

Cognome

Nome

Matricola

ESERCIZIO N°1

6/4 punti

Determinare la tensione di uscita V_U di un invertitore CMOS a vuoto, alimentato con $V_{DD} = 10$ V, con tensione di ingresso $V_{IN} = 4$ V e $V_{IN} = 6$ V ($V_{Tn} = -V_{Tp} = 1$ V; $K_n = -K_p = 2$ mA/V²).

ESERCIZIO N°2

6/4 punti

Realizzare in forma SP una rete combinatoria a 5 ingressi x_4, x_3, x_2, x_1 e x_0 in grado di evidenziare, ponendo a 1 l'uscita, tutti i valori di X (numero binario senza segno i cui bit corrispondono agli ingressi) tali che soddisfano almeno una delle seguenti relazioni:

$X = 20$; $X = 26$; $X > 29$; $X^3 = 64$; $13 \leq X \leq 17$; $40 \leq X^2 \leq 70$; $X = X^2$; $X = 3k$ con $k \in \mathbb{N}$.

ESERCIZIO N°3

6/4 punti

Progettare un riconoscitore di sequenze interallacciate facendo uso di uno shift register, in grado di riconoscere tutte le sequenze di 6 bit in cui i primi 3 bit sono uguali ai secondi 3. Quante sono le sequenze riconosciute?

ESERCIZIO N°4

7/4 punti

Disegnare lo schema logico di un sequenziatore con contatore sincrono (dotato della possibilità di caricamento parallelo) che implementi microcodice specificato nel seguito. Si usi una ROM delle minime dimensioni possibili.

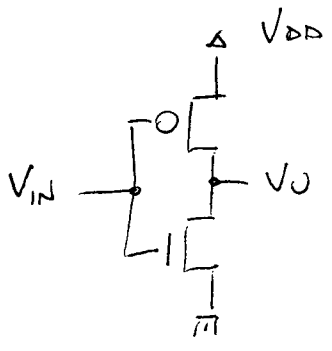
```
A: IF L THEN B ELSE C; OP = 11
B: IF K THEN G ELSE A; OP = 10
C: IF M THEN A ELSE H; OP = 00
D: IF L THEN E ELSE F; OP = 11
E: IF J THEN H ELSE D; OP = 01
F: IF K THEN D ELSE G; OP = 10
G: IF J THEN F ELSE E; OP = 01
H: IF M THEN C ELSE B; OP = 00
```

ESERCIZIO N°5

8/4 punti

Realizzare una subroutine per un microcontrollore della famiglia AVR che esegue l'operazione di modulo 7 sul valore binario rappresentato su 2 byte, contenuto in memoria a partire dall'indirizzo in X , lasciando il risultato in $R0$. Stimare il tempo massimo di esecuzione della subroutine.

1



$V_{DD} = 10V$
 Parametri "simmetrici"

le due situazioni proposte sono complementari

$$V_{IN1} = 4V ; \quad V_{IN2} = 6V = V_{DD} - V_{IN1}$$

è sufficiente studiarne una; per la seconda si avrà

$$V_{O2} = V_{DD} - V_{O1} \quad *$$

le zone dei MOS per il caso 1 è N saturato e P triodo in quanto $V_{IN1} < V_{DD}/2$.

Principio di Kirchhoff per le correnti (e vuoto)

$$\frac{K_M}{2} (V_{IN} - V_{TM})^2 = -\frac{K_P}{2} (V_O - V_{DD}) (V_{IN} - V_{DD} + V_{IN} - V_O - 2V_{TP})$$

pongo $V_O = x$

$$-9 = (x - 10)x$$

$$x^2 - 10x + 9 = 0$$

$$\left\{ \begin{array}{l} x = 9 \\ x = 1 \text{ non accettabile} \end{array} \right.$$

quindi

$$V_{O1} = 9V$$

$$* V_{O2} = 1V$$

2

I valori da identificare sono

{20} {26} {30,31} {4}

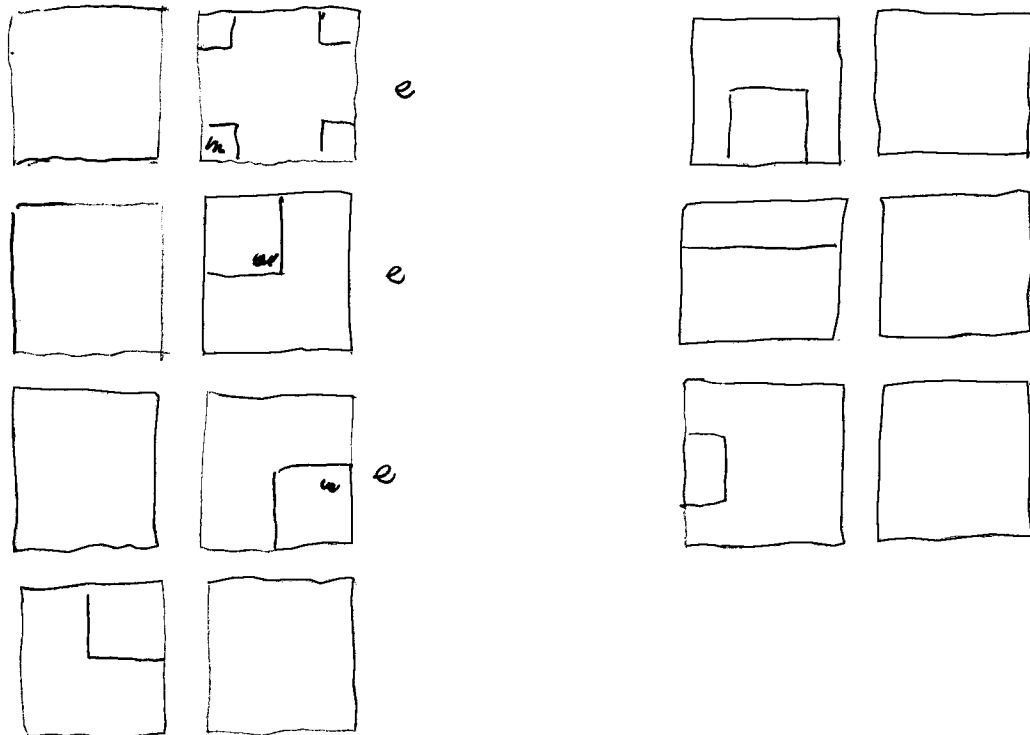
{13,14,15,16,17} {7,8}

{0,1}

{0,3,6,9,12,15,18,21,24,27,30}

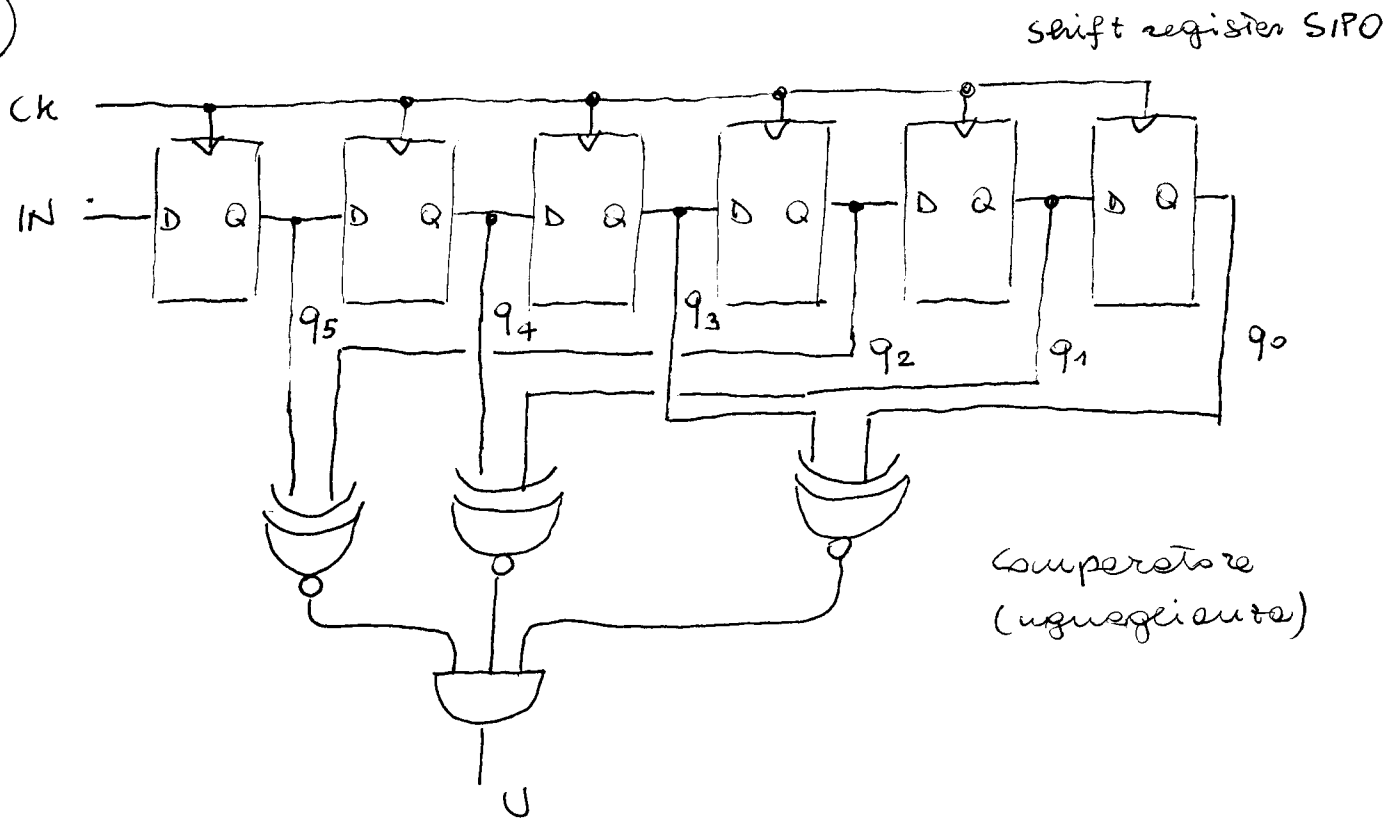
Riparto nella mappa

$x_3 x_2$		$x_4 = 0$				$x_4 = 1$			
		00	01	11	10	00	01	11	10
x_1, x_0	00	1 ⁰	1 ⁴	1 ¹²	1 ⁸	1 ¹⁶	1 ²⁰	0 ²⁸	1 ²⁴
	01	1 ¹	0 ⁵	1 ¹³	1 ⁹	1 ¹⁷	1 ²¹	0 ²⁹	0 ²⁵
	11	1 ³	1 ⁷	1 ¹⁵	0 ¹¹	0 ¹⁹	0 ²³	1 ³¹	1 ²⁷
	10	0 ²	1 ⁶	1 ¹⁴	0 ¹⁰	1 ¹⁸	0 ²²	1 ³⁰	1 ²⁶



$$U = \bar{x}_0 \bar{x}_2 x_4 + \bar{x}_1 \bar{x}_3 x_4 + x_1 x_3 x_4 + \bar{x}_1 x_3 \bar{x}_4 + x_1 x_2 \bar{x}_4 + \bar{x}_0 \bar{x}_1 \bar{x}_4 + x_0 \bar{x}_2 \bar{x}_3 \bar{x}_4$$

3



Le sequenze riconosciute sono 8
(tutte le combinazioni di 3 bit)

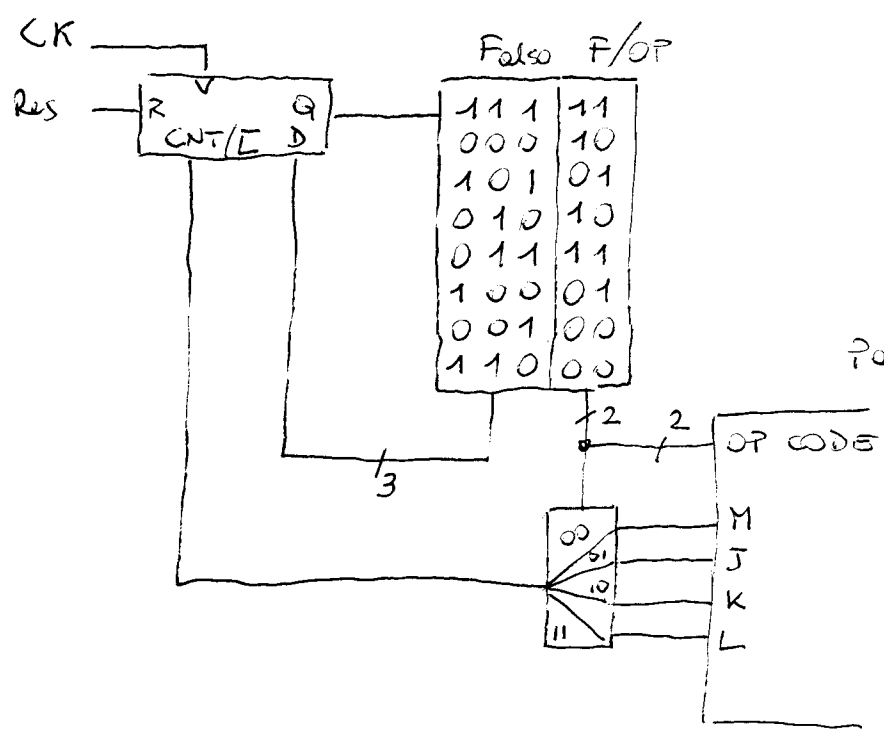
④

Riordino la tabella in modo da evidenziare la seguente ciclica completa

	Stato	Flag	SF se Vero	se Falso	OP
0	A	L	B	C	11
1	B	K	G	A	10
2	G	J	F	E	01
3	F	K	D	G	10
4	D	J	E	F	11
5	E	L	H	D	01
6	H	M	C	B	00
7	C	M	A	H	00

Codifico gli stati in ordine con i valori 0..7 e per il flag uso come codice il valore OP (che casualmente coincide...)

flag	
00	M
01	J
10	K
11	L



5

/* Un modo semplice di fare il modulo 7 di una parola di un byte è sottrarre 7 sino ad avere un numero minore di 7. Per parole su più byte, basta osservare che 256 modulo 7 fa 4.

```
*/
mod7:
    push R16
    push R17
    ld R16,X+           //carica il primo byte
    rcall mod7byte     //esegue modulo 7
    mov R17,R16        //salva in R17 il byte più significativo
    ld R16,X
    sbiw XH:XL,1       //ripristina X
    rcall mod7byte     //esegue modulo 7
    lsl R16
    lsl R16             //moltiplica per 256 mod 7 (4)
    add R16,R17         //somma
    rcall mod7byte     //si riporta ancora tra 0 e 6, trovando il risultato
    mov R0,R16         //salva il risultato
    pop R17
    pop R16
ret

mod7byte:              //esegue r16 mod 7 di un byte
    subi r16,7
    brcc mod7byte
    subi r16,-7

/* La stima delle iterazioni e quindi del numero di cicli richiesti nel caso peggiore può
essere fatta immaginando che entrambi i byte contengano 255.
Per eseguire il modulo 7 di ciascuno occorrono 37 sottrazioni per arrivare al risultato di 3.
La somma dà 15 che richiede altre 3 sottrazioni, tutte svolte nella routine mod7byte. Ogni
iterazione richiede 3 cicli. A questi vanno sommati i cicli richiesti per l'esecuzione delle
altre istruzioni del programma.
*/
```