

Esame **A** di Architetture degli Elaboratori

Soluzione

A.A. 2017-18 — II appello — 20 febbraio 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 $4.\bar{3}$ dalla base 10 alla base 3.

R: (3 pt)

$$\begin{array}{r|l} 4 & 3 \\ \hline 1 & 1 \\ 0 & 1 \end{array} \quad \begin{array}{r|l} .333\dots & 3 \\ \hline 0 & 1 \end{array}$$

e quindi $4.\bar{3} = 11.1_3$. Infatti, $3^{-1} = 1/3 = 0.\bar{3}$, da cui $4.\bar{3} = 1 \cdot 3^1 + 1 \cdot 3^0 + 1 \cdot 3^{-1}$.

2. Sono date le seguenti codifiche in complemento a 2 a 8 bit: $n_1 = 10110111$, $n_2 = 11001100$. Si calcoli la differenza $n_1 - n_2$ e, se possibile, si esprima il risultato nella stessa codifica.

R: (3 pt) Complementando n_2 per cambiarne il segno: $-n_2 = 00110100$, eseguiamo successivamente la somma

$$\begin{array}{r} 10110111 + \\ 00110100 = \\ \hline 11101011 \end{array}$$

che, coinvolgendo un valore positivo e uno negativo, non dà *overflow* ed è quindi codificata come tale.

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 $-1.0101E4$. La codifica richiesta avrà dunque bit di segno asserito, esponente uguale a $127+4 = 131 = 1000011_2$ e infine mantissa uguale a 0101. Sistemando sui 32 bit previsti dallo standard IEEE 754 e convertendo alla base esadecimale:

$$\begin{array}{cccccccccccccccccccccccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & \dots \\ \text{C} & | & 1 & | & \text{A} & | & 8 & | & 0 & \dots \end{array}$$

da cui la codifica richiesta: **0xC1A80000**.

4. Si esprima il numero di bit presenti in una memoria a 512 MB come potenza di 2.

R: (3 pt) Ricordando che $1 \text{ MB} = 2^{20} \text{ byte}$: $8 \cdot 512 \cdot 2^{20} = 2^3 \cdot 2^9 \cdot 2^{20} = 2^{32} \text{ bit}$.

5. Adoperando le regole di equivalenza booleana, calcolare quanto vale E nell'espressione seguente:

$$E = \overline{ABC} + \overline{AC}$$

R: (3 pt) Sfruttando le regole di De Morgan si ha $E = \overline{A} + \overline{B} + \overline{C} + \overline{\overline{A} + \overline{C}} = 1 + \overline{B} + \overline{C} = 1$

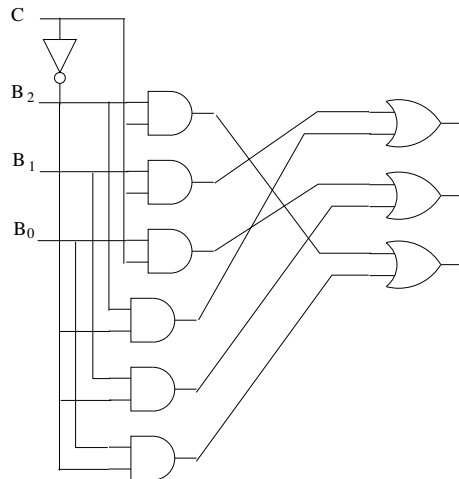
6. [INF] Verificare il risultato ottenuto sopra con una mappa di Karnaugh.

R: (3 pt) La tabella di verità porge una mappa che presenta solo simboli 1:

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

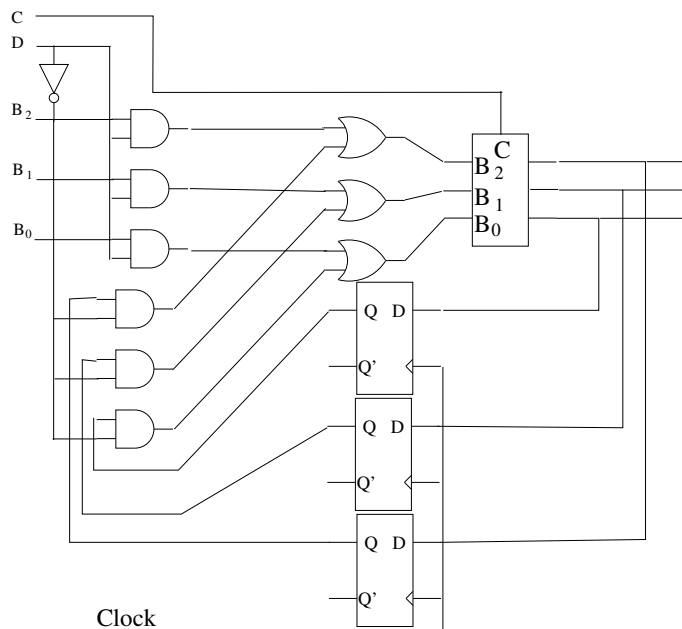
7. Progettare un traslatore rotazionale a 3 bit a sinistra, cioè una rete combinatoria in grado di ruotare un ingresso a 3 bit (B_2, B_1, B_0) verso sinistra di un bit oppure no, rispettivamente se un controllo C a un bit è asserito oppure no. In altre parole, se $C = 1$ allora (B_2, B_1, B_0) ruota in (B_1, B_0, B_2) .

R: (3 pt)



8. [INF] Avendo a disposizione il traslatore di cui sopra, che per comodità può essere denotato come un unico blocco, progettare un circuito sequenziale che, a seconda del valore assunto dal controllo C , a ogni ciclo di clock esegue la rotazione di un bit a sinistra oppure no dei bit (B_2, B_1, B_0) in ingresso oppure dei tre bit in uscita all'istante precedente. Quest'ultima decisione viene presa in base al valore assunto all'istante corrente da un bit etichettato come D .

R: (3 pt)



9. Un codice a ripetizione tripla del carattere codifica le cifre decimali $0, 1, \dots, 9$ in corrispondenti triplette di cifre identiche. Il decodificatore restituisce la cifra c se le tre cifre costituenti la tripletta sono tutte uguali a c , altrimenti segnalando un errore di trasmissione e scartando la tripletta. Si dica in quanti casi una trasmissione errata viene riconosciuta come tale rispetto alla totalità delle codifiche che possono essere ricevute.

R: (3 pt) Su 1000 possibili triplette ricevute, una è quella corretta. Nove triplette contengono tre cifre identiche errate ma non vengono scartate, mentre tutte le altre vengono scartate. Quindi, in 990 casi su 999 una trasmissione errata viene riconosciuta come tale.

10. Un elementare sistema di I/O programmato prevede che ogni 100 ms la CPU impegni 3 ms del proprio tempo nel *polling* di un'unica periferica. Se la periferica non deve essere servita questo tempo viene sprecato, viceversa in caso di servizio i 3 ms vengono impiegati utilmente e la CPU lavora globalmente con un'efficienza del 100%. In queste ipotesi si calcoli quante volte la periferica dovrebbe mediamente richiedere un servizio affinché la CPU lavori con un'efficienza del 99%.

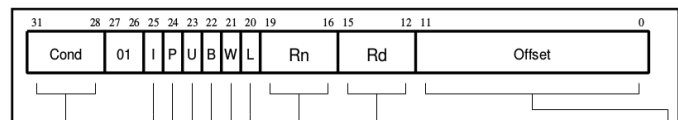
R: (3 pt) Posto x il numero medio di volte in cui la periferica richiede di essere servita, se la periferica non richiede mai il servizio ($x = 0$) l'efficienza è del $100 - 3 = 97\%$. Al contrario, se richiede sempre un servizio ($x = 1$) l'efficienza sale al 100%. Il numero medio di volte si ottiene dunque imponendo che $97 \cdot (1 - x) + 100 \cdot x = 99 \Rightarrow 97 - 97x + 100x = 99 \Rightarrow 3x = 2 \Rightarrow x = 2/3$, cioè due volte su tre.

11. Se la periferica del sistema di I/O presentato in precedenza inoltra mediamente 12 richieste di servizio al secondo, il *polling* può funzionare? Si motivi sinteticamente la risposta.

R: (3 pt) No, poichè il sistema in questione non può servire la periferica mediamente più di 10 volte al secondo.

12. I campi in figura appartengono a un'istruzione macchina per ARM che prevede la possibilità di salvare il contenuto di un registro R_d nell'indirizzo di memoria specificato dal contenuto del registro R_n corretto da un valore di *offset* presente nell'omonimo campo. Quanti kB può valere al massimo lo stesso offset in base alle informazioni presenti in figura?

ARM Processor Instruction Set



R: (3 pt) Essendo il campo *offset* di 12 bit, in assenza di ulteriori informazioni sul funzionamento dell'indirizzamento in ARM il valore massimo in byte che può essere ivi specificato è $2^{12} = 4096 = 4 \cdot 2^{10}$ byte, cioè 4 kB.

13. Quali locazioni di una memoria principale di 64 kB può contenere la linea 1023 di una *cache* a 32 byte composta da 1024 *entry*?

R: Le locazioni dalla $32 \cdot 1023 = 32736$ alla $32 \cdot 1023 + 31 = 32767$ e dalla $32 \cdot 1024 + 32 \cdot 1023 = 65504$ alla $32 \cdot 1024 + 32 \cdot 1023 + 31 = 65535$.

14. [INF] Scrivere un programma in assembly per ARM il quale, caricati nei registri $r2$ e $r3$ rispettivamente due numeri m e n tali che $0 < n \leq m$, calcola il valore $m!/(m-n)! = m(m-1) \dots (m-n+1)$ attraverso una procedura ricorsiva che esegue le $n-1$ moltiplicazioni e deposita il risultato nel registro $r1$.

R: (9 pt)

```
.text
main:
    mov r2, #5          ; read m
    mov r3, #3          ; read n
    subs r3, r3, #1     ; will multiply n-1 times
    blne fattoriale     ; if n!=0 then call
    mov r1, r2          ; copy result in r1
    swi 0x11           ; exit program

fattoriale:
    stmfd sp!, {r4, lr} ; save registers
    mov r4, r2          ; m in r4
    sub r2, r2, #1      ; m <- m-1
    subs r3, r3, #1     ; decrement r3
    blne fattoriale     ; if r3!=0 call again
    mul r2, r4, r2       ; otherwise multiply
    ldmdfd sp!, {r4, lr} ; restore registers
    mov pc, lr          ; restore PC

.end
```