ripetizioni di uno stesso carattere. É gradita la presenza di commenti al codice prodotto.

R: (9 pt) (*continua*)

```

.data
eos:
.asciiz ""                ; contain only end-of-string character
target:
.asciiz "ciao mamma mamma come mi diverto!"
search:
.asciiz "che noia!"
result:
.skip 9+1                ; length(result) <= length(search)
.text
main:
ldr r0, =eos
ldrb r0, [r0]             ; EOS character in r0
ldr r2, =search           ; search head position in r2
ldr r3, =result           ; result head position in r3
loop:
ldrb r7, [r2], #1         ; load search char in r7, inc r2
cmp r7, r0                ; if search char == EOS..
beq exit                  ; ..then exit
ldr r1, =target           ; set target head position in r1
loop_inner:
ldrb r6, [r1], #1         ; load target char in r6, inc r1
cmp r6, r0                ; if target char == EOS..
beq loop                  ; ..then restart outer loop
cmp r6, r7                ; if search char == target char ..
streqb r7, [r3], #1       ; .. then store result char, inc r3
beq loop                  ; .. and restart outer loop (break)
b loop_inner              ; repeat inner loop
exit:
strb r0, [r3]             ; store EOS char in result
swi 0x11                  ; exit
.end

```