

Il testo va riconsegnato

ESERCIZIO N°1

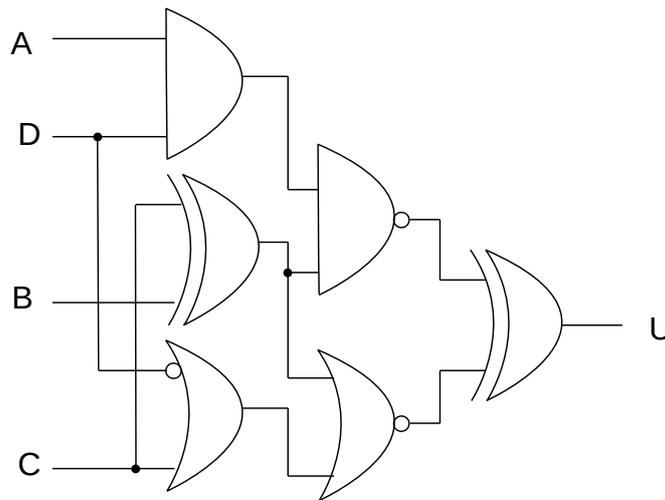
8 punti

Scrivere un sottoprogramma nel linguaggio assembly della famiglia XMEGA AVR che scrive nelle locazioni di memoria da 0x3000 a 0x33AB (compresi gli estremi) i valori in sequenza multipli di 7, a partire da 0, modulo 256 (0, 7, 14, 21, ecc.).

ESERCIZIO N°2

6 punti

Realizzare in forma NOR-NOR ottima la seguente rete combinatoria a 4 ingressi e una uscita; indicare quali implicati sono essenziali, motivando la scelta.



ESERCIZIO N°3

6 punti

Spiegare la differenza tra le istruzioni MUL, MULS e MULSU del linguaggio assembly della famiglia AVR ed esprimere come valore intero con segno in base 10 il valore del risultato delle tre diverse istruzioni (dove è contenuto?) quando gli argomenti sono due registri il cui contenuto è rispettivamente 0xBF e 0xAC.

ESERCIZIO N°4

6 punti

Progettare una rete di Moore a due ingressi (A e l'abilitazione E) e 3 uscite (Q₂, Q₁ e Q₀) in grado di generare (se abilitata) una sequenza in cui il valore successivo è pari al valore precedente più 7, se l'ingresso è 0, e invece è pari al valore precedente meno 3 se l'ingresso è 1. Le somme e le differenze si intendono modulo 8.

ESERCIZIO N°5

7 punti

Individuare il valore della tensione di uscita di una porta NOT CMOS nei due casi, quando l'ingresso vale 0 e quando vale V_{DD}. Il carico della porta è costituito da un resistore da 1 kΩ collegato a massa.

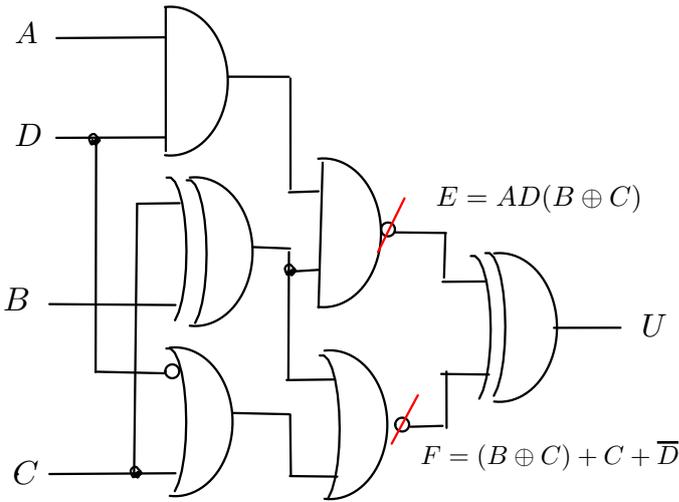
(V_{DD} = 3,3 V; V_{Tn} = |V_{Tp}| = 0,8 V; k_n = |k_p| = 4 mA/V²)

```
/*Scrivere un sottoprogramma nel linguaggio assembly della famiglia  
XMEGA AVR che scrive nelle locazioni di memoria da 0x3000 a 0x33AB  
(compresi gli estremi) i valori in sequenza multipli di 7,  
a partire da 0, modulo 256 (0, 7, 14, 21, ecc.).*/
```

```
write_7n:  
  push R16           //variabile di appoggio  
  push XL  
  push XH  
  ldi XL,low(0x3000)  
  ldi XH,high(0x3000)  
  clr R16           //mette in R16 il valore iniziale  
loop:  
  st X+,R16  
  subi R16,-7       //valore da caricare nella prossima cella  
  cpi XL,low(0x33AB+1)  
  brne loop  
  cpi XH,high(0x33AB+1)  
  brne loop  
  pop XH  
  pop XL  
  pop R16  
ret
```

2

Realizzare in forma NOR-NOR ottima la seguente rete combinatoria a 4 ingressi e una uscita; indicare quali implicati sono essenziali, motivando la scelta.



Posso modificare lo schema, eliminando le NOT. Trovo le mappe di E ed F e quindi la mappa di U.

$$E = AD(B \oplus C)$$

$$F = (B \oplus C) + C + \bar{D}$$

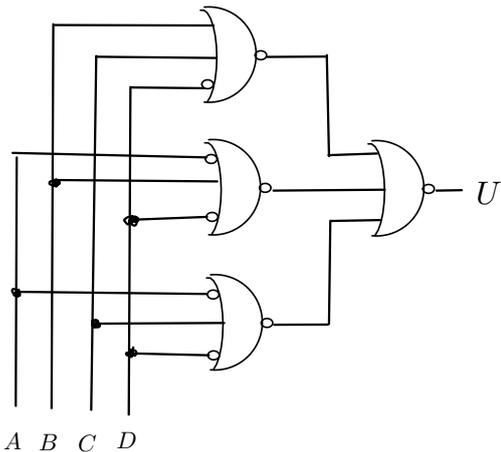
$$U = (E \oplus F)$$

AD \ BC		00	01	11	10
		00	0	0	0
01		0	0	1	0
11		0	0	0	0
10		0	0	1	0

AD \ BC		00	01	11	10
		00	1	0	0
01		1	1	1	1
11		1	1	1	1
10		1	1	1	1

AD \ BC		00	01	11	10
		00	1	*0	0
01		1	1	*0	1
11		1	1	1	1
10		1	1	0	1

Forma NOR-NOR



3

Spiegare la differenza tra le istruzioni MUL, MULS e MULSU del linguaggio assembly della famiglia AVR ed esprimere come valore intero con segno in base 10 il valore del risultato delle tre diverse istruzioni (dove è contenuto?) quando gli argomenti sono due registri il cui contenuto è rispettivamente 0xBF e 0xAC.

Le tre istruzioni eseguono la moltiplicazione tra grandezze contenute in due registri e pongono il risultato su 16 bit nella coppia di registri R1:R0.

Gli operandi vengono interpretati diversamente dalle 3 operazioni e presentano pure delle limitazioni.

MUL Rd,Rr

Gli operandi sono numeri binari senza segno e possono essere registri qualsiasi

MULS Rd,Rr

Gli operandi sono numeri con segno C2 e possono essere registri nel range R16-R31

MULSU Rd,Rr

Rd è un numero con segno (C2) e Rr senza segno.

Gli operandi possono essere registri nel range R16-R23

In tutti i casi viene modificato il flag C, che assume il valore del bit più significativo del risultato.

I valori assegnati, in base 10, corrispondono ai numeri senza segno

Rd=0xBF=191; Rr=0xAC=172

con segno

Rd=0xBF=-65; Rr=0xAC=-84

quindi

MUL Rd,Rr // in R1:R0 viene 32852 (0x8054, negativo considerato in C2: -32684)

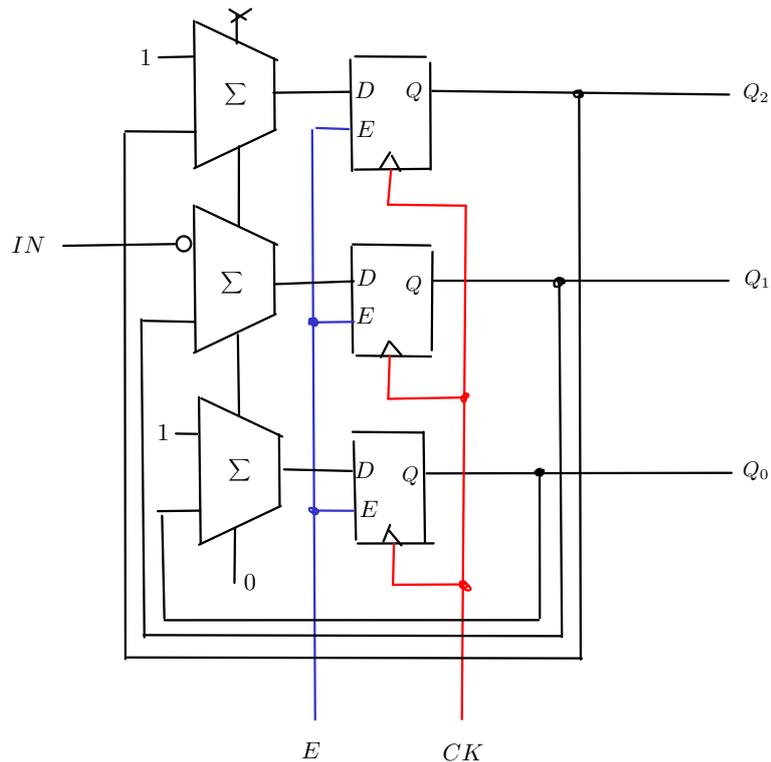
MULS Rd,Rr // in R1:R0 viene 5460

MULSU Rd,Rr // in R1:R0 viene -11180

4

Progettare una rete di Moore a due ingressi (A e l'abilitazione E) e 3 uscite (Q2, Q1 e Q0) in grado di generare (se abilitata) una sequenza in cui il valore successivo è pari al valore precedente più 7, se l'ingresso è 0, e invece è pari al valore precedente meno 3 (ovvero più 5 considerando il modulo 8) se l'ingresso è 1. Le somme e le differenze si intendono modulo 8.

Possiamo costruire la rete di Moore direttamente dalla definizione, considerando l'uscita uguale allo stato presente e valutando lo stato futuro come richiesto dal testo. Per soddisfare il requisito sull'abilitazione, usiamo DE-FF.

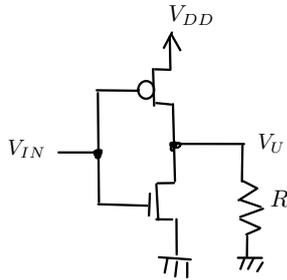


5

Individuare il valore della tensione di uscita di una porta NOT CMOS nei due casi, quando l'ingresso vale 0 e quando vale VDD.

Il carico della porta è costituito da un resistore da 1 kΩ collegato a massa.

$$V_{DD} = 3,3 \text{ V}; \quad V_{Tn} = |V_{Tp}| = 0,8 \text{ V}; \quad k_n = |k_p| = 4 \text{ mA/V}^2$$



Quando l'ingresso $V_{IN}=V_{DD}$, l'nMOS è triodo e il pMOS è interdetto. L'uscita è nulla e nel resistore non scorre alcuna corrente (quindi non ha alcun effetto).

Quando $V_{IN}=0$, il pMOS è acceso (hp: triodo) e l'nMOS interdetto.

Il pMOS non può essere saturo altrimenti si avrebbe una V_U troppo alta, non accettabile. Infatti, in saturazione:

$$I_{SDp} = \frac{|k_p|}{2}(V_{IN} - V_{DD} - V_{Tp})^2 = 12,5 \text{ mA}$$

Scriviamo l'equazione KCL al nodo di uscita con il pMOS triodo, usando V_U come incognita.

$$I_{SDp} = \frac{|k_p|}{2}(V_U - V_{DD})(2V_{IN} - V_{DD} - V_U - 2V_{Tp}) = \frac{V_U}{R}$$

Risolvendo si ottengono 2 soluzioni (una è negativa e non accettabile), di cui quella valida è

$$V_U = 2,982$$

Compatibile con l'ipotesi di pMOS triodo.