

SCHEDA <b>D15_05</b>		Data: <b>11 Giugno 2015</b>
Cognome	Nome	Matricola

### ESERCIZIO N°1

7/4 punti

Progettare una porta logica AOI CMOS che abbia la seguente tabella di verità (Z indica lo stato di alta impedenza e X un valore logico indeterminato causato dalla simultanea conduzione della sezione PMOS e NMOS). Si hanno a disposizione le variabili A, B e C affermate e negate.

A	B	C	U
0	0	0	0
0	0	1	Z
1	0	0	X
1	0	1	Z
0	1	0	1
0	1	1	1
1	1	0	X
1	1	1	0

Determinare quindi la corrente erogata dall'alimentazione nei due casi in cui l'uscita è indeterminata. Si hanno a disposizione NMOS e PMOS con le seguenti caratteristiche:  $V_{Tn} = |V_{Tp}| = 1 \text{ V}$ ;  $k_n = |k_p| = 4 \text{ mA/V}^2$ . Inoltre è  $V_{DD} = 4 \text{ V}$ .

### ESERCIZIO N°2

6/4 punti

Determinare se la seguente espressione logica rappresenta una tautologia.

$$(A+B)(\bar{C}+\bar{D}+E)(\bar{A}+B+\bar{E})(\bar{A}+C+\bar{E})=B\bar{D}+A\bar{D}\bar{E}+B\bar{C}\bar{E}+A\bar{C}\bar{E}+\bar{A}BE+BCE$$

### ESERCIZIO N°3

6/6 punti

Disegnare lo schema elettrico di una porta logica AOI TTL che esegua la funzione

$$U = AB + CD$$

### ESERCIZIO N°4

6/4 punti

Realizzare una macchina sequenziale sincrona secondo il modello di Mealy sincronizzato, con 2 ingressi e 1 uscita che viene posta a 1 (dopo il clock) ogni volta che un ingresso è pari all'ingresso precedente aumentato di 1 (modulo 4).

### ESERCIZIO N°5

8/5 punti

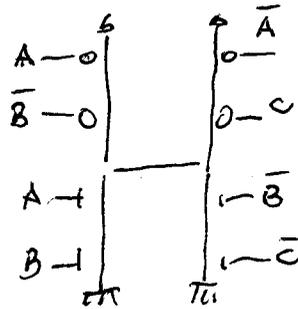
Realizzare un programma in linguaggio assembly per il microcontrollore XMEGA256A3BU, che legga continuamente il numero binario assoluto x posto in ingresso alla porta virtuale 0 e ponga in uscita sulla porta virtuale 1 la parte intera inferiore di  $256/(x + 10)$ . Il programma non deve prevedere l'inizializzazione delle periferiche.

parte n

$\overline{A}\overline{B}\overline{C}$  }  $\overline{BC}$   
 $A\overline{B}\overline{C}$  }  
 $A\overline{B}C$  }  $AB$   
 $A\overline{B}C$  }

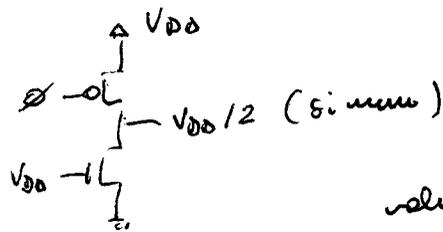
parte p

$\overline{A}BC$  }  $\overline{AC}$   
 $A\overline{B}C$  }  $A\overline{B}$   
 $A\overline{B}\overline{C}$  }  
 $\overline{A}\overline{B}C$  }



Casi 100 2 e 2 per entrambi  
110

$$k_{eq} = \frac{1}{2} k = 2 \text{ mA/V}^2$$



valuto per uMOS

$$I_{DD} = \frac{k_{eq}}{2} \frac{V_{DD}}{2} \left( V_{DD} + \frac{V_{DD}}{2} - 2V_T \right) =$$

$$2(4) = 8 \text{ mA}$$

2

	AB			
CD	00	01	11	10
00	0	1	1	
01	0			
11	0	0	0	0
10	0	1	1	

E=0

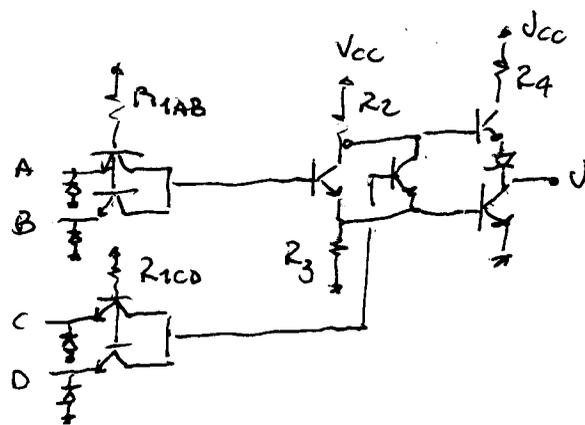
	AB			
CD	00	01	11	10
00	0	1	1	0
01	0		0	0
11	0			0
10	0	1	1	0

E=1

(m0) : 0000  
 A=1  
 B=1  
 C=0  
 D=0  
 E=1

Sx=1  
 Dx=0

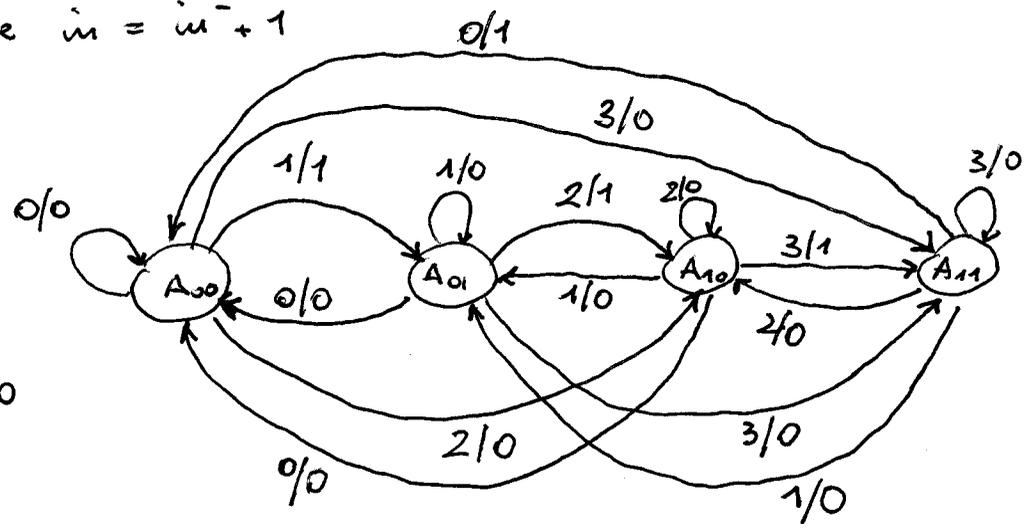
3



4

Se  $i_n = i_{n-1} + 1$

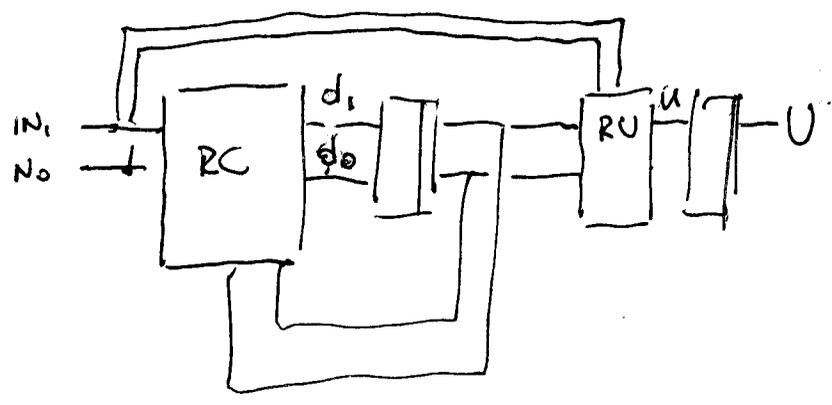
GRAFO



Codici

- A<sub>00</sub> 00
- A<sub>01</sub> 01
- A<sub>10</sub> 10
- A<sub>11</sub> 11
- q<sub>1</sub> q<sub>0</sub>

Architettura



RC

$d_1 = IN_1$   
 $d_0 = IN_0$

RU

q<sub>1</sub> q<sub>0</sub>

	00	01	11	10
00	0	1	0	0
01	0	0	0	1
11	1	0	0	0
10	0	0	1	0

Summa di 4 mintermini

U

$$U = \bar{q}_1 \bar{q}_0 \bar{IN}_1 IN_0 + \bar{q}_1 q_0 IN_1 \bar{IN}_0 + q_1 \bar{q}_0 IN_1 IN_0 + q_1 q_0 \bar{IN}_1 \bar{IN}_0$$

## 5

```
/* Realizzare un programma in assembly per XMEGA256A3BU, che legga continuamente
   il numero binario assoluto x posto in ingresso alla porta virtuale 0
   e ponga in uscita sulla porta virtuale 1 la parte intera inferiore di
   256/(x + 10).
   Il programma non deve prevedere l'inizializzazione delle periferiche.
*/
```

```
main:
rcall configure // viene data per buona
iteration:
  clr R17
  in R16,VP0RT0_IN
  subi R16,-10 // somma 10; in realtà sottrae 246
  brne cont // se x+10=256 in uscita ci va 1
  inc R17
  rjmp fine
cont:
brcc fine // se non carry vuol dire che x>246 e il risultato sarà 0
/*
Ora si vede quante volte ci sta il divisore nel dividendo, che inizialmente
è 0, in quanto 256 non è rappresentabile su 8 bit
*/
mov R18,R16
neg R18 // la prima sottrazione nega x+10
loop
  inc R17 //conta quante volte ci sta
  sub R18,R16
  brcc loop
fine:
out VP0RT1_OUT,R17
rjmp iteration //ripete continuamente

configure:
... //qui ci va l'inizializzazione delle porte
ret
```