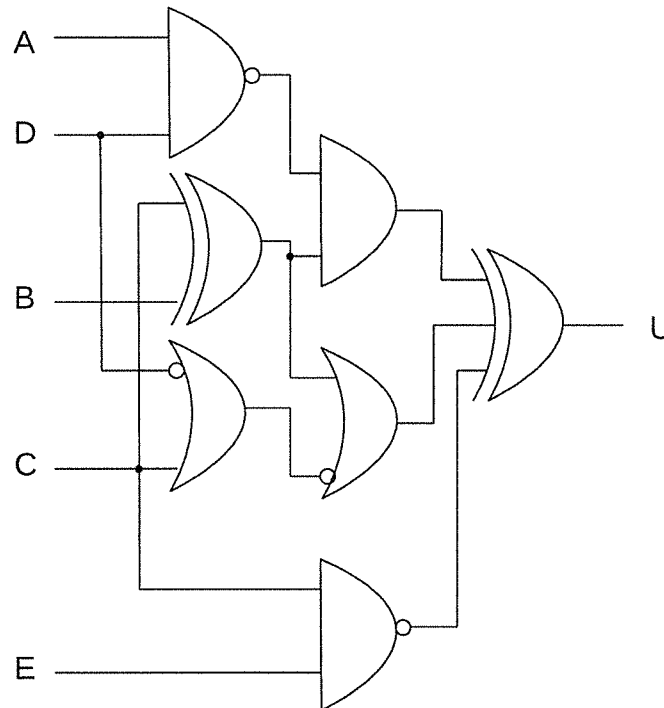


### ESERCIZIO N°1

6 punti

a) Determinare la mappa di Karnaugh della funzione logica  $Y$  realizzata dalla seguente rete.



b) Realizzare con porte logiche elementari in forma PS a minimo numero di letterali la funzione del punto a).

c) Se porte logiche elementari (AND, OR, NOT) a  $K$  ingressi hanno  $T_{pd} = 0,15 \text{ ns} + 0,5 K \text{ ns}$  quale è il  $T_{pd}$  massimo del circuito di cui al punto b)? Data la funzione di ritardo specificata, si può affermare che il tempo di propagazione di una rete aumenta con i livelli di logica usati?

### ESERCIZIO N°2

4 punti

a) Realizzare un multiplexer 8:1, con 3 linee di controllo, avendo a disposizione multiplexer 2:1.

b) Determinare il numero di multiplexer 4:1 (e la quantità di linee di controllo) necessari per realizzare un multiplexer  $N:1$  con  $N$  generico.

### ESERCIZIO N°3

6 punti

Dati i numeri reali  $A = 3\pi$ ,  $B = -e^{1,4}$  e  $C = \sqrt[3]{3}$

a) Determinare quale rappresentazione in virgola fissa in complemento a 2 è in grado di rappresentarli tutti con un errore relativo inferiore all'1%.

b) Determinare la rappresentazione di  $A$ ,  $B$  e  $C$  in virgola mobile in formato standard IEEE 754 singola precisione e valutare l'errore di rappresentazione relativo.

#### **ESERCIZIO N°4**

4 punti

Progettare un contatore sincrono (up/down con abilitazione) modulo 11 facendo uso di T-FF..

#### **ESERCIZIO N°5**

5 punti

Progettare una macchina sequenziale sincrona secondo il modello di Moore con un ingresso e una uscita in grado di generare una forma d'onda di periodo  $8T_{clk}$ , il cui ciclo di lavoro valga  $1/4$  se l'ingresso letto in corrispondenza della fine di ciascun ciclo vale 0 e  $1/2$  altrimenti.

#### **ESERCIZIO N°6**

8 punti

Realizzare una subroutine per un microcontrollore della famiglia AVR che converte in binario le due cifre BCD contenute nel registro R16, lasciando il risultato nello stesso registro R16. Nel caso in cui il dato di partenza non rappresenti un valore BCD valido, il risultato deve essere posto a 0. La subroutine deve, come al solito, lasciare inalterati tutti gli altri registri.

①

Espressione booleana (espansione di Shannon)

$$\begin{aligned}
 Y &= (\overline{AD} \cdot (B \oplus C)) \oplus ((B \oplus C) + (\overline{C + D})) \oplus \overline{CE} = \\
 &= \overline{BC} \cdot \overline{D} + \overline{BC} (\overline{AD} \oplus E) + B\overline{C} \overline{AD} + B\overline{C}E = \\
 &= \overline{BC} \overline{D} + \overline{BC} (AD \cdot E + \overline{E} \overline{A} + \overline{E} \overline{D}) + B\overline{C} \overline{A} + B\overline{C} \overline{D} + B\overline{C}E = \\
 &= \overline{BC} \overline{D} + A\overline{B}CDE + \overline{A}\overline{B}C\overline{E} + \overline{B}C\overline{E} \overline{D} + \overline{A}B\overline{C} + B\overline{C} \overline{D} + B\overline{C}E
 \end{aligned}$$

Mappe

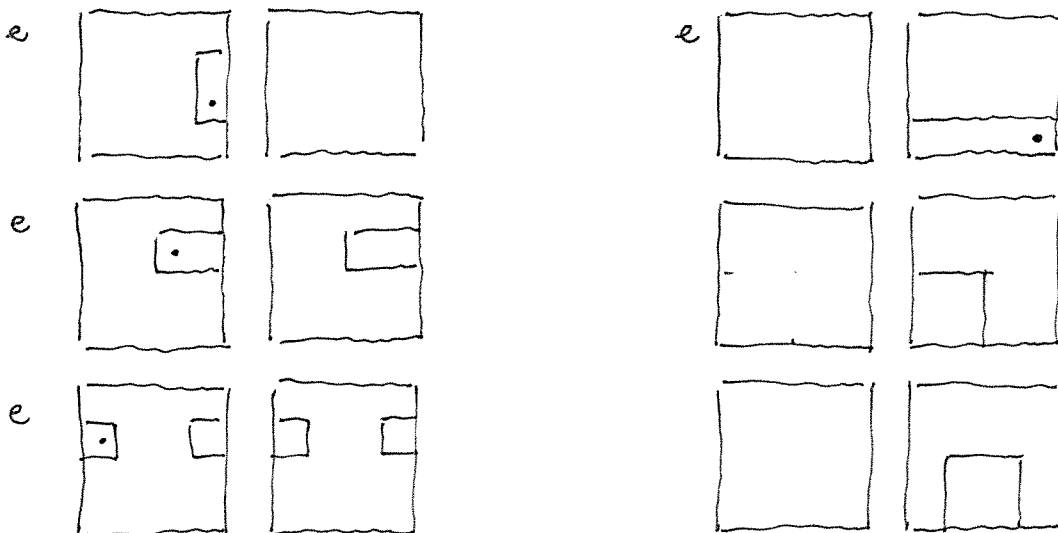
	AB			
CD	00	01	11	10
00	1	1	1	1
01	.	1	.	.
11	1	1	1	.
10	1	1	1	1

E=0

	AB			
	00	01	11	10
00	1	1	1	1
01	.	1	.	.
11				1
10	.	.	.	.

E=1

Sintesi ottime PS (con e sono indicati gli "essenziali" rispetto al MAXTERMINE col pellino)



Gli ultimi due non sono implicati essenziali ma sono i due implicati più grandi a coprire i restanti 3 zeri

Quindi

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

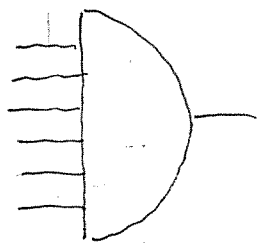
Scheme logico

vedi poi

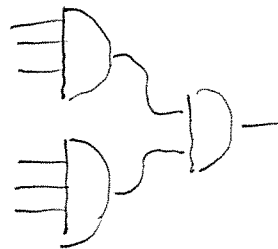
Ritardo (in nanosecondi)

$$t_{pd} = \underset{\text{NOT}}{0,15} + \underset{\text{OR}}{0,5} + \underset{\text{AND}}{0,15} + 4 \cdot 0,5 + 0,15 + 6 \cdot 0,5 = 5,95 \text{ ns}$$

Il ritardo aumenta sicuramente coi livelli di logica?  
NO, perché la quota dipendente dal numero  $k$  degli ingressi è elevata. Per esempio:

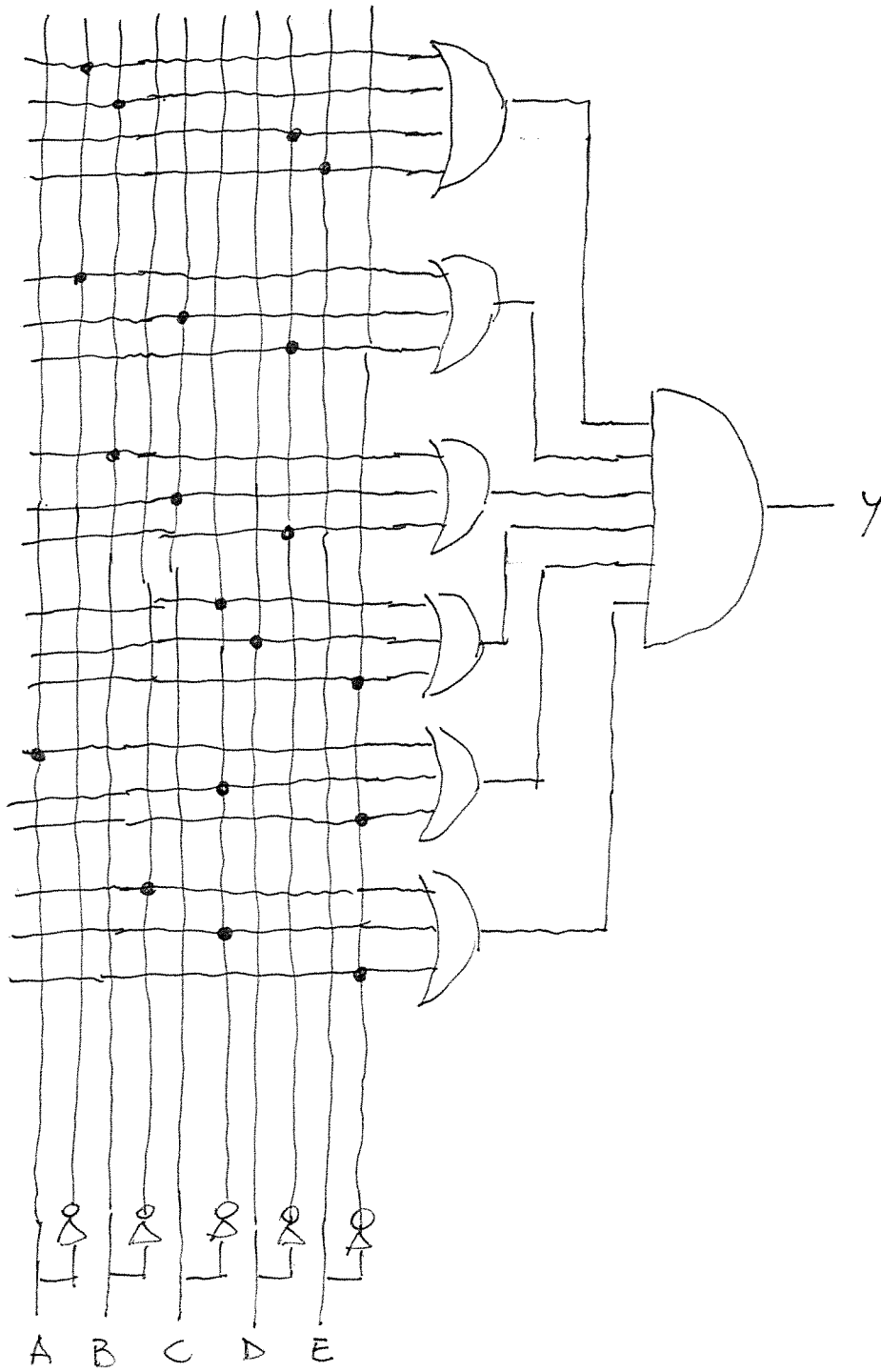


$$t_{pd} = 0,15 + 6 \cdot 0,5 = 3,5 \text{ ns}$$

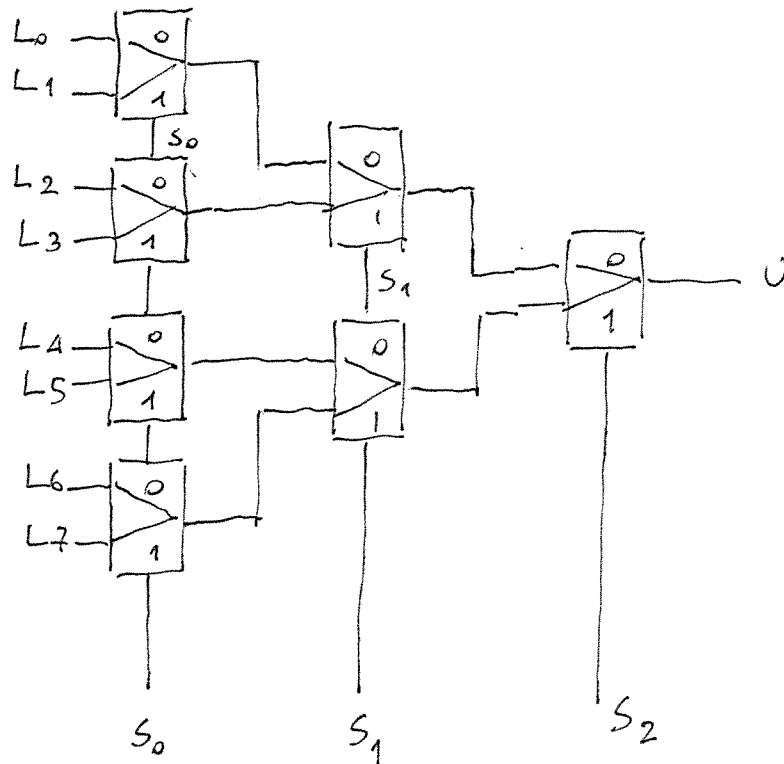


$$t_{pd} = 0,15 + 3 \cdot 0,5 + 0,15 + 2 \cdot 0,5 = 2,8 \text{ ns}$$

Scheme Logics



② MUX 8:1 con 3 linee di selezione ( $S_2 S_1 S_0$ )



Per realizzare un mux  $N:1$  occorrono comunque

$$n = \lceil \log_2 N \rceil \text{ linee di selezione}$$

Con mux base 4:1 (2 linee di selezione) si realizzano multiplexer "completi" da  $4^k$  ingressi (4, 16, 64 ...)

Per un mux  $N:1$  con  $N = 4^k$  occorrono  $m$  mux 4:1 con

$$m = 1 + 4 + 16 \dots + 4^{k-1} = \frac{4^k - 1}{3}$$

Per valori di  $N$  intermedi compresi tra due potenze di 4 si possono risparmiare i mux connessi agli ingressi non usati.

3

Per decidere sulle cifre da usare, occorre esprimere l'errore relativo in funzione delle cifre  $f$  usate per la parte frazionabile

$$\epsilon_r = \frac{\epsilon_A}{|x|} = \left| \frac{2^f x - \text{round}(2^f x)}{2^f x} \right|$$

Valutiamo  $f_A$ :

$$A = 9,4247\dots$$

$$f=5 \quad \epsilon_r = 0,13\% \quad \text{no}$$

$$f=6 \quad \epsilon_r = 0,31\% \quad \text{ok}$$

Valutiamo  $f_B$ : (partiamo da  $f=6$ )

$$B = -4,0552\dots$$

$$f=6 \quad \epsilon_r = 7,30\% \quad \text{no}$$

$$f=7 \quad \epsilon_r = 0,51\% \quad \text{ok}$$

Valutiamo  $f_C$ : (partiamo da  $f=7$ )

$$C = 1,2009\dots$$

$$f=7 \quad \epsilon_r = 1,82\% \quad \text{no}$$

$$f=8 \quad \epsilon_r = 1,43\% \quad \text{no}$$

$$f=9 \quad \epsilon_r = 0,19\% \quad \text{ok}$$

Occorrono quindi 5b per la parte intera con il segno e 9b per la parte frazionabile. In tutto 14b

Per rappresentare i numeri in IEEE 754 (binary 32) ricordiamo la legge di rappresentazione

$$x = (-1)^s \cdot 2^{e-127} \left\{ 1 + \sum_{i=1}^{23} b_{23-i} 2^{-i} \right\}$$

$$A \approx 9,4247... = (-1)^0 \cdot 2^3 \cdot 3\pi/8$$

$$s=0 \quad e=130 \quad B=1493988$$

$$[0 \mid 10000010 \mid 0010110110010111110010]$$

$$E_r = 2,53 \times 10^{-9}$$

$$B \approx -4,0552... = (-1)^1 \cdot 2^2 \cdot e^{1,4}/4$$

$$s=1 \quad e=129 \quad B=115763$$

$$[1 \mid 10000001 \mid 00000011100010000110011]$$

$$E_r = 3,28 \times 10^{-8}$$

$$C \approx 1,2009... = (-1)^0 \cdot 2^0 \cdot \sqrt[6]{3}$$

$$s=0 \quad e=127 \quad B=1685581$$

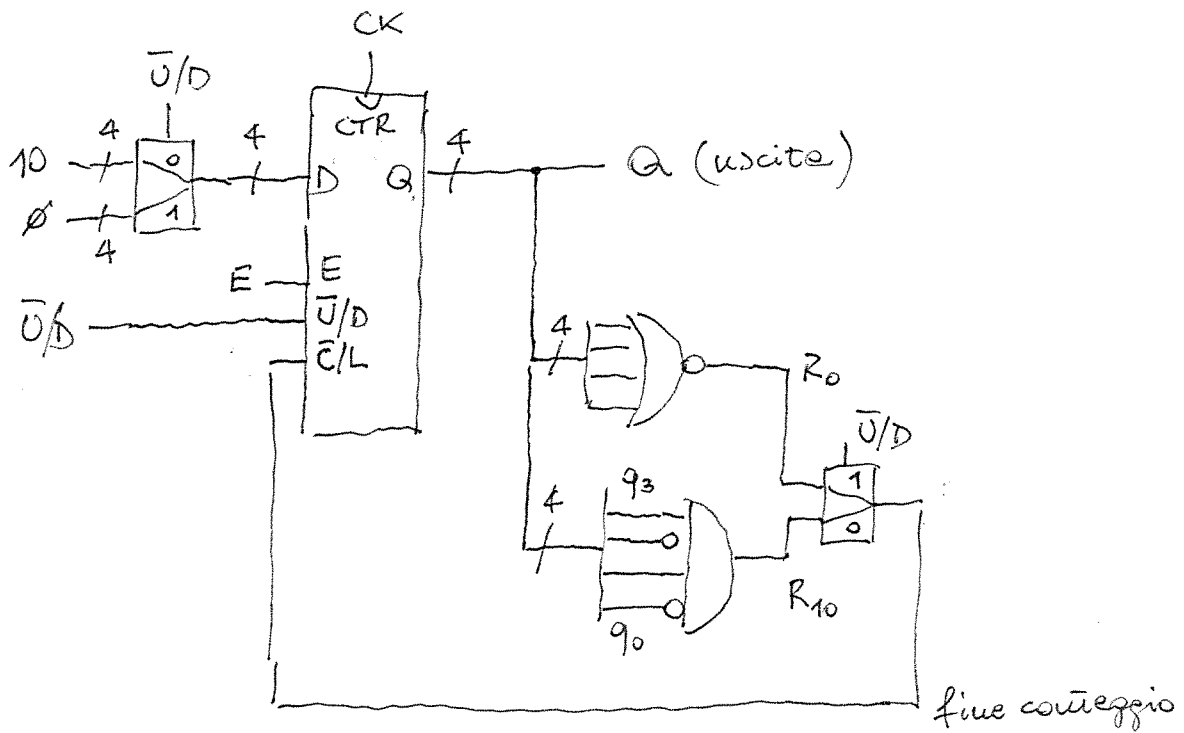
$$[0 \mid 1111111 \mid 0011001101110000100110]$$

$$E_r = 3,47 \times 10^{-8}$$

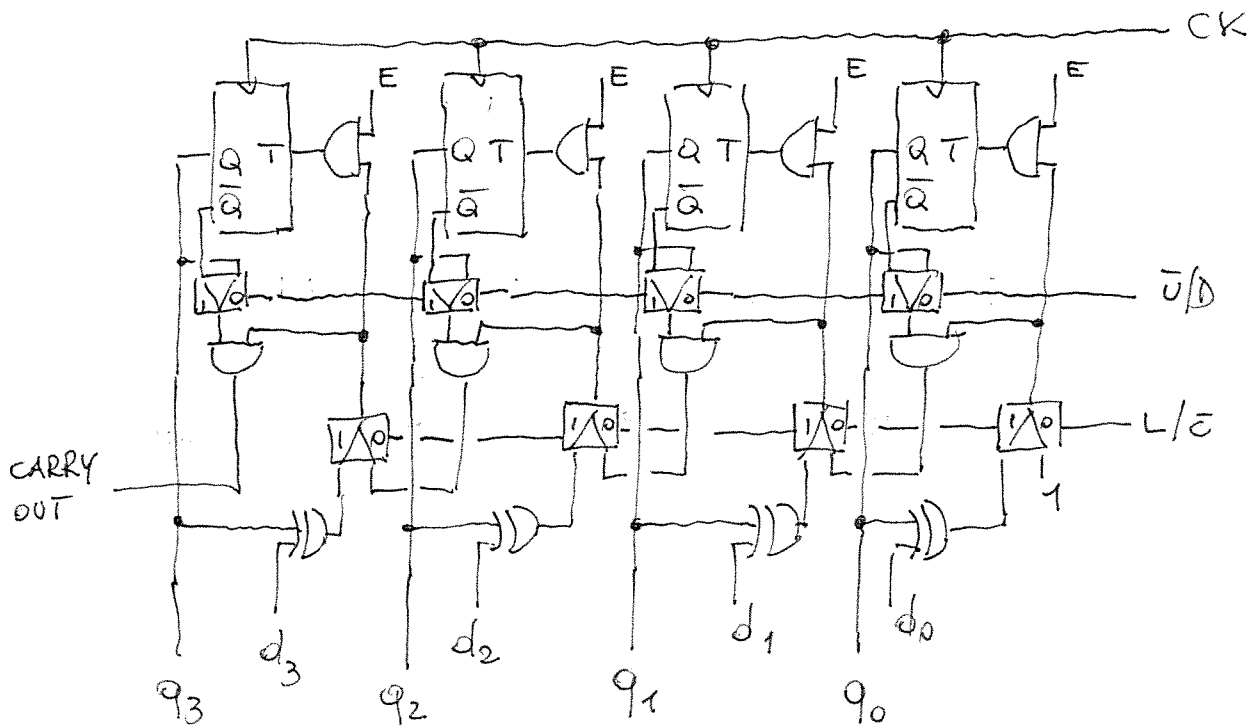


④

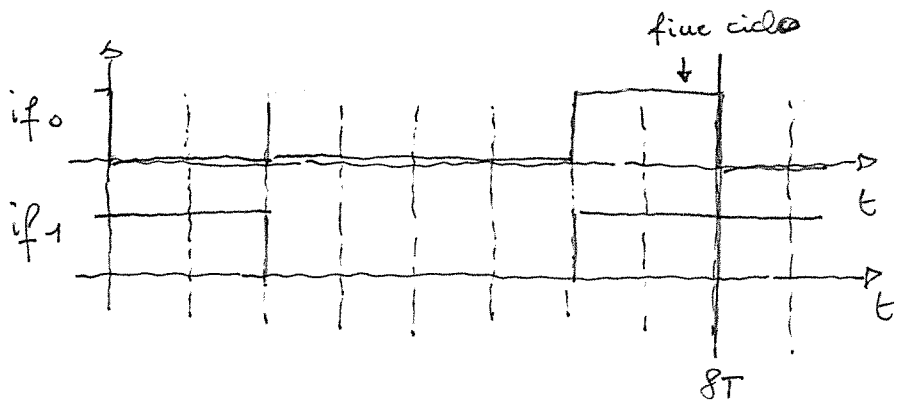
Schema generale, basato su contatore modulo 16  $\overline{UP}/DOWN$  con abilitazione e caricamento parallelo



Il contatore può essere realizzato con T-FF nel modo seguente



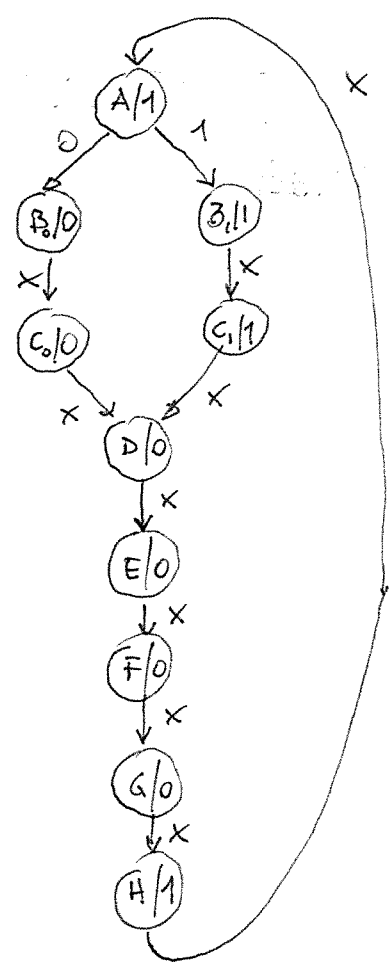
5) Forme d'onda da generare



$S = 1/4$

$S = 1/2$

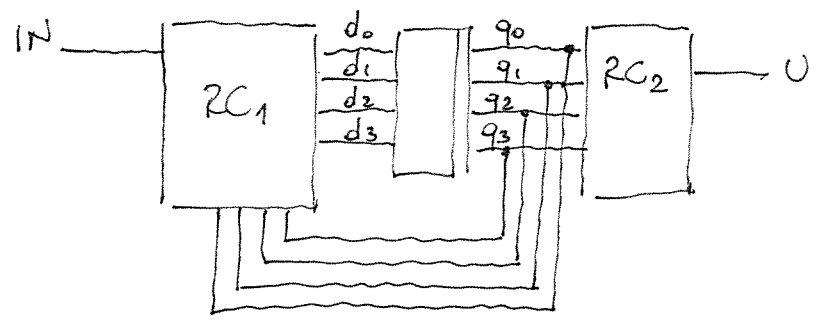
In questo modo il grafo è



Vista la natura ciclica del grafo, codifico gli stati in modo che (parte) dello stato futuro si valuta bile da un contatore

	$q_3$	$q_2$	$q_1$	$q_0$
A	0	1	1	1
$B_0$	0	0	0	0
$C_0$	0	0	0	1
D	0	0	1	0
E	0	0	1	1
F	0	1	0	0
G	0	1	0	1
H	0	1	1	0
$B_1$	1	0	0	0
$C_1$	1	0	0	1

Architettura



Sequenze di conteggio

Sintesi delle rete per l'uscita

	q <sub>3</sub> q <sub>2</sub>			
q <sub>1</sub> q <sub>0</sub> \	00	01	11	10
00	0	0	-	1
01	0	0	-	1
11	0	1	-	-
10	0	1	-	-

$$U = q_3 + q_1 q_2$$

Sintesi per la variabile di stato d<sub>3</sub>

	q <sub>3</sub> q <sub>2</sub>			
q <sub>1</sub> q <sub>0</sub> \	00	01	11	10
00	0	0	-	1
01	0	0	-	0
11	0	0	-	-
10	0	0	-	-

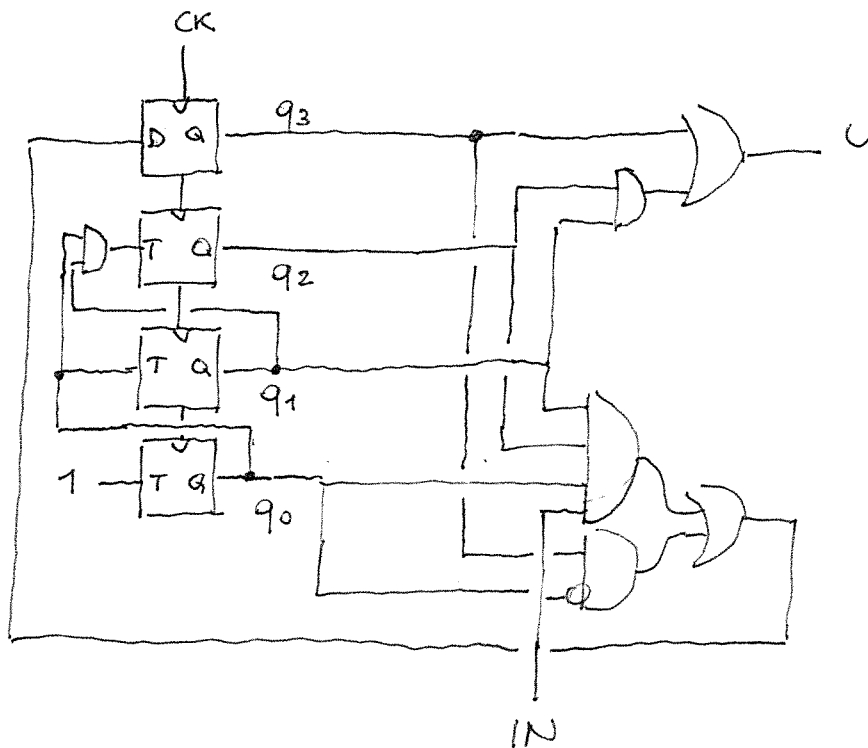
	q <sub>3</sub> q <sub>2</sub>			
q <sub>1</sub> q <sub>0</sub> \	00	01	11	10
00	0	0	-	1
01	0	0	-	0
11	0	1	-	-
10	0	0	-	-

$$d_3 = q_3 \bar{q}_0 + 1N q_2 q_1 q_0$$

1N=0

1N=1

In definitiva:



# 6

```
bcd2bin:
    PUSH R0          //salva i registri usati
    PUSH R1          //salva i registri usati
    PUSH R17
    PUSH R18
    LDI R18,10
    MOV R17,R16      //copia il dato
    ANDI R16,0x0F    //in R16 ci sono le unità u
    CP R16,R18
    BRGE nv          //dato non valido
    ANDI R17,0xF0    //in R17 decine d
    SWAP R17
    CP R17,R18
    BRGE nv          //dato non valido
    MUL R17,R18      //10d
    ADD R16,R0       //10d+u
restore:
    POP R18
    POP R17
    POP R1
    POP R0           ;ripristina i registri salvati
    RET
nv:
    CLR R16
    RJMP restore
```