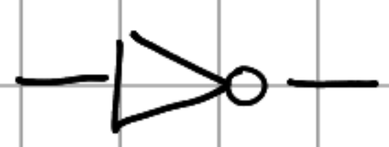
Lez 06

LATCH ≠ FLIP-FLOP → registrano ai cambiamenti di stato
↑
GATED LATCH

- LIMITE PORTE REALI.

- Tempi di risposta → quello fondamentale.

- FAN IN / FAN OUT
(datasheet) ↳ Immagina NOT

 ⇝ va a pilotare qualcos'altro

Che cosa può venda a pilotare? ↵

↳ C'è un limite a quante porte li si possono collegare (così che vengano ben controllate)

- Tempi di propagazione del segnale

Circuito ⟺ formula booleana ⟺ tabella verità

↳ C'è anche un DIAGRAMMA TEMPORALE

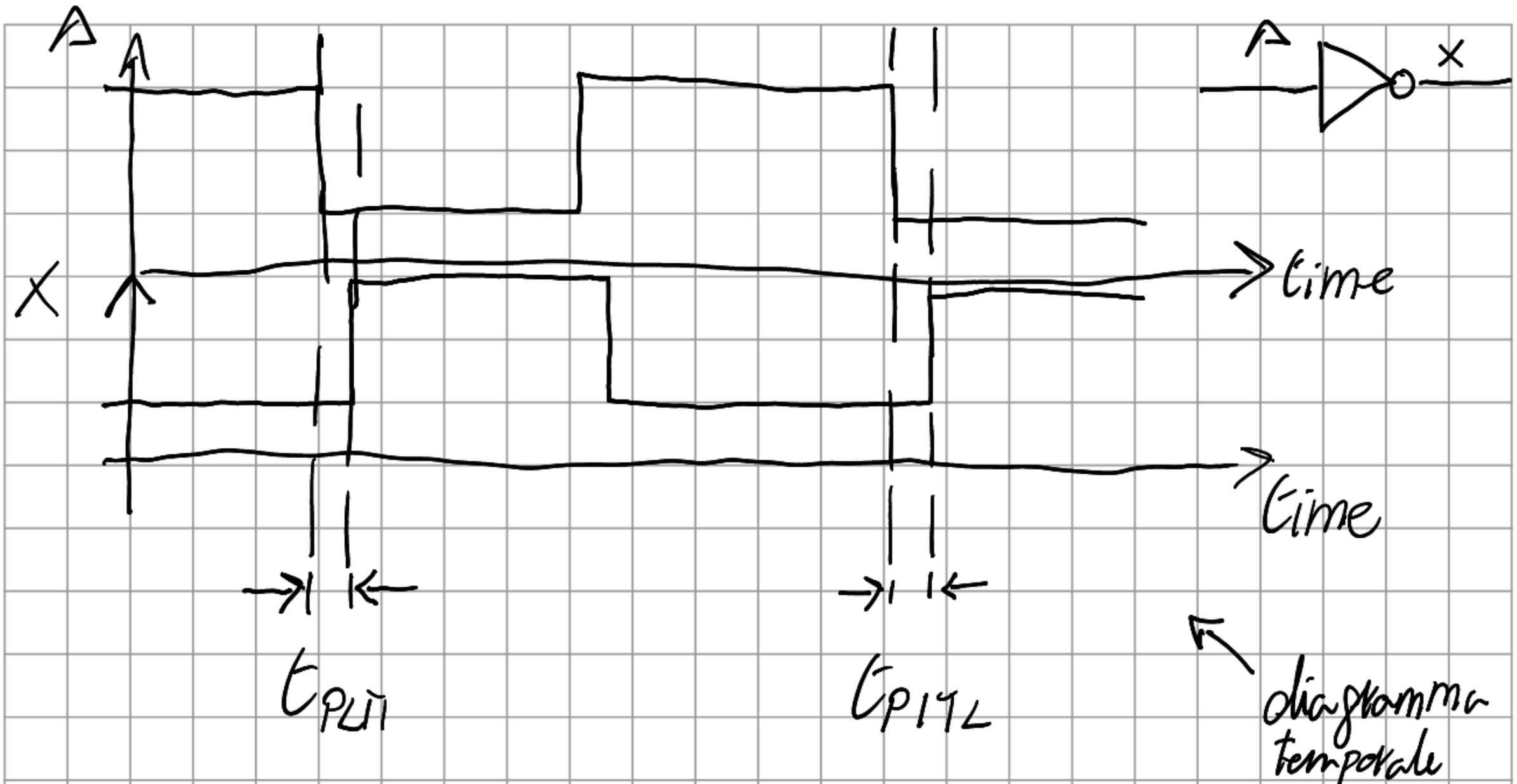
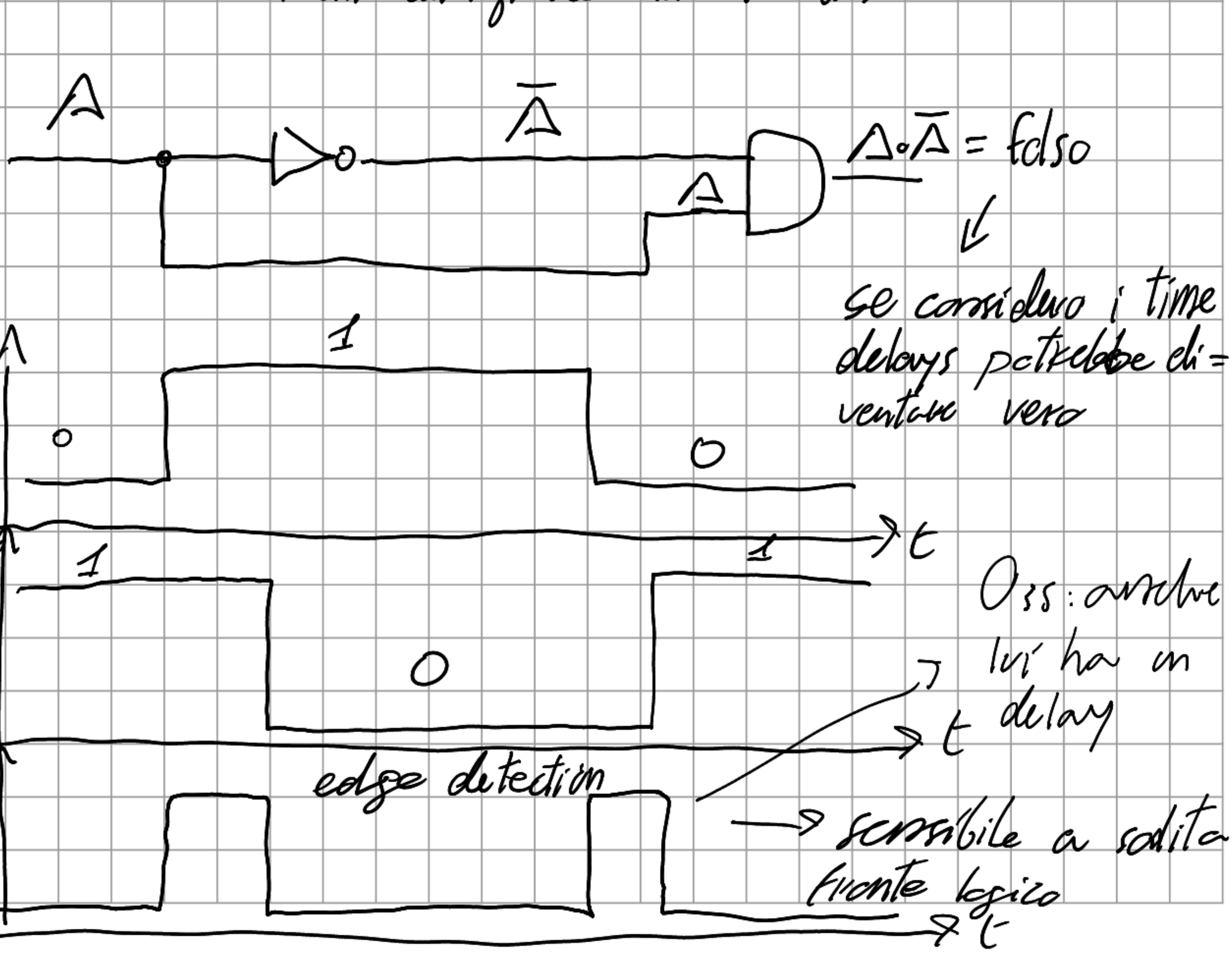


diagramma temporale

↳ porta NON reagisce IMMEDIATAMENTE ai cambiamenti dell'ingresso (nanosecondi)



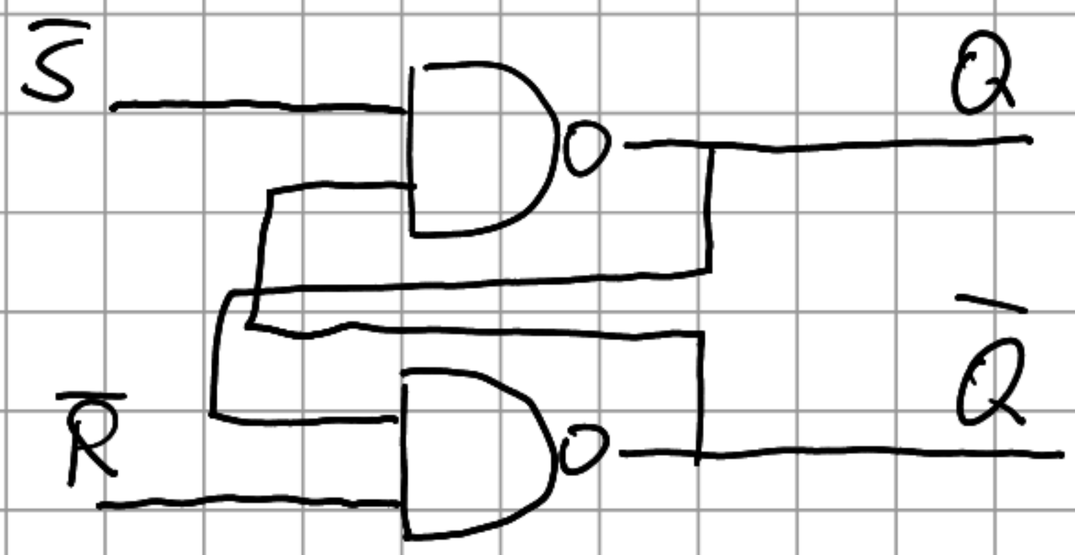
se considero i time delays potrebbe diventare vero

Oss: anche lui ha un delay

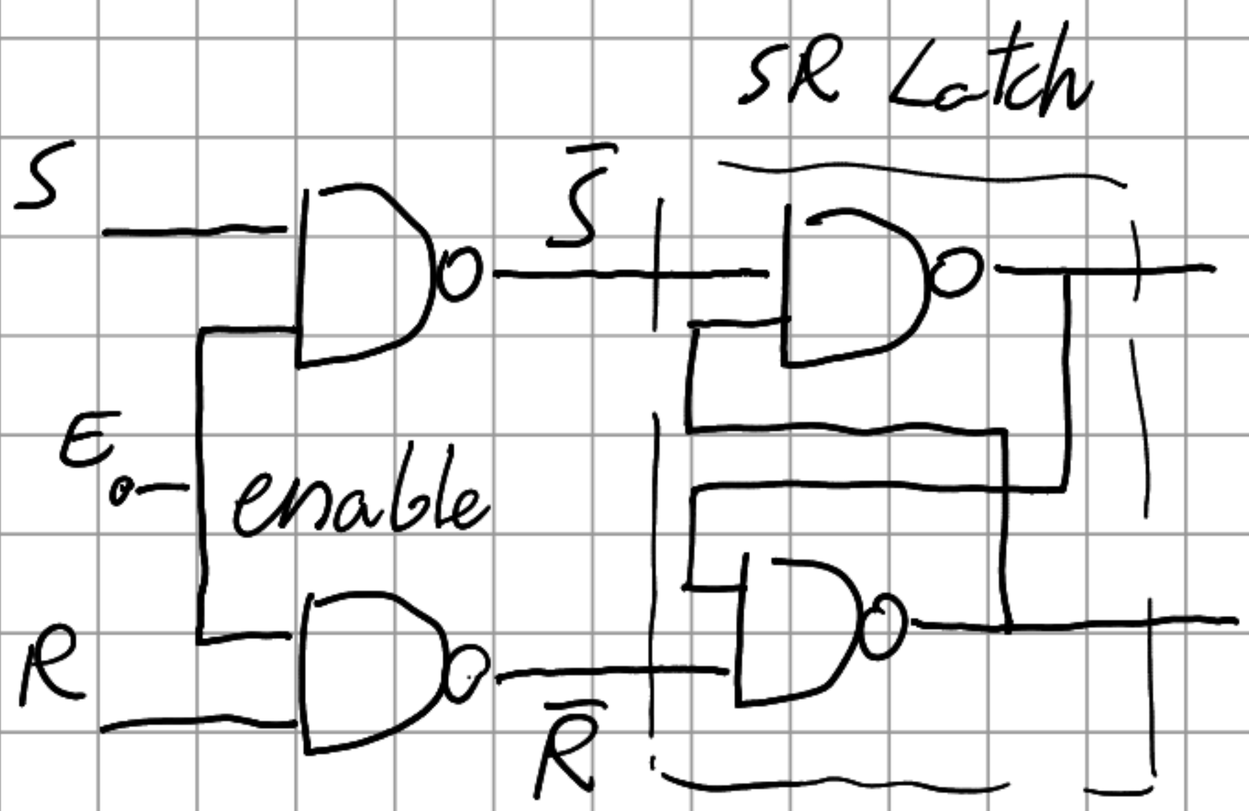
→ sensibile a salita fronte logico

- Si può sfruttare in contesti ben delimitè.
  ↳ Per circuiti complicati diventa difficile quantificarlo.
- Anche tmp. prop. sgn rilevante (☞ 11:40)

- ELEMENTI DI MEMORIA.



Latch



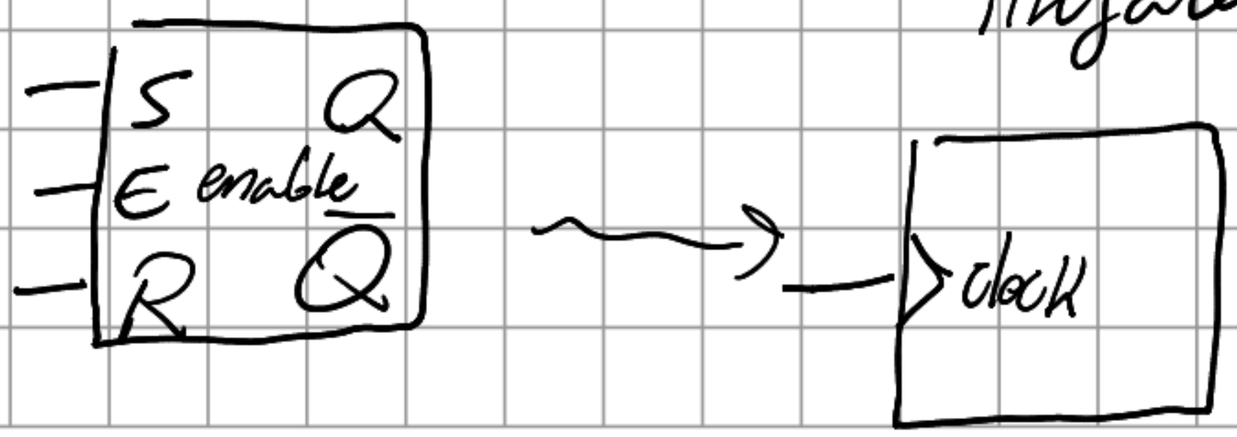
Gated-Latch

E	S	R	Q	Q̄
0	x	x	HOLD	
1	Q	0	HOLD	
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

reset

set

illegale

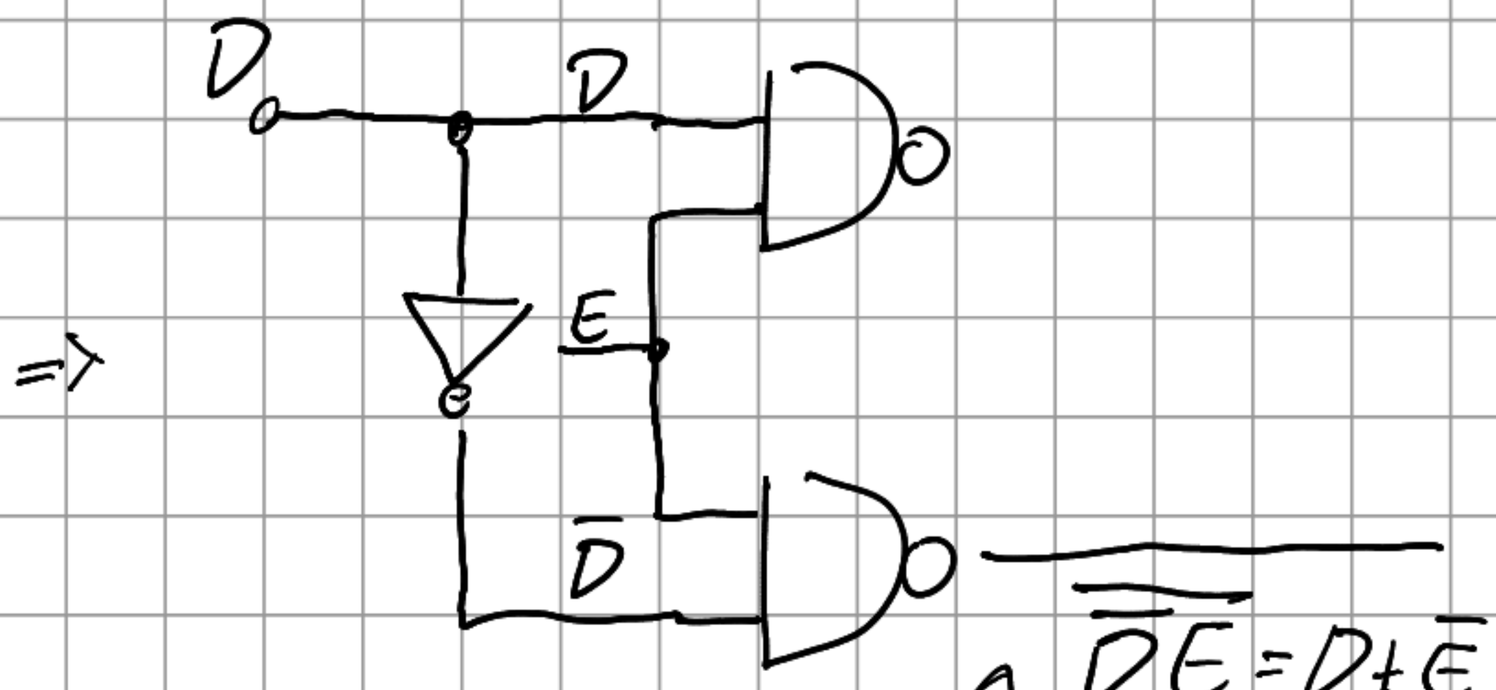


- COME EVITARE LA CONFIG. ILLEGALE?

1. Il "DATA" o "D-LATCH" → Correzione all'ingresso (non chiedo cose che non hanno senso)

↓
Gated d-latch

Non posso chiedere Set e Reset assieme

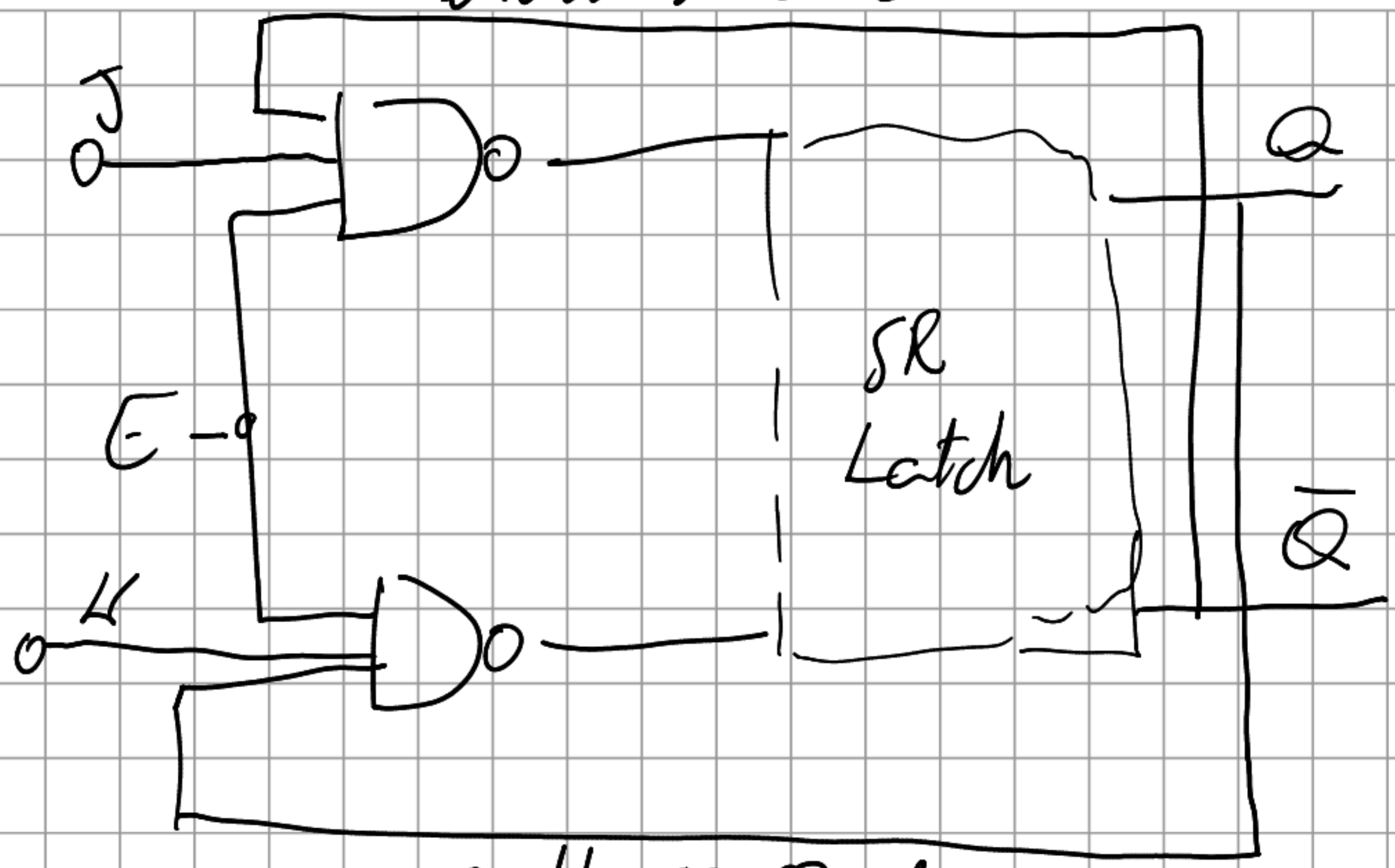


E	D	Q	Q̄
0	x	HOLD	
1	0	0	1
1	1	1	0

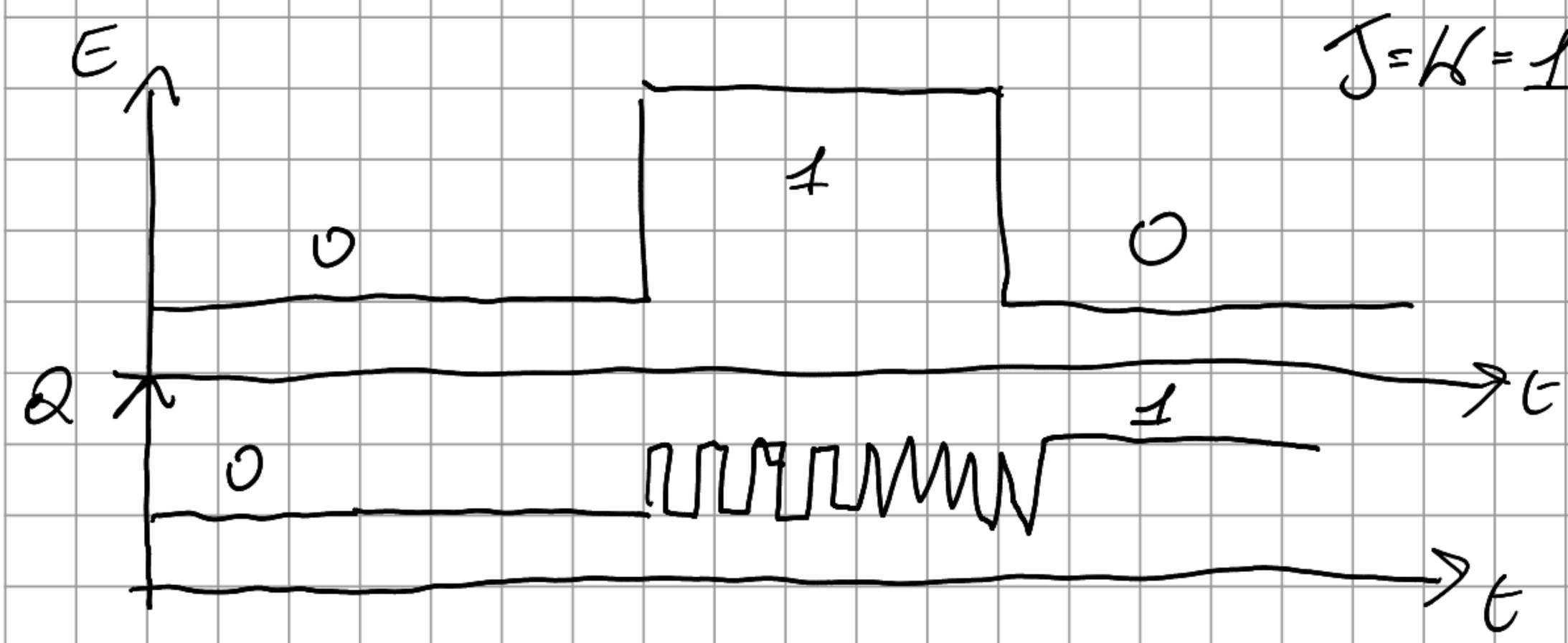
2. Configurazione JK → Disabilito la richiesta insensata

Il mio circ. ha già stato => ignoro richieste inutili

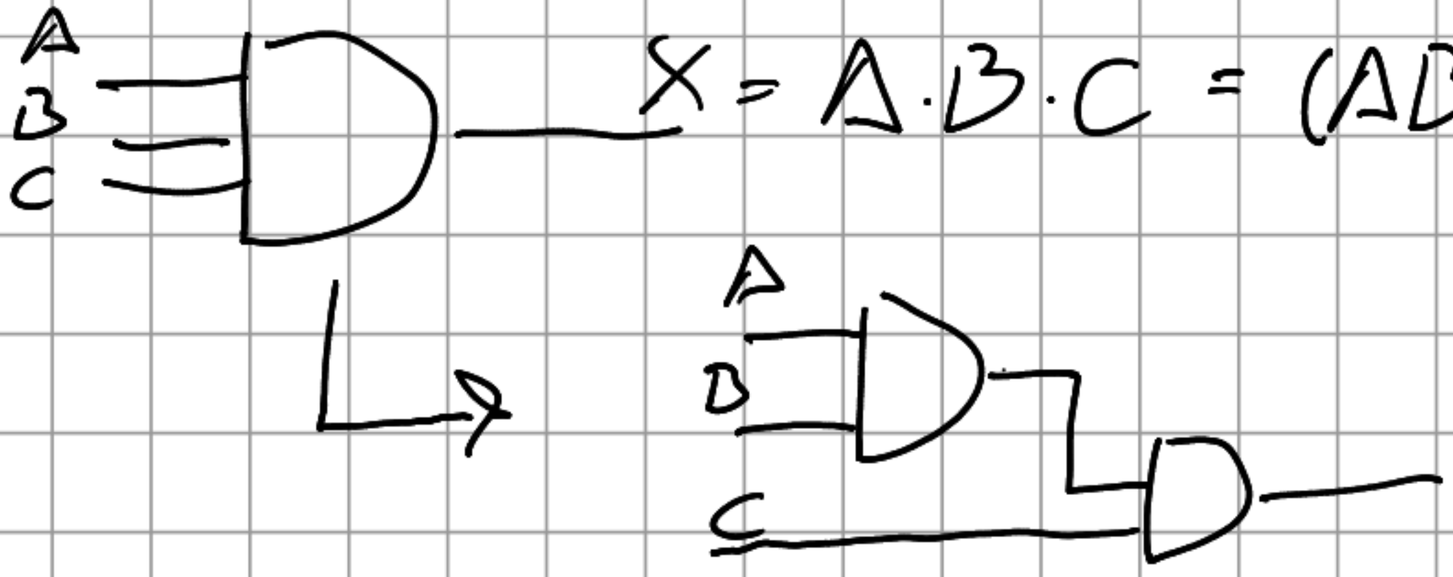
(es: Q = 1, non richiedo di nuovo set)

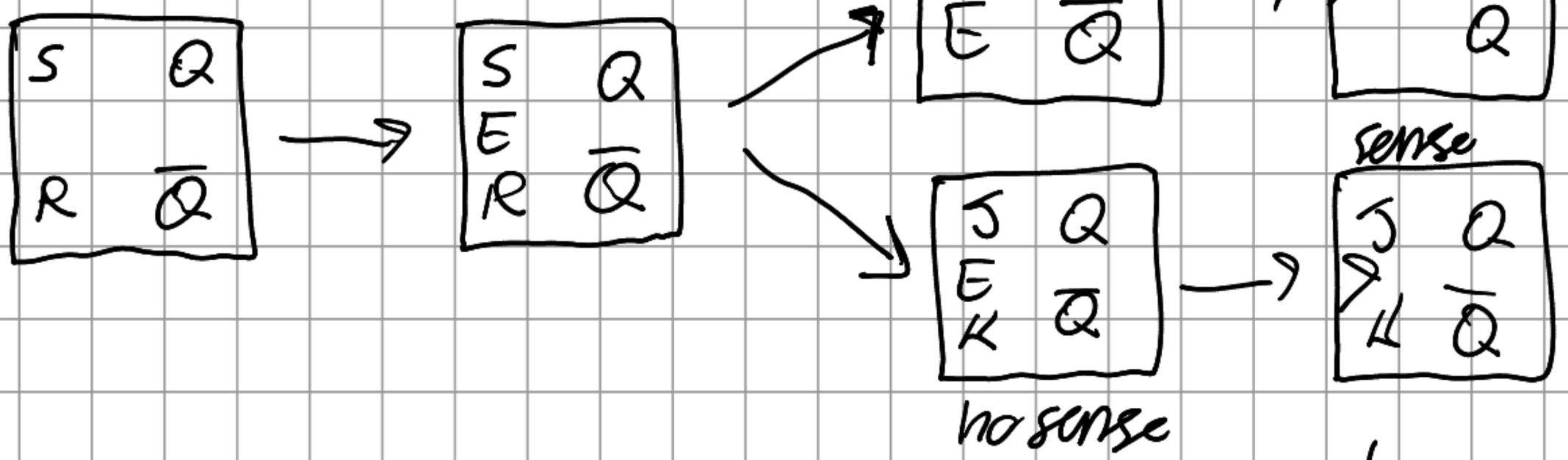


E	J	K	Q	$\bar{Q}$	
0	x	x	HOLD		
1	0	0	HOLD		→ non sto richiedendo nulla
1	0	1	0	1	reset
1	1	0	1	0	set
1	1	1	TOGGLE		



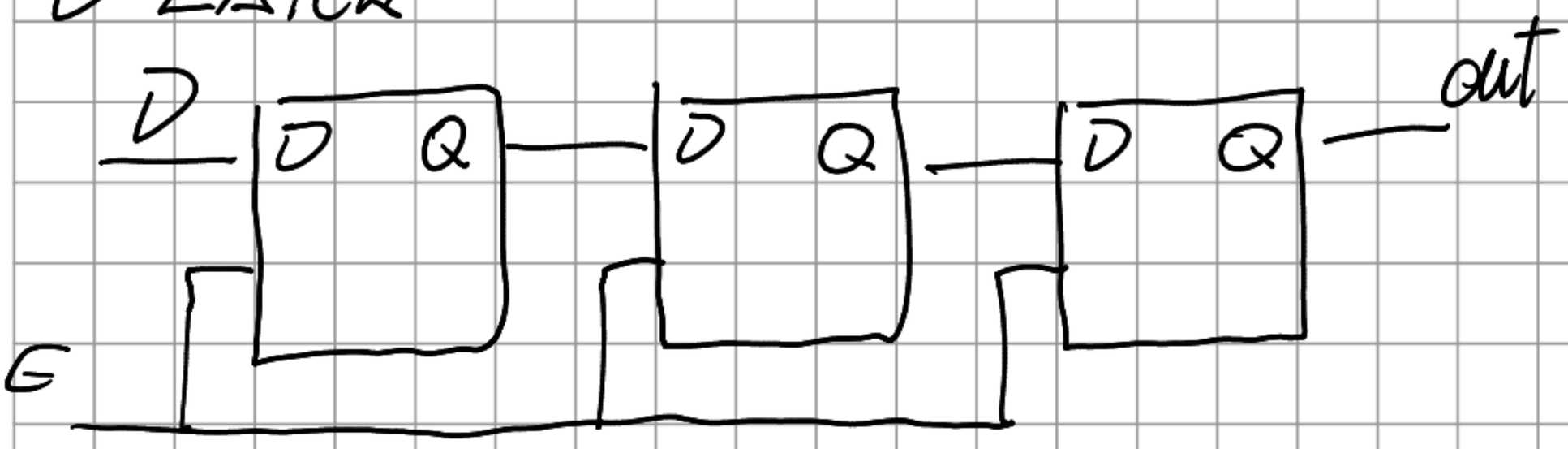
- AND a 3 porte:

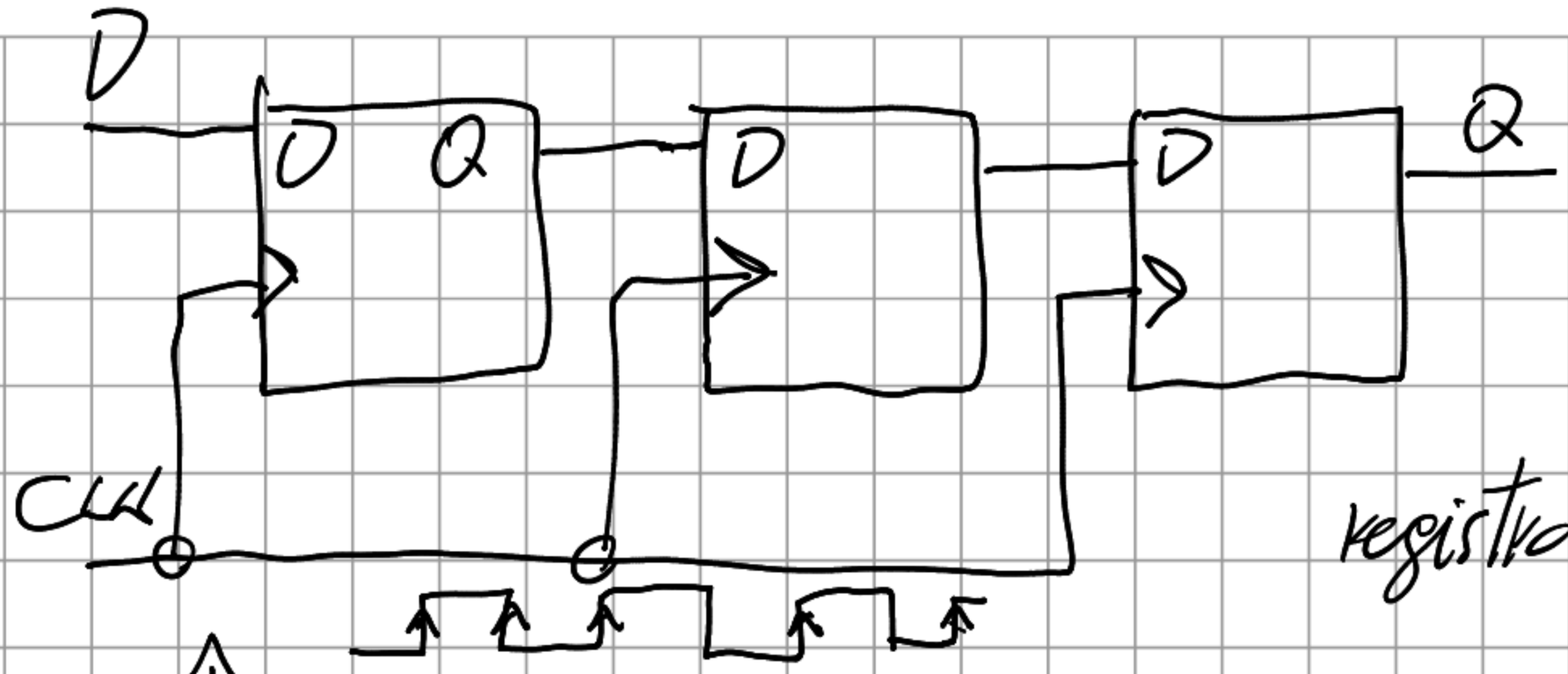




l'operazione avviene 1 volta sola al fronte di Toggle

D-LATCH





registro a scorrimento

fa in modo che al fronte di salita del clock quello che sta in D viene trasferito al seguente alla stessa delay

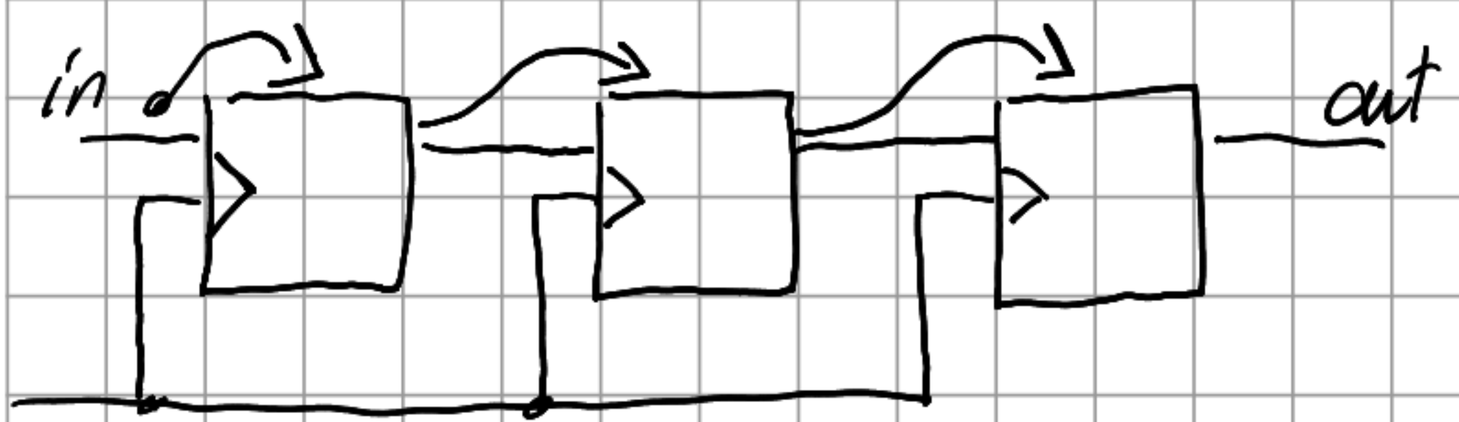
(* 43:00)

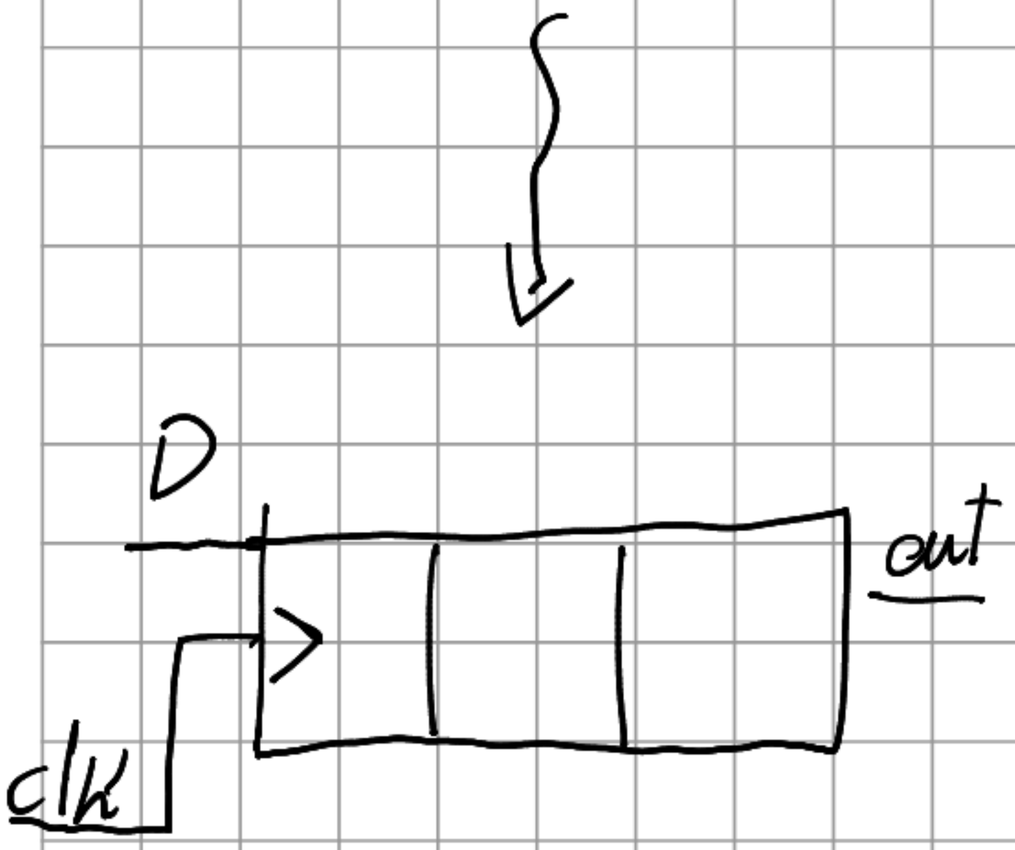
- Arriva a macchina a stati finiti. → elimina race condition

(se loop prop. prende meno di un ciclo di clock)

Lez 07.

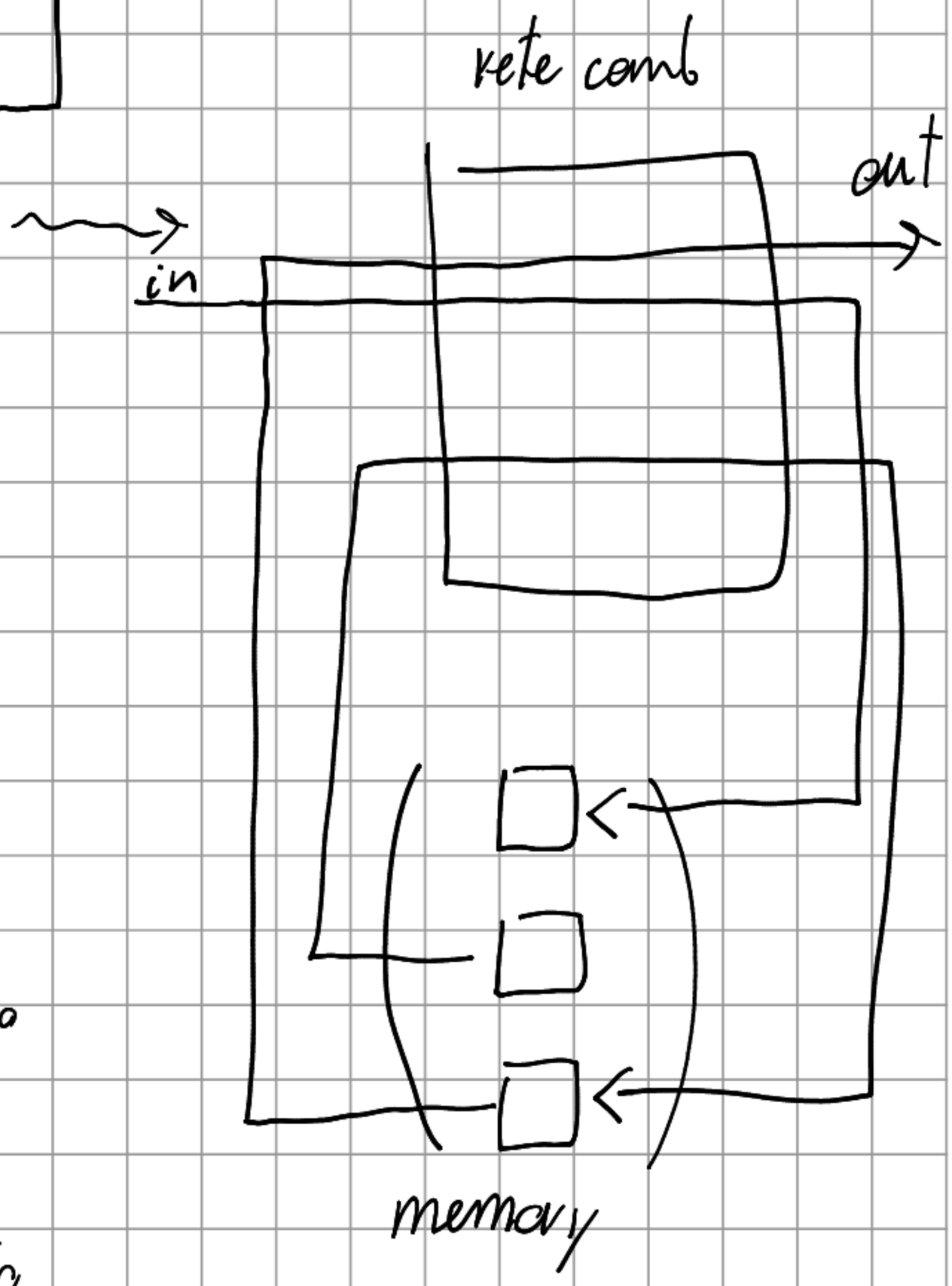
- ESEMPIO: REGISTRO DI SHIFT (di macchina a stati finiti)





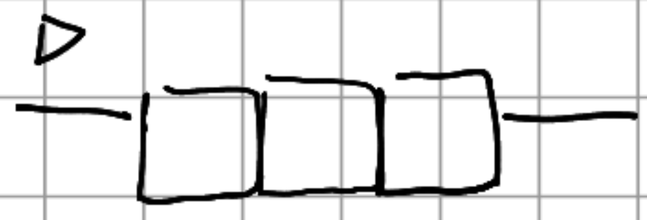
ingresso incamerato nel I° bit e gli altri scorrono

↳ registro a scorrimento

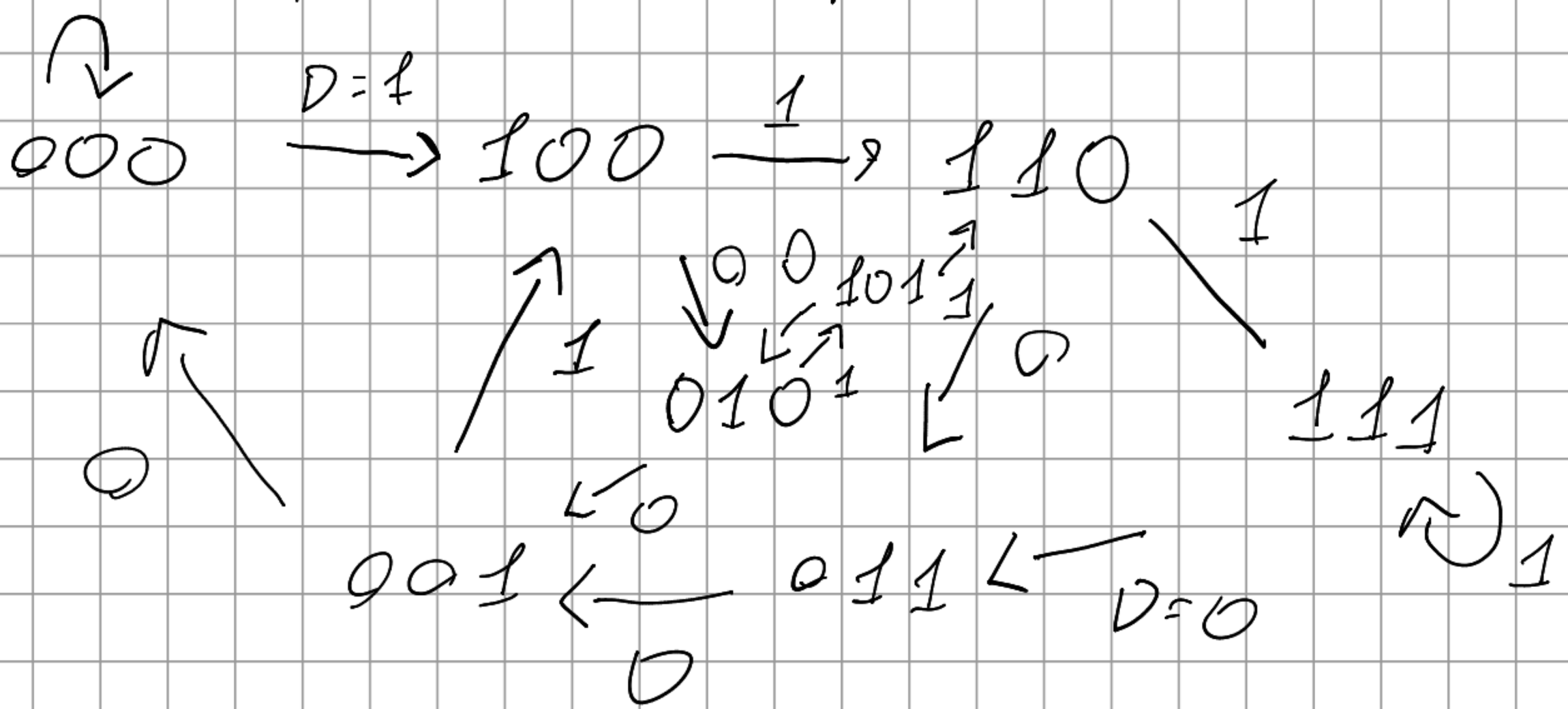


Stato n	IN	Stato n+1	OUT
0 0 0	0	0 0 0	0
0 0 0	1	1 0 0	0
0 0 1	0	0 0 0	0
0 0 1	1	1 0 0	0
0 1 0	0	0 0 1	1
0 1 0	1	0 0 1	1

- Diagramma degli stati:

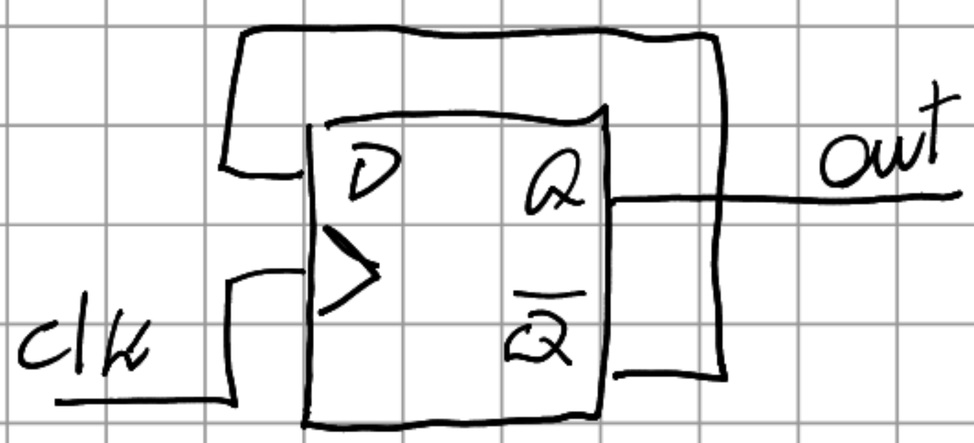


D = 0 (il prox stato va in se stesso)



Qualsiasi sistema digitale (ogni stringa di bit è uno stato, anche la macchinetta del caffè)

- DIVISORI DI FREQUENZA.



continua a saltare da uno stato all'altro

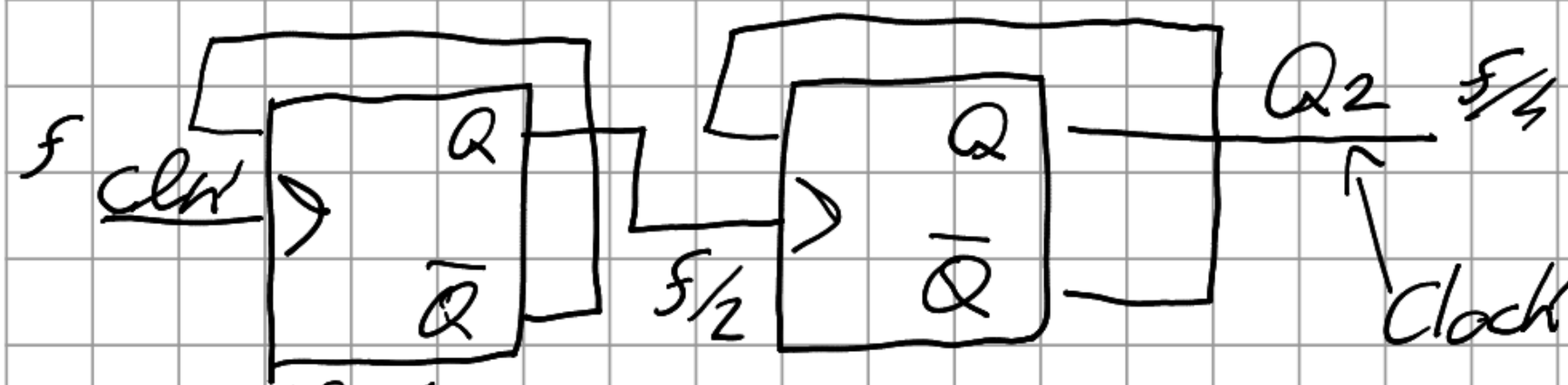
Q2

clk

pallino di fronte a triangolo => flip flop che reagisce a fronti di discesa

→ 1 ← delay

↳ freq. doppia di quella iniziale



0000 → 1000

0001

L'1 propaga ←

0010 ← 0100

Dove uso? CPU che fa + operazioni:

sulla base di dove sta l'1 fa un'op. piuttosto che un'altra.

- Circuiti asincroni → output dell'uno come clock del successivo

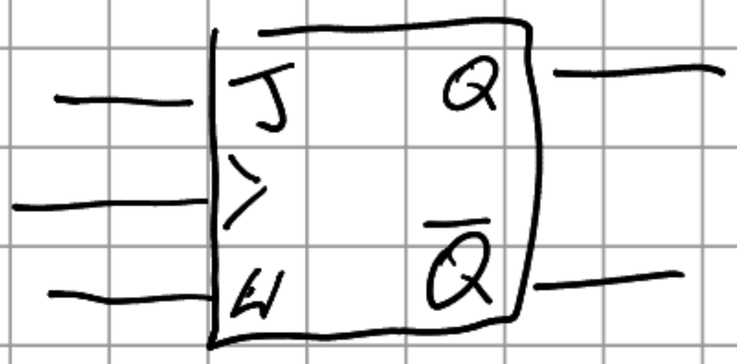
accumulo delay, molto male ↵

↓

Circuito sincrono → tutti i componenti hanno lo stesso clock

- VARIANTE SINCRONA DEL DIV. IN FREQ.

Uso un feedback per eliminare ogni uscita nascosta

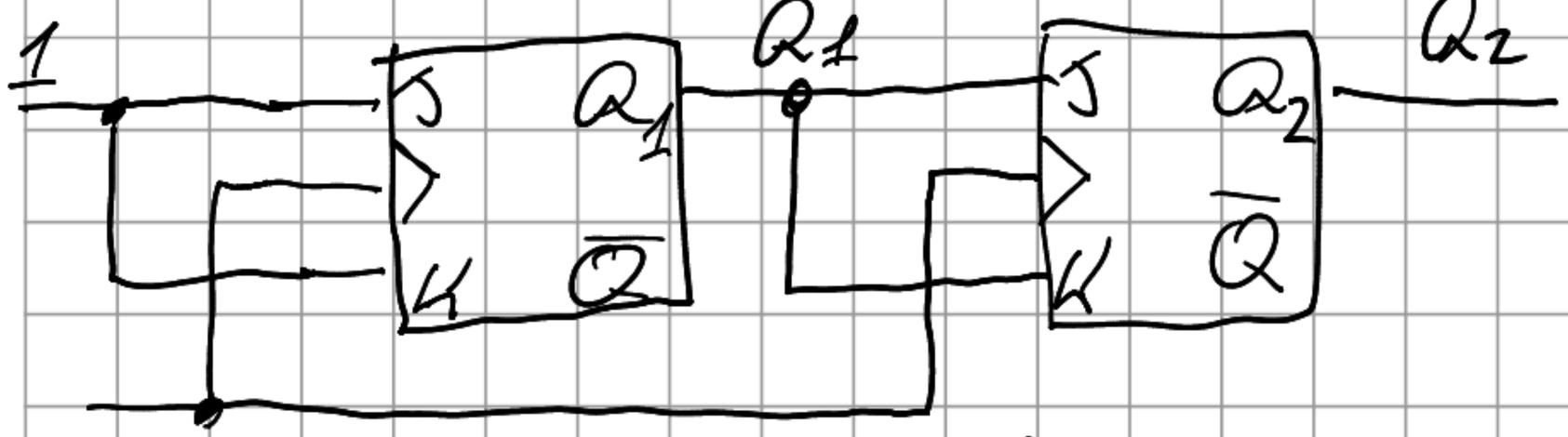


CLK	J	K	Q	$\overline{Q}$	
•	—	—	HOLD		
↑	0	0	HOLD		
↑	0	1	0	1	reset
↑	1	0	1	0	set
↑	1	1	$\overline{Q}$	Q	Toggle

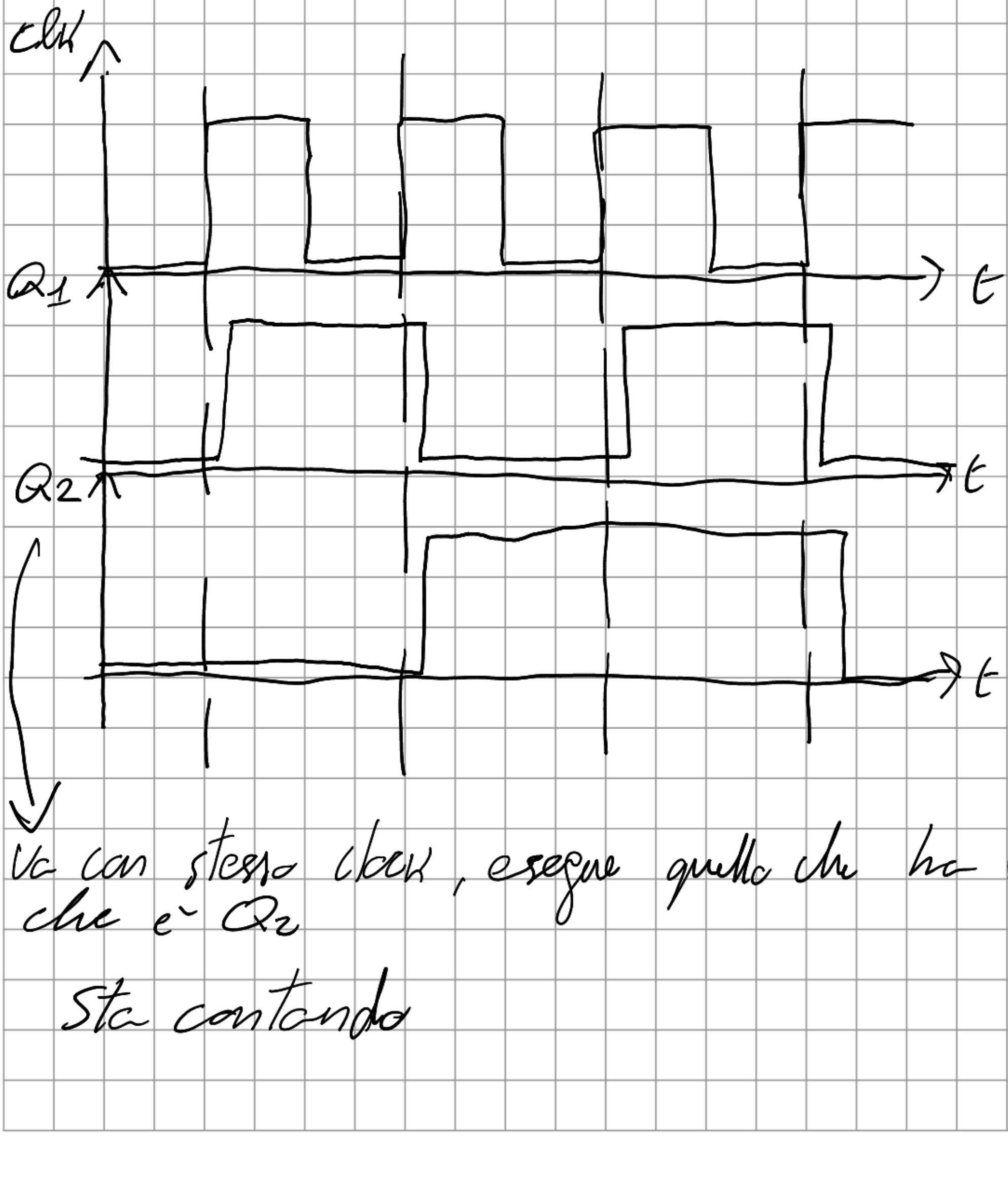
ad ogni ciclo di clock scatta da uno all'altro ↵

- Segue divisore

Toggle config. di default



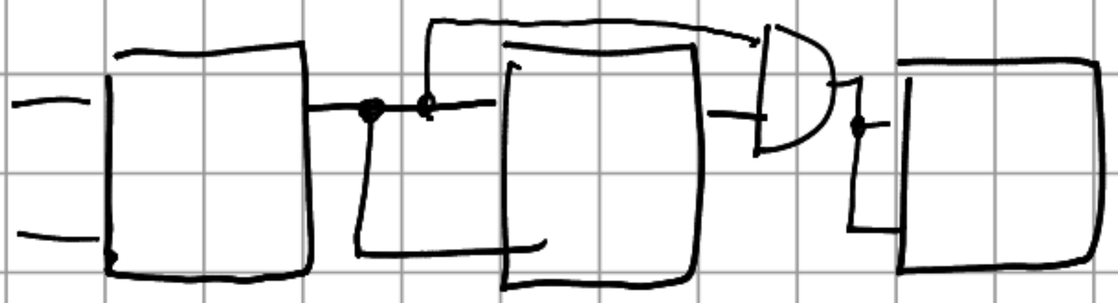
sincrono



stesso delay per tutti

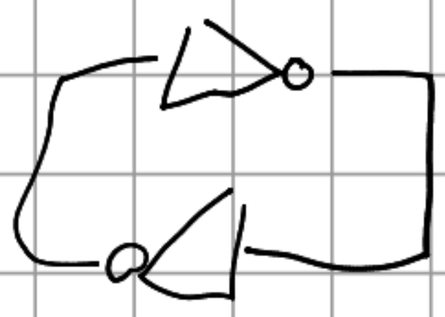
va con stesso clock, esegue quello che ha su J, K che è Q2

Sta contando

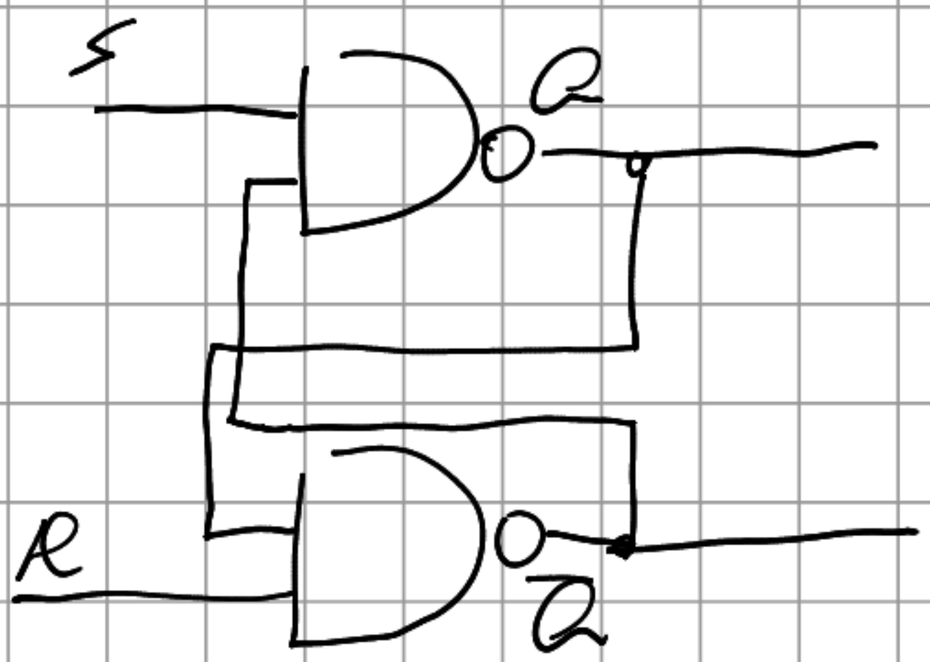


contatore in freq.

Recap, mappe concettuali

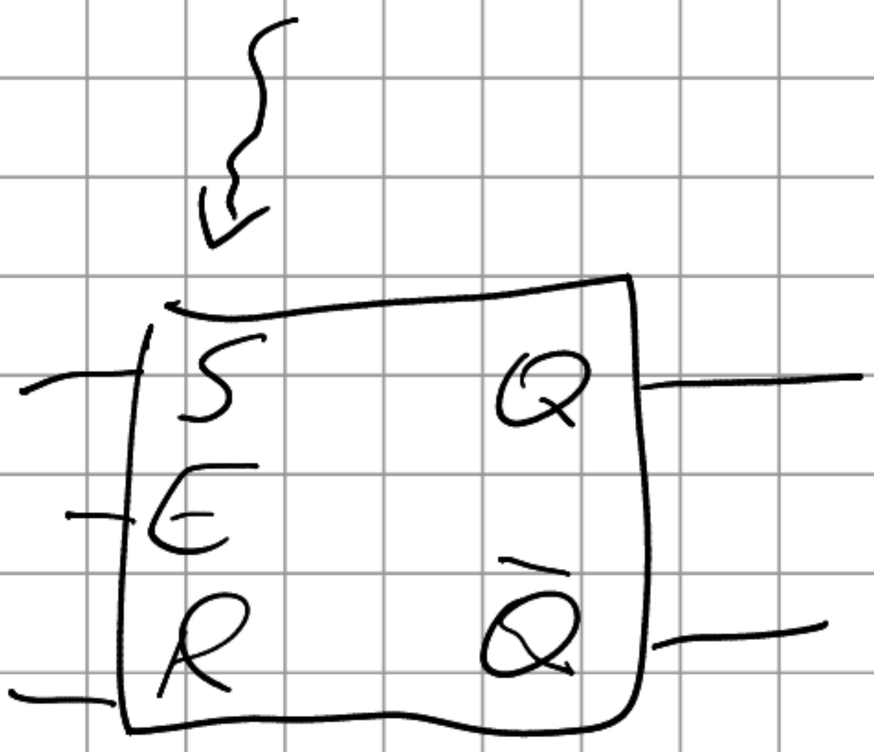


bistabilità se metto feedback (inutile perché circ. resta sempre in stessa config.



LATCH SR

brutto caso illegale

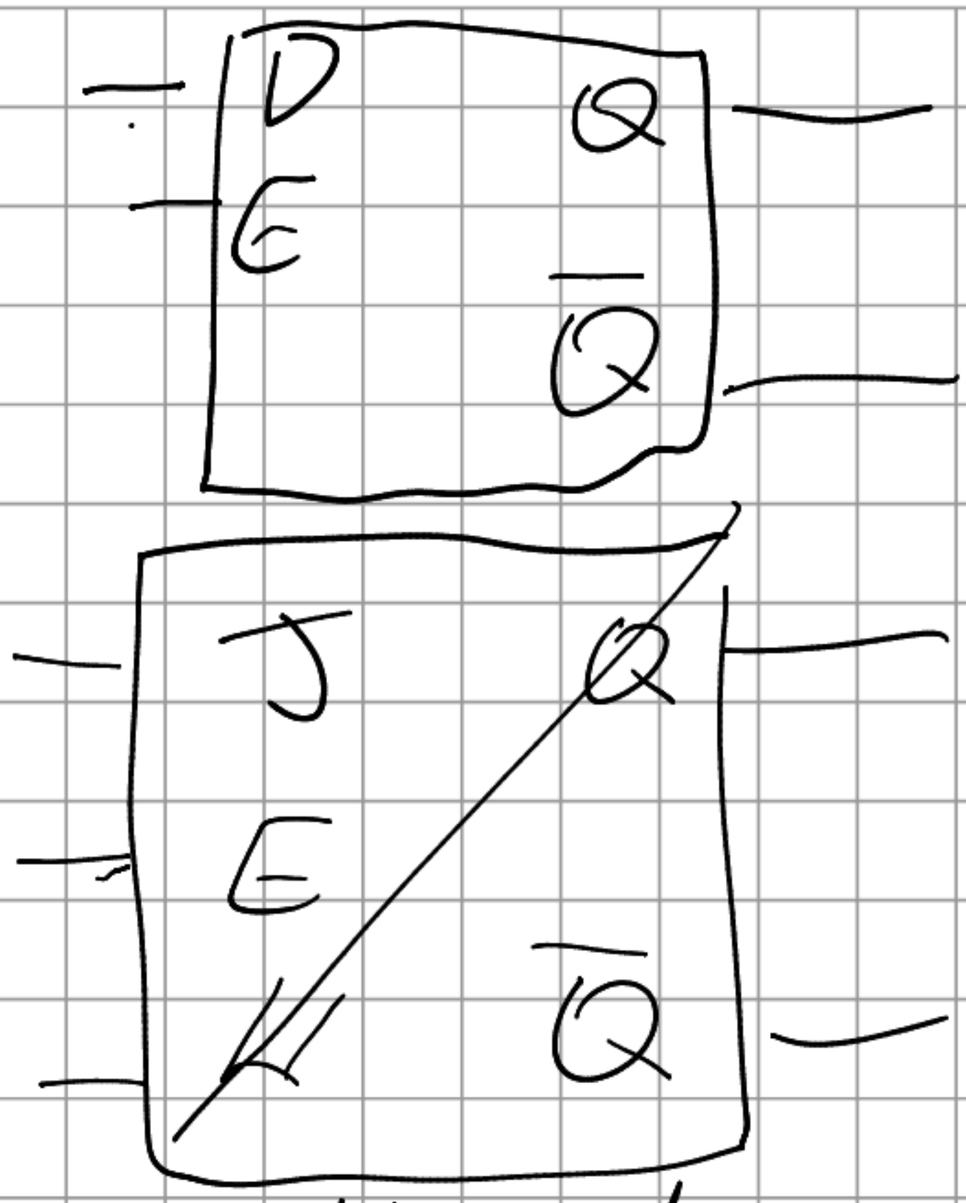


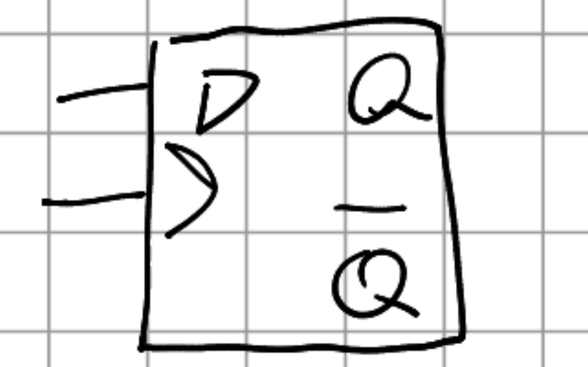
GATED

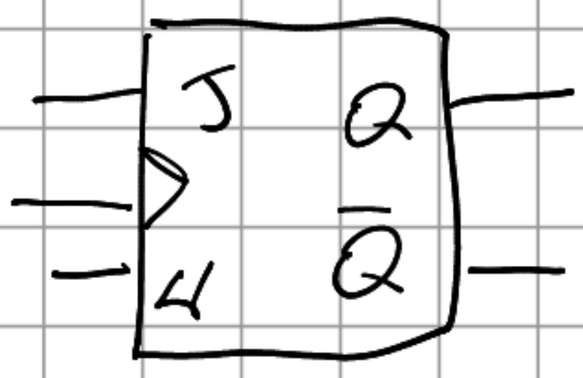
no senso perché
chiedo set-reset insieme

- Diversi modi per risolvere stato illegale

EDGE TRIGGERED







inutile perché
blocca $J=K=1$, ma
rimbalza continuamente
da 1 stato all'altro

FLIP-FLOP

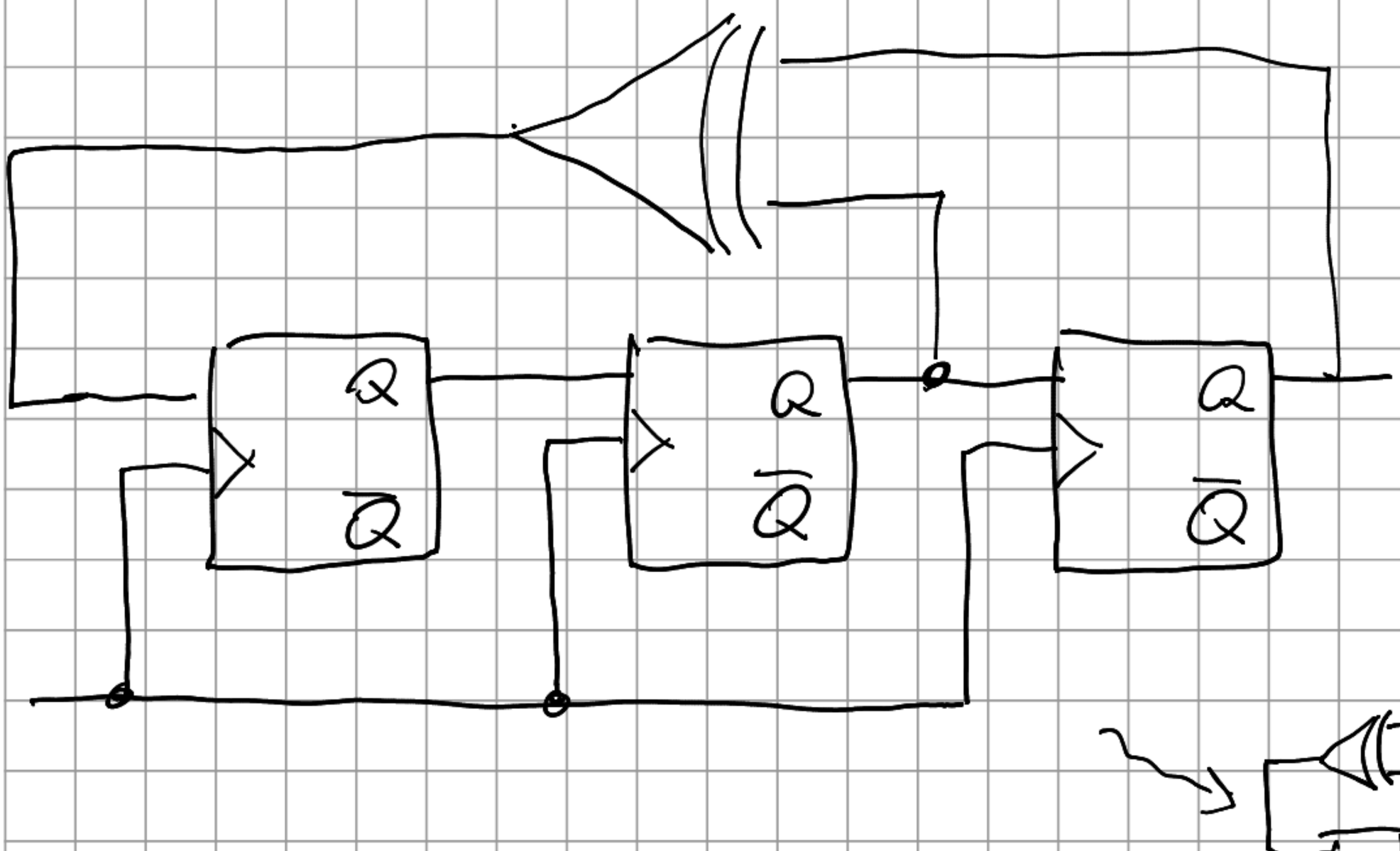
derivazione

↳ flip-flop per
switchare, di
tipo 'T' (toggle)

T flip-flop

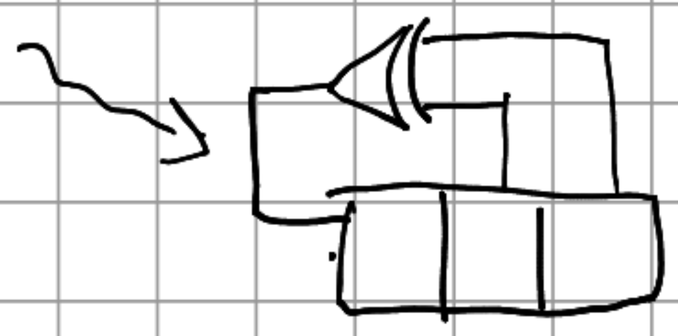
(famiglia di "cose poi trovo nei circuiti logici")

• ESPERIENZA: Linear feedback shift register LFSR.

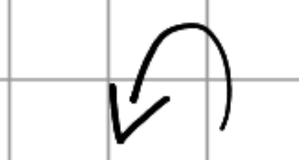


Config. 2+1

Taps

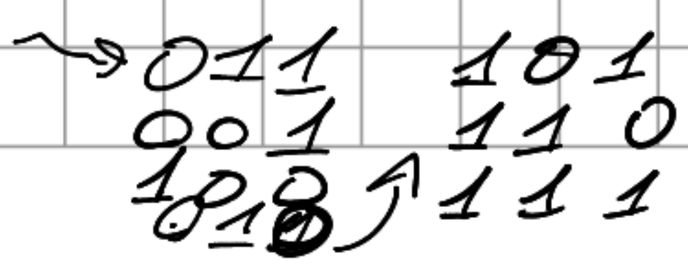


STATO n	XOR	STATO n+1
000	0	000
001	1	100
010	1	101
011	0	001
100	0	010
101	1	110
110	1	111
111	0	011



000

XOR agisce su ultimo bit di stato n
gli altri 2 scattano a destra di 1.



- Seq. pseud. cas. di bit.

CASO 3+1

16 memory config

1 1 1 1
0 1 1 1
0 0 1 1
0 0 0 1 4

1 0 0 0
0 1 0 0
0 0 1 0
1 0 0 1 8

1 1 0 0
0 1 1 0
1 0 1 1
0 1 0 1 12

1 0 1 0
1 1 0 1
1 1 1 0 15
1 1 1 1

time diagram

~> divisore per 15

4 + 1

registri shift con feedback possano essere massimali quando

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

00000

21 config.

7

3

altri 2 cicli

- Seq. bit pseudocasuale → perché non so dove mi trovo nella sequenza se vedo 1

Vedo 0 ... poi?

0

1

1 ... poi? → 0, 1

→ Valgono calcolate P(010); P(110); P(111); P(011)

P(011) = P(111) = 50%

P(010) = 3/7
P(110) = 4/7
) ← BIAS! (Manca freq. con solo zeri)

Predilige gli 1.

P(0|10) = 50%

P(1|10) = 50%

Se trova 4 1 sono a posta, descripto, perché so prevedere gli altri dal feedback.

- Master slave