

Cognome

Nome

Matricola

**ESERCIZIO N°1**

7 punti (4)

Determinare la corrente di uscita  $I_U$  di una porta NOR CMOS che pilota il catodo di un diodo LED ideale ( $V_{Don} = 2$  V costante, anodo a  $V_{DD}$ ), alimentata con  $V_{DD} = 5$  V, con tensioni di ingresso  $V_{IN1} = 1$  V e  $V_{IN2} = 2$  V ( $V_{Tn} = -V_{Tp} = 1$  V;  $K_n = -K_p = 2$  mA/V<sup>2</sup>). Scambiando tra loro i due ingressi, si modifica il valore della corrente?

**ESERCIZIO N°2**

5 punti (4)

Realizzare in forma ottima PS una rete combinatoria in grado di identificare (ponendo 1 in uscita) tutti i numeri primi tra 48 e 63. Gli ingressi della rete sono costituiti dalle 4 cifre binarie meno significative del numero.

(Definizione: "A **prime number** is a natural number greater than 1 that has no positive divisors other than 1 and itself").

**ESERCIZIO N°3**

6 punti

Realizzare una macchina sequenziale sincrona secondo il modello di Mealy sincronizzata con 1 ingresso e 1 uscita che segnali ponendo 1 in uscita l'assenza di commutazioni in ingresso per 2 o più cicli di clock.

**ESERCIZIO N°4**

7 punti

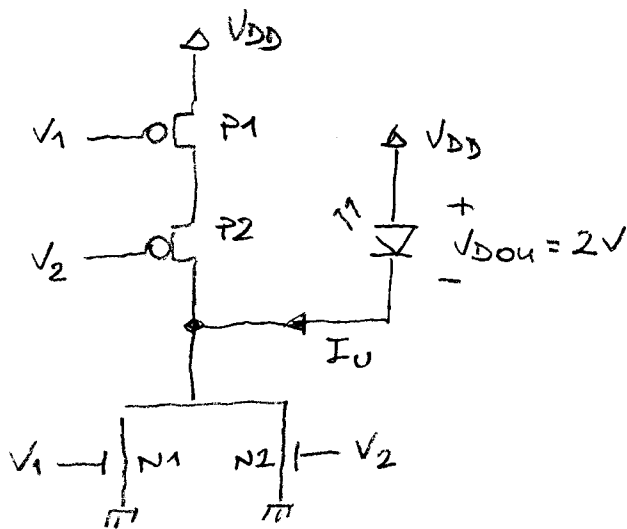
Si hanno a disposizione memorie da 128k x 3 dal costo di 3.8 € e 256k x 2 dal costo di 5.2 €. Realizzare un modulo di costo minimo da 1M x 11.

**ESERCIZIO N°5**

8 punti

Scrivere un sottoprogramma per un microcontrollore Atmel della famiglia AVR che valuta la somma con segno di  $n$  numeri interi ( $n > 0$  è contenuto in R0) da 1 byte, collocati in memoria estesa a partire dall'indirizzo presente in X. Il risultato, da rappresentare su 2 byte, va posto in memoria con il byte meno significativo nella locazione puntata da Y e quello più significativo nella locazione successiva.

1



$$V_1 = 1V$$

$$V_2 = 2V$$

Y due valori in ingresso sono abbastanza bassi ( $< V_{DD}/2$ ).  
 Se LED potrebbe essere spento ( $I_U = 0$ ) e l'uscita alta,  
 maggiore di

$$V_U = V_{DD} - V_{Dou} = 3V \quad (\text{con LED acceso})$$

Per verificare queste ipotesi, usiamo una dimostrazione  
 per assurdo.

Se  $V_U = 3V$ ,  $N_1$  limite interdizione  $I_{N1} = \phi$

$$N_2 \text{ saturato} \quad I_{N2} = \frac{k_M}{2} (V_2 - V_{Tu})^2 = 1mA$$

$P_1$  e  $P_2$  in serie conducono e la loro  
 corrente sarà sicuramente maggiore di quella  
 che si avrebbe con

$$V_1 = V_2 = 2V$$

Questa si calcola facilmente e vale, facendo  
 il MOS equivalente (TRIODO)

$$I_P = -\frac{k_P}{4} (V_{DD} - V_U) (V_2 - V_{DD} + V_2 - V_U - 2V_{TP}) =$$

$$= 2mA$$

Ma è impossibile che sia  $I_P > I_{N1} + I_{N2}$   
 perché  $I_U$  (col verso indicato nello schema)  
 DEVE essere positivo.

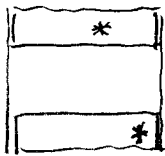
Quindi  $I_U = 0$ , indipendentemente dalla posizione di  
 $V_1$  e  $V_2$ .

2

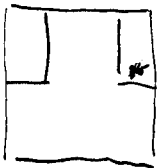
Numeri primi tra 48 e 63 (53, 59, 61)

		$x_3 x_2$			
		00	01	11	10
$x_1 x_0$	00	48 0	52 0	60 0	56 0
	01	49 0	53 1	61 1	57 0
	11	51 0	55 0	63 0	59 1
	10	50 0	54 0	62 0	58 0

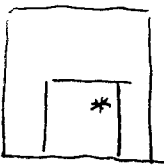
Individuo gli implicanti principali essenziali



$$x_0$$

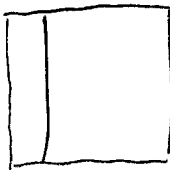


$$x_1 + x_2$$



$$\bar{x}_1 + \bar{x}_2$$

A questi posso aggiungere uno per coprire l'unica cella rimasta fuori (51)



$$x_2 + x_3$$

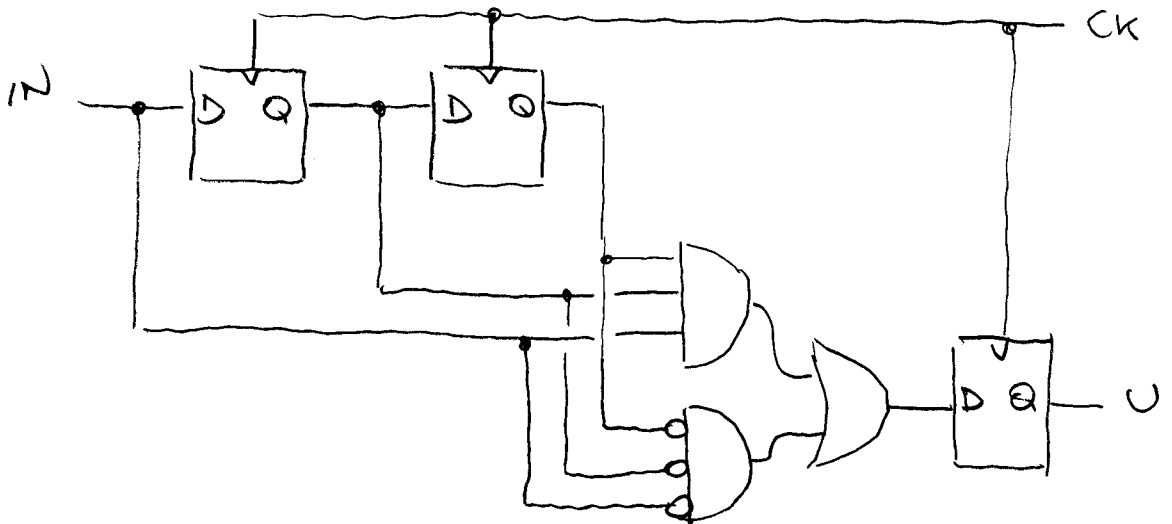
$$U = x_0 (x_1 + x_2) (\bar{x}_1 + \bar{x}_2) (x_2 + x_3)$$

3

Occorre riconoscere le situazioni senza transizioni  
per almeno 2 colpi di clock

000...  
111..

Soluzione di Mesly sincronizzata, ad hoc



④ | due chip hanno un diverso costo per bit.

$$c_3 = \frac{3,8}{128K \cdot 3}$$

$$1,27 \text{ €} / 128Kb$$

$$c_2 = \frac{5,2}{128K \cdot 4}$$

$$1,3 \text{ €} / 128Kb$$

La differenza di costo, meno del 3%, non rende sicuramente conveniente una soluzione con spreco di bit, come quella costituita usando solo chip di costo unitario minore

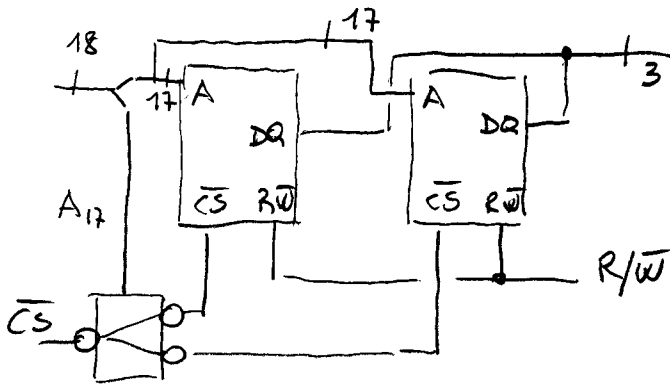
32 chip da  $128K \times 3$  con 1Mb sprecato su 12Mb

conviene quindi usare la soluzione ibrida che usa il maggior numero di chip economici

$$\left. \begin{array}{l} 24 \text{ chip da } 128K \times 3 \\ 4 \text{ chip da } 256K \times 2 \end{array} \right\} \text{ tot } 112 \text{ €}$$

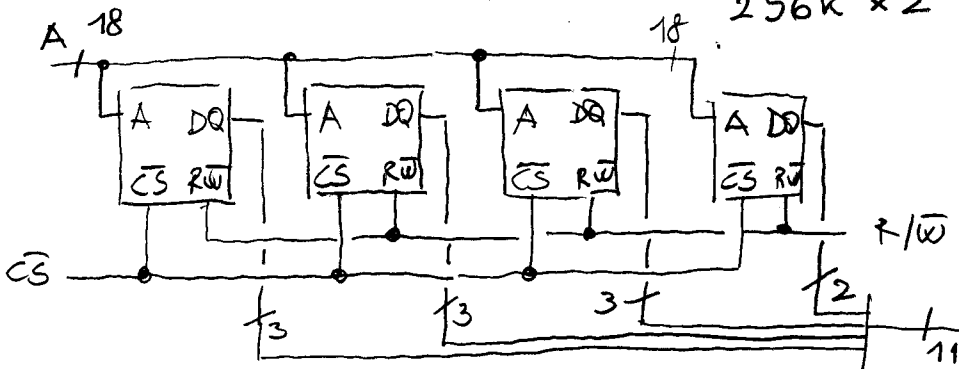
Assemblaggio possibile

$$\left. \begin{array}{l} \text{Raddoppio n° parole } 128K \times 3 \\ 128K \times 3 \end{array} \right\} 256K \times 3$$



(1)

$$\left. \begin{array}{l} \text{Estensione di parole } 256K \times 3 \\ 256K \times 3 \\ 256K \times 3 \\ 256K \times 2 \end{array} \right\} 256K \times 11$$



vedi montaggio (1)

$$\left. \begin{array}{l} \text{linee doppio raddoppio} \\ (19 \text{ e } 20 \text{ fili per A}) \end{array} \right\} \left. \begin{array}{l} 256K \times 11 \\ 256K \times 11 \end{array} \right\} \left. \begin{array}{l} 512K \times 11 \\ 512K \times 11 \end{array} \right\} 1M \times 11$$

5

```
sum:  PUSH R0
      PUSH R16
      PUSH R17
      PUSH R18
      PUSH R19
      PUSH XL
      PUSH XH
```

```
      CLR R18
      CLR R19
```

// registri per il risultato

```
loop: LD R16, X+
      CLR R17
      TST R16
      BRPL poi
      SER R17
      ADD R18, R16
      ADC R19, R17
      DEC R20
      BRNE loop
```

// carica il dato da sommare  
// controlla il segno  
// positivo

// estendi con segno, se negativo

// somma con segno

```
      ST Y+, R18
      ST Y, R19
```

// memorizza il risultato

```
      SBIW YL, 1
```

// ripristina Y

```
      POP XL
      POP XL
      POP R19
      POP R18
      POP R17
      POP R16
      POP R0
      RET
```

// ripristina registri