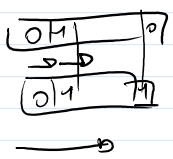
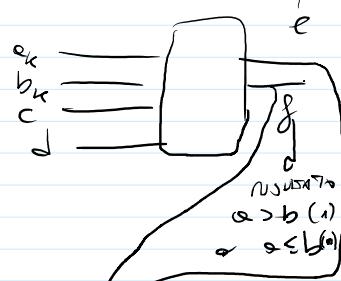


vedere i valori delle cifre successive



scorsiamo e se  
i loro valori  
diversi un  
intercorrispo  
oppure mi fermo quando  
arrivo alle cifre  
meno significative  
(più a dx)

"ho già trovato" /  $a \geq b$   
il risultato  
e  $a > b$   $\downarrow$   
 $a = b$   $\downarrow$   
"vedo  
la cifra  
successiva"



è uscita  
che dicono  
se ho  
trovato  
valori diversi  
e meno

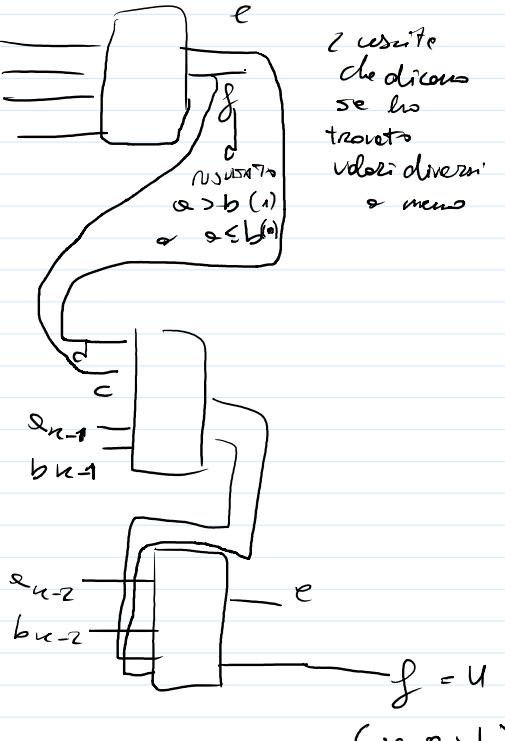
ma di uno dei moduli:

|     | c    | 0    | 0    | 1    | 1    | $\Rightarrow$ | f             | ) come                         |
|-----|------|------|------|------|------|---------------|---------------|--------------------------------|
|     | b    | \    | d    | a    | 1    | 0             | $\Rightarrow$ | ingressi                       |
| 0 0 | 0, 0 | 1, 0 | 1, 1 | 0, 0 |      |               |               | RISULTATO NON TROVATO $\times$ |
| 0 1 | 1, 0 | 1, 0 | 1, 1 | 1, 0 |      |               |               |                                |
| 1 1 | 0, 1 | 0, 0 | 1, 0 | 1, 1 | 0, 0 |               |               |                                |
| 1 0 | 1, 1 | 1, 0 | 1, 0 | 1, 1 | 1, 1 |               |               |                                |
|     | e    | d    | e    | f    |      |               |               |                                |

$e = 1$  risultato trovato  
 $f = 1$  risultato

ripeto "il risultato non  
e' stato ancora  
determinato"

\* non è vero che  $a < b$



\* va bene dirlo per un modulo intermedio,

ma se fosse il modulo finale, anche

se le cifre sono uguali il risultato lo ha ( $a = b$  falso)

Dunque pongo 0 come risultato, anche se

l'uscita - ingresso c indica risultato non trovato

(perché il risultato che ottengo e'  $a_n = b_n$   
se  $c = 0$  (i.e. non trovato))

ma avendo maggiore stretto non ve basta.)

Se cercassi  $a \geq b$   
al posto di zero pongo 1  
perché se  $c = 0$  significa  
che non trovato una  
cifra che le altre sono e  
quindi sono uguali  
(quindi continuo, ma  
intento pongo 1, così  
da avere il risultato,  
altrimenti vero che  $a \geq b$   
 $\rightarrow$  uscire

## Circuiti numerici (moltiplicatore)

giovedì 5 novembre 2020 16:32

Moltiplicatore:  $u = a \cdot b$   
Lo si pensa di sommare tante volte  
un numero (più lento)

2 bit x numero

|       | $b_1$ | 0    | 0    | 1    | 1    |
|-------|-------|------|------|------|------|
| $a_1$ | 0     | 0    | 1    | 1    | 0    |
| 0     | 0     | 0000 | 0000 | 0000 | 0000 |
| 0     | 1     | 0000 | 0001 | 0011 | 0100 |
| 1     | 1     | 0000 | 0011 | 1001 | 0110 |
| 1     | 0     | 0000 | 0010 | 0110 | 0000 |

oche senti

| $a$ | $b$ | $a \cdot b$ |
|-----|-----|-------------|
| 0   | 0   | 0           |
| 0   | 1   | 0           |
| 1   | 0   | 0           |
| 1   | 1   | 1           |

che corrisponde a  
un AND

max rappresentabile è 3, e necessita di 4 cifre binarie (1001)



$$u_3 = a_1 \cdot \bar{b}_1 \cdot b_0 + a_1 \cdot \bar{b}_0 \cdot b_1$$

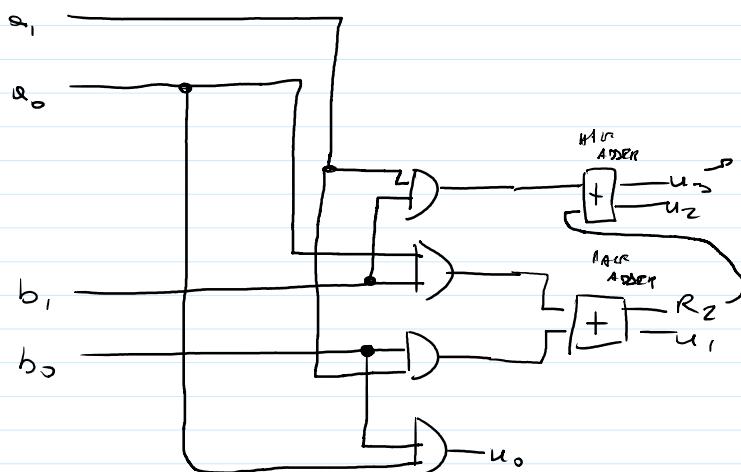
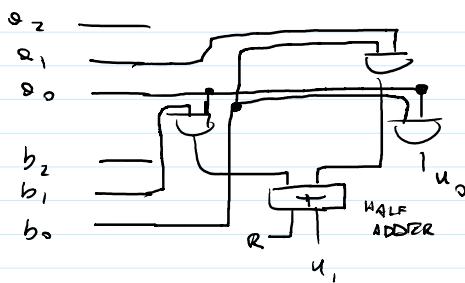
$$u_2 = a_1 \cdot \bar{b}_0 \cdot b_0 + a_1 \cdot \bar{b}_0 \cdot b_1 + a_1 \cdot b_1 \cdot b_0 + a_0 \cdot b_1 \cdot b_0$$

complesso, cerco una  
soluzione modulare

es. di calcolo "a mano"

$$\begin{array}{r} 111 \\ \times 101 \\ \hline 111 \\ 111 \\ 000 \\ \hline 100011 \end{array} = 35$$

Circuito logico



soddisfai questo  
+ zero = 12 parti

$$\begin{array}{r} (R_3) R_2 \\ a_1 b_0, a_0 b_0 \\ a_1 b_1, b_1 b_0 \\ \hline u_3 \quad u_2 \quad u_1 \quad u_0 \end{array}$$

N.B.  $S \cdot 3 = S + S + S$   
 $S \cdot 4 = S + S + S + S$   
 $S \cdot 5 = S + S + S + S + S$

crescita  
lineare

d  
no sono addizionatori  
e' meno efficiente

## Moltiplicazione su 4 bit

per somma dei due sommatori (vedi sopra)

più bit  
di risparmio

$$s_3 \ s_2 \ s_1 \ s_0$$

$$R_5 \quad R_4 \quad b_3 \ b_2 \ b_1 \ b_0 =$$

$$s_3 \ b_1 \ s_2 \ b_2 \ s_1 \ b_3 \ s_0 \ b_0$$

$$s_3 \ b_2 \ s_2 \ b_2 \ s_1 \ b_2 \ s_0 \ b_2$$

$$s_3 \ b_3 \ s_2 \ b_3 \ s_1 \ b_3 \ s_0 \ b_3$$

per somma A & B

ogni "per" e' una funzione "AND"

somma le uscite delle due funzioni AND (i due prodotti)

con un HALF ADDER (2 ingressi e 2 uscite (risparmio))

(risparmio)

nessuna somma, grazie

allo scorrimento di risparmio

4 ingressi (più di un HALF ADDER, ma un addis sommatore completo)

→ Singressi → al risparmio compreso

Sommatore con 5 I e 3 output (per rappresentare tutti i numeri da 0 a 9)

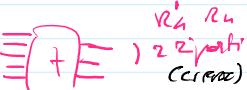
$$u_4 \ u_3 \ u_2 \ u_1 \ u_0$$

HALF HALF

$$(a+b) + (c+d) + e$$

$$b \boxed{+} \boxed{=} \boxed{+}$$

$$d \boxed{+} \boxed{=} \boxed{+}$$



O quando  $\Sigma = 0$   
e prende ogni  $\Sigma = 1$   
( $\text{sum} = S$ )

per sommare:

però con una soluzione modulare

crescita del ritardo (cresce nel migliore dei casi LOGARITMICA)

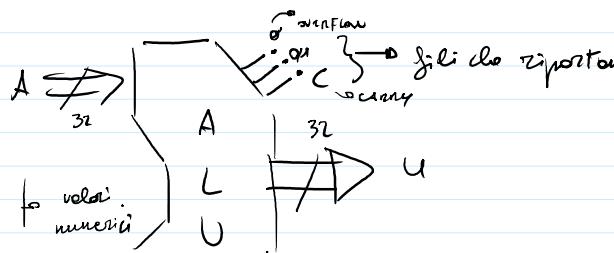
quello dei circuiti sommatori

↳ base 2

N.B. L'eventuale rappresentazione in virgola mobile necessita di un lavoro di denormalizzazione e ri-allineamento delle mantisse.

Spesso nei calcolatori il lavoro e' svolto da un co-processor e non dal processore principale

Si crea un dispositivo che esegue molte più operazioni di calcolo  
[Scelta: lavoriamo su 32 bit]



fili che riportano info aggiuntive (es. "c'è un riporto", che magari non serve)

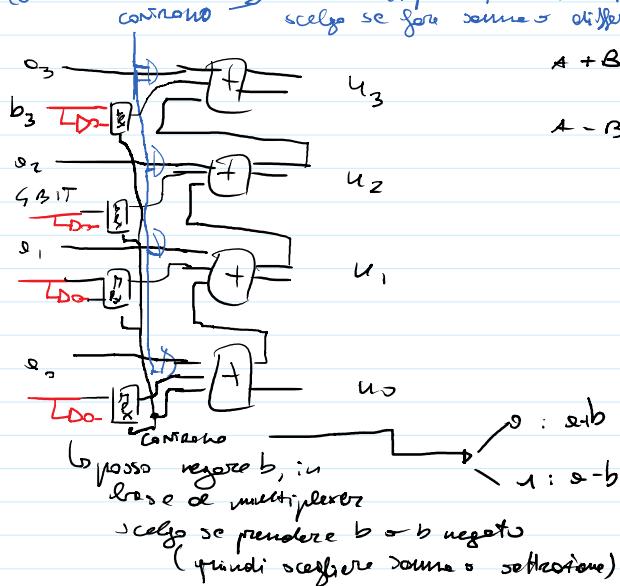
$\xrightarrow{1}$  33° BIT

controlli  $\xrightarrow{3}$  determinano l'operazione da effettuare fra le funzioni implementate (somma, C-Z, tipo LZ, operazioni logiche → elementari (AND, OR, XOR, NOT))  
ognuno ha una certa configurazione

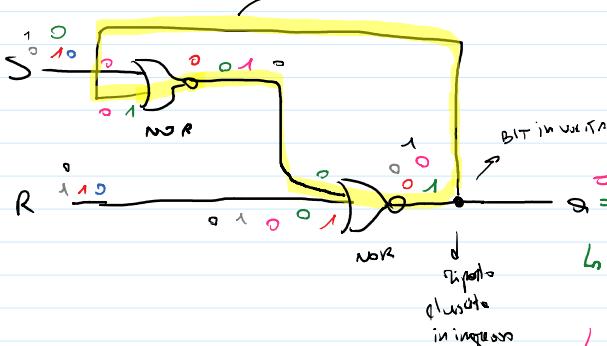
↳ onore in una sezione posso immagazzinare in un il risultato e in ausilio (se zero u) i due numeri sono uguali (1) o diversi (0)  
↳ onore posso tenere dell'overflow

→ se volevi fare operazioni differenti non crea altri circuiti: tramite multiplexer sulla zona di controllo posso fornire le varie configurazioni

(alternativa)



circuito sequenziale per bit chiuso (bisogna anche vedere le fasi di verità)



| S | R | Q                                    |
|---|---|--------------------------------------|
| 0 | 0 | { 0 → può assumere entrambi i valori |
| 0 | 1 | 1                                    |
| 1 | 1 | 1                                    |
| 1 | 0 | 0                                    |

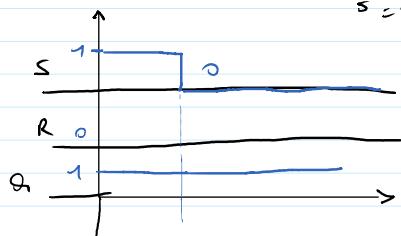
per sceglierli deve avere dei valori in ingresso più

Lo se ipotizzo  
l'uscita sia 1, lo verifico tramite le funzioni

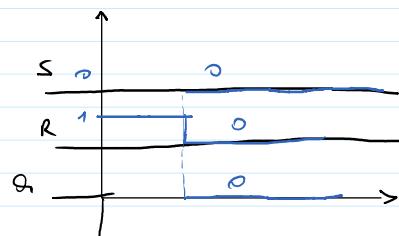
Lo se suppongo  $Q = 0$  allora 0  
la conferma

Dunque posso considerare delle combinazioni di sequenze

$$S = 1, R = 0, Q = 1$$

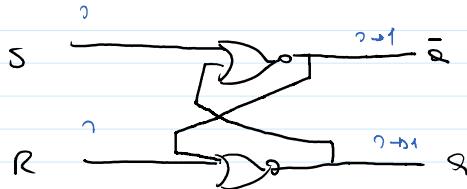


$$S = 0, R = 1, Q = 0$$



Per scoprire il valore di  $Q$  devo andare "indietro nel tempo" e vedere le sequenze di  $S$  e  $R$  precedente, per decidere se  $Q = 1$

Il circuito sequenziale è spesso denotato nel seguente modo:



Il caso 11 è considerato controproducente, si considerano utili solo 3 delle 4 combinazioni possibili, si evita la 4<sup>a</sup>.

Questo perche altrimenti per  $S = 1$  e  $R = 1$  avrei l'uscita dei NOR = 0, dunque  $Q = \bar{Q}$  devo a bordo zero.

FLIP-FLOP di tipo

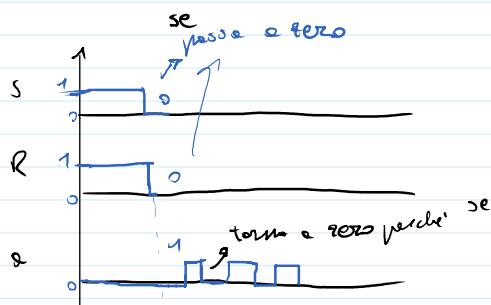
S-R

Set - Reset

forza a 1 forza a zero

$\left\{ \begin{array}{l} \text{Se saggi } S=1, R=1, Q=0 \\ \text{se saggi } S=0, R=0, Q=0 \text{ e } R=1 \\ \text{quando } S=0, R=0 \text{ il dispositivo} \\ \text{memorizza il valore} \\ \text{che aveva precedentemente eseguito} \\ (\text{l'uscita perde il valore in memoria perché} \\ \text{prima di entrare nel FLIP-FLOP}) \end{array} \right.$

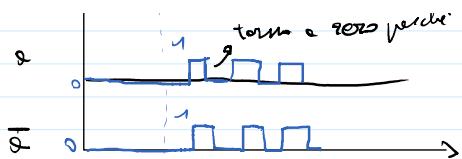
con un ritardo E ottengo in uscita lo stesso valore in memoria



$Q = 1$  allora anche gli ingressi non cambiano → è continuo e essendo negati

il sistema sta alternando periodicamente 0 e 1

ha creato una specie di generatore



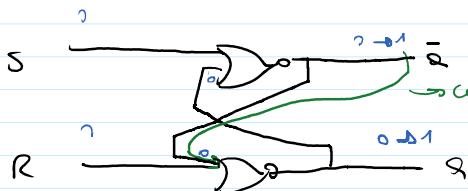
l'oscillazione sta alternando periodi lenti e i  
elli massimi velocità  
possibile

→ In questi anni  
specie di generatore  
di clacson


 E E E  
 dopo un certo  
 ritardo il dispositivo  
 si accorge e cambia  
 i valori

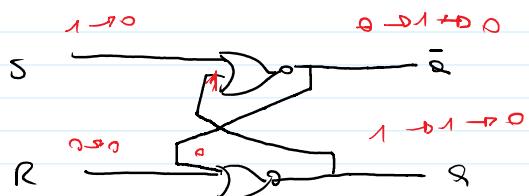
d  
lesiono  
evitare questi ] **INSTABILE**  
continuo rientro di  
veloci  
(raggi anteriori da 0 a 11 e 00 (SR))  
per questo 11 non si usa

prime or  $n = 1, 1$  or  $0, 1$



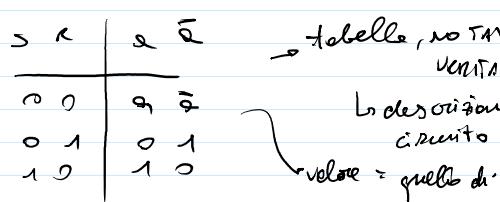
essi dunque o, 1 ingresso binario in uscita, non 1, comincia con zero. Ha di nuovo lo stesso valore in entrata le uscite (perché entrambi si aggiornano dopo un certo ritardo)

N.B. Non mi sfidacco se vorrai correre veloce in ingresso.  
(se è diverso)



per la situazione e' OK (e' 1/1 dei da problemi)

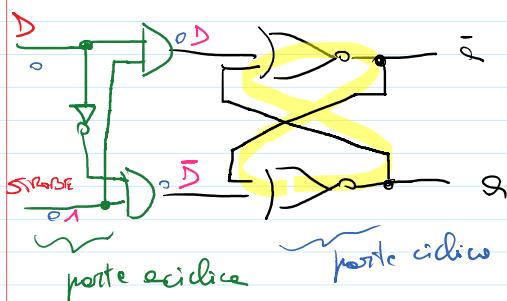
| A | B |  | work |
|---|---|--|------|
| 0 | 0 |  | 1    |
| 0 | 1 |  | 0    |
| 1 | 1 |  | 0    |
| 1 | 0 |  | 0    |



La descrizione del circuito sequenziale  
volte = quello di prima

↳ diagnostic evidence indicates the key  
per se cognitive goal can be very questionable

## Für für die Tiere



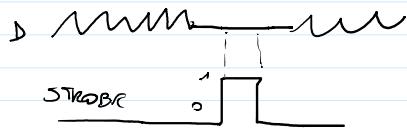
aggiunge al FET MOS J-FET un circuito che entri  
il corso 1,1.

] ho ingresso nel circ. scq.  
uguali a 0,0

è la base per costituire una memoria;

portando con  $stroke = 1$ , in base  
al valore di  $\lambda$  si avrà il risultato di

portando con strobe = 1, in base  
al valore di  $D$  conservo il valore di  
 $D$  e meno, "fotografando" il suo valore  
precedente e riportandolo alle sue



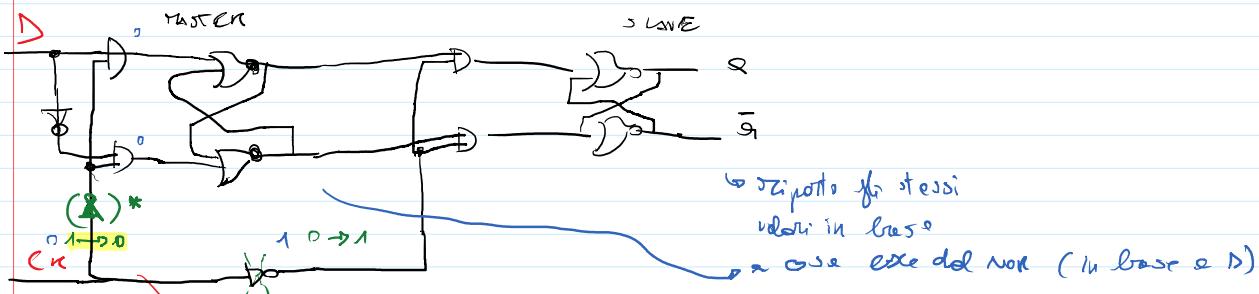
$\triangleright$  puoi cambiare una  
quando strobe passa  
da 0 a 1  
 $\triangleright$  due mettersi fissi,  
dopo di che puoi tornare  
ad alternare i valori  
quando strobe passa  
da 1 a 0  
(pensa a una macchina fotografica,  
vedi contatto)

Lo vantaggio è il valore fisso +  $D$  per brevi intervalli  
di tempo (= inclusi di zero, dispositivi lenti)

$\hookrightarrow$  elice oare una specie di "otturatore" (con tempo con le lame, esponendo un'istantanea, di luce, di un oggetto in movimento).

Nei circuiti si utilizza una configurazione detta **MASTER-SLAVE**.

consiste nel replicare lo stesso dispositivo FLIP-FLOP SR



se  $CK = 1$ , il master regole le uscite di  $D$

"di due istantanee"  $\rightarrow$  dice momento del master e lo slave lascia il valore precedente per scegliere e mantenere in memoria  $D$

l'uscita rimane stabile anche nel caso in cui cambia l'ingresso del master che, mentre elice la variazione di  $D$ , ha già scambiato con il photocoupler lo stato ma ancora in memoria il valore

La  $D$  non deve più rimanere fisso  
può variare continuamente e lo "fotocoupler"  
la scatta su una delle varianti

il tempo di "scatto" può  
essere il doppio dei dispositivi  
più veloci, ma non dipende  
da quanto faccio mantenere  
un certo valore all'ingresso  
clock (quando viene fatto  
la foto)

| $D$ | $CK$ | $S$ | $\bar{Q}$ |
|-----|------|-----|-----------|
| 0   | 0    | 0   | 1         |
| 1   | 1    | 1   | 0         |

$\rightarrow$  le uscite rimangono  
in memoria

\* aggiungendo un NOT ottengo  
la sensibilità in salita  
(aggiunto  $\rightarrow$  del  $CK$ )

**CORPO DI TIERRA EDGE TRIGGERED**  
(sensibile alla variazione, in  
discesa (valori vicini) o in  
salita)

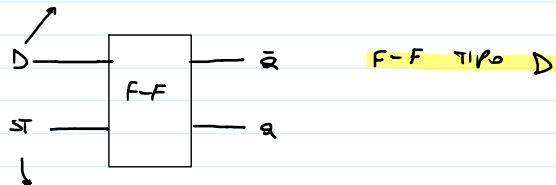
I flip flop possono essere così rappresentati:



(il tipo di F-F posso specificarlo sul nome delle scatole o, come in figura, lasciarlo agli ingegneri)

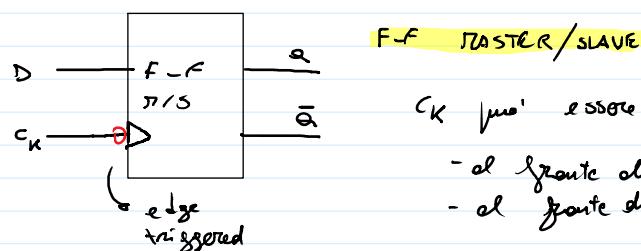
N.B. In ingresso ho 1,1

porto il valore da memorizzare in  $q$



memorizzazione tramite variazione da 0 a 1 di STROBE.

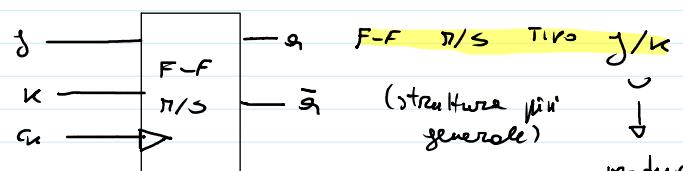
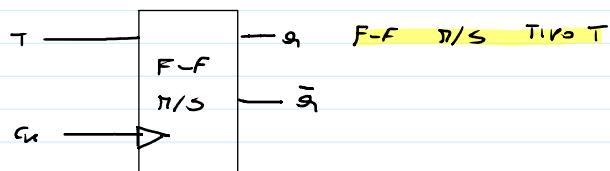
Svantaggio: tenere fisso D mentre STROBE varia può determinare LENTITÀ o problemi di vario genere in altri dispositivi che usano D



$C_K$  puo' essere sensibile:

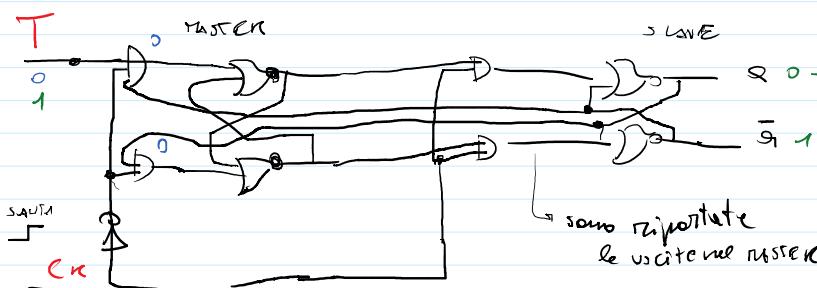
- al fronte di salita (valori. da 0 a 1)
- al fronte di discesa (valori. da 1 a 0) → in questo caso aggiungo un pull-up

N.B. Il funzionamento non e' del valore ma sulla variazione (memorizzo sempre D in  $q$ )



produce comportamenti diversi

TIPO T



$Q: 0 \rightarrow 1 \rightarrow 0$        $T=1$   
 $\bar{Q}: 1 \rightarrow 0 \rightarrow 1$        $T=0$

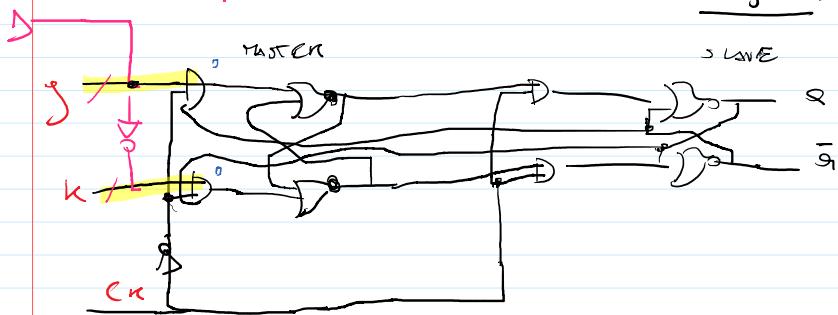
si mantengono le uscite  
comportamento simile  
a quello di S-R  
ma la velocità e'  
lavora sulle velocità  
di variazione di CK

d'uso prevalente da re viene fatto per avere circuiti

centatori (passa centore gli impulsi di clock e rappresentare in binario il numero di impulsi in un certo lasso di tempo)

un solo impulso centrale gli impulsi di classe e  
d'impulsore in lavoro (il numero di impulsi in un certo lasso di tempo)

se voleassi ripetizione  
il tipo D



TIR  $\gamma - \alpha$

S WNE

ottengo gli stessi risultamenti  
 del tipo T ma posso  
 avere INVERTITI i valori g e K,  
 potendo rappresentare anche il  
 tipo D ( resistenze in uscita o > 1  
 cambiando configurazione  
    g-K

| j | k | ca | a | b |
|---|---|----|---|---|
| 0 | 0 | 5  | 0 | 0 |
| 0 | 1 | 5  | 0 | 1 |
| 1 | 0 | 5  | 1 | 0 |
| 1 | 1 | 5  | 0 | 0 |

$\rightarrow$  equivalent  $T \otimes T$

→ " TWO D (inverted o in e)

→ " Tiro > (inscrição e link)

$\rightarrow^{\wedge}$  TIPRO T

le ero prima  
t'ero e d'altra

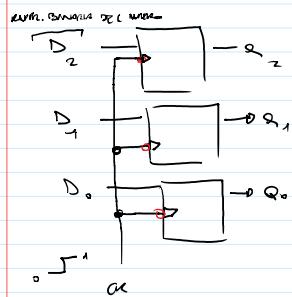
Je was in jasse

N. 3.

Rimarrebbe INTROVATO  
(voce assente in solita)

## REGISTRI TIPO D (DATA)

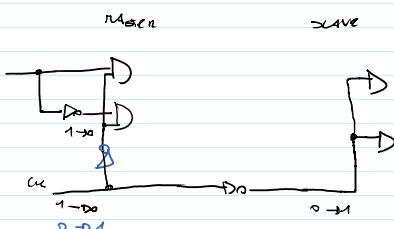
Per registro si intende una sequenza di flip-flop (in questo caso di tipo D)



Memorizza all'interno del flip-flop il valore di D quando ha una variazione del clock (grande al salire nell'sequenza)

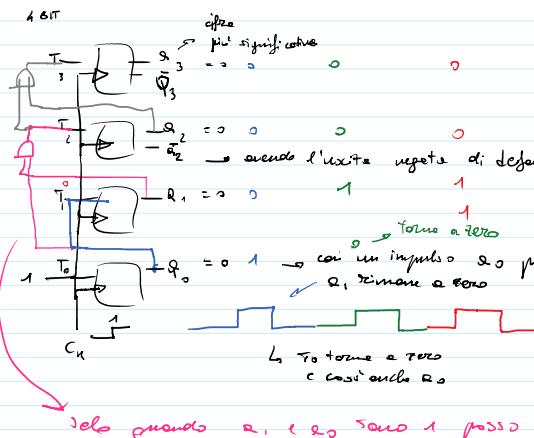
→ un nuovo valore si registra quando si ha una nuova variazione: prendendo il valore costante se il valore D rimane in memoria

fine di discesa / punto  
della fine del



se cambio il circuito aggiungendo la negazione del clock, non metto il polino o sulla rappresentazione è scelta vera.

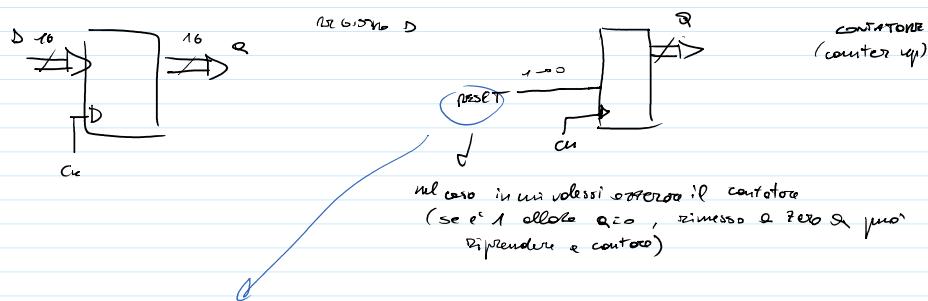
## REGISTRI TIPO CONTATTORE (TIPO T (toggle, commutatore))



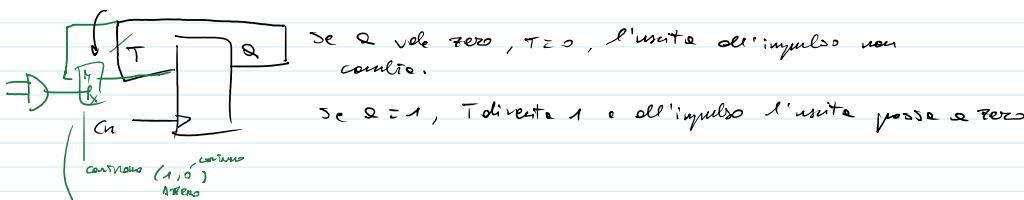
ho ottenuto la sequenza 0, 1, 10, 11

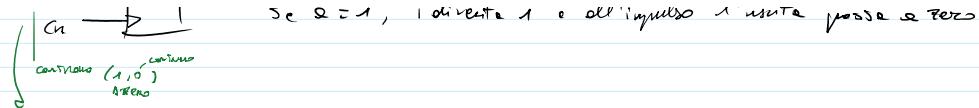
avendo l'uscita negata di default nel flip-flop tipo T non c'è bisogno di inserire funzioni di negazione se voglio il valore opposto all'uscita

Potrei costruire il registro anche in questo modo:



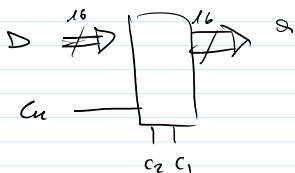
per ottenere il F-F tipo T





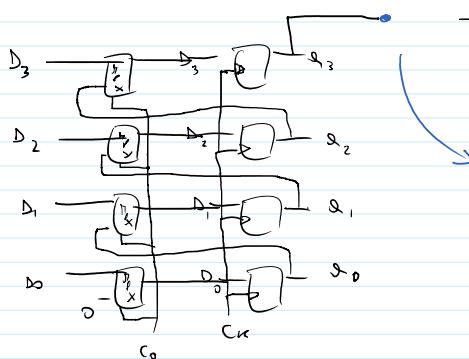
con un multiplexer posso scegliere se continuo conto o arrestare

aggiungendo più vir di controllo posso creare funzionamenti diversi per il registro  
contatore (es. incrementare il valore, arrestare, contare in esatti, contare all'indietro)



### REGISTRO A SCORRIMENTO

F-F TIPO  $\Rightarrow$



se  $C_0 = 1$  posso incrementare un valore  
 se  $C_0 = 0$  e faccio variare il clock  
 se inserisco uno zero al fondo  
 dei bit con uno scorrimento verso  
 l'alto (cioè ora  $d_0$  ora  $d_1$  ecc.)  
 $d_0 \rightarrow d_1 \rightarrow d_2 \rightarrow d_3$

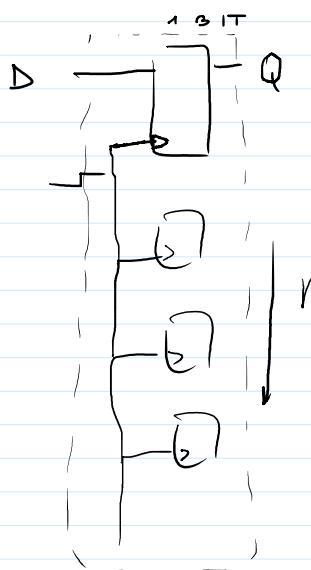
Può servire per trasmettere informazioni  
 codificate attraverso un unico filo  
 (magari ricavante mediante una altra macchina)  
 attraverso più cicli (es. 1 BIT per un ciclo,  
 per 32 BIT avrò 32 cicli)

Facendo scorrere per più di una posizione verso l'alto (es dr 2, dr 4 ecc...) posso

realizzare un circuito moltiplicatore (x2) : sfrutto le somme con scorrimento

(lo svantaggio è di richiedere un numero di cicli di clock crescente rispetto al n° di BIT)

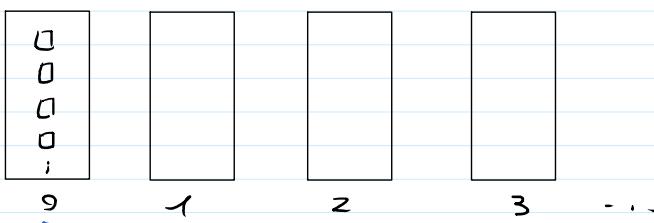
N.B. Se scorro verso il basso posso ottenere la divisione per 2.



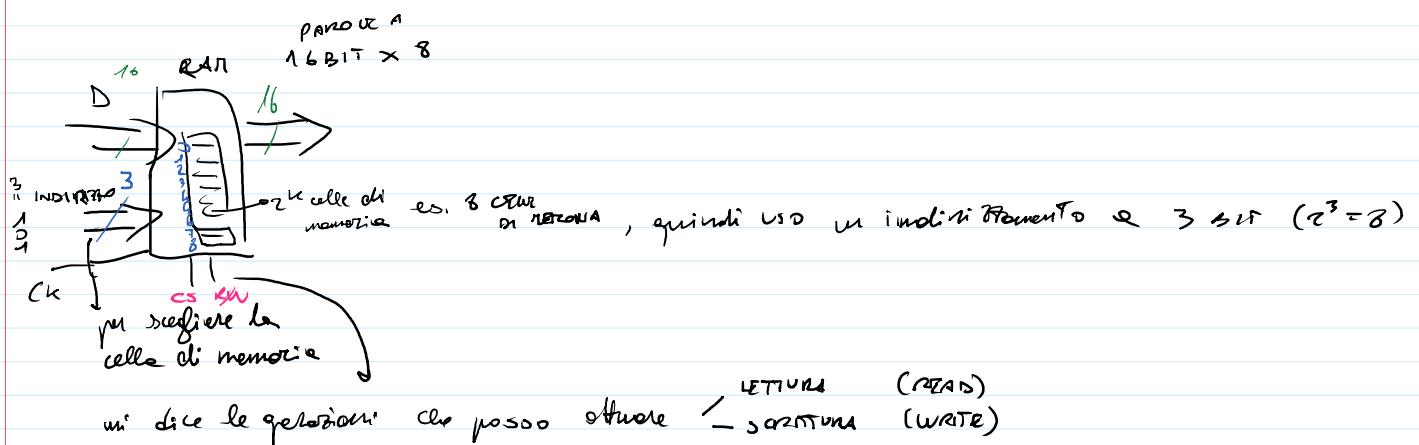
con un impulso memorizzo il valore

posso farla replicando il registro, composto da 4 bit  
(es. con 4 rappresento  $2^4$  numeri senza segno))

Quando tanti registri posso memorizzare tanti numeri diversi:



→ tale numero è detto **INDIRIZZO** e ci permette di accedere in lettura o scrittura a tali registri (che sono quindi identificati da tale numero) **CELLE DI MEMORIA**



- Quando sono silenzio che il controllo vuole scrivere e l'indirizzo è salto alla posso all'impulso di clock

- Per le letture scelgo l'indirizzo, posiziono il controllo sulle sequenze di scrittura e mi aspetto di ricevere un'informazione a 16 bit in uscita (dopo una attesa di ingresso di 16 bit)

• Posso anche effettuare NO-OP, nessuna operazione e 3 bit non bastano

Con i controlli:  $CS - \text{chip select} = 0 \rightarrow \text{non utilizzo la RAM}$  anche se ci sono dati in memoria

$CS = 1 \rightarrow \text{dispositivo attivo}$

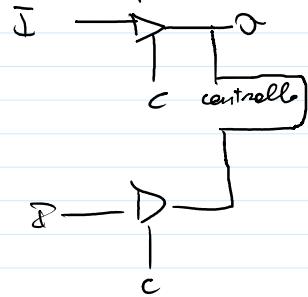
L'elenco  $\Sigma$  de  $RW = 1 \rightarrow \text{lettura}$

L'elenco  $RW = 0 \rightarrow \text{scrittura}$

Tale configurazione ci permette di esporre lo spazio, in particolare gli i fili verso l'esterno, posso economizzare i fili da 32 a 16 utilizzando  $D$  per la lettura e  $D$  per la scrittura. E' però difficile da realizzare.

La selezione e' l'argomento di dispositivo a 3 stati

OPERAZIONI IDENTICHE ( $D = J$ )



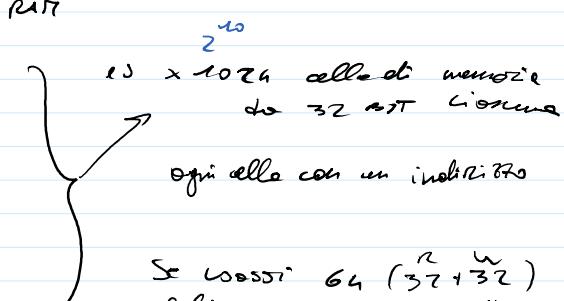
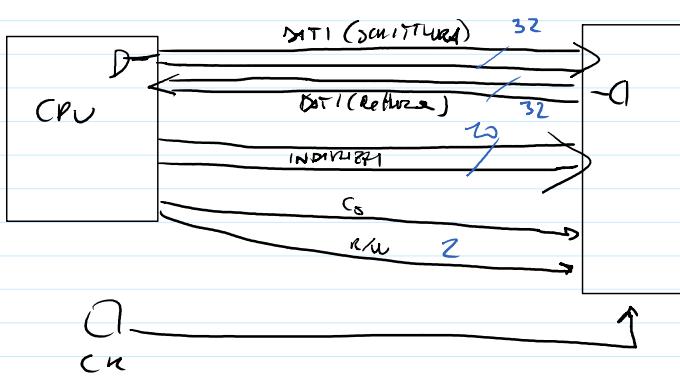
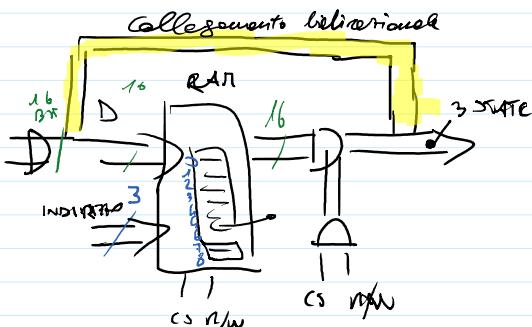
| I | C | D                         |
|---|---|---------------------------|
| 0 | 0 | l'uscita non e' collegata |
| 0 | 1 | 0                         |
| 1 | 0 | non collegato             |
| 1 | 1 | 1                         |

l'uscita non e' collegata  
non collegato

quando stocca  
non ha valori  
in uscita

CONDIZIONE DI FG

Lo permette di collegare più uscite 3-STATE su uno stesso filo senza creare malfuncionamenti avviamente abilitando un solo dispositivo per volta (gestibile tramite il controllo, ma più diversi = 1)

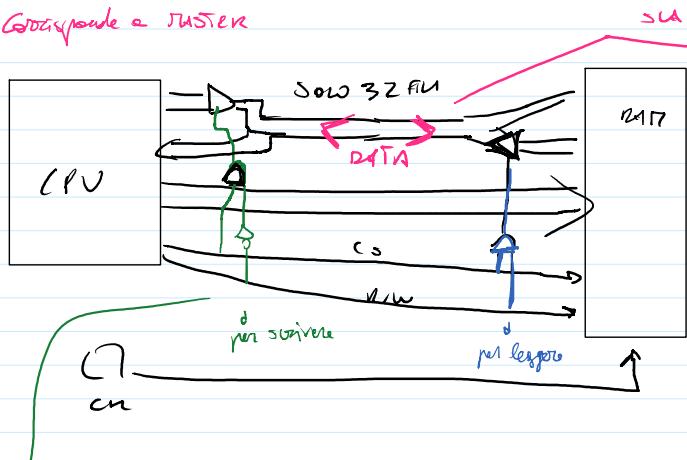


Se ho solo 64 ( $32 \times 32$ ) fili, posso fare scrittura e lettura contemporaneamente. Ma perché voglio solo una delle quattro per volta, ne uso solo 32

Corrisponde a JUSTIN

- SLAVE

Corrisponde a JASTER



e' bidirezionale, in base  
a cosa voglio fare  
(R o W)

Così facendo solo un dispositivo  
di 3 stati e' attivo per volta.

R/W

/ \ scrittura  
1 lettura

quindi un dispositivo  
3 stato serve per  
e l'altro per W

→ il not e' dato dal fatto di scegliere