

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 - 3 pt. Si scelgano le 3 cifre non identiche più piccole in **N = N₅N₄N₃N₂N₁N₀** e, mantenendone l'ordine dato all'interno di **N**, si attribuisca al numero **M** ottenuto giustapponendo le stesse cifre la base **B** più piccola che ne permette una rappresentazione simbolicamente sensata. A questo punto si converta lo stesso numero dalla base **B** alla nuova base 11, mostrando i calcoli necessari.

RISPOSTA: per esempio se **N = 473200**, allora **M = 320** e **B = 4**. In tal caso la conversione alla base 11 si ottiene da

$$320_4 = 3 \cdot 4^2 + 2 \cdot 4^1 + 0 \cdot 4^0 = 48 + 8 + 0 = 56$$
$$56 = 5 \cdot 11^1 + 1 \cdot 11^0 = 51_{11}.$$

2 - 3 pt. Si consideri il numero **M** dell'esercizio precedente nella base **B** determinata come sopra. Quanti bit sono necessari per rappresentare in complemento a due l'intervallo numerico **[-M_B, M_B]**?

RISPOSTA: nel caso dell'esercizio precedente si aveva **M = 320₄**. Di qui si ha subito **320₄ = 111000₂**. Ricordando la necessità di un bit aggiuntivo per il segno, si nota subito che 7 bit sono sufficienti per rappresentare l'intervallo in questione.

3 - 3 pt. Un chip di memoria ha una capacità di **N** Byte. Se ogni sei anni la densità di transistor nell'unità di memoria raddoppia, quale sarà la capacità del chip tre anni dopo a parità di superficie? Mostrare i calcoli necessari.

RISPOSTA: Poiché in sei anni la capacità raddoppia, ogni anno il fattore d'incremento è uguale alla radice sesta di due. A ogni

triennio dunque la capacità cresce del cubo della radice sesta di 2, cioè $2^{3/6} = 2^{1/2} = \sqrt{2}$.

4 - 3 pt. Si consideri il numero $K = N_5N_4$ formato dalle due cifre più significative di $N = N_5N_4N_3N_2N_1N_0$. Lo si converta nel corrispondente numero binario K_2 . Infine, si realizzi lo schema di un circuito logico avente uscita uguale a 1 solo se gli ingressi formano una sequenza uguale a K_2 oppure uguale a K_2+1 .

RISPOSTA: per esempio se $N = 473200$ si ha $K = 47$ e $K_2 = 101111$. A questo punto basta realizzare di un circuito logico individuato dalla seguente tabella di verità:

ABCDEF	U
000000	0
...	0
101110	0
101111	1
110000	1
110001	0
...	0
111111	0

Il circuito si può subito realizzare per esempio in forma canonica (una porta OR a 2 ingressi di 2 porte AND a 6 ingressi).

5 - 3 pt. Quanti transistor occorrono per realizzare il circuito dell'esercizio precedente?

RISPOSTA: nel caso della realizzazione canonica il circuito adopera 5 porte NOT, 2 porte AND a 6 ingressi e una porta OR a 2 ingressi. Occorrono dunque $5 + 2 \cdot 7 + 3 = 22$ transistor.

6 - 3 pt. Si progetti un codice binario a lunghezza fissa, in grado di rappresentare un numero di lettere dell'alfabeto occidentale sufficienti per codificare il proprio nome e cognome.

RISPOSTA: ad esempio nel caso del nome e cognome del docente, il nome contiene i caratteri A,C,D,E,F,I,O,N,R,T, i quali possono essere codificati con lunghezza fissa adoperando 4 bit:

A = 0000
C = 0001
D = 0010
E = 0011
F = 0100
I = 0101

O = 0110
N = 0111
R = 1000
T = 1001

7 - 3 pt. Adoperando il codice all'esercizio precedente, durante la trasmissione del **nome e cognome** si verifica una serie di errori che casualmente mutano la sequenza, che viene viceversa ricevuta come **cognome e nome**. Si faccia vedere quali bit hanno cambiato valore.

RISPOSTA: sempre adoperando l'esempio dell'esercizio precedente, si ha:

F	E	D	E	R	I	C	O	F	O	N	T	A	N	A
010000	1100	1000	1110	0001	01000	101100	1000	1100	111100	100000	1110000			
01000	1100	111100	100000	111000	001000	01100	1000	11100	00101000	0101000	10110			
F	O	N	T	A	N	A	F	E	D	E	R	I	C	O

8 - 3 pt. E' possibile bloccare via software gli interrupt nel microcontrollore di Arduino? Motivare la risposta.

RISPOSTA: sì, per esempio a livello globale impostando il bit 7 del registro SREG. Le istruzioni per cambiare il valore di tale bit sono rispettivamente cli() [clear interrupt] per disabilitare gli interrupt, e sei() [set interrupt] per ripristinare lo stato precedente all'arrivo di un'interruzione.

9 - 3 pt. In un'architettura a 32 bit si consideri una memoria cache con 1024 linee di 32 byte associativa a **K** vie, in cui **K = N₅** è la cifra più significativa di **N = N₅N₄N₃N₂N₁N₀**. Quanti **bit** occupa questa memoria? Mostrare i calcoli necessari:

RISPOSTA: Tenendo conto del campo TAG che in nel caso specifico occupa 17 bit più il bit di validità, entrambi presenti in ogni via, l'occupazione totale è di $K * 1024 * (32 * 8 + 17 + 1)$ bit.

10 - 3 pt. In un processore provvisto di un'unica ALU il tempo per leggere il contenuto di un registro è di 2 μs e il tempo di scrittura in un registro è di 6 μs. Se non si tiene conto dei tempi ulteriori necessari allo svolgimento un'istruzione macchina, quanto impiega nel migliore dei casi il processore in questione per scambiare il contenuto di due registri?

RISPOSTA: Nel migliore dei casi il processore deve i) leggere il registro A, ii) scrivere il suo contenuto in un registro ausiliario C, iii) leggere il registro B, iv) scrivere il

contenuto di B in A, v) leggere il registro C, vi) scrivere il contenuto di C in B. In totale occorrono $2+6+2+6+2+6 = 22 \mu s$.

11 [INF] - 3pt. Considerate le sei cifre $MN_4N_3N_2N_1N_0$ che formano il numero $N = MN_4N_3N_2N_1N_0$, convertire il numero $-N_4 - 2 - 8^{-M}$ ("meno N_4 meno 2 meno 8 alla $-M$ ") in codifica *floating point IEEE 754* a 32 bit. Mostrare i calcoli.

RISPOSTA: Il numero in questione è uguale a $-(N_4+2+8^{-M})$. Quindi, può essere convertito in un binario negativo contenente una parte intera uguale a N_4+2 e una parte frazionaria uguale a 8^{-M} , subito rappresentabile in binario come $0.0...01_2$ avendo posto $3 \cdot M - 1$ zeri a destra del punto decimale. Il resto dell'esercizio segue la tradizionale procedura di conversione nella codifica *floating point* richiesta.

12 [INF] - 3 pt. È data la seguente mappa di Karnaugh

AB	00	01	11	10
CD	-----			
00	M	1	1	M
01	0	0	0	0
11	P	0	0	P
10	M	0	0	M

in cui P vale 0 se la cifra **più** significativa N_5 in N è pari oppure 1 se è dispari, e M 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 P e M, avremo:

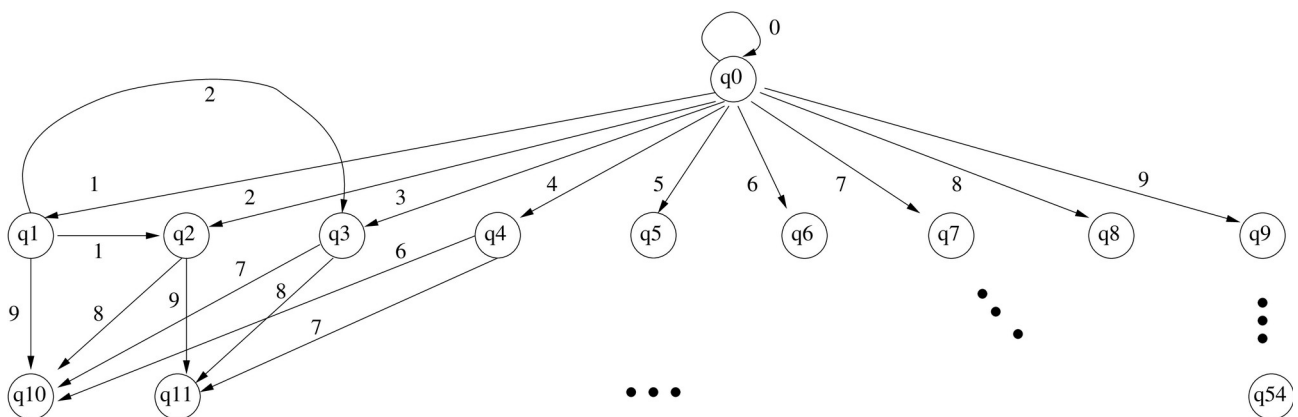
- 1 copertura di "uni" se $P=0$ e $M=0$: $E = BC'D'$
- 2 coperture di "uni" se $P=1$ e $M=0$: $E = B'CD + BC'D'$
- 2 coperture di "uni" se $P=0$ e $M=1$: $E = B'D' + C'D'$
- 2 coperture di "uni" se $P=1$ e $M=1$: $E = B'C + C'D'$

Erano ovviamente accettate anche soluzioni basate sulla copertura degli zeri.

13 [INF] - 3 pt. Disegnare le parti significative del grafo, oppure illustrare a parole il funzionamento di una macchina di Moore definita sull'insieme delle cifre decimali $I=\{0,1,...,9\}$, in grado di calcolare la somma decimale delle cifre da N_0 a N_5 in $N = N_5N_4N_3N_2N_1N_0$ via via ricevute in ingresso. 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: Etichettiamo ogni stato come qS , in cui S è la somma delle cifre che la macchina ha ricevuto in ingresso fino a quel momento. La macchina al solito parte da uno stato iniziale $q0$, e di qui prosegue su uno stato scelto tra dieci possibili $q0, \dots, q9$ a seconda del valore N_0 in ingresso compreso tra 0 e 9. A questo punto, sulla base del valore N_1 procede su uno degli stati tra $q0$ e $q18$ sulla base della somma $N_1 + N_0$. Il processo continua identico fino a quando la sesta cifra N_5 non è arrivata. Dopo l'arrivo della sesta cifra la macchina sarà necessariamente in uno stato compreso tra $q0$ e $q54$, rappresentativo della somma di tutte le cifre ricevute in ingresso. Nella figura sotto è indicata solo una parte del grafo della macchina, per non ingombrare il disegno con un numero eccessivo di nodi e archi.



14 [INF] - 9 pt. Scelta la cifra più grande B in $N = N_5N_4N_3N_2N_1N_0$, scrivere un programma in assembly per ARM il quale converte N , inteso in base 10, nella nuova base B . Le cifre che costituiscono il numero nella nuova base sono scritte nel verso corretto in locazioni successive di memoria appartenenti a un array, via via che sono calcolate. É gradita la presenza di commenti al codice prodotto.

RISPOSTA:

```
@ ++++++++ conversione di N in base B ++++++++
```

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

```
.data
```

```
n:      .word 63 ; 473200
```

```
b:      .word 2 ; 7
```

```
result: .skip 20      ; contains binary digits until value 2^20
```

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

```
.text
```

```
main:
```

```

        ldr r4, =result+20-1      ; load result array bottom in r4
        ldr r1, =n                 ; load number address in r1
        ldr r1, [r1]               ; load number in r1
        ldr r2, =b                 ; load base address in r2
        ldr r2, [r2]               ; load base in r2
loop_1: mov r6, #0                 ; reset result in r6
loop_0: cmp r1, r2                 ; ...
        subge r1, r1, r2           ; if n>=b then n <= n-b
        addge r6, r6, #1           ; if n>=b increment result
        bge loop_0                 ; if n>=b repeat loop_0
        strltb r1, [r4], #-1       ; if n<b byte-store remainder
        movlt r1, r6               ; if n<b copy previous result
        cmp r6, #0                 ; if result == 0 ...
        beq exit                   ; ... then exit
        b loop_1                   ; else start again
exit:
        swi 0x11

```