

A.A. 2020-21

Cognome e Nome _____

Matricola _____

Soluzione

Per tutta la verifica, **N** sarà uguale alle cinque o sei cifre del numero della matricola dello studente, dapprima privata di eventuali lettere, e poi trascritta nel verso opposto aggiungendo infine zeri fino a raggiungere un numero di sei cifre.

Es.: se la matricola è 237424, allora **N = 424732**

se la matricola è 237400, allora **N = 473200**

se la matricola è I-37424, allora **N = 424730**.

1. Si supponga che del numero **N** non si conosca la base **B** di rappresentazione. In altre parole, chi lo ha scritto non ha specificato in che base lo ha rappresentato. Si chiede

a) qual è la base **più piccola** in cui esso potrebbe essere stato rappresentato così come è stato scritto. RISULTATO: **B = _____**

b) assumendo che **N** sia rappresentato in base **B**, di convertirlo in base 10.

RISULTATI: La base minima dev'essere sufficientemente grande da includere la cifra massima presente in **N**. Per esempio, se **N = 424730** allora **B = 8**. La conversione alla base 10 segue dalla definizione vista a lezione. Per esempio, se **N = 424730**:

$$424730_8 = 4 \cdot 8^5 + 2 \cdot 8^4 + 4 \cdot 8^3 + 7 \cdot 8^2 + 3 \cdot 8^1 + 0 \cdot 8^0$$

2.

a) Si prenda la cifra **più** significativa di **N** e la si converta in un numero binario. RISULTATO: _____

b) Dopo averlo cambiato di segno, si rappresenti lo stesso numero in aritmetica in complemento a due a **10 bit**

RISULTATI: La conversione di una cifra decimale in binario è immediata. Per esempio, se **N = 424730** allora il numero richiesto è $4_{10} = 100_2$.

A questo punto dal cambio di segno segue, per definizione di aritmetica in complemento a due a 10 bit,
 $-100_2 = 1111111011_2 + 1_2 = 1111111100$ che è la rappresentazione richiesta.

3. Un chip di memoria ha una capacità di N Byte. Se ogni anno la densità di transistor nell'unità di memoria quadruplica, qual era la capacità del chip tre mesi prima a parità di superficie?

RISPOSTA: Poiché in un anno la capacità quadruplica, a ogni trimestre l'incremento di capacità è uguale alla radice quarta di 4, cioè $4^{1/4}$. L'incremento annuale in altre parole evolve nel corso dei trimestri secondo la progressione

... $\rightarrow 4^{-1/4}N \rightarrow N \rightarrow 4^{1/4}N \rightarrow 4^{2/4}N \rightarrow 4^{3/4}N \rightarrow 4^{4/4}N \rightarrow \dots$

Quindi, la quantità di cui dev'essere ridotta la capacità è uguale a $4^{-1/4} = 1/4^{1/4} = 1/\sqrt[4]{2} = 0.707$.

4.

a) Si calcoli il **resto intero di N/16** e lo si converta in un numero binario a 4 bit. RISULTATO: _____

b) Si costruisca una tabella di verità, la cui unica uscita assume valori uguali alle cifre costituenti il risultato al punto a) a partire dalla cifra meno significativa a quella più significativa nel verso crescente del valore degli ingressi.

RISPOSTA: dato che il resto R in binario di N/16 si compone di 4 bit, la corrispondente tabella di verità può assumere 4 combinazioni diverse degli ingressi binari, che quindi sono due:

A	B		E
0	0		C_0
0	1		C_1
1	0		C_2
1	1		C_3

in cui C_3, C_2, C_1, C_0 , sono le cifre binarie che compongono il resto della divisione: $R = C_3C_2C_1C_0$.

5. Realizzare:

a) un circuito combinatorio che, adoperando solamente **porte NAND**, realizza la tabella di verità trovata all'esercizio 4

RISPOSTA: conviene rispondere prima alla domanda b)

b) l'espressione booleana corrispondente al circuito appena realizzato

RISPOSTA: dalla realizzazione in forma canonica della tabella di verità precedente è immediata la scrittura della corrispondente espressione booleana come OR di AND. A questo punto, per quanto visto nei lucidi a lezione, è sufficiente calcolare la doppia negazione di questa espressione per trovare subito una nuova

espressione che contiene solo operatori di tipo NAND. Da quest'ultima scaturisce immediato il circuito combinatorio.

6. Le ultime 10 lettere dell'alfabeto internazionale sono QRSTUVWXYZ. Alla lettera Z corrisponde la codifica esadecimale ASCII **5A**. Si indichi la sequenza ASCII corrispondente alla parola di 6 caratteri generata da N, adoperando ogni cifra di N come posizione (numerata da 0 a 9) della lettera appartenente a QRSTUVWXYZ. Per esempio, se $N = 424730$ allora la parola è USUXTQ.

RISPOSTA: tenendo conto delle posizioni dei caratteri nel codice ASCII, per esempio nel caso USUXTQ si ha subito la sequenza 55, 53, 55, 58, 54, 51.

7. Quanto valgono i bit di parità pari nella sequenza ASCII trovata all'esercizio precedente?

RISPOSTA: riscrivendo le codifiche trovate in formato binario a 7 bit, si trova immediatamente l'ottavo bit di parità ASCII che occupa logicamente la parte più significativa di ogni ottetto: 01010101, 01010011, 01010101, 11011000, 11010100, 11010001.

8. Nell'ambiente di programmazione Arduino un'istruzione di alto livello è eseguita dal microcontrollore con un numero variabile di istruzioni macchina.

a) E' possibile per lo sviluppatore ridurre il numero di istruzioni macchina generate da un'istruzione di alto livello?

Risposta: _____

b) Nel caso la risposta a) sia negativa si spieghi il meccanismo che impedisce di controllare il numero di istruzioni macchina generate; nel caso la risposta a) sia affermativa si descriva una metodologia per ridurre il numero di istruzioni macchina generate da un'istruzione di alto livello.

RISPOSTA:

a) sì, è possibile al contrario di quanto avviene di norma in un processore general-purpose quale ARM.

b) ci sono differenti metodologie per controllare il numero di istruzioni macchina eseguite; per esempio è possibile scrivere delle istruzioni assembly per ARM e farle eseguire al microprocessore con la specifica chiamata "asm volatile {}" oppure è possibile utilizzare le operazioni sui registri del microcontrollore messe a disposizione per il processore AVR.

9. Una memoria cache ad accesso immediato è formata da 4096 entry numerate rispettivamente da 0 a 4095, ciascuna di 16 Byte numerati rispettivamente da 0 a 15. Quale entry e quale byte occuperà il contenuto della locazione di memoria di indirizzo N (compreso tra 000000 a 999999) quando è presente in cache?

RISPOSTA: Ricordando come è strutturata una memoria cache ad accesso immediato formata da 4096 righe ciascuna lunga 16 Byte, se ad esempio è $N = 424730$ si ha che, assumendo come sempre che ogni locazione occupi un Byte, la divisione intera $N/(4096*16)$ è uguale a 6 con resto 31514. Ci chiediamo ora che byte nella cache occupi la locazione $6*4096*16 + 31514$. Ripetendo lo stesso ragionamento sulle entry, si ha che $31514/16 = 1969$ con resto 10. E dunque, la locazione in oggetto occupa il **byte 10** (numerati da 0 a 15) nella **entry 1969** (numerate da 0 a 4095).

10. Si dia un esempio schematico di **pipelining** delle operazioni, spiegando se esso ha oppure no effetti benefici sui tempi di risposta del processore.

RISPOSTA: E' sufficiente scegliere uno degli esempi presenti nei lucidi dalle lezioni, e ricordare che il pipelining accelera l'esecuzione complessiva delle istruzioni che però, essendo ciascuna sottoposta alle stesse elaborazioni effettuate da un processore privo di pipelining, individualmente non ricevono alcun beneficio in termini di velocità di esecuzione.

11 [INF]. Detta **M** la cifra **meno** significativa in N, convertire il numero $N + 2^{-M-4}$ ("N più 2 alla -M-4") in codifica *floating point IEEE 754* a 32 bit.

RISPOSTA: Essendo N intero, esso può essere convertito in un binario contenente un numero finito di cifre. D'altra parte, 2^{-M-4} è subito rappresentabile in binario come $0.0...01_2$ avendo posto M-3 zeri a destra del punto decimale. Il resto dell'esercizio segue la tradizionale procedura di conversione nella codifica *floating point* richiesta.

12 [INF]. È data la seguente mappa di Karnaugh

AB		00		01		11		10	
CD		-----							
00		1		1		S		1	

01		0		S		0		0	

11		0		1		0		0	

10		0 1 D D	

in cui S vale 0 se la cifra **più** significativa N_5 in N è pari oppure 1 se è dispari, e D vale 0 se la cifra **meno** significativa N_0 in N è pari oppure 1 se è dispari. Qual è l'espressione booleana dell'uscita E dal circuito combinatorio descritto da questa mappa?

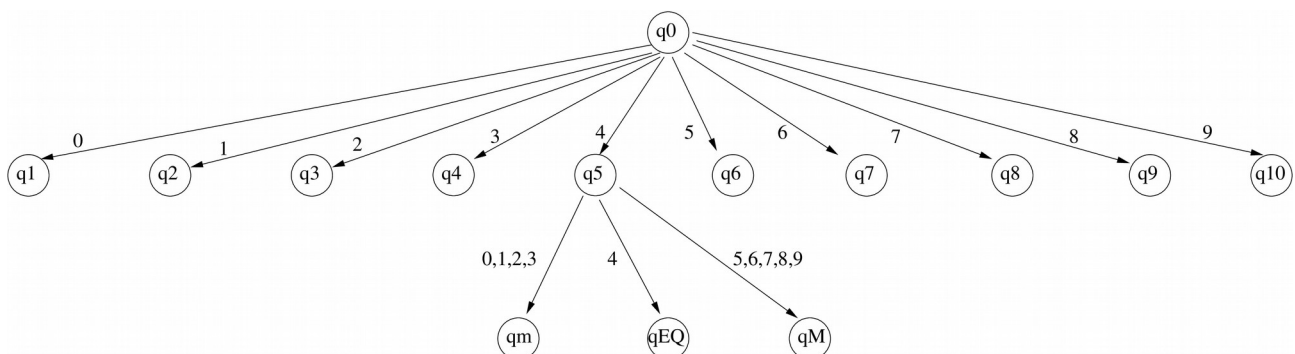
RISPOSTA: in dipendenza dai valori assunti da S e D, avremo:

- 3 coperture se S=0 e D=0: $E = A'C'D' + B'C'D' + A'BC$
- 2 coperture se S=1 e D=0: $E = A'B + C'D'$
- 4 coperture se S=0 e D=1: $E = A'C'D' + A'BC + ACD' + AB'D'$
- 3 coperture se S=1 e D=1: $E = A'B + C'D' + AD'$

Erano ovviamente accettate anche soluzioni basate sulla copertura degli zeri.

13 [INF]. Disegnare il grafo, oppure illustrare a parole il funzionamento di una macchina di Moore definita sull'insieme delle cifre decimali $I=\{0,1,\dots,9\}$, in grado di riconoscere se la prima cifra N_0 in ingresso alla macchina è **maggiore, minore o uguale** alla seconda cifra N_1 in ingresso, in cui N_0 è la cifra meno significativa in N e N_1 è la cifra immediatamente alla sua sinistra: $N = N_5N_4N_3N_2N_1N_0$. I nodi devono essere tutti etichettati in modo consistente. A questo punto si dica quali nodi la specifica macchina ha percorso durante il funzionamento.

RISPOSTA: La macchina al solito parte da uno stato iniziale q_0 , e di qui prosegue su uno stato scelto tra dieci possibili, chiamiamoli q_1, \dots, q_{10} a seconda del valore N_0 in ingresso compreso tra 0 e 9. Sia N_{0+1} l'etichetta numerica dello stato in cui la macchina è arrivata a seguito di questo ingresso. Da questo stato, tracciando gli archi corrispondenti è semplice determinare che la macchina terminerà in q_m se N_0 è minore di N_1 , in q_{EQ} se N_0 è uguale a N_1 , o infine in q_M se N_0 è maggiore di N_1 . A questo punto è immediato elencare i **tre nodi** che la macchina ha percorso in dipendenza dai dati N_0 e N_1 pervenuti all'ingresso. Nella figura sotto è indicato con completezza il solo caso in cui $N_0 = 4$, per non ingombrare il disegno con un numero eccessivo di archi.



14 [INF]. Scrivere un programma in assembly per ARM il quale accede a sei locazioni in memoria inizializzate rispettivamente con le cifre $N_0 N_1 N_2 N_3 N_4 N_5$ che compongono il numero $N = N_5 N_4 N_3 N_2 N_1 N_0$, e determina:

i) la cifra C_0 più grande tra N_0 , N_1 e N_2 (se ci sono due o tre cifre più grandi identiche il programma può scegliere C_0 tra esse a piacere)

ii) la cifra C_1 più grande tra N_3 , N_4 e N_5 (se ci sono due o tre cifre più grandi identiche il programma può scegliere C_1 tra esse a piacere)

iii) la cifra C più grande tra C_0 e C_1 (se C_0 e C_1 sono identiche il programma può sceglierne una a piacere).

Infine, il programma salva il risultato C nella memoria.

Si chiede anche, sulla base dello specifico numero N , di indicare quanti confronti tra cifre il programma in questione avrà effettuato al termine dell'esecuzione.

È gradita la presenza di commenti al codice prodotto.

```
@ ++++++ estrae la più grande di sei cifre ++++++
```

```
@ ***** data segment *****
```

```
.data
```

```
n:          .word 7,4,3,5,6,3      ; for instance N = 743563
```

```
massima:    .skip 4
```

```
@ ***** code segment *****
```

```
.text
```

```
main:
```

```
    ldr r9, =n
    ldr r5, [r9],#4      ; load N5 and point to next
    ldr r4, [r9],#4      ; load N4 digit and point to next
    ldr r3, [r9],#4      ; load N3 digit and point to next
    ldr r2, [r9],#4      ; load N2 digit and point to next
    ldr r1, [r9],#4      ; load N1 digit and point to next
    ldr r0, [r9],#4      ; load N0 digit and point to next
```

```
                                ; three-valued maximum search
```

```
    cmp r0, r1              ; if N0 < N1...
    movlt r0, r1             ; ...then copy N1 in r0
    cmp r0, r2              ; if r0 < N2
    movlt r0, r2             ; ...then copy N2 in r0
    mov r6, r0              ; store result in r6
```

```
    cmp r3, r4              ; if N3 < N4...
    movlt r3, r4             ; ...then copy N4 in r3
```

```

    cmp r3, r5           ; if r3 < N5
    movlt r3, r5         ; ...then copy N5 in r3
    mov r7, r3           ; store result in r7

                           ; two-valued maximum search

    cmp r6, r7           ; if r6 > r7...
    strgt r6, [r9]        ; ...store r6...
    strle r7, [r9]        ; ...else store r7
exit:
    swi 0x11

```

Si noti che il programma **in ogni caso** esegue $2+2+1 = 5$ confronti.