

ESERCIZIO N°1

5 punti

Determinare la rappresentazione in base 2, 3, 5, 12 e 16 del proprio numero di matricola M .

ESERCIZIO N°2

8 punti

Realizzare una subroutine terne per il microcontrollore AVR XMEGA256A3BU che determina quante terne di byte consecutivi nello spazio di memoria dalla locazione 0x2000 alla locazione 0x20BB contengono nel digit meno significativo $\{b_3, b_2, b_1 \text{ e } b_0\}$ i codici di una terna di valori BCD appartenenti a cifre consecutive nel proprio numero di matricola (partendo dalla meno significativa; quindi se M vale per esempio 543210 le terne di cifre da considerare sono 012; 123; 234 e 345). Il risultato deve essere lasciato in R0. Vanno considerate anche terne eventualmente interallacciate.

ESERCIZIO N°3

5 punti

- a) Usando blocchi logici noti (porte logiche elementari, full-adder, mux e demux...) realizzare una rete in grado di sottrarre in virgola fissa a 8 bit un numero in traslazione da uno C1, fornendo il risultato in modulo e segno.
- b) Realizzare la rete che valuta l'overflow nella sottrazione precedente.

ESERCIZIO N°4

5 punti

Disegnare il grafo di flusso e quindi lo schema logico corrispondente di una macchina sequenziale sincrona secondo il modello di Moore con 1 ingresso, in grado di riconoscere la sequenza $\{m_2, m_5, m_7, m_{11}\}$ e anche la sequenza $\{m_{11}, \overline{m_5}, m_9, m_2\}$ non interallacciate in alcun modo. La macchina deve essere dotata di reset asincrono, che la riporta in uno stato di partenza. Non è richiesta ottimizzazione. I valori m_0, m_1, \dots sono le cifre binarie di M .

ESERCIZIO N°5

5 punti

Sintetizzare in forma normale ottima SP e disegnare lo schema logico della una rete combinatoria a 5 ingressi (X_4, X_3, X_2, X_1, X_0) individuata dai valori di uscita, corrispondenti alla sequenza ordinata degli ingressi, pari al valore corrispondente delle prime 32 cifre binarie di $(123456789 + M^2)$.

ESERCIZIO N°6

5 punti

Disegnare lo schema logico di un contatore $\overline{\text{UP/DOWN}}$ modulo N con abilitazione E, ove $N = (9 + |M|_7)$.

① Matricola di esempio 654321

BASE 16 : 8FBF1 (calcolatrice)

BASE 2: 1001'1111'1011'1111'0001 (dall'hex)

BASE 3: 1020020120010

BASE 5: 131414241

BASE 12: 2767A9

Metodologia: con la calcolatrice divisioni progressive con troncamento e poi, a ritroso, i resti

Es. base 3

654321		0
218107		1
72702		0
24234		0
8078		2
2692		1
897		0
299		2
99		0
33		0
11		2
3		0
1		1

↑
cifre

2

Realizzare una subroutine terne per il microcontrollore AVR XMEGA256A3BU che determina quante terne di byte consecutivi nello spazio di memoria dalla locazione 0x2000 alla locazione 0x20BB contengono nel digit meno significativo $\{b_3, b_2, b_1 \text{ e } b_0\}$ i codici di una terna di valori BCD appartenenti a cifre consecutive nel proprio numero di matricola (partendo dalla meno significativa; quindi se M vale per esempio 543210 le terne di cifre da considerare sono 012; 123; 234 e 345). Il risultato deve essere lasciato in R0. Vanno considerate anche terne eventualmente interallacciate.

terne:

```
push R16 //registri per i valori dalla memoria
push R17
push R18
push R19 //contatore
push R20 //registri per i numeri della matricola
push R21
push R22
push R23
push R24
push R25
push XL //puntatore alla memoria
push XH
clr R0
ldi XL,low(0x2000)
ldi XH,high(0x2000)
ldi R19,0xBB-0x00+1-2 //il +1 per gli estremi, il -2 perché terne
ldi R20,1 //caso della matricola 654321, terne tutte diverse
ldi R21,2
ldi R22,3
ldi R23,4
ldi R24,5
ldi R25,6
ld R17,X+
andi R17,0x0F
ld R18,X+ //inizializzo due dei registri
andi R18,0x0F
loop:
  mov R16,R17 //il valore all'indirizzo minore esce, gli altri scorrono
  mov R17,R18
  ld R18,X+ //il valore preso dalla memoria va per indirizzo maggiore
  andi R18,0x0F //prendo il digit meno significativo
  rcall trovaterna //non indispensabile, ma rende più pulito il codice
  dec R19
  brne loop
pop XH
pop XL
pop R25
pop R24
pop R23
pop R22
pop R21
pop R20
pop R19
pop R18
pop R17
pop R16
ret
```

trovaterna: //caso delle terne tutte diverse

```
cp R16,R20
cpc R17,R21
cpc R18,R22
breq incrementa
cp R16,R21
cpc R17,R22
cpc R18,R23
breq incrementa
cp R16,R22
cpc R17,R23
cpc R18,R24
breq incrementa
cp R16,R23
cpc R17,R24
cpc R18,R25
breq incrementa
ret //uscita senza aver trovato la terna
```

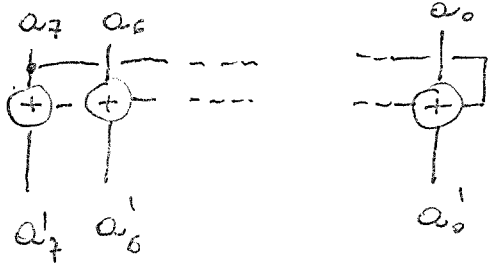
incrementa: //uscita dopo aver trovato la terna

```
inc R0
ret
```

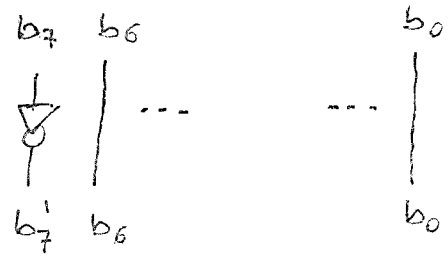
3

Converto tutto in C2. Sottraggo. Penso da C2 a MS.
 Si può avere overflow nella sottrazione oppure nella conversione finale

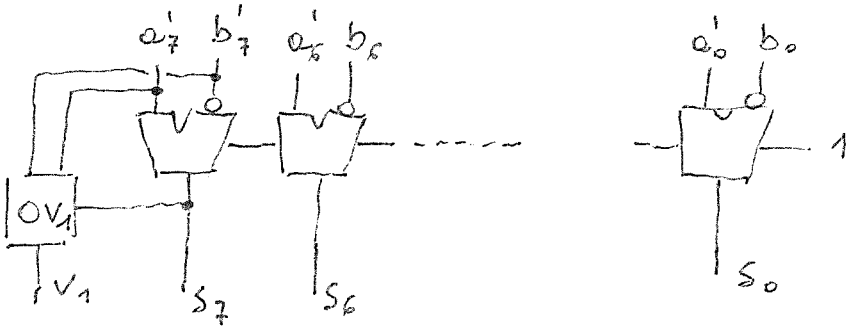
A (C1 → C2)



B (T → C2)

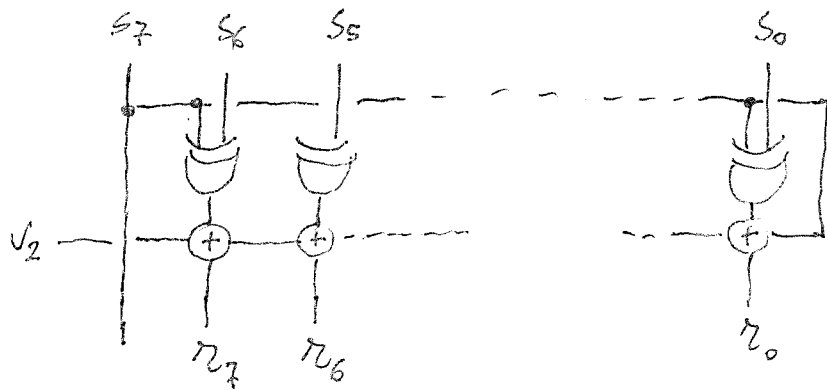


Sottrattore

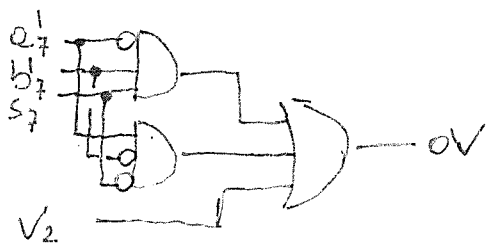


$$V_1 = \overline{a'_7} b'_7 s_7 + a'_7 \overline{b'_7} s_7$$

Conversione C2 → MS



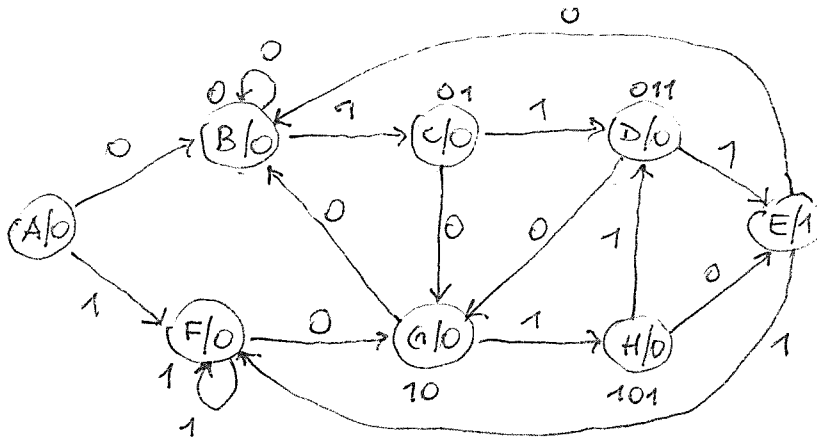
Si ha carry out solo nel caso 10000000, che non è convertibile in MS (-128)



④ sequenze NON interscacciate (... 10111 1110001)
 $m_{11} m_9 m_7 m_5 m_2 m_0$

0111 e 1010

grafo



codifica (verb.)

	$Q_2 Q_1 Q_0$	U
iniz. A	000	0
B	001	0
C	010	0
D	011	0
E	111	1
F	110	0
G	101	0
H	100	0

$U = Q_2 Q_1 Q_0$
 rete per l'uscita

Mappe delle transizioni

$Q_2 \backslash N$	$Q_1 Q_0$			
	00	01	11	10
00	001	001	101	101
01	110	010	111	011
11	011	100	110	110
10	111	001	001	101

$Q_2 \backslash N$	$Q_1 Q_0$			
	00	01	11	10
00	0	0	1	1
01	1	0	1	0
11	0	1	1	1
10	1	0	0	1

$$D_2 = \overline{1N} \overline{Q_2} Q_1 + 1N \overline{Q_2} \overline{Q_1} \overline{Q_0} + \overline{Q_2} Q_1 Q_0 + 1N Q_2 Q_0 + 1N Q_2 Q_1 + \overline{1N} Q_2 \overline{Q_0}$$

$Q_2 \backslash N$	$Q_1 Q_0$			
	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	0	1	1
10	1	0	0	0

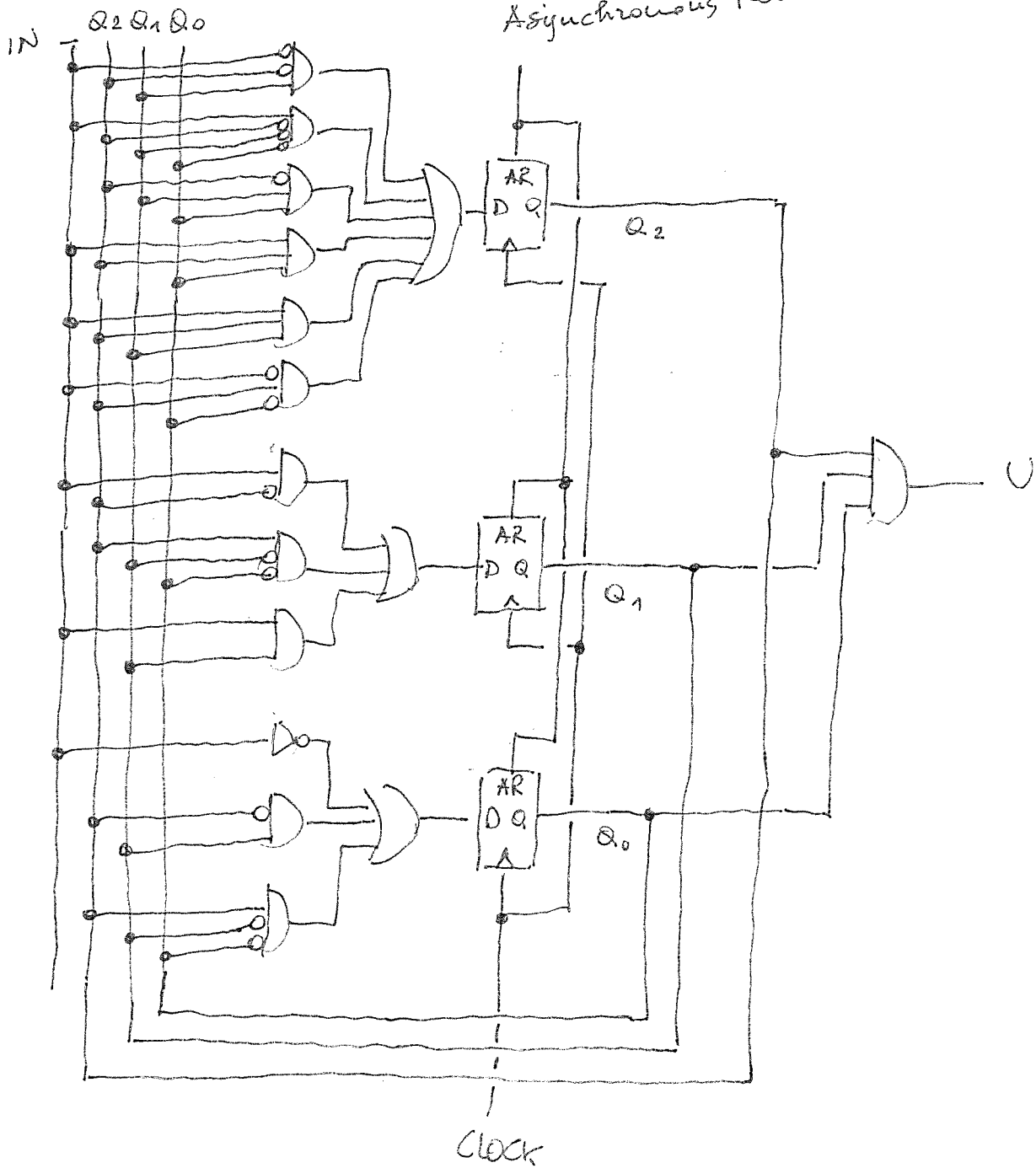
$$D_1 = 1N \overline{Q_2} + Q_2 \overline{Q_1} \overline{Q_0} + 1N Q_1$$

$Q_2 \backslash N$	$Q_1 Q_0$			
	00	01	11	10
00	1	1	1	1
01	0	0	1	1
11	1	0	0	0
10	1	1	1	1

$$D_0 = \overline{1N} + \overline{Q_2} Q_1 + Q_2 \overline{Q_1} \overline{Q_0}$$

Scheme logics

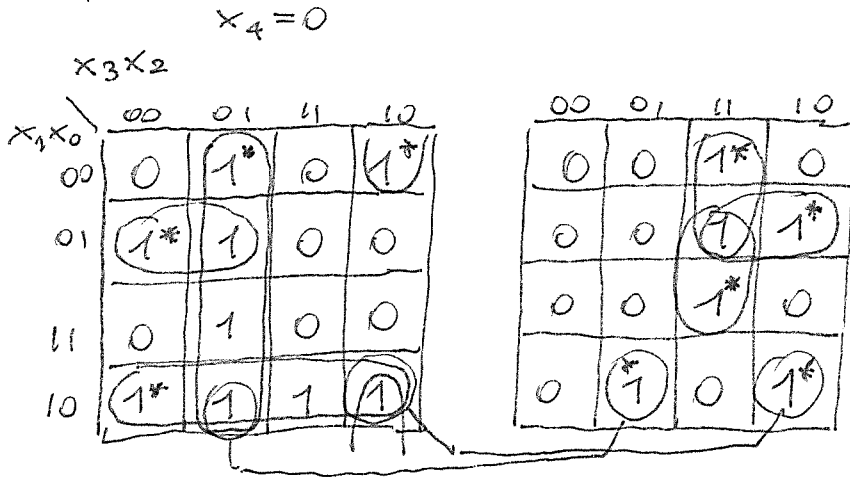
Asynchronous Reset



⑤ Le 8 cifre esadecimali meno significative di $M^2 + 123456789$ sono B64045F6

(il calcolo si può fare anche con calcolatrici semplici, purché siano in grado di gestire correttamente numeri interi fino a 4 miliardi)

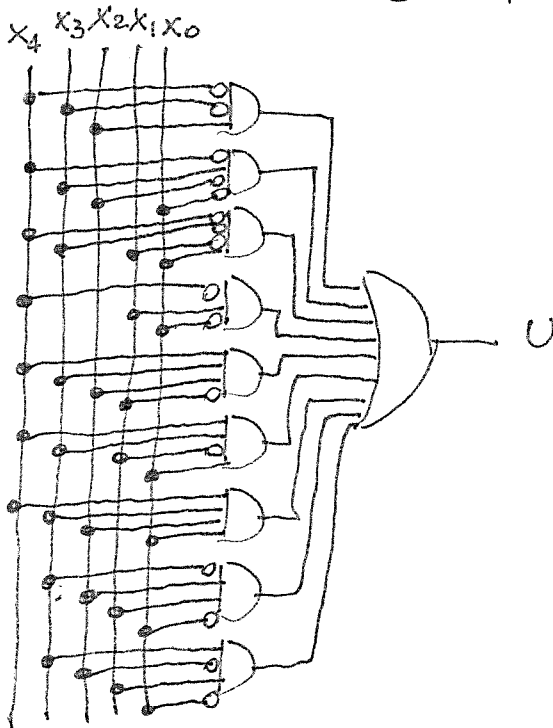
Mappe



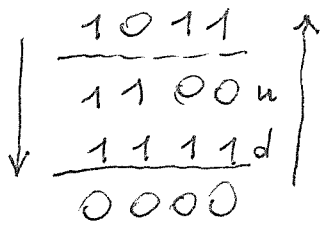
hex column 6 F 4 5 0 4 B 6

Siccome ottime $\Sigma \Pi$: tutti implicanti ESSENZIALI

$$\begin{aligned}
 U = & \bar{x}_4 \bar{x}_3 x_2 + \bar{x}_4 x_3 \bar{x}_2 \bar{x}_0 + \bar{x}_4 \bar{x}_3 \bar{x}_1 x_0 + \bar{x}_4 x_1 \bar{x}_0 + \\
 & + x_4 x_3 x_2 \bar{x}_1 + x_4 x_3 \bar{x}_1 x_0 + x_4 x_3 x_2 x_0 + \bar{x}_3 x_2 x_1 \bar{x}_0 + \\
 & + x_3 \bar{x}_2 x_1 \bar{x}_0
 \end{aligned}$$



6) Contatore UP/DOWN modulo 12



si può prendere FB --
per entrambi

$\sim B$ -- down (oppure F)
FB -- up

