

SCHEDA ASE2503		Data: 10 Febbraio 2025
Cognome	Nome	

ESERCIZIO N°1

8 punti

Realizzare un sottoprogramma per il microcontrollore AVR XMEGA256A3BU, in grado di invertire una stringa s_x (scambiare il primo con l'ultimo carattere, il secondo con il penultimo e così via). Una stringa, la cui massima lunghezza può essere di 255 caratteri (codici ASCII su 8 bit), è composta da un primo byte che contiene l'indicazione della lunghezza della stringa stessa (nel caso di stringa vuota l'indicazione di lunghezza è 0) seguito dai codici dei caratteri che compongono la stringa. La stringa s_x è posta in memoria a partire dall'indirizzo contenuto in X.

ESERCIZIO N°2

4 punti

Determinare la rappresentazione nel formato binary 32 dello standard IEEE754-2008 dei seguenti numeri (nel caso di valori non rappresentabili esattamente, si scelga la rappresentazione che meglio approssima il valore proposto):

99!

$-e^{3e}$

$\log_2(99!)$

$-7/2^{99}$

ESERCIZIO N°3

5 punti

Realizzare in forma NOR-NOR ottima una rete combinatoria in grado di valutare il bit di peso 0 nella somma tra due numeri espressi in MS su 3 bit. Anche la somma deve essere espressa in MS.

ESERCIZIO N°4

6 punti

Disegnare lo schema circuitale di un contatore up/down modulo 11 con abilitazione.

ESERCIZIO N°5

5 punti

Disegnare lo schema logico di un sommatore a 3 bit secondo l'approccio carry look-ahead. Si hanno a disposizione porte elementari, half-adder e full-adder.

ESERCIZIO N°6

5 punti

Realizzare una macchina sequenziale sincrona secondo il modello di Mealy sincronizzato con 4 ingressi, rappresentanti le cifre binarie di una cifra esadecimale, e 1 uscita, in grado di evidenziare la presenza in ingresso di una sequenza (non interallacciata) di 3 valori maggiori strettamente di 9. Non è richiesta l'ottimizzazione della codifica degli stati.

/*Realizzare un sottoprogramma per il microcontrollore AVR XMEGA256A3BU, in grado di invertire una stringa sX. Una stringa, la cui massima lunghezza può essere di 255 caratteri (codici ASCII su 8 bit), è composta da un primo byte che contiene l'indicazione della lunghezza della stringa stessa (nel caso di stringa vuota l'indicazione di lunghezza è 0) seguito dai codici dei caratteri che compongono la stringa. La stringa sX è posta in memoria a partire dall'indirizzo contenuto in X.*/

```
string_invert:
    push R2
    push R16
    push R17
    push R18
    push XL
    push XH
    push YL
    push YH
    ld R16,X+           //in R16 lunghezza di stringa
    cpi R16,2
    brcs fine         //stringa nulla o carattere singolo
    clr R2            //registro di appoggio nullo
    movw YH:YL,XH,XL //replica il puntatore
    add YL,R16
    adc YH,R2        //Y punta oltre la fine della stringa
    lsr R16          //numero di inversioni
loop:
    ld R17,X
    ld R18,-Y
    st X+,R18       //copia i caratteri scambiati
    st Y,R17
    dec R16
    brne loop
fine:
    pop YH
    pop YL
    pop XH
    pop XL
    pop R18
    pop R17
    pop R16
    pop R2
    ret
```

2

legge di rappresentazione per binary 32 normalizzati

$$x = (-1)^s 2^{(E-127)} (1 + T 2^{-23})$$

99! è oltre il range di rappresentabilità - $(9,33 \times 10^{155})$
può essere rappresentato come +∞

s	E	T
0	11111111	000000000000000000000000

$$-e^{3e} \approx -3480,20154 \dots \quad (-1) \cdot 2^{11} \cdot (1 + 5866298 \cdot 2^{-23})$$

s	E	T
1	10001010	101100110000001100111010

$$\log_2(99!) = \ln(99!) / \ln(2) = 518,121137 \dots \quad 2^9 (1 + 100289 \cdot 2^{-23})$$

s	E	T
0	10001000	000000110000011111000001

$$-7 \cdot 2^{-99} = (-1) 2^{-97} \cdot 1,75$$

s	E	T
0	00011110	1100000000000000000000

* NB: chi non riesce a valutare 99! con la calcolatrice, può eseguire la valutazione con

$$[\ln(69!) + \ln(70 \cdot 71 \cdot \dots \cdot 99)] / \ln(2)$$

3

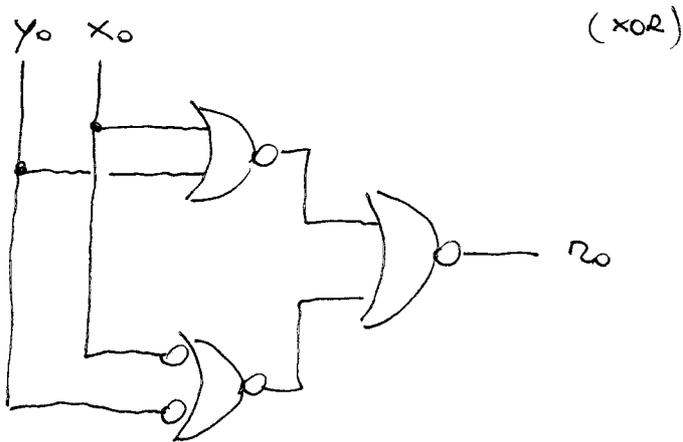
Sul risultato possiamo avere effetto solo i segni e i bit di peso 0 degli addendi.

(Anzi, i segni non modificano r_0)

$x_0 y_0$	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

$$\begin{array}{r} s_x x_1 x_0 + \\ s_y y_1 y_0 \\ \hline s_r r_2 r_1 r_0 \end{array}$$

$$r_0 = (x_0 + y_0)(\bar{x}_0 + \bar{y}_0) = \overline{(x_0 + y_0)} + (\bar{x}_0 + \bar{y}_0)$$



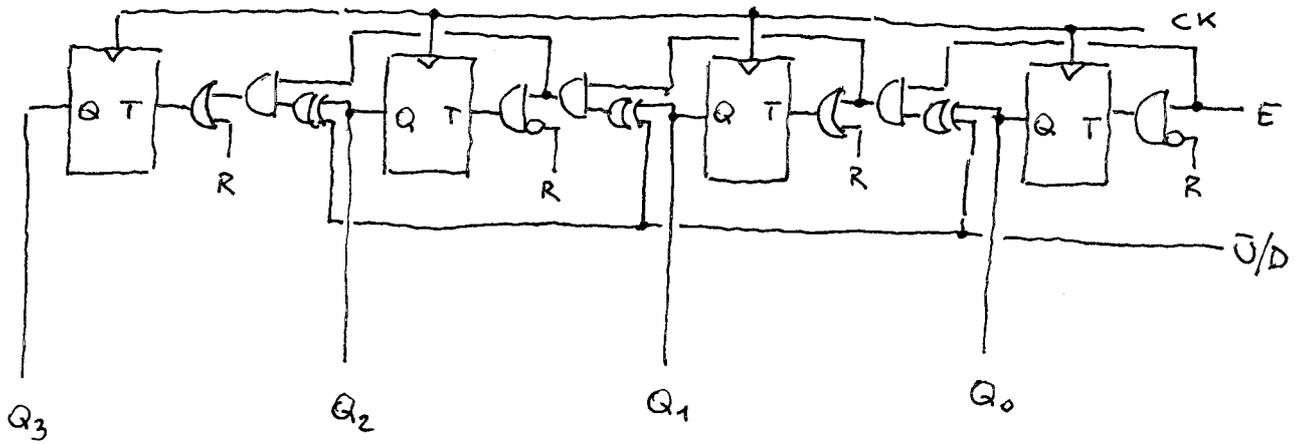
④

Sequenza

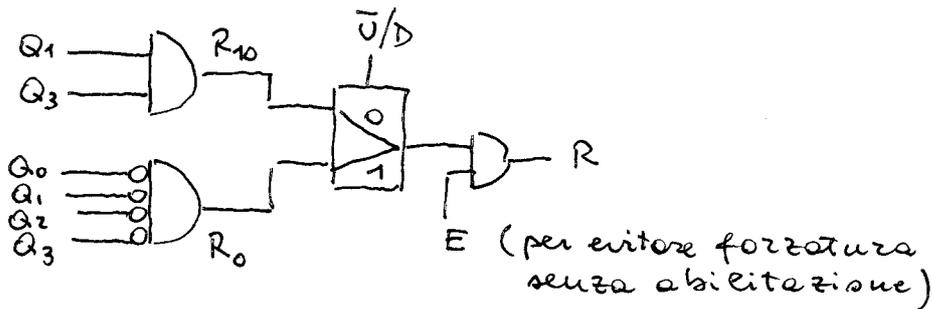
UP 1 0 0 1
 1 0 1 0
 ↓ - 1 0 1 1 -
 - 1 1 1 1 - ↑
 0 0 0 0 DOWN

F B F B questo intervento va bene per entrambe le direzioni di conteggio

Uso come base un contatore \bar{U}/D mod 16 con Enable



Rete per la valutazione di R (fine conteggio)



5

$$G_0 = A_0 B_0$$

$$P_0 = \emptyset \text{ (non c'è carry in ingresso)}$$

$$G_1 = A_1 B_1$$

$$C_{in1} = G_0$$

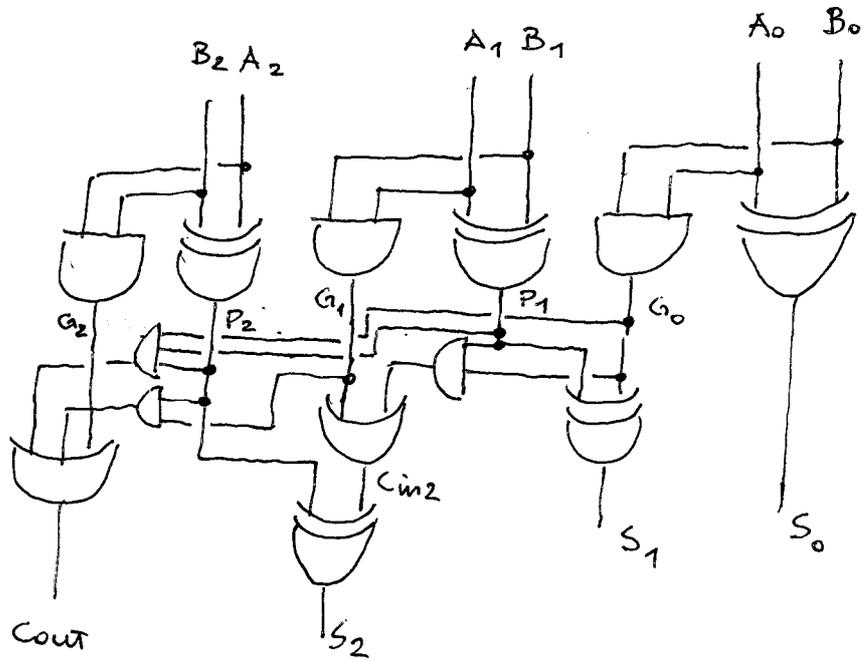
$$P_1 = A_1 \oplus B_1$$

$$G_2 = A_2 B_2$$

$$C_{in2} = G_1 + P_1 G_0$$

$$P_2 = A_2 \oplus B_2$$

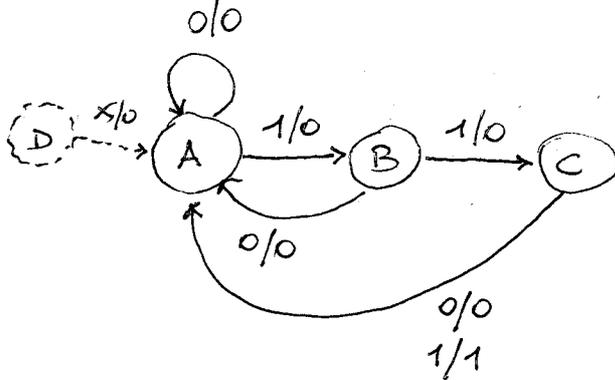
$$C_{out} = G_2 + P_2 G_1 + P_2 P_1 G_0$$



6

la variabile in ingresso (cifre esadecimali) può essere confrontata subito con 9 e originale l'ingresso effettivo della rete (sequenza non interrelacciata di 3 valori 1)

A: $N > 9$



Codifica

A 00
B 01
C 11
q, q₀

q₁q₀

A	00	01	11	10
0	00/0	00/0	00/0	-
1	01/0	11/0	00/1	-

Rete confronto per A

x ₃ x ₂	00	01	11	10
x ₁ x ₀	00	0	1	0
01	0	0	1	0
11	0	0	1	1
10	0	0	1	1

$$q_1^+ = \bar{q}_1 q_0 A$$

$$q_0^+ = \bar{q}_1 A$$

$$U = q_1 q_0 A$$

$$A = x_3 x_2 + x_3 x_1$$

