

SCHEDA ASE1907		Data: 15 Luglio 2019
Cognome	Nome	

ESERCIZIO N°1

8 punti

Realizzare una subroutine `substring_clear` per il microcontrollore AVR XMEGA256A3BU che elimina N caratteri (N contenuto in R16) dalla stringa puntata da X a partire dal valore indicato in R17 (che contiene il numero di caratteri da mantenere dall'inizio, nella stringa originale). La stringa è costituita da un primo byte che contiene la lunghezza della stringa stessa (da 0 per la stringa nulla a un massimo di 255) seguito dai caratteri codificati ASCII su 8 bit (quindi 1 byte per carattere). La stringa risultante, che deve essere aggiornata anche per quanto riguarda il nuovo valore della lunghezza, deve sostituire quella originale.

ESERCIZIO N°2

5 punti

Disegnare lo schema logico in forma normale a minimo numero di letterali (scegliendo la migliore tra SP e PS) della rete combinatoria che calcola il valore di pilotaggio del segmento "e" in un decoder BCD-7segmenti.

ESERCIZIO N°3

5 punti

Realizzare un decoder BCD 7segmenti usando un decoder 4:16 e porte OR oppure NOR. Minimizzare il numero complessivo degli ingressi delle porte utilizzate.

ESERCIZIO N°4

5 punti

Disegnare lo schema logico di un contatore sincrono up/down modulo 12 con Reset sincrono.

ESERCIZIO N°5

6 punti

Progettare una macchina di Moore con un ingresso e una uscita, che viene posta a 1 dopo che sono arrivati complessivamente 5 ingressi a 1. Dopo aver riconosciuto la condizione di set, la macchina ritorna nello stato iniziale.

ESERCIZIO N°6

4 punti

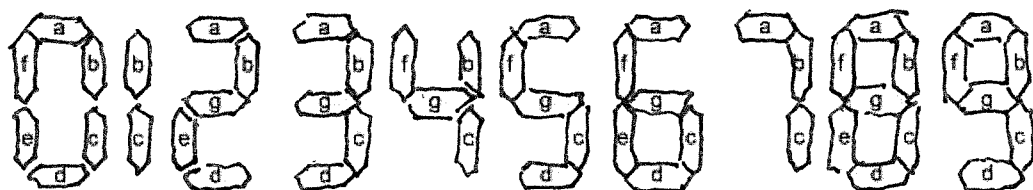
Determinare la rappresentazione nel formato binary 32 (IEEE754) dei seguenti numeri, arrotondando il valore effettivo al codice più vicino, e poi valutare l'errore assoluto con segno definito da $(\hat{x} - x)$ con 3 (strettamente) cifre significative

$$-250!/240!$$

$$4\log_{11}(e^{500})$$

$$-2^{-130}$$

$$\sqrt[56]{(8344608)^{49}}$$



1

Realizzare una subroutine `substring_clear` per il microcontrollore AVR XMEGA256A3BU che elimina N caratteri (N contenuto in R16) dalla stringa puntata da X a partire dal valore indicato in R17 (che contiene il numero di caratteri da mantenere dall'inizio, nella stringa originale). La stringa è costituita da un primo byte che contiene la lunghezza della stringa stessa (da 0 per la stringa nulla a un massimo di 255) seguito dai caratteri codificati ASCII su 8 bit (quindi 1 byte per carattere). La stringa risultante, che deve essere aggiornata anche per quanto riguarda il nuovo valore della lunghezza, deve sostituire quella originale.

```
substring_clear:
    push R2 //registro di appoggio nullo
    push R18
    push R20
    push R22
    push XL //puntatore alla stringa
    push XH
    push YL //puntatore alla regione da spostare
    push YH
    clr R2 //registro nullo di appoggio
    ld R18,X //lunghezza della stringa: L
    cp R17,R18 //se P>=L non occorre fare nulla e la stringa resta immutata
    brcc end
    sub R18,R17 //caratteri che si possono eliminare: M=L-P
    cp R16,R18 //se N>=M non occorre spostare nulla e P è la nuova lunghezza
    st X,R17
    brcc end
    sub R18,R16 //lunghezza della coda da spostare C
    mov R20,R18 //la usa come contatore per lo spostamento
    add R18,R17 //lunghezza effettiva della nuova stringa C+P
    st X+,R18
    add XL,R17 //puntatore all'inizio della zona da eliminare sommando P
    adc XH,R2
    movw YH:YL,XH:XL //replica il puntatore
    add YL,R16 //lo sposta all'inizio del pezzo da tirare su sommando N
    adc YH,R2
loop: //riporta in alto il pezzo di stringa che non va eliminato
    ld R22,Y+
    st X+,R22
    dec R20
    brne loop
end:
    pop YH //ripristina i puntatori e i registri
    pop YL
    pop XH
    pop XL
    pop R22
    pop R20
    pop R18
    pop R2
    ret
```

2

la mappa è (per il segmento "e")

	$x_3 x_2$			
	00	01	11	10
$x_1 x_0$	00	01	11	10
00	1 ⁰	0 ⁴	-	1 ²
01	0 ¹	0 ⁵	-	0 ⁹
11	0 ³	0 ⁷	-	-
10	1 ²	1 ⁶	-	-

SP

1	0	-	1
0	0	-	0
0	0	-	-
1	1	-	1

2 implicanti di ordine 2

4 lettere

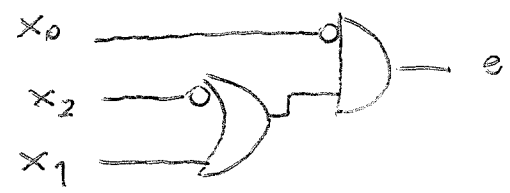
PS

1	0	-	1
0	0	-	0
0	0	-	-
1	1	-	-

1 implicato di ordine 3
1 implicato di ordine 2

3 lettere: OTTIMO

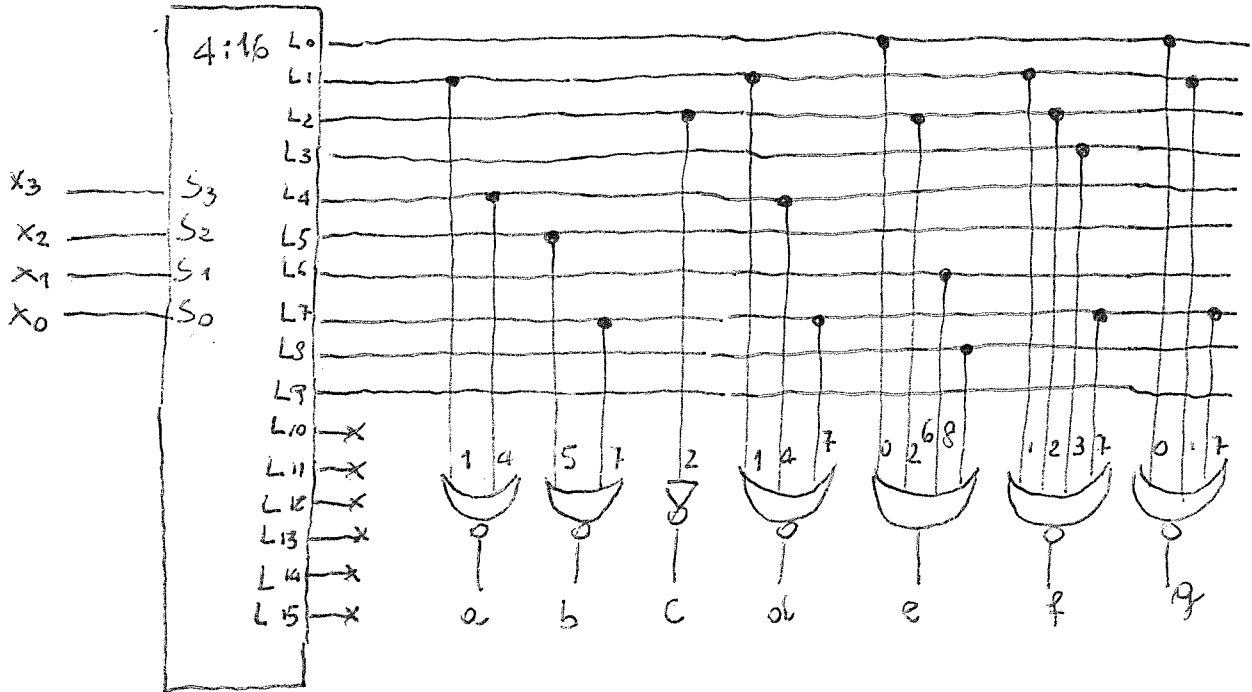
$$e = \bar{x}_0 \cdot (\bar{x}_2 + x_1)$$



③ Decoder BCD-7seg u

il decoder 4:16 calcola i MINTERMINI

sceglie OR / NOR per individuare 1 / 0 della funzione di uscita MINIMIZZANDO i termini di ingresso da usare

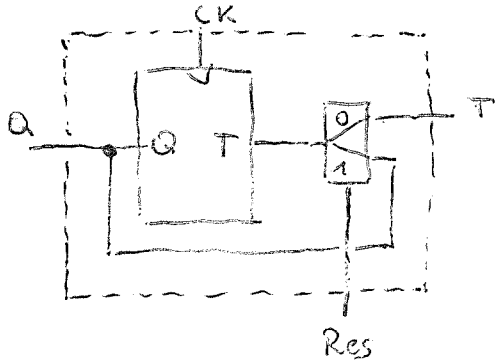


in totale si hanno 19 ingressi.

Le linee che corrispondono a codici NON VALIDI (10..15) vanno lasciate SCOLLEGATE

4 Contatore up/down modulo 12 con Reset sincrono

Uso T-FF con reset come elemento base

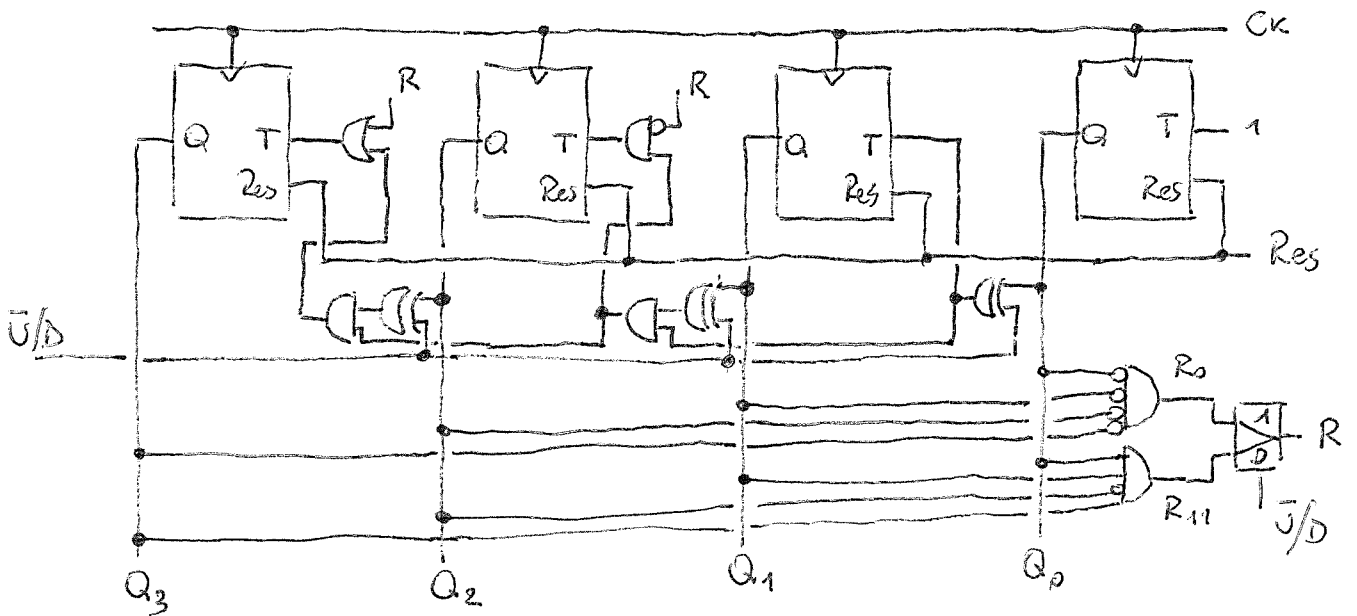


Sequenza
 :
 1 0 1 0
 1 0 1 1

 ↓ ↑ 1 1 0 0 UP
 ↓ ↑ 1 1 1 1 DOWN

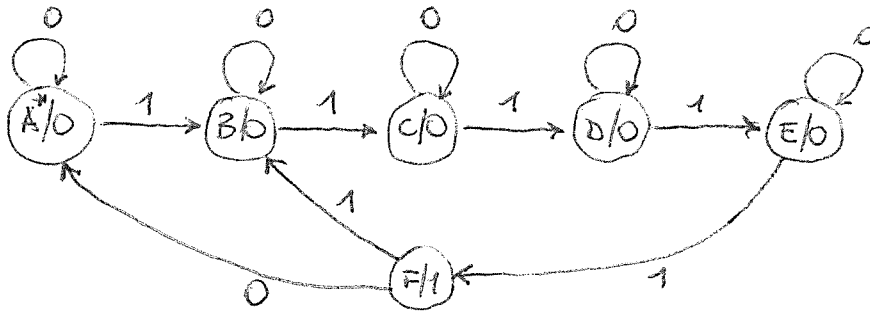
 0 0 0 0
 0 0 0 1
 :
 :

(FB --) Comandi buoni per entrambi le direzioni



5

GRAZO



codifica

	q_2	q_1	q_0	U
A	0	0	0	0
B	0	0	1	0
C	0	1	1	0
D	0	1	0	0
E	1	1	0	0
F	1	0	0	1

Rete per U

q_2	q_1	q_0	
	00	01	11
0	0	0	0
1	1	1	0

$$U = q_2 \bar{q}_1$$

Rete per lo stato futuro

q_2	q_1	q_0	
	00	01	11
0	000	001	011
1	001	011	010
1	001	-	-
0	000	-	-

0	0	0	0
0	0	0	1
0	-	-	1
0	-	-	1

$$d_2 = q_1 \bar{q}_0 (N + q_2)$$

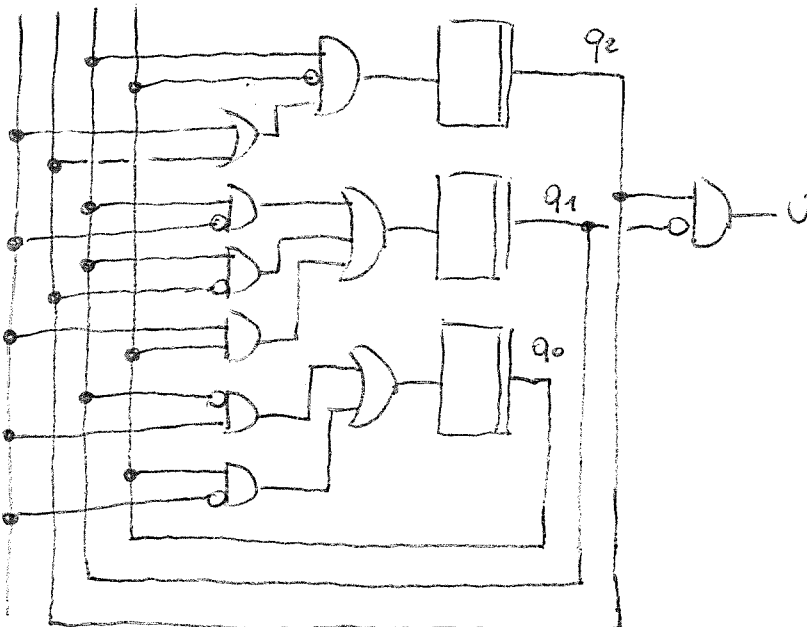
0	0	1	1
0	1	1	1
0	-	-	0
0	-	-	1

$$d_1 = q_1 \bar{N} + q_1 \bar{q}_2 + q_0 \bar{N}$$

0	1	1	0
1	1	0	0
1	-	-	0
0	-	-	0

$$d_0 = \bar{q}_1 \bar{N} + q_0 \bar{N}$$

\bar{N} q_2 q_1 q_0



6

Legge di rappresentazione NORMALIZZ.

$$x = (-1)^S 2^{(E-127)} (1 + T \cdot 2^{-23}) \quad E \neq 0; E \neq 255$$

NON NORMALIZZ.

$$x = (-1)^S 2^{-126} \cdot T \cdot 2^{-23} = (-1)^S T 2^{-149} \quad E = \phi$$

A) $-(250 \cdot 249 \cdot 248 \dots \cdot 241) \approx -7,947265 \dots \cdot 10^{23}$ (usare tutta la precisione della calc.)

$$= -2^{79} \cdot \left(1 + \frac{2640438 \cdot 2^{-23}}{284436 \text{ hex}}\right)$$

$$S \left| \begin{array}{c} E \\ 1 \mid 11001110 \end{array} \right| \begin{array}{c} T \\ 010.1000.0100.1010.0011.0110 \end{array}$$

$$E = -2,83 \cdot 10^{16}$$

B) $4 \log_{11}(e^{500}) = \frac{2000}{\ln 11} \approx 834,0648 \dots$

$$= 2^9 \left(1 + \frac{5276709 \cdot 2^{-23}}{508425 \text{ hex}}\right)$$

$$0 \mid 10001000 \mid 101.0000.1000.0100.0010.0101$$

$$E = -2,45 \cdot 10^{-5}$$

C) -2^{130} non normalizz. $T = 2^{19}$

$$1 \mid 00000000 \mid 000.1000.0000.0000.0000.0000$$

$E = \phi$ (non ha senso parlare di cifre significative in questo caso)

D) $8344608^{49/50} \approx 1138230 \dots = 2^{20} \cdot \left(1 + \frac{717236 \cdot 2^{-23}}{AF1B4 \text{ hex}}\right)$

$$0 \mid 10010011 \mid 000.1010.1111.0001.1011.0100$$

$$E = 0,0560$$