

SCHEDA D08_05		Data: 05 Giugno 2008	
Cognome	Nome	Matricola	

ESERCIZIO N°1

7 punti

Determinare la tensione di uscita V_U di un invertitore CMOS ϵ vuoto, alimentato con $V_{DD} = 3,3$ V, con tensione di ingresso $V_{IN} = 1,1$ V ($V_{Tn} = -V_{Tp} = 0,5$ V; $K_n = -K_p = 1$ mA/V²).

ESERCIZIO N°2

6 punti

Determinare se la seguente espressione booleana è una identità.

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

ESERCIZIO N°3

6 punti

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

ESERCIZIO N°4

6 punti

Si deve realizzare una memoria da 1Gx32 avendo a disposizione moduli da 512Mx9 e 256Mx5. Indicare quanti moduli servono di ciascun tipo e come eseguire l'assemblaggio.

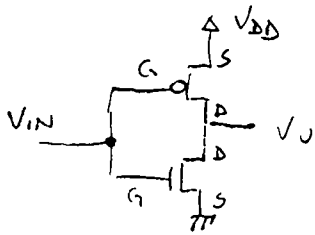
ESERCIZIO N°5

8 punti

Scrivere un sottoprogramma nel linguaggio assembly della famiglia AVR che esegue la somma (senza segno) di 8 byte contenuti in 8 locazioni di memoria consecutive. L'indirizzo del primo byte è contenuto in X. Il risultato, valutato su 16 bit, va posto in due locazioni consecutive, con la parte meno significativa che precede quella più significativa, all'indirizzo contenuto in Y. Il sottoprogramma non deve alterare i registri di lavoro.

①

I due transistori hanno caratteristiche simmetriche.
Con $V_{IN} < V_{DD}/2$ avremo p triodo e n saturo



$$I_{DSM} = \frac{k_{N1}}{2} (V_{IN} - V_{TM})^2$$

$$I_{DSP} = \frac{k_{P}}{2} (V_{O} - V_{DD}) (V_{IN} - V_{DD} + V_{IN} - V_{O} - 2V_{TP})$$

Equazione alle correnti nel nodo di uscita ($V_{O} = x$)

$$0,36 = (x - 3,3)(-0,1 - x)$$

$$x^2 - 3,2x + 0,03 = 0$$

$$x = 1,6 + \sqrt{2,53} = 3,19 \text{ V}$$

2) Visto che non è immediato trovare un controesempio dell'affermazione proposta, verifichiamo tutti i casi identificando le mappe associate ai due membri dell'identità

	AB				
DE	00	01	11	10	
00	1		1	1	$A\bar{D}\bar{E}$
01	1	1		1	
11	1			1	
10	1			1	$\bar{A}\bar{D}\bar{E}$

forme SP $\bar{A}\bar{D}\bar{E}$ \bar{B}

	AB				
DE	00	01	11	10	
00		0			
01			0		$\sim(\bar{A}+\bar{B}+E)$
11		0	0		
10		0	0		

$(\bar{B}+\bar{D})$ $(A+\bar{B}+E)$
forme PS

Conclusione: poiché le mappe coincidono è un'identità

3

La macchina potrà essere realizzata con 8 stati, identificati dal valore dell'ingresso ultimo arrivato e da quello dell'uscita. Possiamo individuare la tabella di flusso

Stato	IN ($x_1 x_0$)				Out	Codifica $q_2 q_1 q_0$
	0	1	2	3		
S_{00}	S_{00}	S_{11}	S_{20}	S_{30}	0	0 0 0
S_{01}	S_{00}	S_{11}	S_{20}	S_{30}	1	0 0 1
S_{10}	S_{00}	S_{10}	S_{21}	S_{30}	0	0 1 0
S_{11}	S_{00}	S_{10}	S_{21}	S_{30}	1	0 1 1
S_{20}	S_{00}	S_{10}	S_{20}	S_{31}	0	1 0 0
S_{21}	S_{00}	S_{10}	S_{20}	S_{31}	1	1 0 1
S_{30}	S_{01}	S_{10}	S_{20}	S_{30}	0	1 1 0
S_{31}	S_{01}	S_{10}	S_{20}	S_{31}	1	1 1 1

Out = q_0

Reti per lo stato futuro RC

$q_1 q_0$	$x_1 x_0$			
	00	01	11	10
00	0 0 1 1			
01	0 0 1 1			
11	0 0 1 1			
10	0 0 1 1			

0 1 1 0
0 1 1 0
0 1 1 0
0 1 1 0

$q_1 q_0$	$x_1 x_0$			
	00	01	11	10
00	0 1 0 0			
01	0 1 0 0			
11	0 0 0 1			
10	0 0 0 1			

$q_2 = 0$

$q_1 q_0$	$x_1 x_0$			
	00	01	11	10
00	0 0 1 1			
01	0 0 1 1			
11	0 0 1 1			
10	0 0 1 1			

0 1 1 0
0 1 1 0
0 1 1 0
0 1 1 0

0 0 1 0
0 0 1 0
1 0 0 0
1 0 0 0

$q_2 = 1$

$d_2 = x_1$

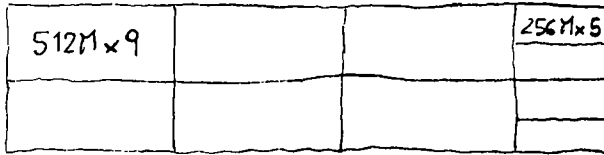
$d_1 = x_0$

$$d_0 = \bar{x}_1 \bar{x}_0 \bar{q}_1 \bar{q}_2 + x_1 \bar{x}_0 q_1 \bar{q}_2 + \bar{x}_1 \bar{x}_0 q_1 q_2 + x_1 x_0 \bar{q}_1 q_2$$

Architettura



④ Assemblaggio complessivo

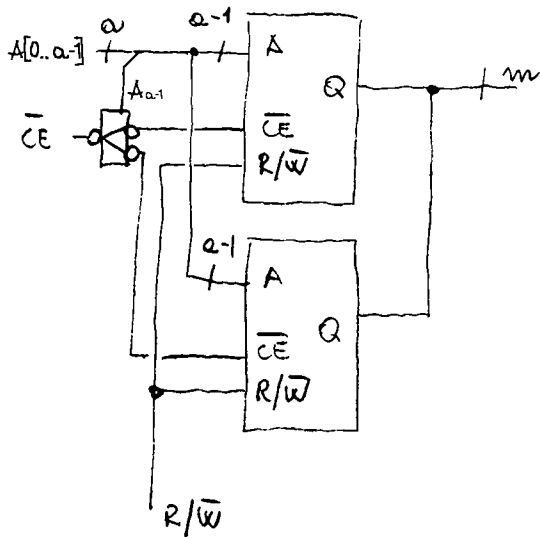


In totale servono

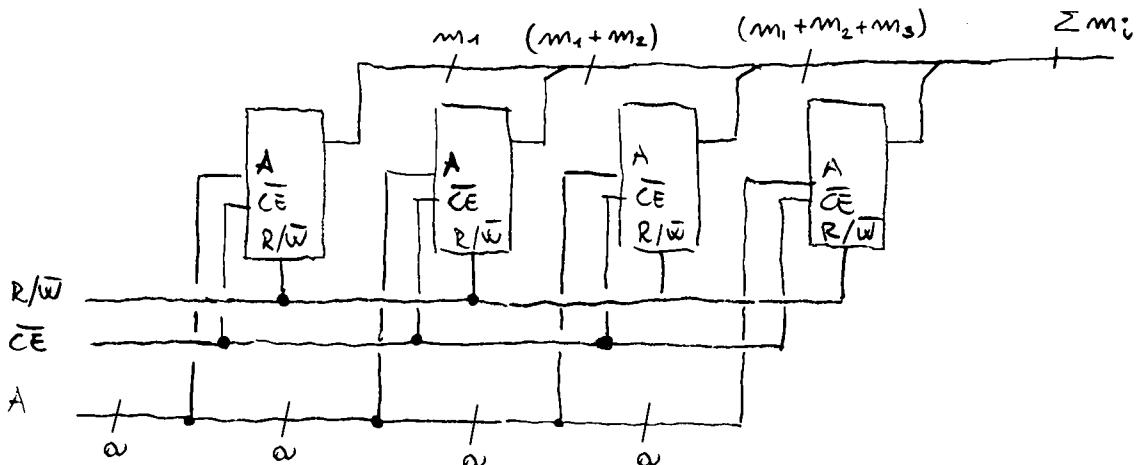
6 memorie 512Mx9
4 memorie 256Mx5

- 1) Uso coppie da 256Mx5 per avere 512Mx5 $\left. \begin{array}{l} m=5 \\ a=29 \end{array} \right\}$
- 2) Estendo le parole a 512Mx32
- 3) Raddoppio il numero di parole a 1Gx32 $\left. \begin{array}{l} m=32 \\ a=30 \end{array} \right\}$

Raddoppio del n° di parole da m



Somma del n° di bit



5

Sub:

```
PUSH R16 ; salva registri
PUSH R17
PUSH R18
PUSH R19
PUSH R20
PUSH XL
PUSH XH
PUSH YL
PUSH YH
```

```
LDI R20, 8 ; contatore
CLR R16
CLR R17
CLR R19
```

loop:

```
LD R18, X+
ADD R16, R18 ; sistema l'eventuale carry
ADC R17, R19
DEC R20
BRNE loop ; sistema il risultato
ST Y+, R16
ST Y+, R17
```

```
POP YH ; ripristina
POP YL
POP XH
POP XL
POP R20
POP R19
POP R18
POP R17
POP R16
RET
```