

Laboratorio di Segnali e Sistemi - Capitolo 8 -

Elettronica digitale: circuiti sequenziali



Claudio Luci
SAPIENZA
UNIVERSITÀ DI ROMA

last update : 070117

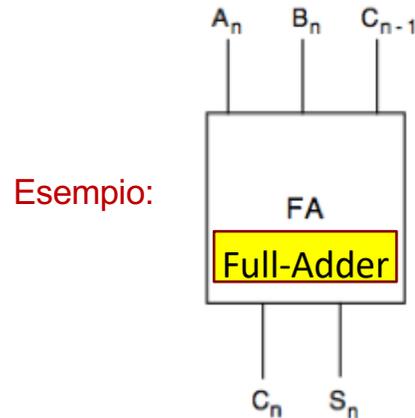
Sommario del capitolo:

- Circuiti sequenziali: introduzione
- Latch e Flip-Flop: S-R, J-K, D e T
- Shift register
- Contatori
- DAC: a pesiera e con rete R-2R
- ADC

Circuiti sequenziali

Circuiti sequenziali

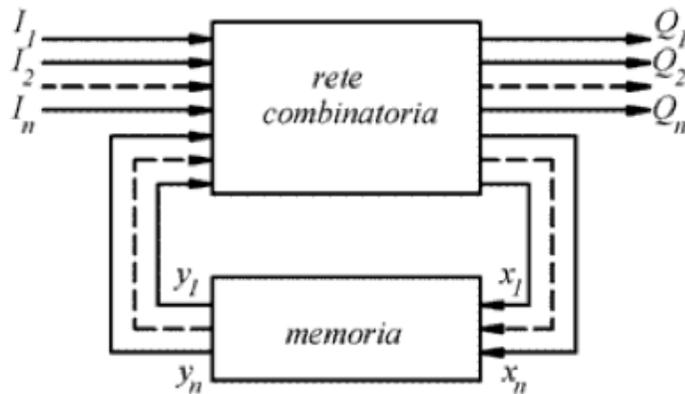
- Tutti i circuiti digitali studiati finora (porte logiche, sommatore, encoder, etc...) appartengono alla categoria dei **circuiti logici combinatori**.



I valori presenti in uscita dipendono esclusivamente dalle combinazioni binarie in ingresso e non dalla sequenza temporale con cui tali combinazioni vengono fornite al circuito.

Se cambiano gli ingressi cambiano anche i valori in uscita (non teniamo conto dei ritardi intrinseci dei vari componenti).

- Nei **circuiti sequenziali** le uscite dipendono anche dalla sequenza, cioè dall'ordine con cui vengono forniti gli ingressi e dallo stato pregresso del circuito, tenuto in conto da un elemento di **memoria**.



Le uscite Q sono una funzione degli ingressi I e dello stato passato del circuito registrato nella memoria.

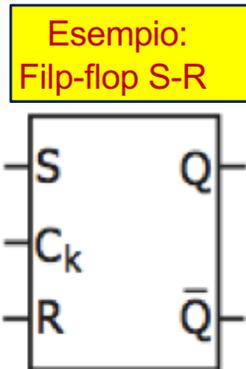
Occorre tener conto della propagazione dei segnali attraverso i vari elementi. È necessaria una sincronizzazione esplicita che viene fatta tramite un segnale di clock.

Il "building block" dei circuiti sequenziali è il flip-flop.

Latch e flip flop

Latch e flip flop

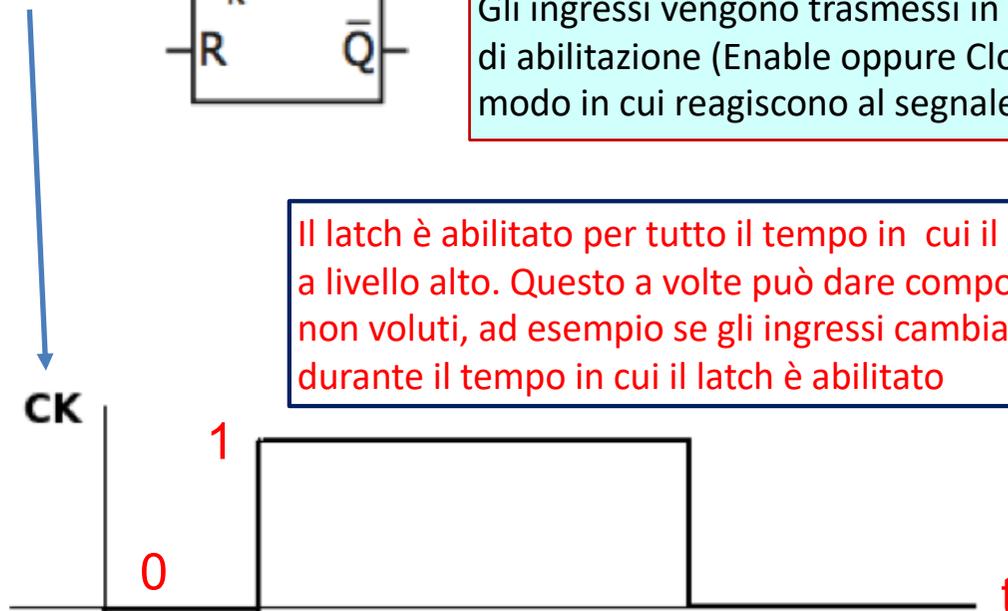
- L'elemento base dei circuiti sequenziali è il **latch** (lucchetto) oppure il **flip-flop** (vedremo tra un attimo la differenza tra i due). Essi sono dei dispositivi in grado di memorizzare un singolo bit, che può assumere i valori 0 oppure 1.



I latch e i flip flop hanno due uscite non indipendenti, Q e il suo negato \bar{Q} . L'uscita dipende dal valore presente ai suoi due ingressi, come specificato dalla tabella della verità.

Gli ingressi vengono trasmessi in uscita solo quando è presente un segnale di abilitazione (Enable oppure Clock). Latch e flip-flop si distinguono dal modo in cui reagiscono al segnale di abilitazione.

Segnale di clock:



Il latch è abilitato per tutto il tempo in cui il clock è a livello alto. Questo a volte può dare comportamenti non voluti, ad esempio se gli ingressi cambiano durante il tempo in cui il latch è abilitato

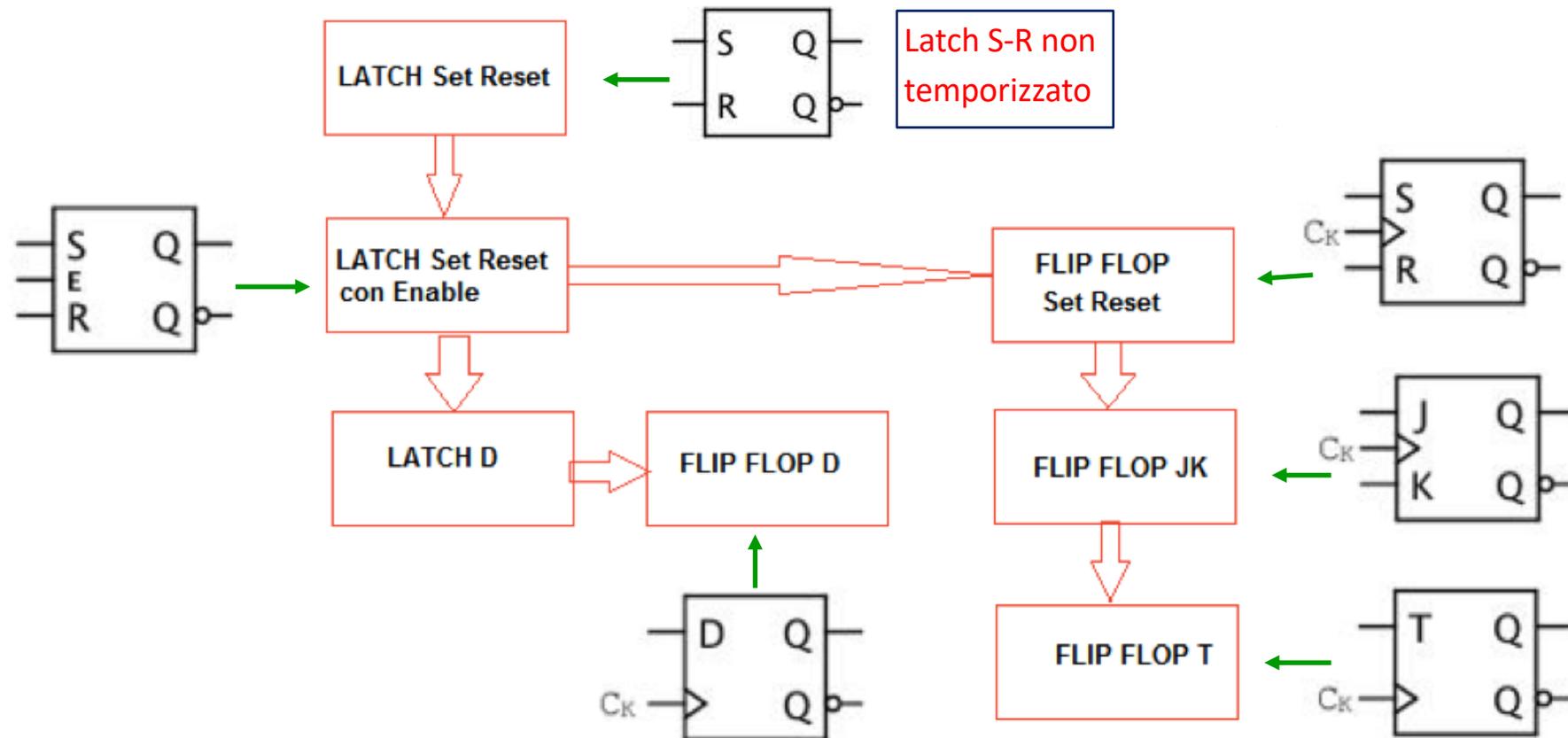
Nei flip-flop invece gli ingressi sono abilitati solo durante il tempo di salita (o di discesa) del clock, quindi per un tempo che può essere molto più breve di quello di un latch. Si dice che sono dei dispositivi "edge triggered".

Tabella della verità del flip-flop S-R

S_n	R_n	Q_{n+1}
0	0	Q_n
1	0	1
0	1	0
1	1	?

Latch e flip flop

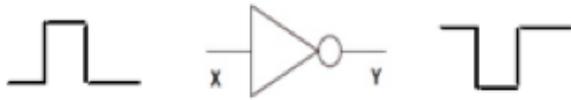
- L'elemento più semplice è il **latch S-R non temporizzato**, ovvero che non è dotato di un ingresso di abilitazione
- A partire da esso si possono costruire diversi tipi di latch/flip-flop che sono poi utilizzati per formare circuiti sequenziali più complessi.



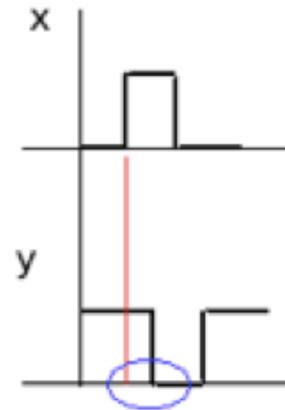
N.B. nel libro di Nigro viene chiamato Latch S-R solo quello non temporizzato, mentre tutti gli altri vengono chiamati flip-flop, distinguendo se sono "edge triggered" oppure no.

Come "triggerare" sul fronte del clock

- Per selezionare il fronte di salita del clock si può, ad esempio, sfruttare il ritardo di propagazione del segnale in una porta NOT.



Nel caso ideale l'uscita dovrebbe commutare immediatamente, ma in realtà lo fa con un certo ritardo. Come si vede, per un certo periodo sia l'ingresso che l'uscita del NOT si trovano a livello alto.

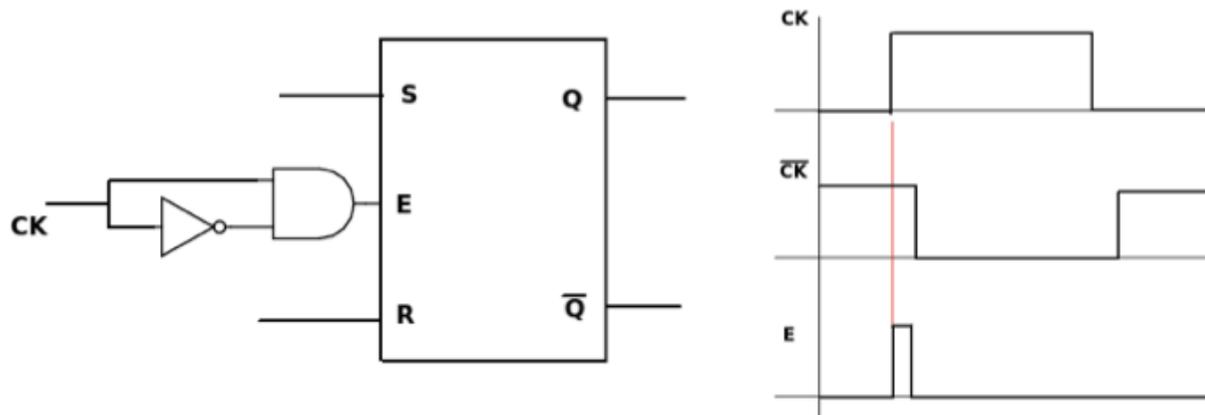


ritardo di propagazione

Famiglia logica	Ritardo (ns)
CMOS 4000	300
HCMOS	23
CMOS ACL	9,5
TTL LS	15
TTL FAST	6



- Si può trasformare un latch S-R con Enable, attivo su tutto il fronte del clock, in flip-flop S-R attivo solo sul fronte di salita del clock usando lo schema di riferimento seguente:



Il flip-flop è abilitato solo durante il tempo di propagazione del segnale all'interno del NOT.

Utilizzando un NOR al posto dell'AND è possibile avere un flip-flop che viene abilitato sul fronte di discesa del clock.

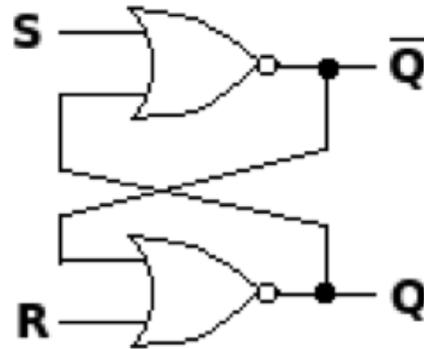
Un altro modo per avere un flip-flop a partire da un latch è quello di costruire la versione master-slave

Latch S-R (set-reset) non temporizzato

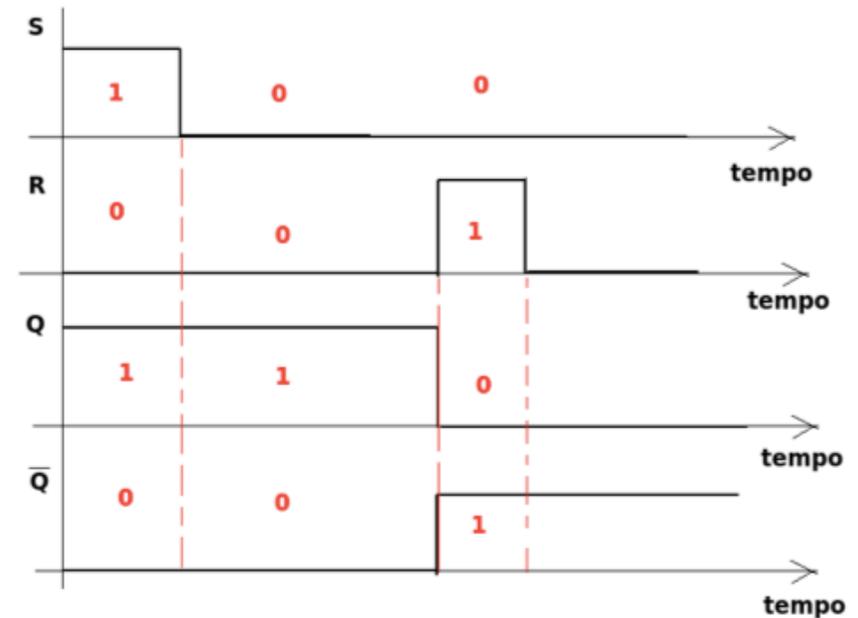
- Costruiamo un latch S-R con due porte NOR (sul libro di Nigro sono invertite le posizioni S e R):

L'uscita di un NOR viene riportata all'ingresso dell'altro

N.B. Possono anche essere usate due porte NAND, ma la logica viene invertita.



Set – Reset



- La tavola della verità del Latch S-R è la seguente:

La combinazione 0 – 0 non fa cambiare lo stato delle uscite.

La combinazione S = 0 e R = 1 porta l'uscita Q a 0

La combinazione S = 1 e R = 0 porta l'uscita Q a 1

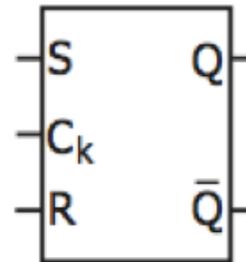
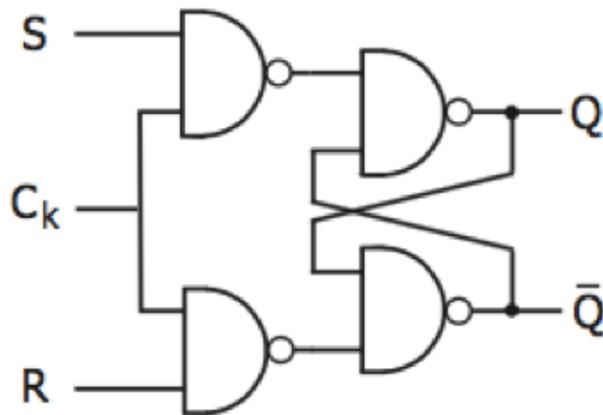
S	R	\bar{Q}	Q
0	0	\bar{Q}_0	Q_0
0	1	1	0
1	0	0	1
1	1	0	0

La combinazione S = 1 e R = 1 non può essere usata.

- La condizione S=1 e R=1 non è stabile. Entrambe le uscite rimangono a 0 solo finché in ingresso c'è 1-1; non appena l'ingresso torna a 0-0, l'uscita si porterà in uno stato non predicibile.

Flip-Flop S-R (set-reset)

- Partendo dal latch S-R non temporizzato si costruisce il latch/Flip-Flop S-R dotato di un ingresso di clock (Ck) che abilita il circuito per fare la transizione.

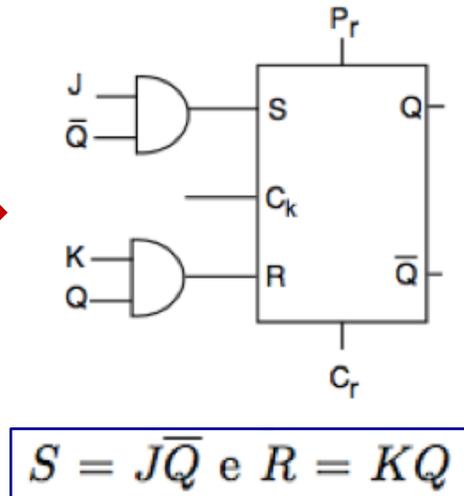
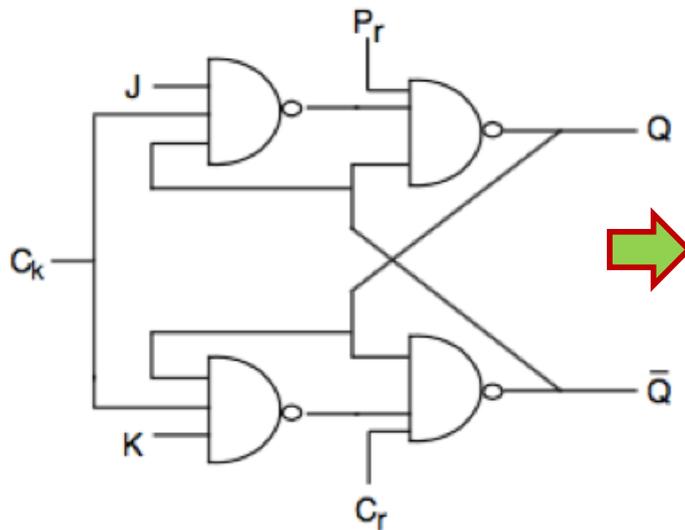


S_n	R_n	Q_{n+1}
0	0	Q_n
1	0	1
0	1	0
1	1	?

- Quando il clock è uguale a zero, l'uscita dei primi due NAND vale sempre 1, di conseguenza le uscite Q e Q-negato non cambiano stato.
- Per farlo funzionare si impostano i due ingressi S e R ai valori voluti, si porta il Clock a 1 e le uscite si portano ai valori riportati nella tabella della verità, dove gli indici n e n+1 indicano la sequenza.
- Lo stato 1-1 rimane ancora indeterminato, perché durante il clock entrambe le uscite si portano a 1, ma quando il clock torna a zero, un'uscita va a zero e l'altra a 1 in maniera imprevedibile, perché dipende da quale delle due è più veloce a commutare.
- Occorre quindi fare un'ulteriore modifica al flip-flop e passare al flip-flop J-K

Flip-Flop J-K

- Utilizzando il circuito seguente non avremo più uscite indeterminate, come si vede dalla tavola della verità:



J_n	K_n	Q_n	\bar{Q}_n	S_n	R_n	Q_{n+1}
0	0	0	1	0	0	Q_n
0	0	1	0	0	0	Q_n
1	0	0	1	1	0	1
1	0	1	0	0	0	$Q_n = 1$
0	1	0	1	0	0	$Q_n = 0$
0	1	1	0	0	1	0
1	1	0	1	1	0	$\bar{Q}_n (= 1)$
1	1	1	0	0	1	$\bar{Q}_n (= 0)$

- L'uscita viene riportata anche all'ingresso del circuito di abilitazione, in modo tale che il valore del Set e Reset dipendono anche dallo stato dell'uscita precedente all'impulso di clock.
- La tabella della verità del flip-flop J-K può essere riassunta nel modo seguente:

J_n	K_n	Q_{n+1}
0	0	Q_n
1	0	1
0	1	0
1	1	\bar{Q}_n

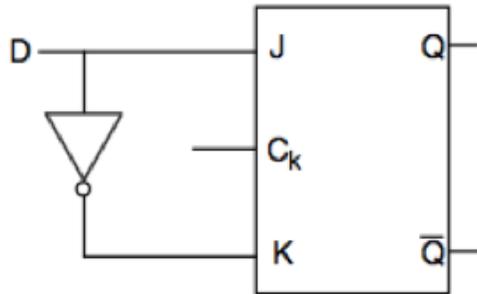
Il flip-flop J-K ha altri due ingressi: Pr (pre-set) e Cr (clear) che servono a fissare lo stato iniziale del circuito indipendentemente dal clock, ovvero in modo asincrono.

C_r	P_r	Q	\bar{Q}
0	1	0	1
1	0	1	0

Dopo aver fissato lo stato desiderato, occorre mantenere Cr e Pr ad 1 per abilitare il flip-flop (Pr=0, Cr=0 non deve essere usato perché porta a situazioni ambigue)

Flip-Flop di tipo D e di tipo T

Flip-Flop di tipo D



- Un flip-flop S-R (oppure J-K) dove gli ingressi sono collegati come in figura realizza il cosiddetto tipo D (delay).
- Ora gli ingressi sono sempre opposti tra loro, quindi la tavola della verità è la seguente:

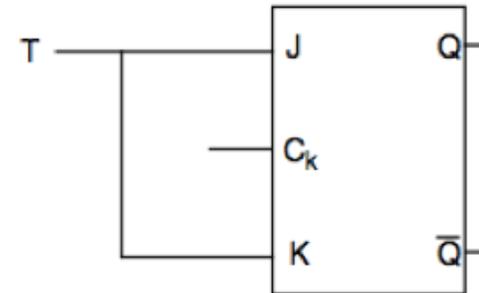
J_n	K_n	Q_{n+1}
1	0	1
0	1	0

➔

D_n	Q_{n+1}
1	1
0	0

- In sostanza questo circuito produce un ritardo di un ciclo di clock (da cui il nome).
- Esso rappresenta una “memoria” ad 1 bit.
- L’informazione viene scritta sul circuito abilitando il clock e permane su di esso.

Flip-Flop di tipo T



- Invece quando gli ingressi di un flip-flop J-K (master-slave) sono connessi direttamente tra loro si ha il cosiddetto tipo T (toggle).
- la sua tavola della verità è la seguente:

J_n	K_n	Q_{n+1}
0	0	Q_n
1	1	$\overline{Q_n}$

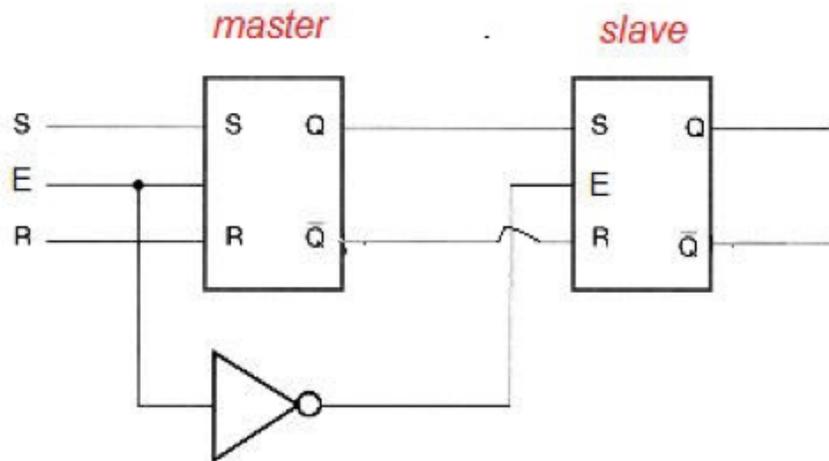
➔

T_n	Q_{n+1}
1	$\overline{Q_n}$
0	Q_n

- Quando $T=0$ il flip-flop non fa nulla;
- Quando $T=1$ l’uscita cambia da Q a Q-negato ad ogni ciclo di clock

Flip-flop S-R master-slave

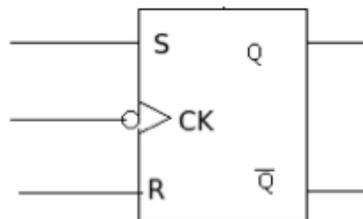
- È possibile realizzare un flip-flop che viene abilitato solo sul fronte di salita/discesa del clock utilizzando la configurazione detta master-slave. Nella figura è mostrato un flip-flop S-R master-slave:



Si hanno due latch Set-Reset collegati in cascata. Le uscite del primo latch (*master*) sono collegate alle entrate del secondo latch (*slave*).

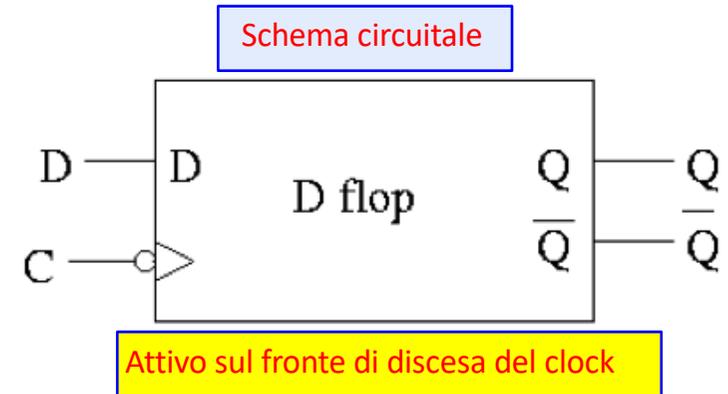
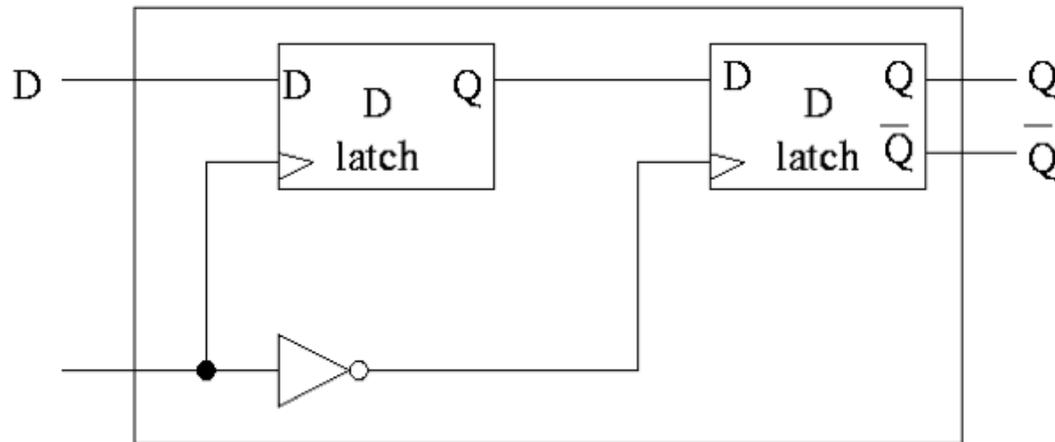
Il segnale di abilitazione (clock) arriva direttamente al master, mentre giunge allo slave dopo essere stato invertito dalla porta NOT. Ciò significa che il master e lo slave non sono mai contemporaneamente abilitati.

- Durante l'impulso di clock il secondo stadio è inibito e le uscite non possono commutare. Invece quando cessa l'impulso di clock, il primo stadio è inibito e lo stato delle uscite del master si trasferiscono sullo slave.
- In sostanza il circuito realizza (in maniera diversa rispetto al flip-flop edge triggered) un flip-flop Set-Reset pilotato sul fronte di discesa del clock.

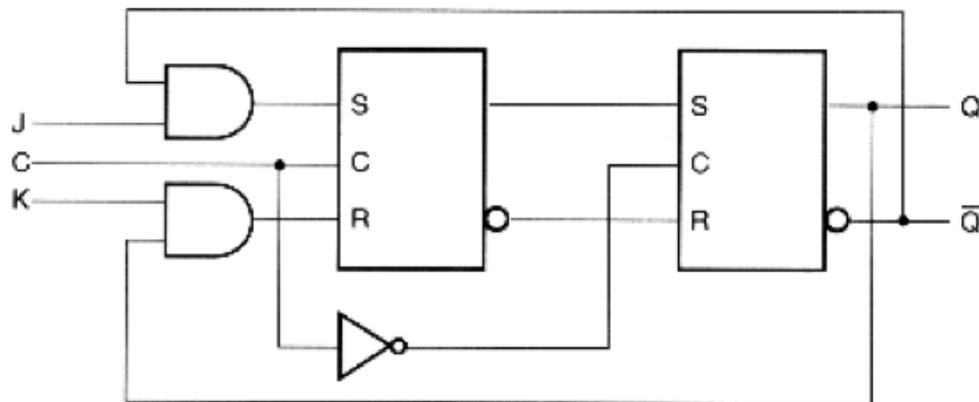


Flip-flop D and J-K master-slave

- Si possono anche realizzare flip-flop di tipo D and J-K master-slave:



- J-K master-slave:

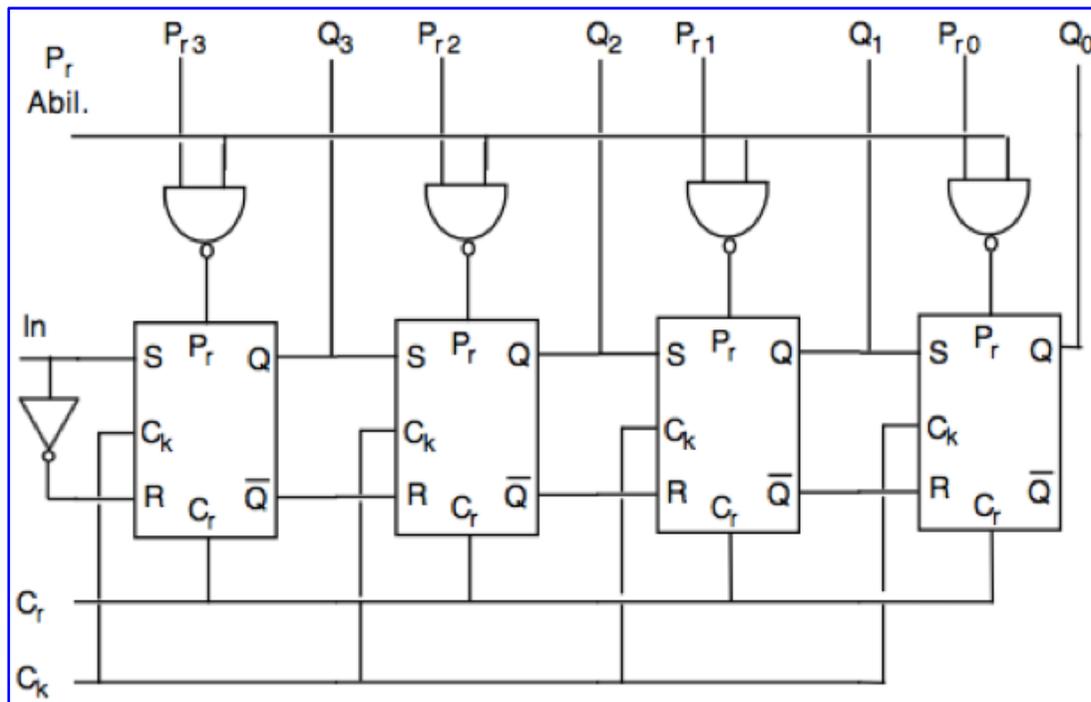


J	K	Next State of Q
0	0	Q
0	1	0
1	0	1
1	1	\bar{Q}

Shift register

Shift register (registro a scorrimento)

- Lo shift register è formato da n flip-flop di tipo S-R master-slave; il primo flip-flop è di tipo D. Questo circuito consente di memorizzare n bit, che possono essere caricati sia in modo seriale attraverso l'ingresso di sinistra, sia in parallelo attraverso gli ingressi di preset.
- Il contenuto può essere riletto sia in parallelo che in serie.



Azzeramento:

si pone $C_r = 0$, $P_{r\text{Abil}} = 0$

1) Caricamento parallelo:

$C_r = 1$, $P_{r\text{Abil}} = 1$, $P_{rj} = 1/0$:

i vari flip-flop vengono caricati con il contenuto presentato agli ingressi P_{rj}

2) Lettura parallela:

si effettua dalle uscite Q_i

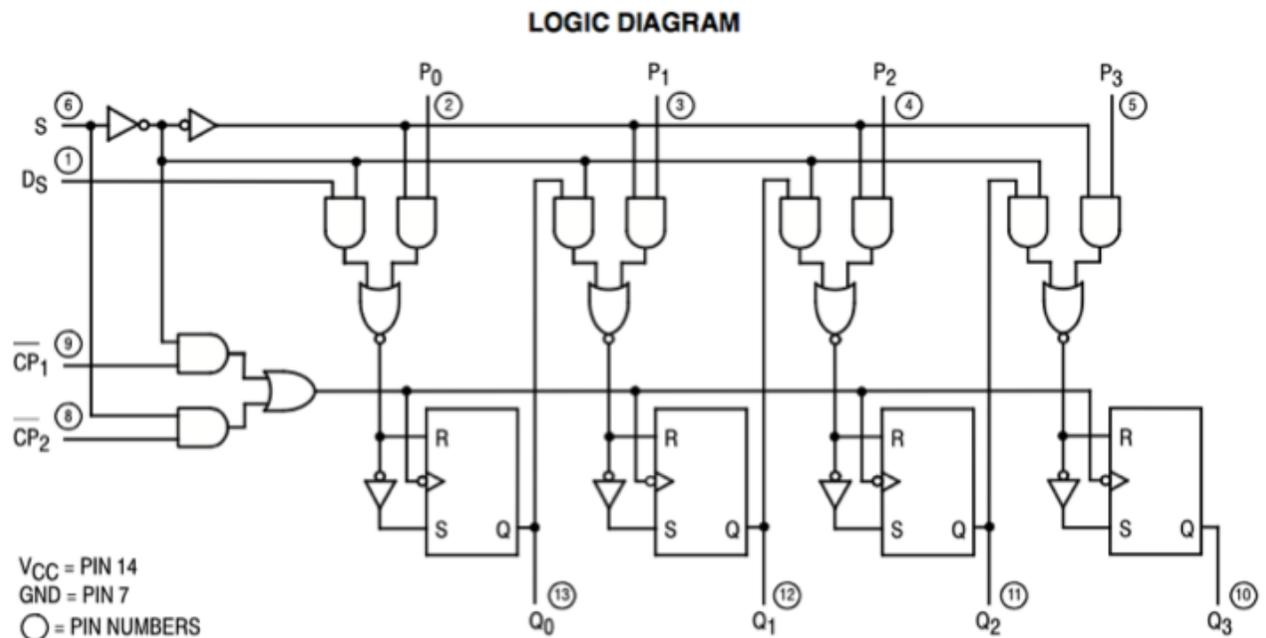
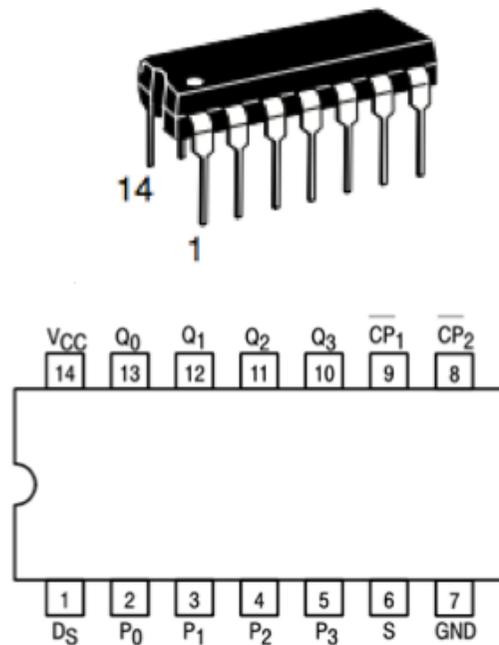
C_r	P_r	Q	\bar{Q}
0	1	0	1
1	0	1	0

3) Caricamento seriale: si presenta il primo bit da caricare sull'ingresso seriale; fornendo un impulso di clock l'informazione viene caricata sul flip-flop F4; successivamente si presenta il secondo bit e si fornisce un secondo impulso di clock: il primo bit passa al flip-flop F3 e in F4 entra il secondo bit; continuando si possono caricare tutti i flip-flop;

4) Lettura seriale: fornendo ulteriori impulsi di clock i bit memorizzati si presentano successivamente sull'uscita Q_0

Shift register

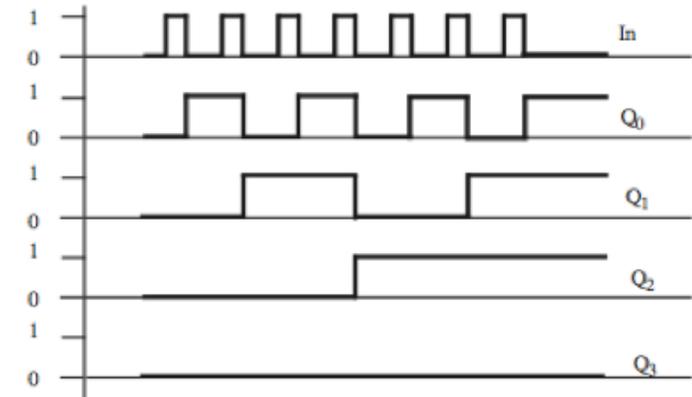
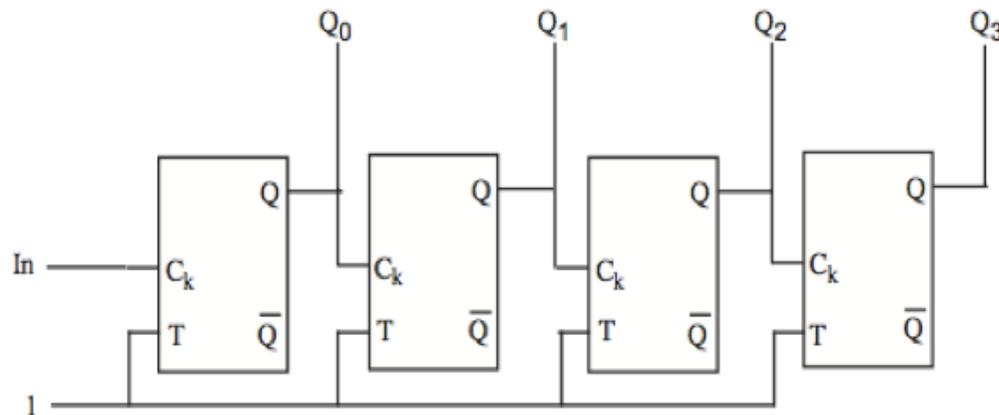
- Si possono costruire registri in grado di effettuare scorrimenti verso destra o verso sinistra. Essi possono essere utilizzati per effettuare moltiplicazioni e divisioni, che richiedono questo tipo di manipolazione.
- Lo shift register può essere usato come memoria, come convertitore serie-parallelo, parallelo-serie; ritardo digitale, etc., e costituisce quindi un circuito molto importante in una grande varietà di applicazioni.
- L'integrato **74LS95** è un esempio di shift register a 4 bit, capace di compiere sia scorrimenti verso destra che verso sinistra, con ingressi e uscite sia seriali che paralleli.



Contatori binari

Contatore asincrono

- Si può costruire un contatore binario utilizzando dei flip-flop di tipo T master-slave.
- Tutti gli ingressi vengono posti a 1; gli impulsi da contare entrano nel clock del primo stadio, mentre ogni uscita fornisce il clock allo stadio successivo. Poiché i flip-flop sono master-slave, ogni uscita commuta sul fronte di discesa del clock ricevuto all'ingresso, come si vede in figura, dove abbiamo supposto che inizialmente tutti i flip-flop siano a zero:

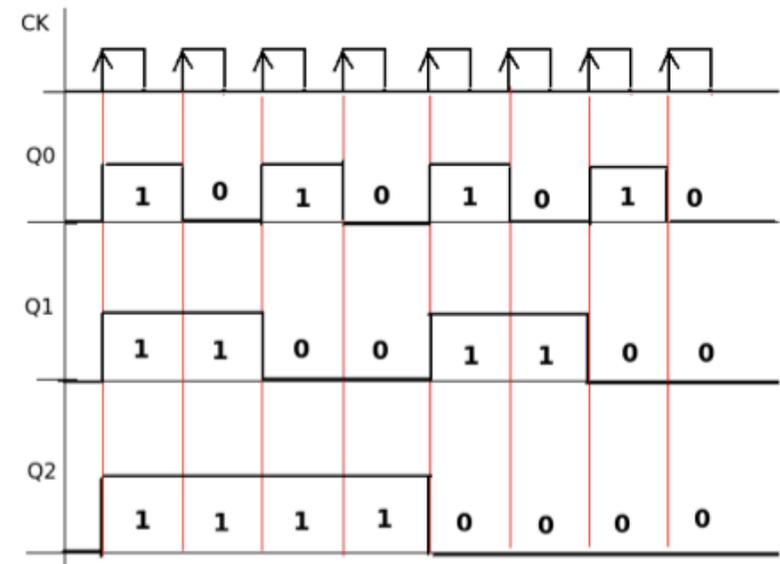
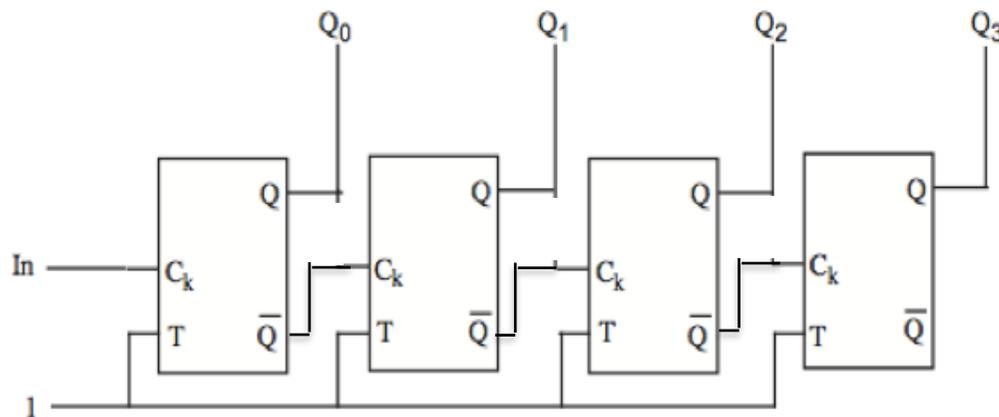


- L'insieme delle uscite costituisce un numero a 4 bit (Q_0 è il bit meno significativo).
- Un contatore a n stadi conta fino a 2^n e poi ricomincia da zero.**
- Si può anche notare come le frequenze di commutazione vengano dimezzate man mano, quindi questo circuito può anche essere usato come un divisore di frequenza, per potenze di due.

Numero di impulsi	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	1
...				
15	1	1	1	1
16	0	0	0	0

Contatore avanti-indietro

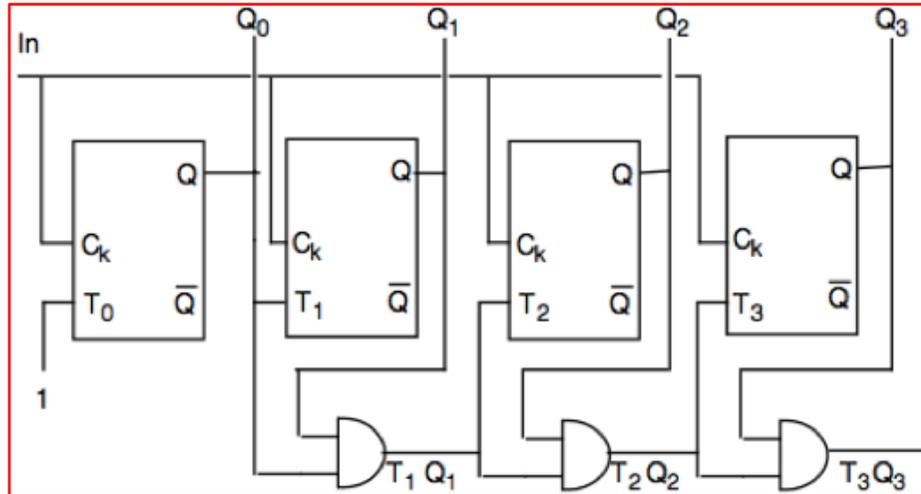
- Un contatore che possa effettuare un conteggio in entrambe le direzioni è detto avanti-indietro.
- Per realizzare un contatore che conti all'indietro occorre che l'ingresso di clock di ogni stadio sia collegato all'uscita \bar{Q} dello stadio precedente, mentre il primo stadio resta invariato.
- In questo modo il primo stadio commuta ad ogni impulso, mentre gli stadi successivi commuteranno quando l'uscita \bar{Q} proveniente dallo stadio precedente passa da 1 a 0;
- In altri termini è come se il flip-flop commutasse sul fronte di salita del clock e non più sul fronte di discesa come nel caso del contatore in avanti:



- Nella tabella vi è l'esempio di un contatore a tre bit.
- Partendo da uno stato 000, al primo clock tutti i flip-flop commutano e si portano nello stato 111;
- al secondo clock commuta solo il primo flip-flop (Q_0) portandosi da 1 a 0, ma questo cambio non fa scattare il secondo flip-flop (sensibile ora al fronte di salita) e così via.

Contatore sincrono

- In un contatore sincrono gli impulsi da contare vengono inviati simultaneamente a tutti i flip-flop:



N.B. Un Flip-Flop di tipo-T commuta solo se $T=1$

Q_0	commuta ad ogni impulso;	$T_0 = 1$
Q_1	solo se $Q_0 = 1$;	$T_1 = Q_0$
Q_2	solo se $Q_0 = Q_1 = 1$;	$T_2 = Q_0 Q_1$
Q_3	solo se $Q_0 = Q_1 = Q_2 = 1$;	$T_3 = Q_0 Q_1 Q_2$

- Il tempo di commutazione non è più dipendente dal numero di stadi ed è molto più breve del caso precedente, infatti esso è uguale a:

$$T = T_F + (n - 2)T_G$$

T_F : ritardo di propagazione di un Flip-Flop
 T_G : ritardo di propagazione di una porta AND
 dove $T_G \ll T_F$

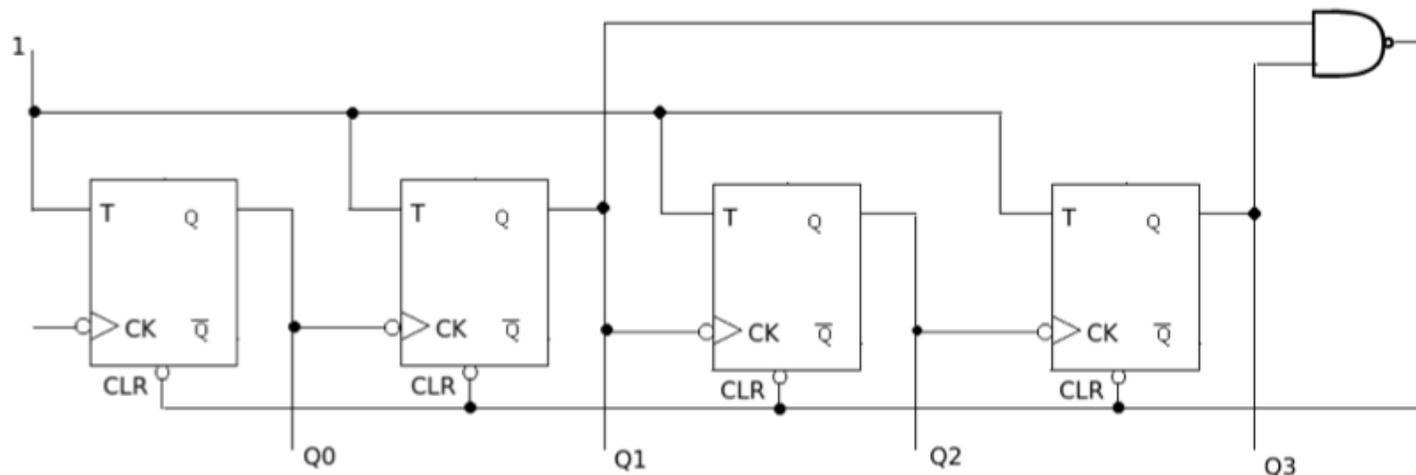
- È possibile migliorare ulteriormente la velocità utilizzando una tecnica di riporto in parallelo delle varie uscite Q invece della serie di AND utilizzati in questo schema. Questo richiederebbe l'uso di AND a molti ingressi ed una logica un po' più complessa, ma consentirebbe di arrivare ad un tempo complessivo di commutazione pari a:

$$T = T_F + T_G$$

Contatore modulo 10

Contatore modulo 10

- Usando uno degli schemi che abbiamo appena visto è possibile realizzare un contatore con modulo $M = 2^n$, dove n è il numero di flip-flop utilizzati.
- Vediamo ora come realizzare un contatore con modulo qualsiasi, ad esempio in modulo 10, il quale conta da 0000 (zero in base 10) fino a 1001 (nove in base 10) e poi si riavvia.



La porta NAND manda un impulso di clear (CLR=0) quando si raggiunge la combinazione 1010 (=10), cioè quando $Q_3=1$ e $Q_1=1$

- All'arrivo dell'undicesimo fronte di discesa del clock, i flip-flop commutano alla configurazione 1010. Tale configurazione provoca però immediatamente l'azzeramento di tutti i flip-flop (a causa della porta NAND collegata all'ingresso di Clear).
- La configurazione 1010 compare in uscita per un tempo bravissimo (idealmente nullo), appena sufficiente per azzerare i flip-flop e far ripartire la sequenza da 0000

Clock Pulse	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Esercizio sul contatore

Esercizio sul contatore (129)

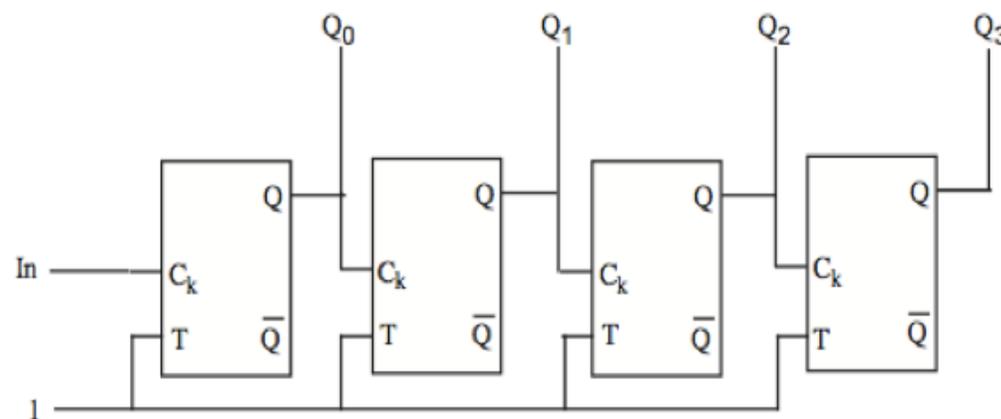
Esercizio 129

Il numero di persone, n , presenti in una stanza, viene continuamente sorvegliato, grazie a cellule fotoelettriche poste all'ingresso e ad un contatore up/down a 8 bit. Se una persona entra il contenuto del contatore aumenta; se una persona esce il contenuto del contatore diminuisce. Utilizzando le uscite del contatore costruire una variabile logica A ed una variabile logica B tali che:

$$A = 1 \text{ se } n \geq 192$$

$$192 = 128 + 64 = 1100\ 0000$$

$$B = 1 \text{ se } n < 192$$



Soluzione Esercizio Contatore

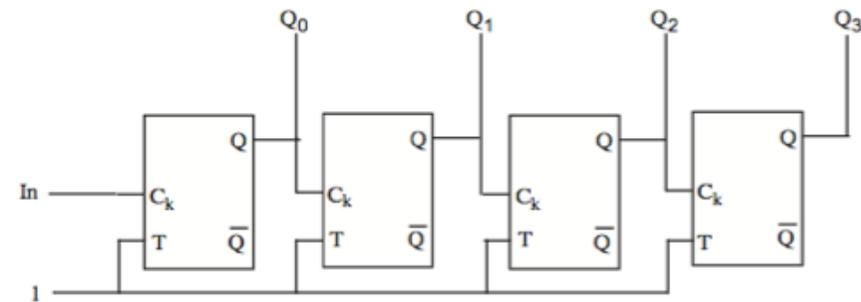
Esercizio 129

Il numero di persone, n , presenti in una stanza, viene continuamente sorvegliato, grazie a cellule fotoelettriche poste all'ingresso e ad un contatore up/down a 8 bit. Se una persona entra il contenuto del contatore aumenta; se una persona esce il contenuto del contatore diminuisce. Utilizzando le uscite del contatore costruire una variabile logica A ed una variabile logica B tali che:

$$A = 1 \text{ se } n \geq 192$$

$$B = 1 \text{ se } n < 192$$

$$192 = 128 + 64 = 1100\ 0000$$



Esercizio 129

Chiamiamo D_0, \dots, D_7 le 8 uscite del contatore (D_0 è la meno significativa). Se il contenuto del contatore è maggiore o uguale a 192, le uscite D_7 e D_6 sono sicuramente ad 1. Quindi la variabile logica A è semplicemente data da

$$A = D_6 D_7$$

Ovviamente si ha $B = \bar{A}$.

Convertitore Digitale-Analogico (DAC)

Conversione Digitale-Analogica (DAC)

- Il mondo dell'elettronica digitale non è (completamente) separato da quello dell'informazione analogica ed esiste spesso l'esigenza di passare dall'uno all'altro.
- Questo avviene ogni qualvolta il valore di una grandezza analogica, corrente o tensione, deve essere controllato numericamente.
- La conversione digitale-analogica è un'operazione attraverso la quale si costruisce una tensione V (o una corrente I) proporzionale ad un numero binario A dato. Si ha quindi:

$$V = K \sum_{0}^{n-1} 2^i a_i$$

a_i sono i bit (0 o 1) che costituiscono il numero A
 K è una costante di proporzionalità (ha la dimensione di V)

Se tutti gli a_i sono 1 si ha:



$$V_{max} = K(2^n - 1)$$

V_{max} è il valore massimo esprimibile con n bit

- In generale è più comodo scrivere:

$$V = \frac{V_{max}}{2^n - 1} \sum_{0}^{n-1} 2^i a_i$$

- I dispositivi che realizzano questa conversione si chiamano (in inglese) Digital to Analog Converter, brevemente chiamati DAC;
- La formula qui sopra riportata rappresenta il comportamento di un dispositivo ideale. Vediamo ora quali sono i parametri che qualificano le prestazioni di un DAC reale.

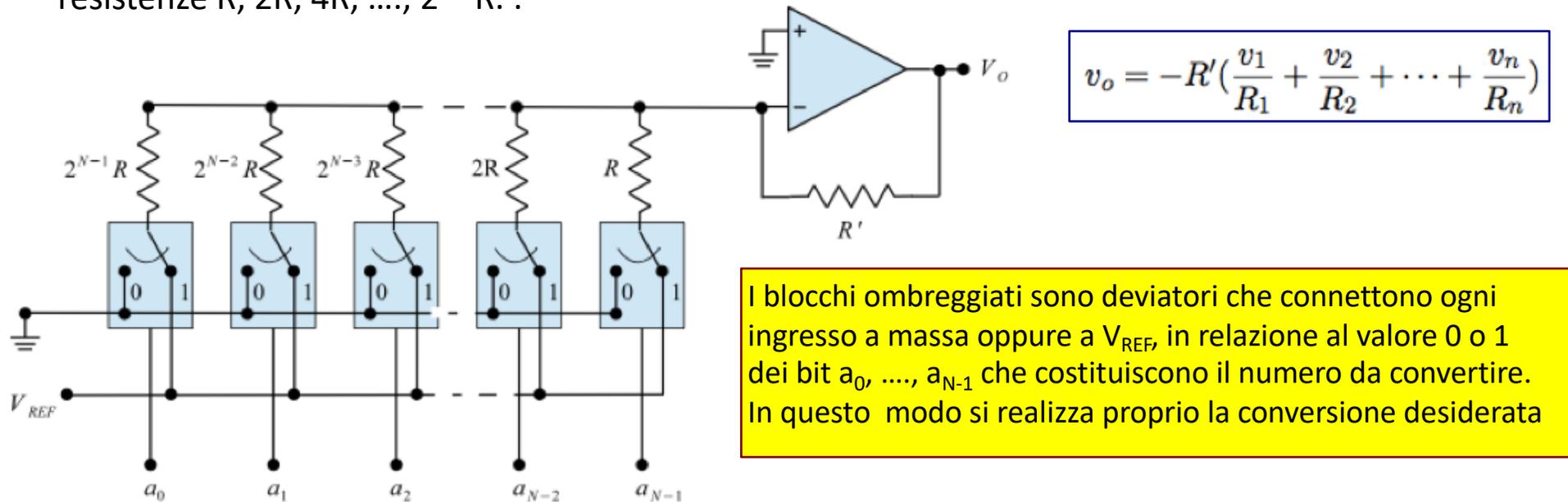
Caratteristiche di un DAC reale

- ❑ In un DAC reale occorre tener conto dei seguenti parametri:
 - ❑ **Risoluzione:** la minima variazione possibile della tensione di uscita. Rappresenta anche il valore della tensione di uscita associata al bit meno significativo;
 - ❑ **Offset:** l'uscita potrebbe non essere 0 quando il numero in ingresso è uguale a zero;
 - ❑ **Errore di guadagno:** la differenza tra il valore massimo teorico e quello reale della tensione in uscita una volta annullato l'offset;
 - ❑ **Errore di linearità:** la differenza tra la caratteristica di trasferimento ideale e quella reale, una volta annullati gli errori precedenti;
 - ❑ **Tempo di assestamento:** tempo impiegato dall'uscita ad assestarsi (entro una certa percentuale del valore finale) in corrispondenza di una variazione della parola d'ingresso.

DAC a pesiera

DAC a pesiera

- Il primo e più semplice esempio di DAC è quello in figura. Si tratta in sostanza di un sommatore analogico, in cui le tensioni da sommare sono pesate secondo le potenze di due, grazie alle resistenze $R, 2R, 4R, \dots, 2^{N-1}R$.



- L'uscita del convertitore non varia in modo continuo, bensì "a gradini", passando dalla tensione 0 alla tensione V_{max} ed è data da:

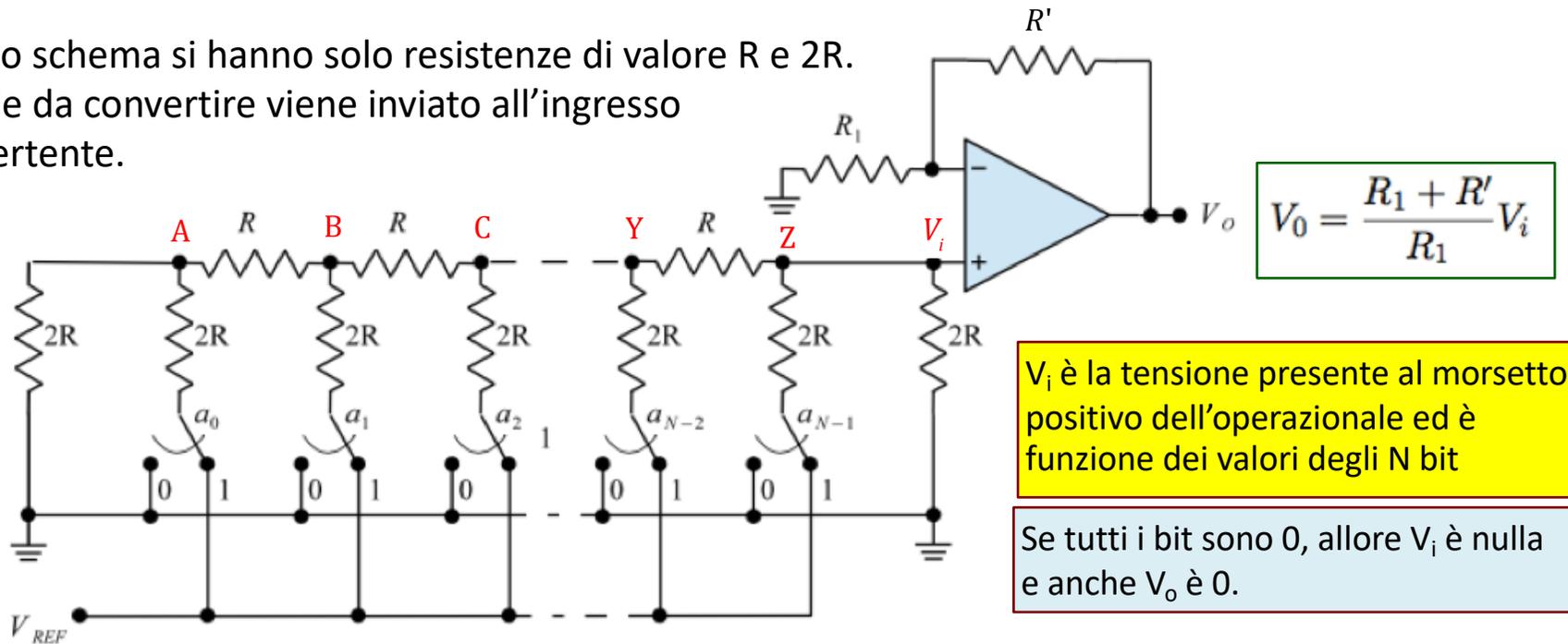
$$V_0 = -\frac{R'}{R} V_{REF} \left(a_{N-1} + \frac{a_{N-2}}{2} + \dots + \frac{a_0}{2^{N-1}} \right)$$

- La linearità di questo dispositivo è legata alla precisione con cui i valori dei resistori sono scalati; si comprende come questo renda notevolmente critico realizzare in pratica un dispositivo in grado di convertire correttamente numeri con molti bit.

DAC con rete R-2R

DAC con rete R-2R

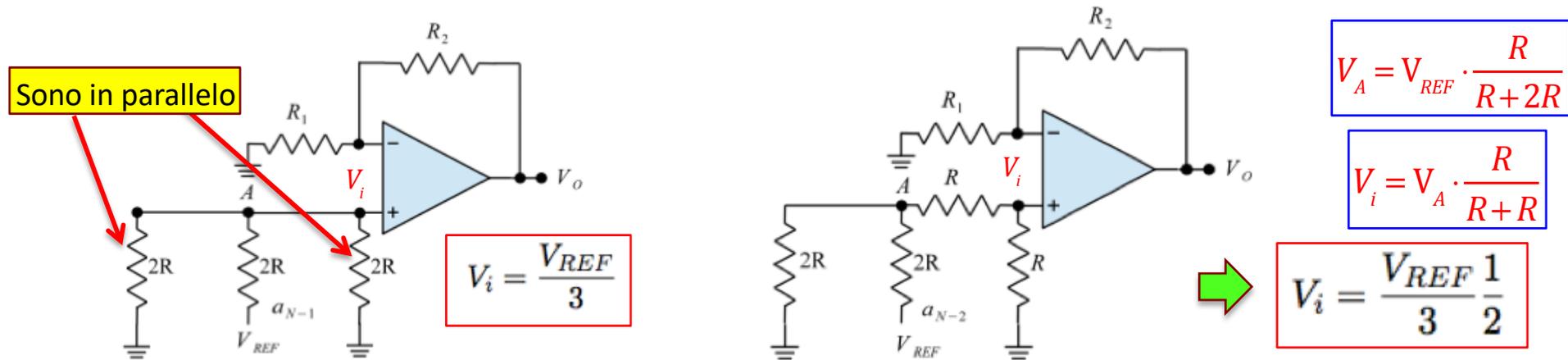
- In questo schema si hanno solo resistenze di valore R e 2R. Il segnale da convertire viene inviato all'ingresso non invertente.



- Se uno solo dei bit è pari ad 1 il corrispondente nodo si porta alla tensione $V_{REF}/3$, perché qualunque sia la sua posizione, il nodo vede alla sua sinistra e alla sua destra una resistenza $2R$ verso massa.
- Infatti supponiamo ad esempio che solo il bit a_0 sia ad uno mentre tutti gli altri a zero; il nodo A vede alla sua sinistra una resistenza $2R$ verso massa, mentre a destra vede una resistenza R in serie alla resistenza equivalente tra il nodo B e massa che è ancora uguale a R . Per dimostrare quest'ultima cosa partiamo dal nodo Z e ricordiamo che a_{N-1} è collegato a massa; si vede che abbiamo due resistenze $2R$ in parallelo, che equivalgono a una resistenza R . Spostiamoci ora sul nodo Y, alla sua destra abbiamo una resistenza R in serie alla resistenza che c'è tra il nodo Z e massa, che abbiamo detto essere R , quindi fa una resistenza $2R$. E così via fino al nodo B, come volevasi dimostrare.
- Questo ragionamento si può applicare per ciascun nodo.

DAC con rete R-2R

- ❑ Vediamo ora quanto vale la tensione V_i quando uno solo dei bit d'ingresso vale 1.
- ❑ Nella figura di sinistra vediamo cosa succede quando il bit $N-1$ è uguale a 1, mentre nella figura di destra è illustrato il caso in cui sia uguale a 1 il bit $N-2$.
- ❑ Ricordiamo che la tensione del nodo che ha il bit a 1 vale $V_{REF}/3$ (qualunque sia il bit settato a 1).



- ❑ Se prendiamo il caso del bit $N-3$, vediamo che V_i è ancora divisa per 2 e così via: $V = \frac{V_{REF}}{3} \cdot \frac{1}{4}$
- ❑ Iterando questo ragionamento si comprende che il peso di ogni bit è via via ridotto di un fattore 2 ogni qualvolta ci si sposta verso sinistra;
- ❑ Il comportamento non cambia nel caso generale grazie al principio di sovrapposizione, per cui:

$$V = \frac{V_{REF}}{3} \left(\frac{a_{N-1}}{2^0} + \frac{a_{N-2}}{2^1} + \dots + \frac{a_0}{2^{N-1}} \right)$$

- ❑ Questa soluzione è più facile da implementare perché abbiamo bisogno solo di resistenze R e $2R$

Esercizio sul DAC

Esercizio DAC R-2R

Esercizio 127

Progettare un convertitore digitale analogico (DAC) a 8 bit, con rete R-2R, che fornisca una uscita di tensione negativa compresa tra 0 e -15 V. Questo significa che quando gli 8 ingressi sono tutti 0 l'uscita è 0 V, mentre quando tutti gli ingressi sono 1, l'uscita è -15 V. Il numero a 8 bit da convertire è fornito da un normale registro a 8 bit, con uscite TTL in logica positiva (0 – 5 V).

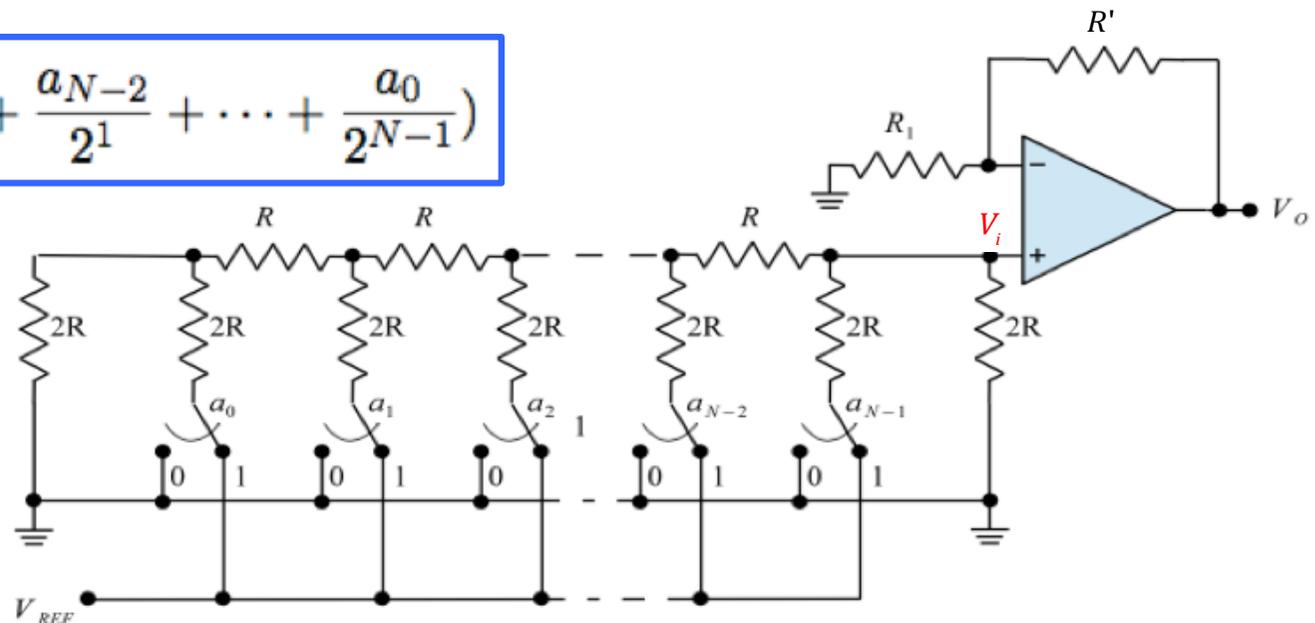
Nota: è naturalmente possibile utilizzare più operazioni, se necessario.

$$\frac{255}{128} = \frac{1}{1} + \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{128}$$

$$V_i(FF) = \frac{5}{3} \cdot \frac{255}{128} = \frac{425}{128} = 3.32 \text{ V}$$

$$V = \frac{V_{REF}}{3} \left(\frac{a_{N-1}}{2^0} + \frac{a_{N-2}}{2^1} + \dots + \frac{a_0}{2^{N-1}} \right)$$

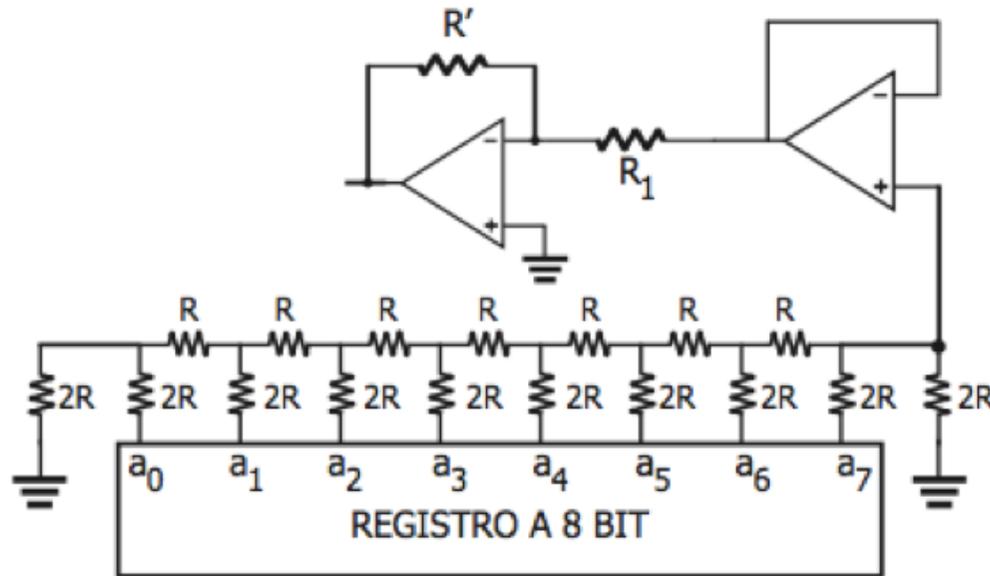
$$V_{REF} = 5 \text{ V}$$



Soluzione Esercizio DAC R-2R

Esercizio 127

$$V = \frac{V_{REF}}{3} \left(\frac{a_{N-1}}{2^0} + \frac{a_{N-2}}{2^1} + \dots + \frac{a_0}{2^{N-1}} \right)$$



$$V_{REF} = 5 \text{ V}$$

$$V(FF) = \frac{5}{3} \cdot \frac{255}{128} = \frac{425}{128} = 3.32 \text{ V}$$

$$\frac{425}{128} \times \frac{R'}{R} = 15$$



$$\frac{R'}{R} = \frac{128}{425} \times 15 = 4.518$$

Questo esercizio può naturalmente essere risolto in vari modi. In figura è mostrata una possibile soluzione. L'uscita del primo operazionale è una tensione positiva, variabile tra 0V (per ingressi tutti a zero) e circa +3.3V (per ingressi tutti ad uno). Il secondo operazionale inverte il segno e porta il range di tensione al livello voluto, grazie ad una opportuna scelta di R_1 e R' .

Convertitore Analogico-Digitale (ADC)

Conversione Analogico-Digitale (ADC)

La conversione analogico-digitale e' un processo attraverso il quale si vuole convertire una grandezza elettrica continua (tipicamente una tensione) in un numero (binario) ad essa proporzionale. Nel mondo di oggi questa operazione e' fondamentale in innumerevoli campi.

I dispositivi che realizzano questa funzione sono comunemente chiamati ADC, *Analog to Digital Converter*, e possono essere realizzati in vari modi. I principali parametri che caratterizzano un convertitore analogico digitale sono:

- intervallo di conversione: l'intervallo di tensione che puo' essere convertito in un numero;
- risoluzione: indica il numero dei valori discreti che esso puo' produrre. Ad esempio, un convertitore che codifica l'ingresso in 256 livelli discreti ha una risoluzione di 8 bit. Puo' anche essere espressa in termini di tensione, ovvero come l'intervallo di tensione che separa due numeri adiacenti;
- errore di linearita': rappresenta la discrepanza tra risposta ideale (proporzionale all'ingresso) e quella reale ottenuta;
- errore di quantizzazione: e' dovuto intrinsecamente alla risoluzione del dispositivo;
- tempo di conversione: e' il tempo che il dispositivo impiega per fornire la risposta digitale alla sua uscita.



SAPIENZA
UNIVERSITÀ DI ROMA

Fine del capitolo 8