Piero Malcovati

Dipartimento di Ingegneria Elettrica
Università di Pavia

Corso di Misure Elettriche

Capitolo 9

Conversione Analogico/Digitale

http://ims.unipv.it/~piero/Misure.html

9.1. Generalità

In un ***convertitore analogico/digitale***, il problema di fondo consiste nello stabilire la ***corrispondenza*** tra la ***grandezza analogica*** di ingresso (che ha andamento continuo nel tempo) e la ***grandezza digitale*** d'uscita (che, in quanto numerica, ha andamento discreto, per gradini)

La ***corrispondenza*** deve essere univoca e il grado di dettaglio raggiunto nella ***discretizzazione*** del segnale da esaminare determina l'***incertezza*** della ***conversione***.

Questa ***incertezza*** non deve essere confusa con quella del valore riguardante il ***misurando*** della quale è uno solo dei ***termini***

Un altro ***problema*** è la scelta del ***codice*** mediante il quale esprimere ***numericamente*** i dati che, all'atto pratico, ricade sempre su quelli ***binari***, cioè di quei metodi di numerazione, che impiegano due stati (lo stato 0 e lo stato 1)

Si osserva che la ***codifica*** avviene praticamente in assenza di ***errori***, così come la eventuale ***trasmissione*** a distanza del segnale

Negli ***strumenti*** che ricorrono a queste ***tecniche***, l'***incertezza della misura*** è fondamentalmente legata alla fase di ***conversione A/D***

9.2. Codici Binari

È a tutti noto che il ***sistema decimale*** esprime i ***numeri*** mediante ***dieci simboli*** (le cifre da 0 a 9), ordinati secondo le ***potenze*** intere decrescenti di 10

Ad ***esempio***:

$$683.21 = 6\times10^{2} + 8\times10^{1} + 3\times10^{0} + 2\times10^{-1} + 1\times10^{-2}$$

Il ***sistema binario*** esprime invece i ***numeri*** mediante ***due simboli*** (i digit 0 e 1), ordinati secondo ***potenze*** intere decrescenti di 2

Ad ***esempio***, per il numero 100 in codice binario si ha

$$100 = 1100100 = 1\times2^{6} + 1\times2^{5} + 0\times2^{4} + 0\times2^{3} + \\ + 1\times2^{2} + 0\times2^{1} + 0\times2^{0}$$

mentre lo stesso numero in codice decimale risulta

$$100 = 1\times10^{2} + 0\times10^{1} + 0\times10^{0}$$

È evidente la differente ***lunghezza*** della ***parola*** che rappresenta il ***dato*** nei due ***sistemi***, ma nel secondo caso i ***digit*** sono soltanto ***due***, contro i ***dieci*** del primo

Nel ***sistema binario*** i ***digit*** vengono detti ***bit*** (binary digits)

Con il ***sistema binario*** si possono formare ***codici*** diversi, con opportune combinazioni di ***gruppi*** di ***bit***

Uno dei più semplici è il ***NBCD*** (Natural Binary-Coded Decimal) che rappresenta ciascuno dei ***dieci digit*** del ***codice decimale*** con una diversa sequenza di ***quattro bit***

N	***Codice NBCD***	***N***	***Codice NBCD***
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

Rimanendo nel campo dei ***codici BCD*** (Binary-Coded Decimal), nessuno impedisce di stabilire altre corrispondenze fra ***gruppi*** di ***quattro bit*** e ***cifre decimali***, che non siano quella naturale e ricavabile direttamente dalla espressione generale

$$N = \sum_{i=1}^{n} d_i 2^{i-1}$$

dove d_i può assumere solo i valori 0 ed 1

Esistono infatti altri ***sistemi*** che usano come dati i digit 0 ed 1: ad esempio il ***sistema ottale*** che ordina i digit secondo potenze intere decrescenti di 8, usato nella tecnica dei calcolatori

Se si esaminano le ***quartine*** del ***codice NBCD***, si nota che in alcuni passaggi da un ***decimale*** al ***successivo*** deve essere modificato lo stato di più di un bit

Ad esempio, nel passaggio da 7 a 8 si deve cambiare lo stato di tutti i bit

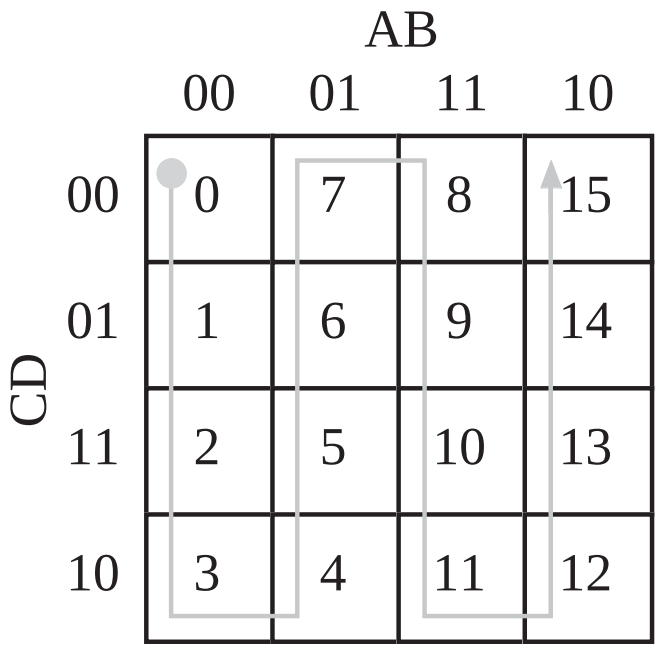
Questa non è un ***problema arbitrario***, poiché in fase di ***conversione analogico/digitale*** è assai più probabile l'introduzione di ***errori*** (dovuti a simultaneità di operazioni o a imperfezione dei circuiti) qualora il passaggio da una ***cifra*** alla ***successiva*** comporti il cambiamento di più di un bit della quartina

Sono perciò stati individuati altri ***codici***, che realizzano quanto desiderato, come il ***codice XS-3 Gray***

N	***Codice XS-3 Gray*** ***ABCD***
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100

N	***Codice XS-3 Gray*** ***ABCD***
9	1101
10	1111
11	1110
12	1010

Una regola mnemonica utile per scrivere le ***quartine*** del ***codice Gray*** è data dalla gabbia di Garnaugh



Qualora si debba rappresentare in ***codice*** un'informazione ***alfanumerica***, cioè contenente non soltanto ***cifre*** ma anche ***lettere*** di alfabeto o ***simboli***, e non sia più sufficiente una ***quartina di bit*** per codificare univocamente ogni dato, si ricorre a codici che utilizzano gruppi di ***otto bit*** (detti ***byte***)

Valga per tutti l'esempio del ***codice ASCII*** (American Standard Code for Information Interchange) nel quale con soli 7 bit si esaurisce il messaggio, lasciando disponibile l'ottavo per l'introduzione di un ***bit di controllo***, assai utile nella fase di eventuale trasmissione a distanza (***controllo di parità***)

9.3. Campionamento e Quantizzazione

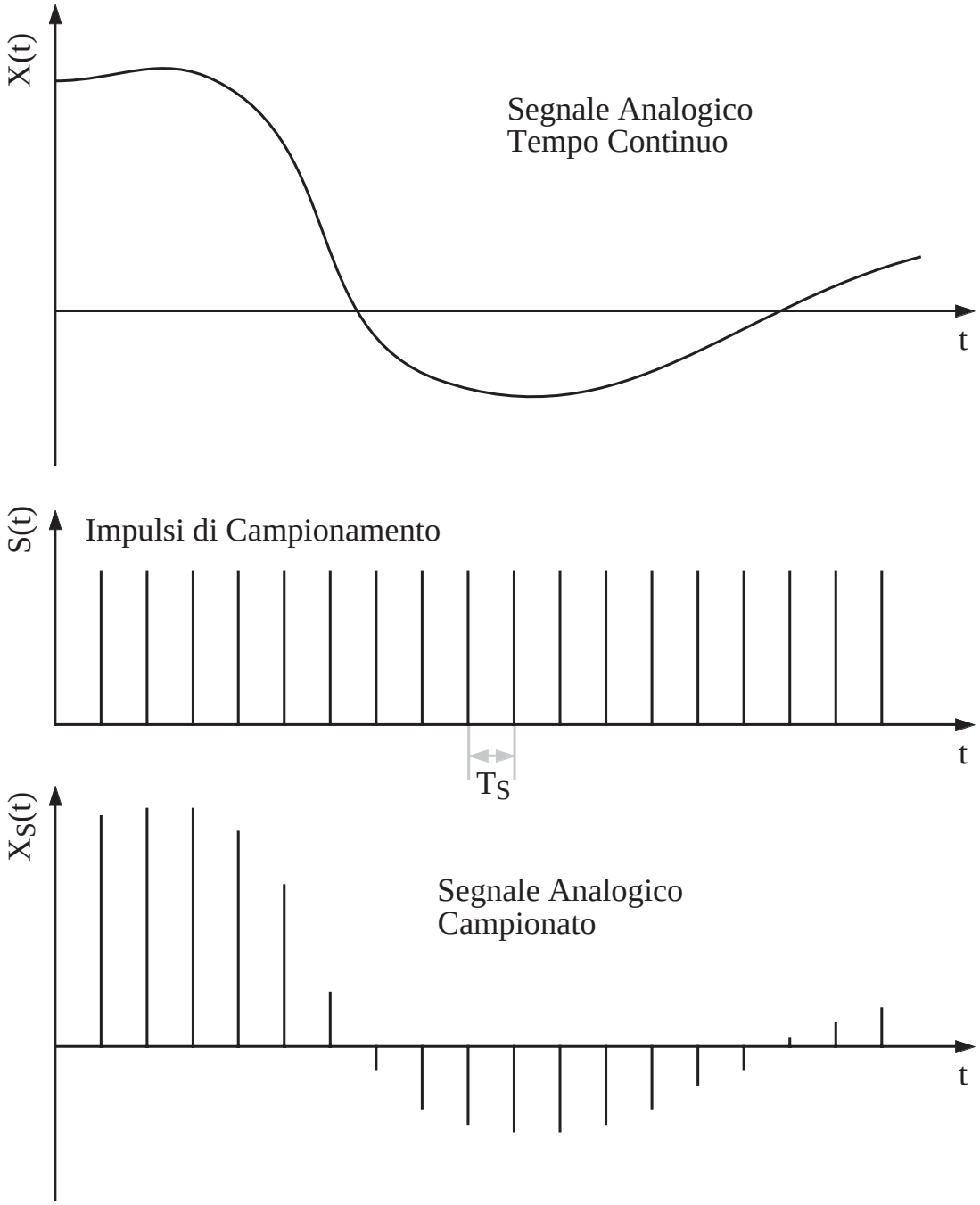
La ***digitalizzazione*** di un ***segnale analogico tempo-continuo*** coinvolge due processi di ***discretizzazione***: un processo di ***discretizzazione*** nel dominio del ***tempo*** (***campionamento***) e un processo di ***discretizzazione*** in ***ampiezza*** (***quantizzazione***)

Una delle ***fasi*** più critiche della ***trasformazione*** di un segnale ***analogico*** in uno ***digitale*** è quella del ***campionamento*** che consiste nel convertire una grandezza variabile nel ***tempo*** in modo ***continuo*** in una ***discreta*** che rappresenti la prima in modo univoco e con la introduzione di errori trascurabili

Un ***segnale analogico*** che varia in funzione del ***tempo*** viene ***moltiplicato*** per un segnale di tipo ***impulsivo*** di ampiezza costante (ad esempio, unitaria) che si ripete con ***frequenza*** preordinata e pure costante, detta ***frequenza di campionamento***

Il ***periodo*** T_S corrispondente alla citata ***frequenza*** è detto ***intervallo di campionamento***

Il risultato che si ottiene dal prodotto è rappresentato ovviamente da una ***serie di impulsi*** modulati in ampiezza che rappresentano in una certa misura il ***segnale analogico*** assegnato



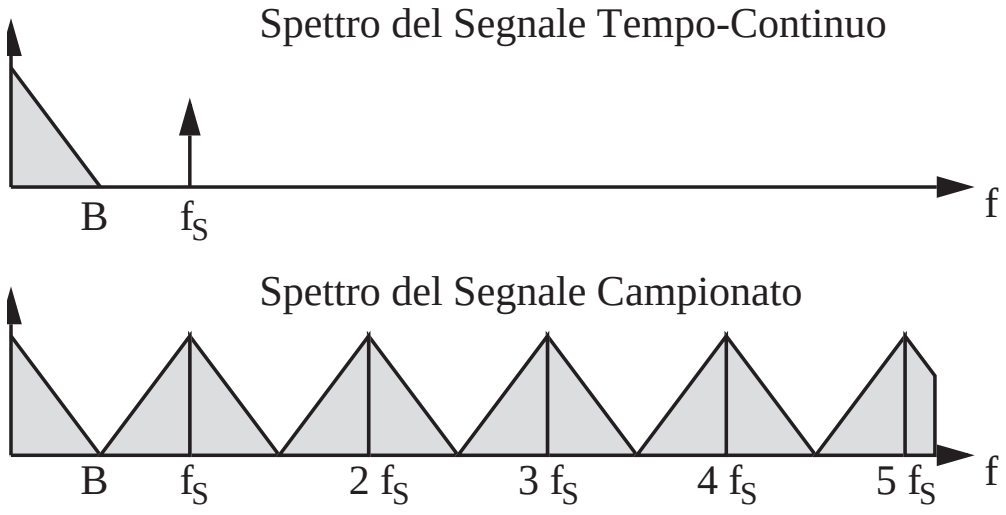
Un altro ***parametro*** tipico del ***campionamento*** è la ***finestra di osservazione*** ovvero il tempo totale del ***campionamento***

Affinché il ***segnale campionato*** sia una rappresentazione scevra da grossolane ***incertezze*** è necessario che siano rispettate alcune ***regole*** fondamentali

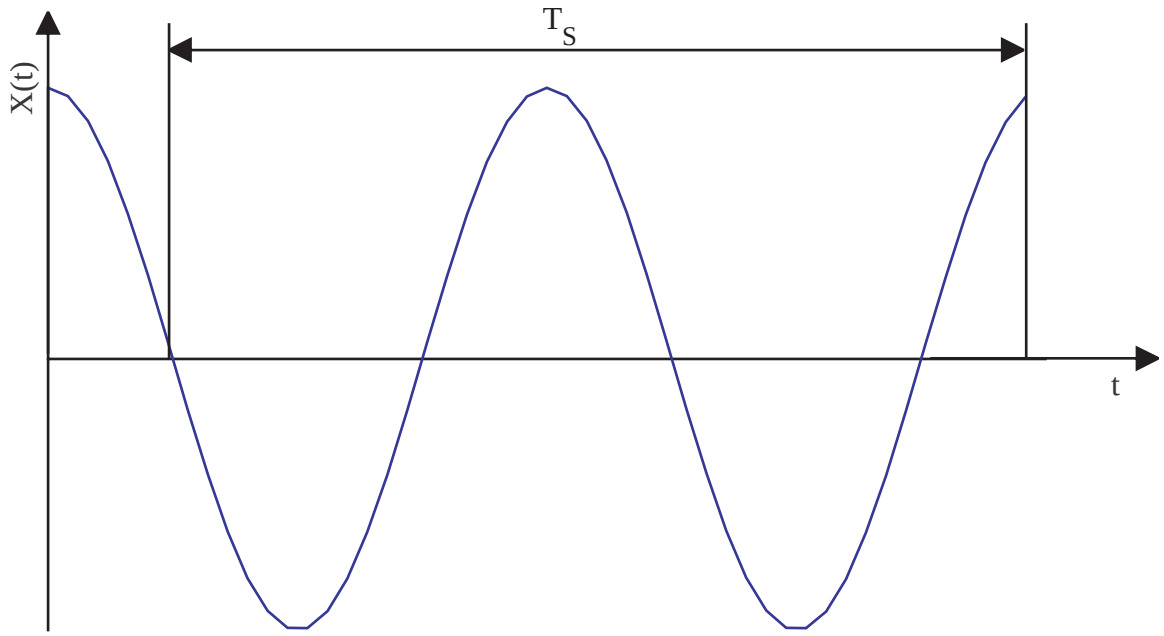
Gli errori che si commettono sono fondamentalmente dovuti a due fatti: l'***incompatibilità*** tra le frequenze del ***segnale analogico*** e quella di ***campionamento*** e il ***troncamento*** del segnale

Nel primo caso lo ***spettro di frequenza*** del ***segnale campionato*** differisce da quello del segnale analogico per la presenza di ***componenti armoniche spurie*** dovute alle sovrapposizione delle "***immagini***" del ***segnale*** introdotte dal ***campionamento*** intorno ai ***multipli interi*** della f***requenza di campionamento***

Per eliminare il ***fenomeno*** è necessario scegliere la ***frequenza di campionamento*** rispettando il ***teorema di Shannon*** secondo il quale detta ***frequenza*** deve essere almeno pari a ***due volte la banda del segnale analogico***

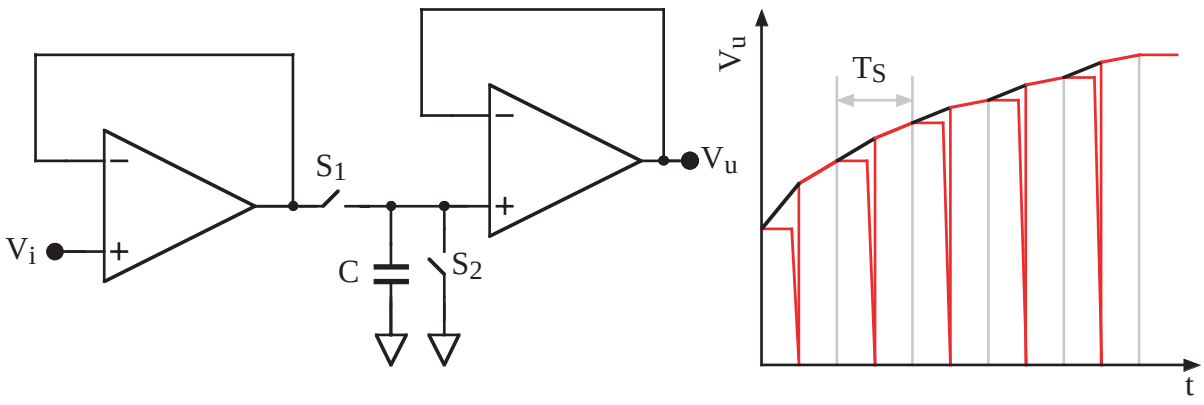


Gli ***errori*** di ***troncamento*** si verificano invece quando la ***finestra di osservazione*** non contiene tutto il ***segnale analogico*** originale (o un numero ***non intero*** di ***periodi*** di un segnale alternato), in quanto vengono introdotte delle false ***discontinuità*** e quindi ***frequenze spurie*** nello spettro



Infine si deve tenere presente che un ***sistema reale*** non è in grado di generare una ***funzione di Dirac***, ma solo ***impulsi*** di ***durata finita***

Per ovviare alle difficoltà sopra ricordate, si usano allora ***sistemi di campionamento***, detti ***sample and hold***



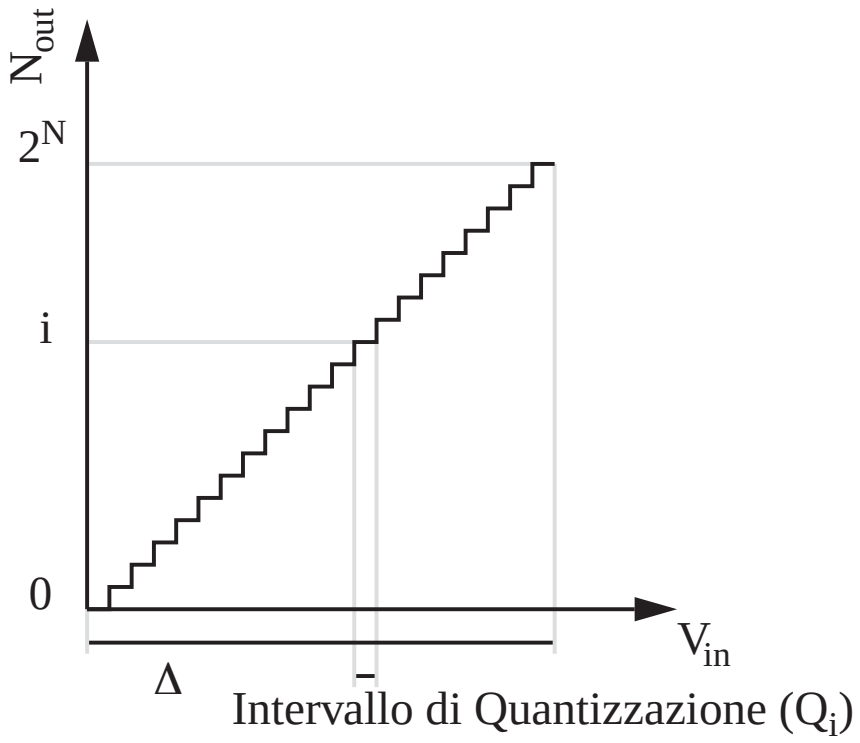
In questi ***circuiti*** si utilizzano ***interruttori*** rapidi (tempi di intervento dell'ordine dei nanosecondi) per escludere il ***segnale*** d'***ingresso*** dal ***circuito di conversione*** per il tempo T_S durante il quale V_i è mantenuta dal ***condensatore*** C

Il processo di ***discretizzazione*** in ***ampiezza*** o ***quantizzazione***, a differenza del ***campionamento***, introduce inevitabilmente un'***incertezza***, detta ***errore di quantizzazione***

Il ***segnale digitale*** in uscita da un ***convertitore A/D***, infatti, è per definizione costituito da un ***numero finito*** di ***bit*** (N) che identificano $2^N - 1$ ***intervalli di quantizzazione***, ciascuno di ampiezza $\Delta / (2^N - 1)$, dove Δ denota l'***ampiezza massima*** del segnale

Pertanto, tutti i ***livelli analogici*** compresi in un particolare ***intervallo di quantizzazione*** dopo la ***conversione A/D*** risultano ***indistinguibili***, provocando una ***perdita*** di ***informazione***

L'entità dell'***errore di quantizzazione*** risulta tanto minore quanto maggiore è la ***risoluzione*** del ***convertitore A/D***, definita dal ***numero*** *N* di ***bit*** in uscita

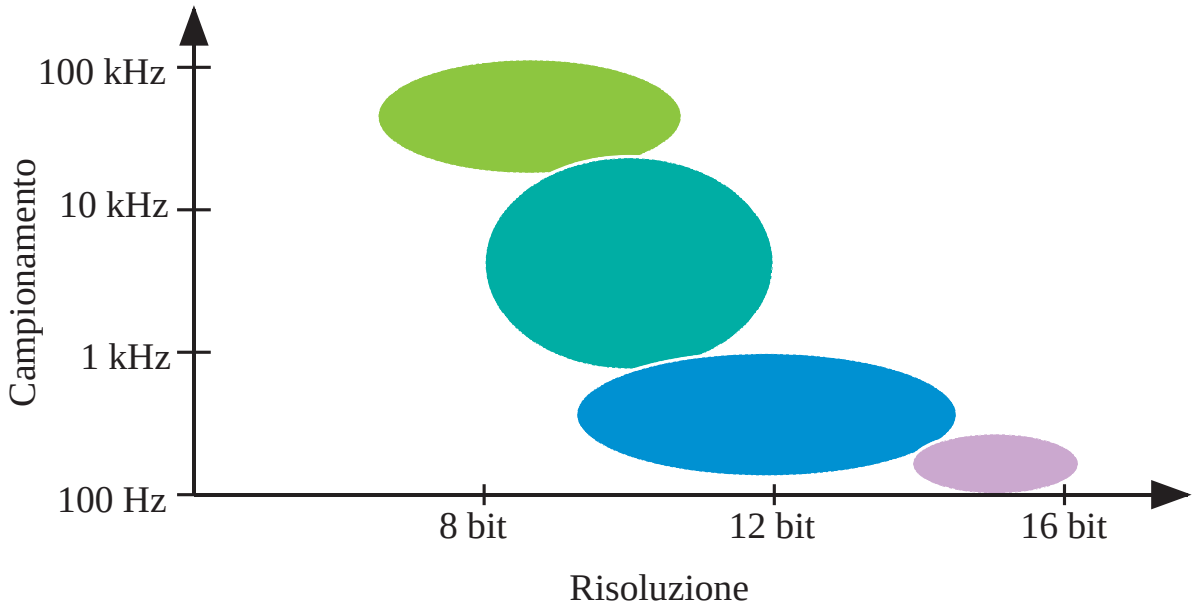


9.4. Conversione A/D

Una volta ***campionato*** il ***segnale analogico***, per completare il processo di ***conversione A/D*** occorre stabilire in quale ***intervallo di quantizzazione*** si trovi ciascun ***campione***, in modo da ottenerne la ***rappresentazione numerica***

Esistono numerose ***tecniche circuitali*** per effettuare questa ***operazione***

In generale, le diverse ***tecniche*** permettono di ottenere una ***elevata risoluzione*** con ***bassa frequenza di campionamento*** oppure una ***bassa risoluzione*** con una ***elevata frequenza di campionamento*** (il contenuto informativo per unità di tempo resta grossomodo costante)

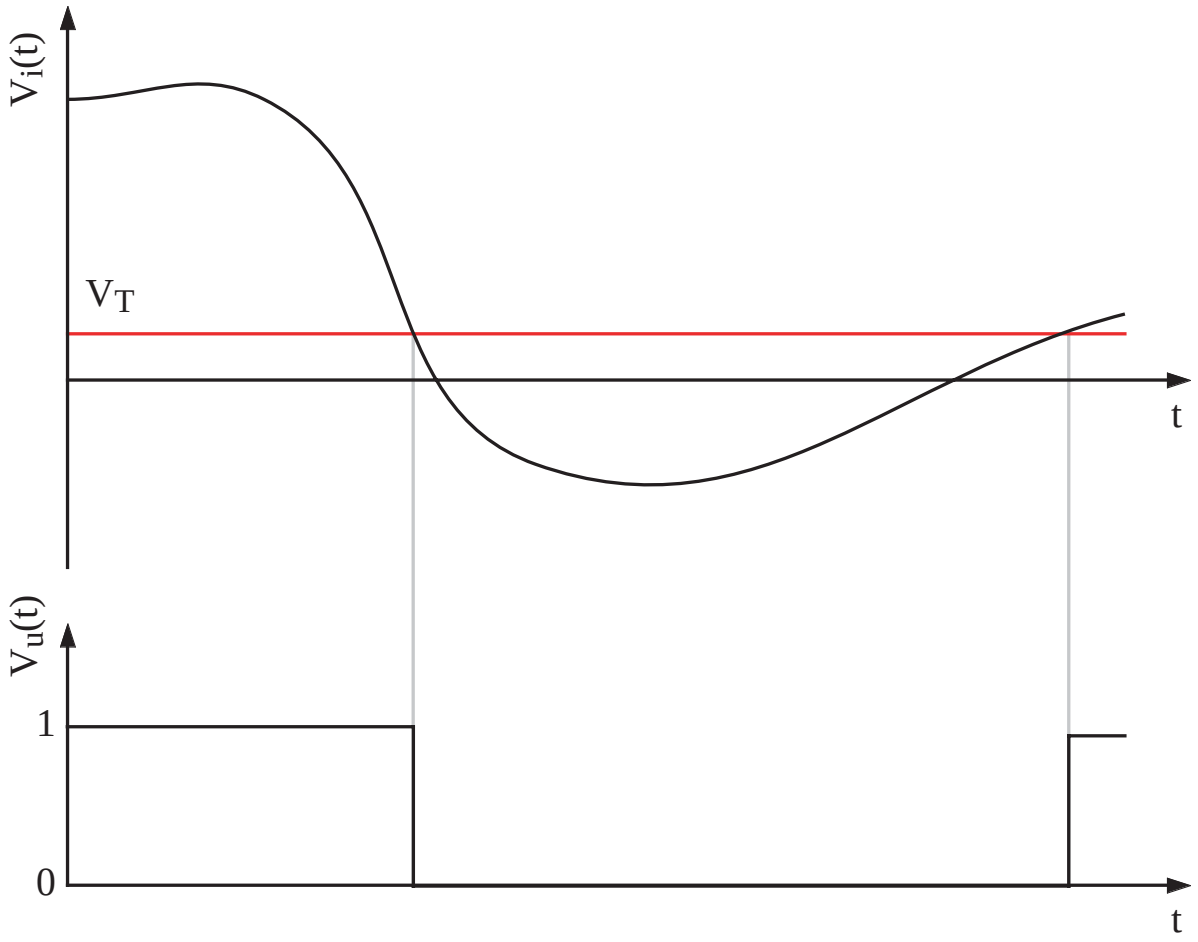


A seconda dei casi, quindi, occorre scegliere la ***tecnica*** di ***conversione*** che garantisce il miglior ***compromesso*** tra ***frequenza di campionamento*** e ***risoluzione***

Prima di considerare alcune ***tecniche*** di ***conversione A/D*** particolarmente adatte per gli ***strumenti di misura***, è utile esaminare alcuni ***blocchi base*** di uso comune nei ***convertitori A/D***

9.4.1. Comparatori

In ogni ***convertitore A/D*** è presente almeno un ***comparatore***



Un ***comparatore*** riceve in ingresso un ***segnale analogico*** (V_i) e fornisce in uscita un ***segnale digitale*** (V_u) di livello "0" se $V_i < V_T$ oppure di livello "1" se $V_i > V_T$

Esso costituisce quindi un ***quantizzatore*** con ***risoluzione*** di un bit

Il ***segnale*** in uscita può essere un ***impulso*** o un ***livello di tensione***, a seconda dei casi

L'***incertezza*** legata alla ***operazione*** di ***comparazione*** dipende dalla ***sensibilità*** del ***comparatore***, cioè la minima variazione di V_i che determina la commutazione del livello del segnale di uscita e dalla sua ***rapidità di risposta***

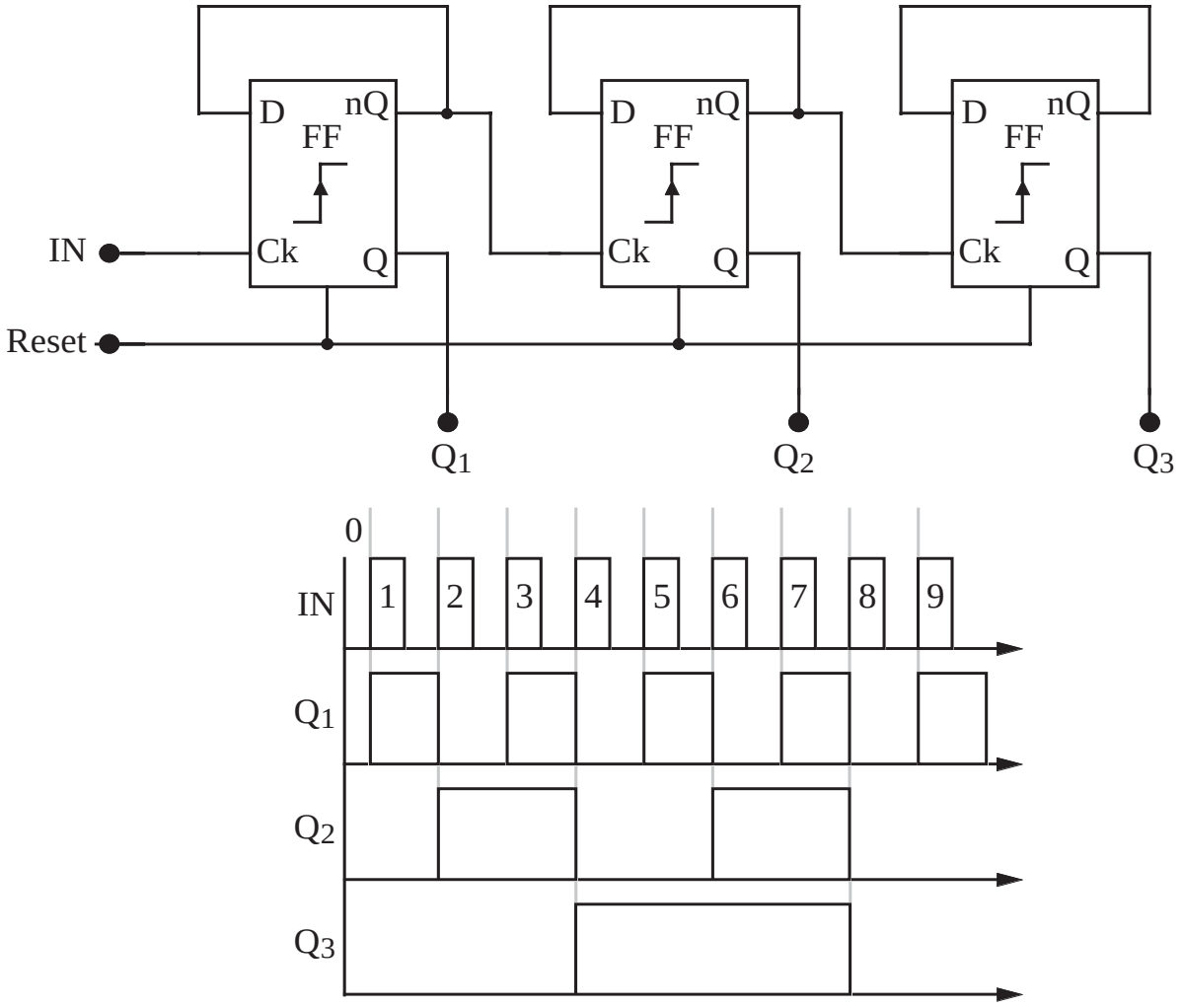
La ***tecnologia*** attuale impiega nei ***comparatori*** circuiti ***amplificatori*** ad alto guadagno con ***ingresso differenziale***, costituiti da ***circuiti integrati*** e quindi molto stabili e sensibili, con buona velocità di risposta

9.4.2. Contatori

Nella ***conversione A/D*** un ruolo importante hanno i ***sistemi di conteggio*** degli ***impulsi*** (contatori di impulsi)

Per questi ***sistemi*** si ricorre solitamente all'uso di ***catene*** di ***flip-flop*** collegati in ***cascata***

Per comprendere qualitativamente il funzionamento di un ***contatore di impulsi***, si può fare riferimento a un caso ***semplice*** con solo tre ***flip-flop*** (*FF*)



Il primo ***impulso*** inviato in ingresso al primo ***flip-flop*** (*IN*) provoca la transizione di Q_1 che passa dal livello “0” al livello “1”, senza che il secondo ***flip-flop*** cambi lo stato della propria uscita

Quando un secondo ***impulso*** viene applicato al primo ***flip-flop***, Q_1 passa dallo stato “1” allo stato “0”, mentre Q_2 passa dallo stato “0” allo stato “1”

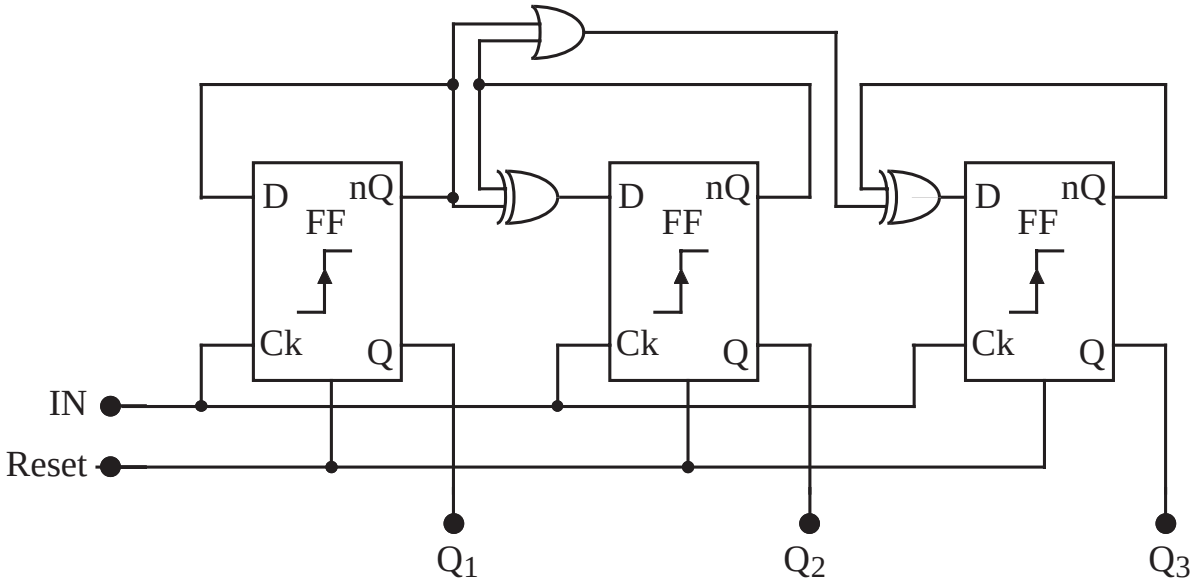
Con ragionamento analogo si può comprendere cosa succede per il terzo ***flip-flop***

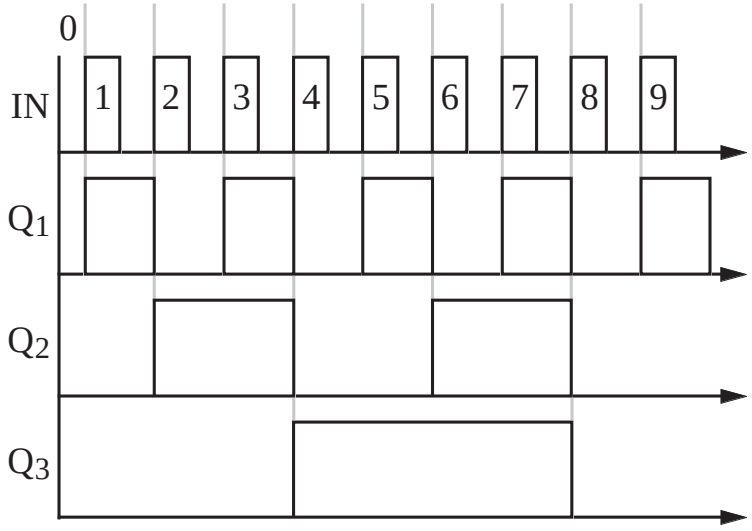
Si può osservare che in ragione del ***meccanismo*** descritto la ***catena*** è in grado di contare fino a $2^3 = 8$ ***impulsi*** e che la parola digitale $Q_3\ Q_2\ Q1$ rappresenta la ***codifica binaria*** del ***numero*** di ***impulsi*** ricevuti

Più generalmente, un ***contatore*** con una ***cascata*** di N ***flip-flop*** può contare fino a 2^N ***impulsi***

Un gruppo di ***flip-flop*** collegati funzionalmente come descritto sopra costituisce un ***modulo*** del ***contatore***

A differenza del tipo di ***contatore*** descritto, che è detto ***asincrono***, si può ricorrere al tipo in cui gli ***impulsi*** da contare sono inviati contemporaneamente a tutti i ***flip-flop*** con il vantaggio di ridurre i ***tempi di propagazione*** dei ***segnali*** (***contatore sincrono***)





I ***contatori*** sono generalmente integrati da un ***dispositivo*** di ***azzeramento*** (*Reset*) che consente automaticamente o manualmente di riportare ogni ***modulo*** alle condizioni iniziali

9.4.3. Memorie

I sistemi di ***memoria*** hanno la funzione di ***raccogliere*** in forma ***ordinata*** le ***informazioni*** che devono essere successivamente utilizzate

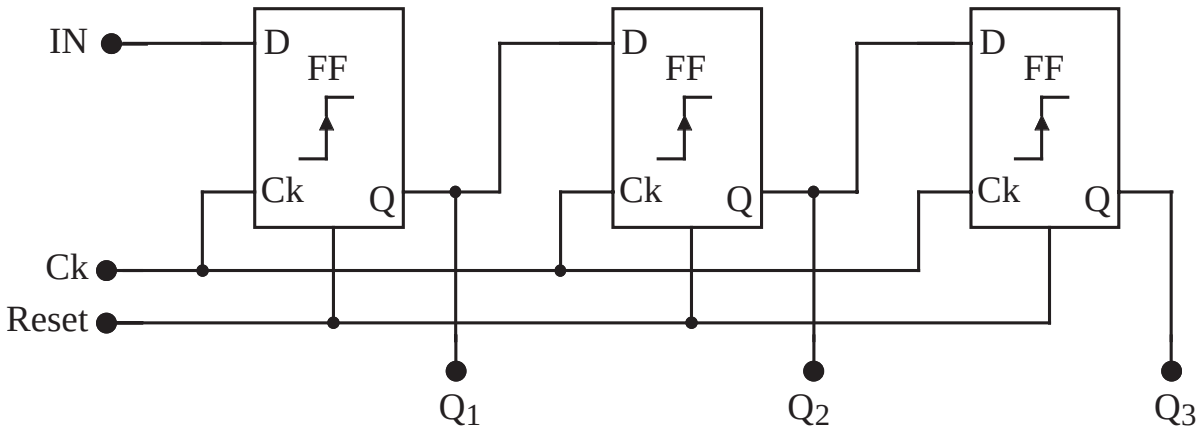
In esse possono essere immagazzinate anche le ***istruzioni*** per eventuali programmi di ***elaborazione dei dati***

Le ***memorie*** sono costituite da un numero, generalmente elevato, di ***celle*** ciascuna delle quali è in grado di ***accogliere*** e ***conservare*** l'***informazione*** di un solo bit

Un ***sistema*** di questo tipo è più pregiato se molto ***veloce*** nell'acquisizione dei ***dati*** e a ***basso consumo***

Esistono due ***categorie*** fondamentali di ***memorie*** che si differenziano nel modo con il quale possono essere ***gestite***

Un primo esempio è lo ***shift-register*** (registro a scorrimento) che utilizzando un certo numero di ***flip-flop*** consente di ottenere ***memorie temporanee*** di limitata capacità



Memorie con ***capacità*** molto più elevate sono ottenute ricorrendo a ***flip-flop*** raccolti in strutture monolitiche da 16 a 1024 elementi, impiegando transistori MOS o logiche ELC

Le ***memorie RAM*** (Random-Access Memory) sono volatili in quanto il loro contenuto può essere letto, modificato e integrato da istruzioni

A differenza di quanto avviene negli ***shift-register***, le operazioni di ***scrittura*** e ***lettura*** sono indipendenti dalla ***posizione*** della ***parola***

Nella ***memoria RAM*** l'informazione è ***immagazzinata*** secondo una ***disposizione*** a ***righe*** e ***colonne*** e può essere di tipo ***statico*** o ***dinamico*** (in questo secondo caso l'informazione resta in memoria per un tempo limitato)

Le ***memorie ROM*** (Read-Only Memory) sono anch'esse realizzate in forma di ***matrice*** ma sulle informazioni contenute non è possibile alcuna ***alterazione*** o istruzione, per cui esse mantengono la configurazione ad esse data in origine (salvo cancellarne il contenuto)

Si ricorda che la ***capacità*** di una ***memoria*** può essere espressa in termini di numero di ***righe*** e ***colonne***, ma più frequentemente si preferisce esprimerla mediante il prodotto dei due termini sopra indicati (ad esempio, 256 bit)

9.4.4. Conversione dal Codice Binario al Decimale

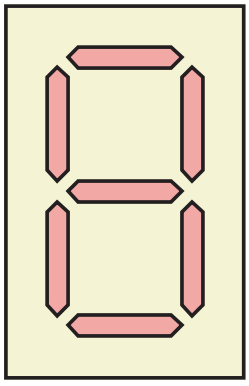
Qualora il ***risultato*** di una ***misurazione*** debba essere espresso visivamente in ***forma digitale*** è necessario provvedere ad una ***transcodifica*** in modo che il risultato sia espresso in un ***linguaggio*** comprensibile ad un normale ***operatore*** e cioè quello ***decimale***

Il ***passaggio*** risulta alquanto ***complesso*** dal punto di vista operativo anche se ***concettualmente*** piuttosto ***semplice*** ed ***intuitivo***

In pratica, si utilizzano ***catene*** di ***flip-flop***, in numero opportuno ed opportunamente collegati, a seconda del ***codice*** che si vuole convertire

In tal modo è possibile ***trasformare*** il ***codice binario*** in una sequenza di ***quartine di bit*** (corrispondenti al ***codice BCD***) e inviarla ad un opportuno ***dispositivo di visualizzazione***

Negli ***strumenti*** moderni si usano ***diodi luminescenti*** o ***cristalli liquidi***, che descrivono le ***cifre decimali*** tramite ***sette segmenti*** che possono essere attivati o disattivati



9.4.5. Circuiti di Interfaccia

La maggior parte dei ***sistemi numerici*** comprende un ***oscillatore*** che genera una ***sequenza*** continua di ***impulsi*** detta ***clock*** (di forma opportuna) distanziati l'uno dall'altro di un tempo T_{Ck} costante: al solito si tratta di uno ***oscillatore al quarzo*** di buona precisione e stabilità

La ***trasmissione*** dei ***dati*** in uscita da un ***convertitore A/D*** verso i blocchi di ***elaborazione*** successivi può avvenire attraverso un ***singolo canale*** di trasmissione (in ***serie***), per cui per ogni ***informazione elementare*** è necessario un tempo T_{Ck} (***trasmissione seriale***)

In alternativa, la ***trasmissione*** può essere effettuata per mezzo di n vie in ***parallelo***, per cui la durata di trasmissione si riduce a T_{Ck}, a scapito di una maggiore complessità delle connessioni (***trasmissione parallela***)

Eventuale ***rumore*** presente in fase di ***trasmissione*** può far comparire ***impulsi spuri***

Tra i vari modi per ***rivelare*** questi ***disturbi***, quello più semplice consiste nell'inserire, al termine di ciascun ***messaggio***, uno "0" o un "1" ***supplementare***, a seconda che esso contenga un numero ***pari*** o ***dispari*** di "1"

All'arrivo si controlla che il ***numero*** di "1" sia sempre ***pari*** (***controllo di parità***) in quanto ciò è ritenuto sufficiente, considerando assai improbabile un doppio errore in un solo passaggio

9.4.6. Convertitori D/A

La ***conversione digitale/analogica*** rappresenta l'operazione ***inversa*** di quella ***analogico/digitale***

Negli ***strumenti*** essa si rende necessaria sia, come vedremo, nella ***realizzazione*** di alcuni ***convertitori A/D*** sia quando il ***misurando*** deve essere fornito in uscita in ***forma analogica***, ad esempio per la rappresentazione analogica di una tensione sullo schermo di un ***oscilloscopio***

In generale è richiesto un ***segnale analogico*** sotto forma di ***tensione*** o ***corrente***, per cui l'operazione consiste nel ***convertire*** le ***informazioni numeriche*** in tante ***unità di base*** del segnale di uscita e di provvedere poi alla loro ***somma***

L'***unità di base***, espressa nell'***unità di misura*** della ***grandezza analogica*** cercata, rappresenta il ***minimo valore*** in uscita dal ***convertitore*** e quindi la sua ***risoluzione***

Per una ***tensione*** si può scrivere la espressione

$$V_u = \frac{V_r}{2^N} \sum_{i=0}^{N-1} b_i 2^i$$

nella quale b_i è il ***singolo bit*** e può quindi essere 0 o 1, N è il ***numero di bit*** disponibili e $V_r / 2^N$ è l'***unità di base***

Poiché il ***massimo*** di questa ***funzione*** si ha quando tutti i ***bit*** sono uguali a 1, resta anche definito il ***valore*** di ***fondo scala*** della ***grandezza analogica*** (V_r)

Da quanto esposto appare evidente che tanto più ***elevato*** è il ***numero di bit***, tanto ***migliore*** risulta la ***risoluzione*** del ***segnale analogico***

Questo aspetto, che incide notevolmente sul ***costo*** dello ***strumento***, è di fondamentale importanza quando si desideri ottenere un ***segnale analogico*** affetto da modesta ***incertezza*** e con elevata ***definizione***

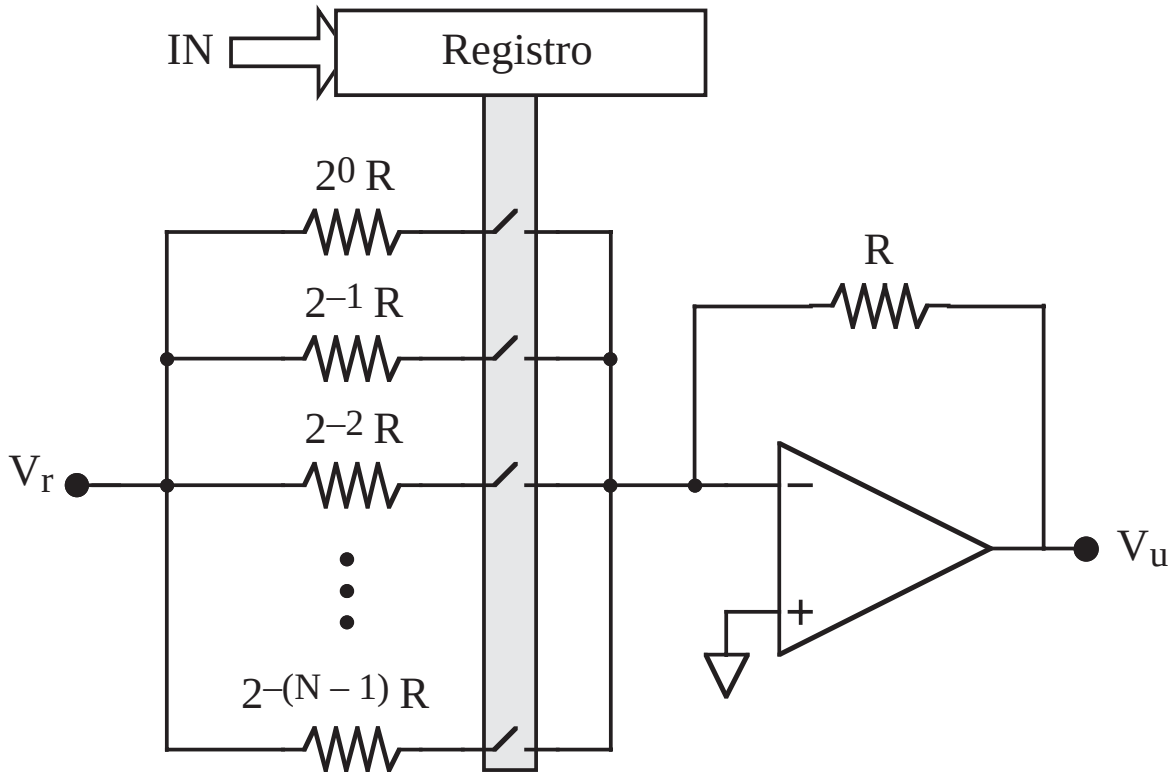
Un esempio di ***convertitore D/A*** è costituito da:

Una ***sorgente*** interna di ***tensione continua*** di riferimento fortemente stabilizzata

Un certo numero di ***interruttori analogici***

Resistori di precisione di valori diversi per pesare i ***singoli bit*** (in termini di corrente)

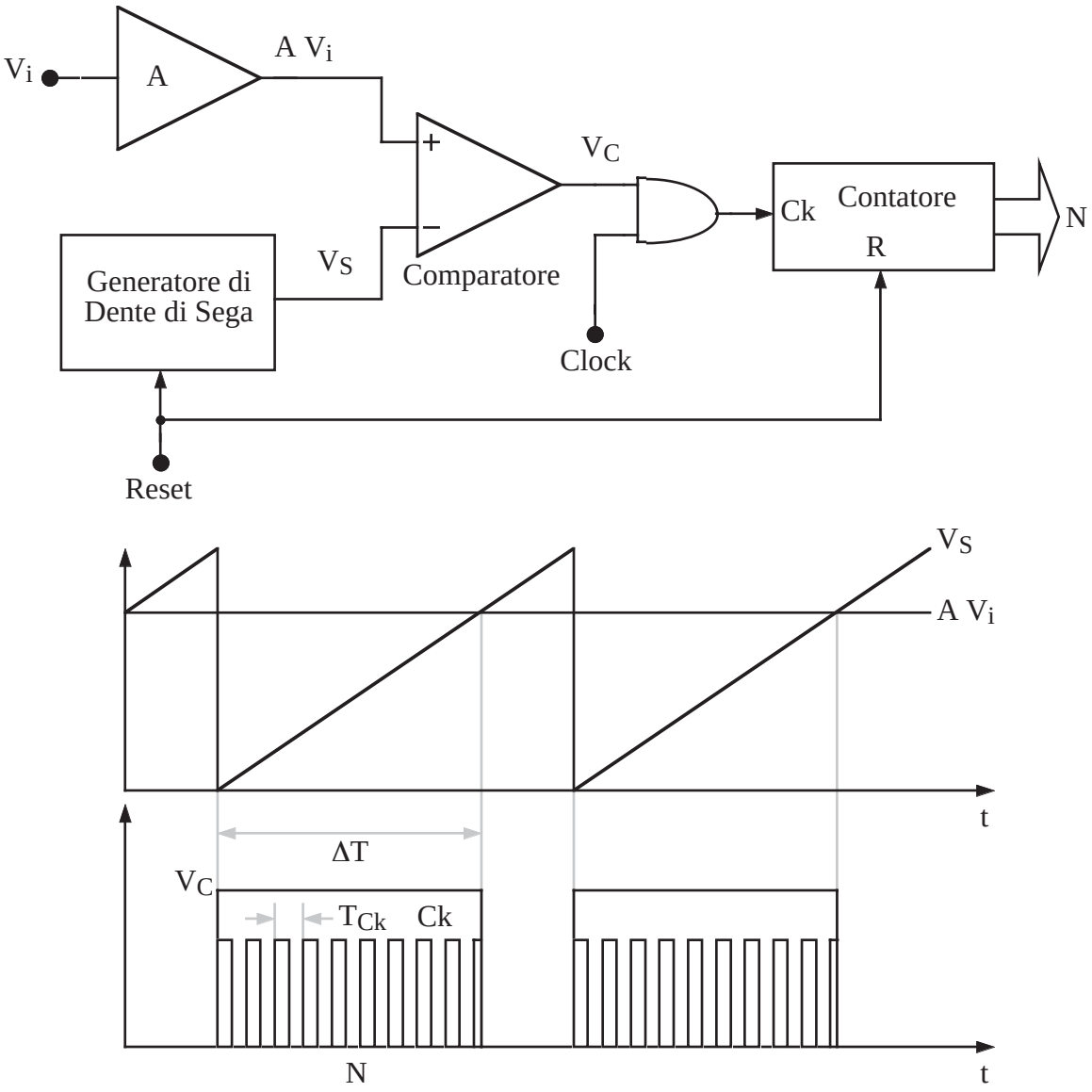
Un ***amplificatore*** operazionale ***sommatore***



Gli ***interruttori analogici*** devono essere comandati da segnali controllati dai ***bit*** della ***parola*** da convertire

9.4.7. Convertitore A/D a Dente di Sega o a Rampa Lineare

In un ***convertitore A/D*** a ***dente di sega*** o a ***rampa lineare*** la ***tensione*** di ingresso V_i, opportunamente ***amplificata*** (A), viene ***confrontata***, tramite un ***comparatore*** con una ***tensione*** a ***dente di sega*** ($V_S = k\,t$)



La ***tensione*** di uscita del ***comparatore*** V_C si trova al ***livello logico alto*** fintanto che $A\,V_i > V_S$, mentre passa al l***ivello logico basso*** non appena $A\,V_i = V_S$

Un ***contatore di impulsi*** determina il ***numero*** di ***impulsi di clock*** di periodo T_{Ck} contenuti nell'***intervallo di tempo*** ΔT in cui la tensione V_C si trova al ***livello logico alto***, fornendo in uscita un numero N a n bit dato da

$$N = \frac{\Delta T}{T_{Ck}} = \frac{AV_i}{kT_{Ck}} = k_N V_i$$

dove k_N rappresenta la ***costante*** del ***convertitore A/D***

La ***risoluzione*** del ***convertitore*** è essenzialmente determinata dalla ***pendenza*** del ***dente di sega*** k e dal ***periodo*** T_{Ck} ***del clock*** utilizzato

In particolare, per ***aumentare*** la ***risoluzione*** occorre ***ridurre*** il più possibile sia k sia T_{Ck}

Supponendo di utilizzare la ***massima frequenza di clock*** possibile (T_{Ck} minimo), quindi, il ***tempo massimo*** necessario per effettuare una ***conversione***, dato da

$$\Delta T_{max} = \frac{AV_{i,\,max}}{k} = 2^n T_{Ck}$$

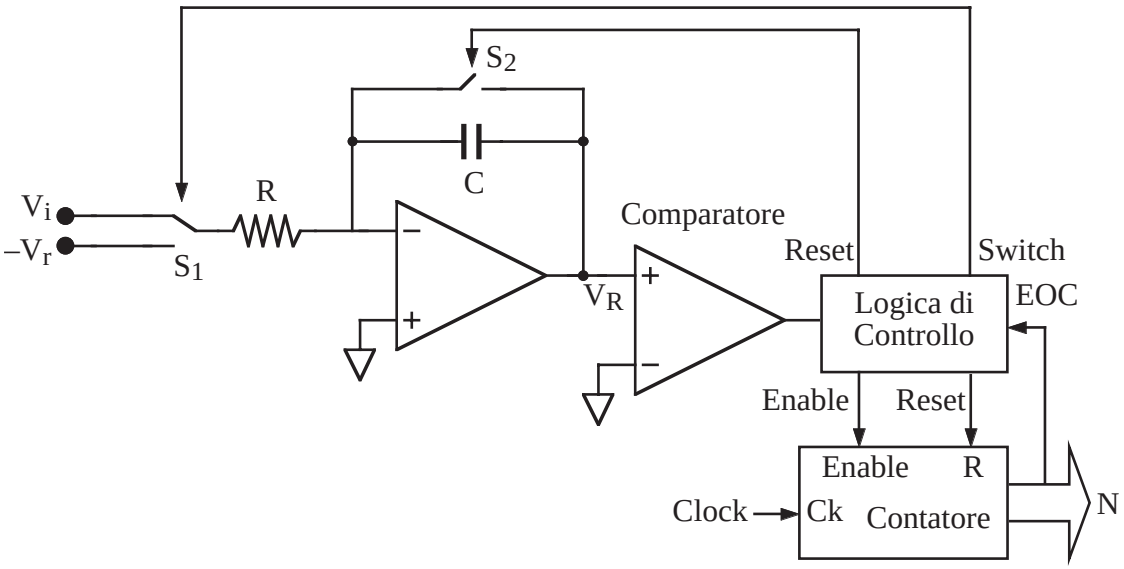
cresce al crescere della ***risoluzione*** richiesta (k diminuisce)

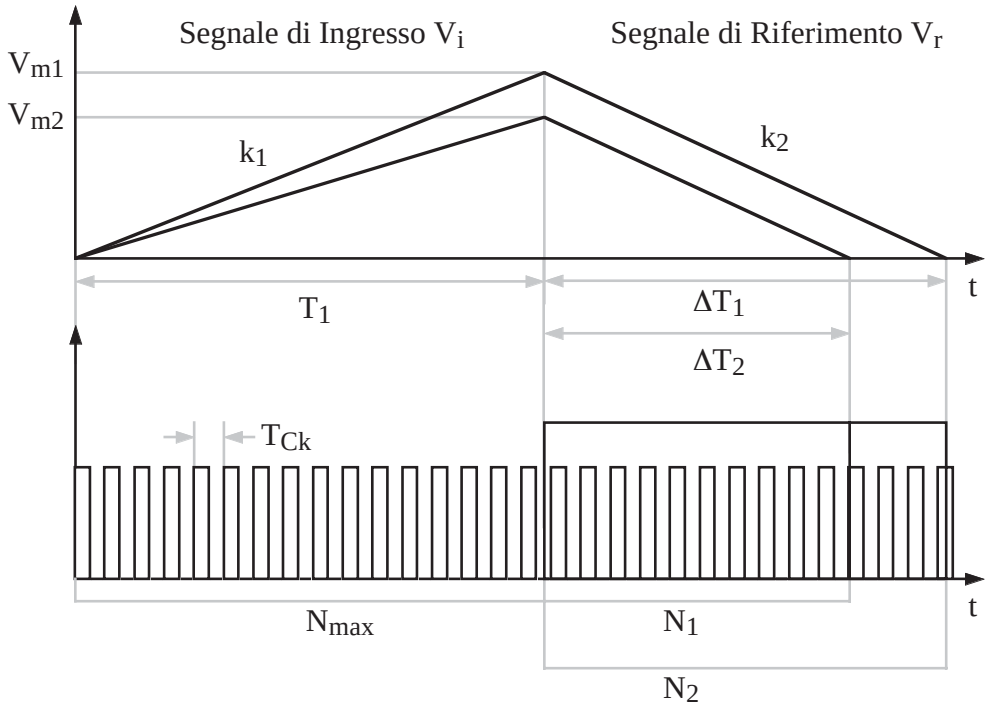
La ***linearità*** e la ***precisione*** della ***conversione*** dipendono ovviamente dalla ***linearità*** del ***dente di sega*** e dalla ***costanza*** del ***periodo del clock***

Nella maggioranza dei casi la ***pendenza*** della ***rampa*** prodotta dal ***generatore di dente di sega*** rappresenta il contributo dominante all'***incertezza*** del ***convertitore***, in quanto essa dipende tipicamente da una ***costante di tempo*** $R\,C$ difficile da controllare

9.4.8. Convertitore A/D a Doppia Rampa Lineare

Il ***convertitore A/D*** a ***doppia rampa lineare*** è basato sullo stesso principio di funzionamento del ***convertitore*** a dente di sega, ma è in grado di raggiungere ***prestazioni*** nettamente ***migliori***, in quanto il risultato della ***conversione*** viene reso indipendente dalla ***costante di tempo*** *R C* utilizzata per generare la ***rampa*** lineare





Il ***ciclo*** di ***conversione*** di un ***convertitore*** a ***doppia rampa lineare*** è diviso in ***due fasi*** distinte

In una ***prima fase***, tramite un ***modulo integratore***, viene generata una ***rampa*** la cui pendenza è ***proporzionale*** al ***segnale di ingresso*** data da

$$V_R = \frac{V_i}{RC} t = k_1 t$$

All'inizio della ***seconda fase***, dopo un ***intervallo di tempo*** prefissato T_1, l'***interruttore*** S_1 viene ***commutato*** connettendo l'***ingresso*** dell'***integratore*** alla ***tensione*** $-V_r$

L'***uscita*** dell'***integratore***, che nel frattempo ha raggiunto il ***valore***

$$V_{R,\,max} = V_m = \frac{V_i}{RC} T_1 = k_1 T_1,$$

inizia quindi a ***scendere*** secondo l'equazione

$$V_R = V_m - \frac{V_r}{RC} t = V_m - k_2 t$$

Contemporaneamente viene abilitato un ***contatore di impulsi*** (*Enable*) pilotato da un opportuno ***segnale di clock*** (*Clock*)

Quando la ***tensione*** V_R raggiunge lo zero il ***comparatore*** cambia ***stato*** e il ***conteggio*** viene fermato

L'***intervallo di tempo*** necessario a ***scaricare*** completamente la ***capacità*** C si ricava dall'equazione

$$V_m - k_2 \Delta T = 0$$

e risulta pari a

$$\Delta T = \frac{V_m}{k_2} = \frac{k_1 T_1}{k_2} = \frac{\frac{V_i}{RC} T_1}{\frac{V_r}{RC}} = \frac{V_i}{V_r} T_1$$

Si può notare come il ***valore*** di ΔT sia ***indipendente*** dalla ***costante di tempo*** $R\,C$

Scegliendo $T_1 = 2^n T_{Ck}$, il ***codice digitale*** che si ottiene in uscita al ***convertitore*** risulta

$$N = \frac{\Delta T}{T_{Ck}} = \frac{V_i}{V_r} \frac{T_1}{T_{Ck}} = \frac{V_i}{V_r} 2^N$$

La ***precisione*** del ***convertitore***, quindi, in questo caso dipende solo dalla ***precisione*** con cui si realizza la ***tensione di riferimento*** V_r, mentre la dipendenza dal ***periodo del clock*** e dalla ***costante di tempo*** $R\,C$ viene eliminata

Questo ***miglioramento*** della ***precisione*** del ***convertitore*** viene ottenuto a spese di un ***periodo di conversione*** più ***lungo*** che non in un ***convertitore*** a ***dente di sega***

Il ***tempo*** necessario per ottenere il ***codice digitale*** in uscita risulta infatti pari a

$$\Delta T_{max} = 2^n T_{Ck} + 2^n T_{Ck} = 2^{n+1} T_{Ck}$$

L'***interruttore*** S_2 controllato dal segnale *Reset* permette di ***resettare*** completamente l'***integratore*** prima di iniziare un ***ciclo*** di ***conversione***

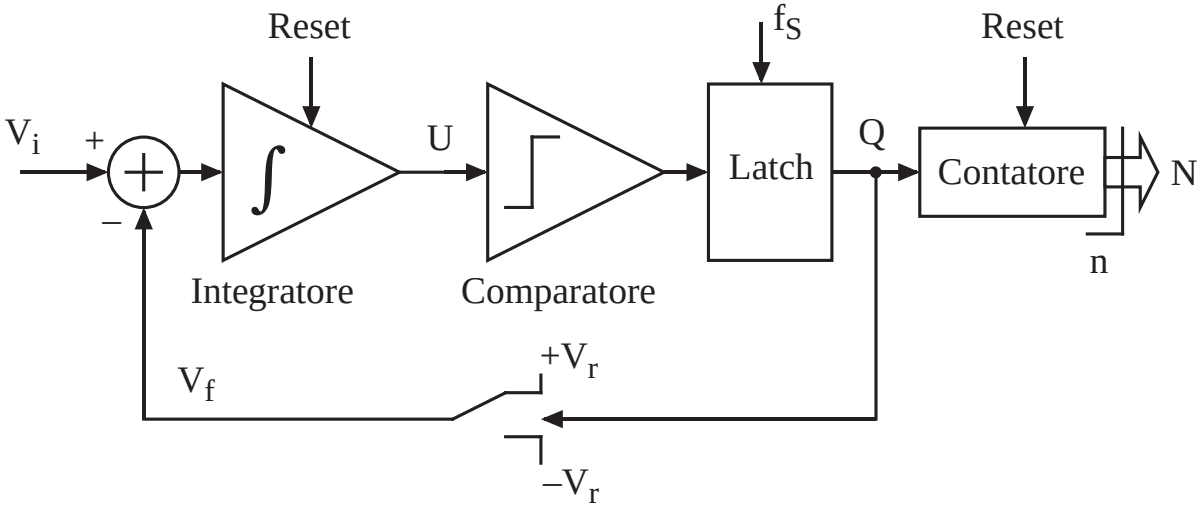
Contestualmente viene ***resettato*** anche il ***contatore***

Una opportuna ***logica*** di ***controllo*** si occupa di generare tutti i ***segnali*** necessari al funzionamento del ***convertitore*** (*Reset*, *Enable*, *Switch*)

Il segnale *EOC*, fornito dal ***contatore*** alla ***logica*** di ***controllo*** viene utilizzato per identificate l'***istante di tempo*** T_1

9.4.9. Convertitore A/D Incrementale

Un ***convertitore A/D incrementale*** contiene gli stessi ***elementi costitutivi*** di un ***convertitore*** a ***doppia rampa lineare***, ovvero un ***integratore***, un ***comparatore*** e un ***contatore***, il principio di ***funzionamento*** è tuttavia diverso



L'***equazione*** che regola il ***comportamento*** di un ***convertitore incrementale***, infatti, è data da

$$\begin{cases} U_0 = V_i \\ U_{k+1} = U_k + (V_i - (-1)^{Q_k + 1} V_r) \end{cases}$$

In pratica ad ogni ***colpo di clock*** (indice k) il ***comparatore*** verifica il ***segno*** del ***segnale di uscita*** dell'***integratore*** U, determinando se al ***colpo di clock*** successivo la ***tensione di riferimento*** V_r deve essere ***sommata*** ($U < 0$) o ***sottratta*** ($U > 0$) al ***segnale di ingresso*** V_i

Trattandosi di un ***anello*** di ***reazione negativa*** con ***elevato guadagno*** per le ***basse frequenze*** (per via dell'***integratore***), il ***segnale*** V_f tenderà ad ***uguagliare*** in ***media*** il ***segnale di ingresso*** V_i

Pertanto, il ***contatore***, accumulando il ***segnale digitale*** Q (legato a V_f a meno della tensione V_r), dopo 2^n colpi di clock fornirà in uscita una ***parola digitale*** N data da

$$N = \sum_{k=0}^{2^{n-1}} Q_k = 2^n \overline{Q} = 2^n \frac{\overline{V_f}}{V_r} = 2^n \frac{V_i}{V_r}$$

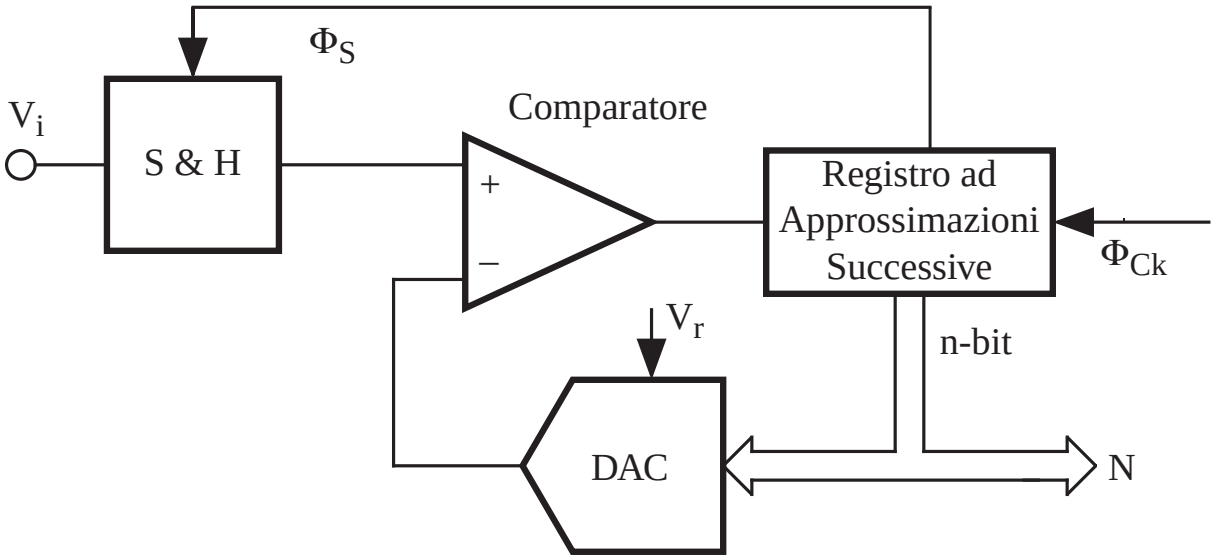
dove $\overline{Q}$ e $\overline{V_f}$ rappresentano i ***valori medi*** di Q e V_f

Ovviamente, il ***tempo di conversione*** risulta pari a 2^n ***periodi di clock***

Il ***segnale*** *Reset* permette di ***azzerare*** l'***integratore*** e il ***contatore*** prima di ogni ***conversione***

9.4.10. Convertitore A/D ad Approssimazioni Successive

Un ***convertitore A/D*** ad ***approssimazioni successive*** è costituito da un circuito di ***sample and hold***, un ***comparatore***, un ***convertitore D/A*** (DAC) e un blocco digitale denominato ***registro*** ad ***approssimazioni successive*** (SAR)



Il ***principio di funzionamento*** di questo ***convertitore*** è basato sul ***metodo*** delle ***bisezioni***, che permette di determinare la ***parola digitale*** a *n* bit che rappresenta il ***segnale di ingresso*** in soli *n* ***periodi di clock*** (contro i 2^n periodi di clock dei ***convertitori*** a ***rampa lineare*** o ***incrementali***)

All'inizio di ogni ***ciclo di conversione*** il segnale di ingresso V_i viene ***campionato*** dal ***sample and hold***

Successivamente, il ***segnale di ingresso*** viene ***confrontato*** con la ***tensione analogica*** fornita dal ***DAC*** che corrisponde al ***bit più significativo***

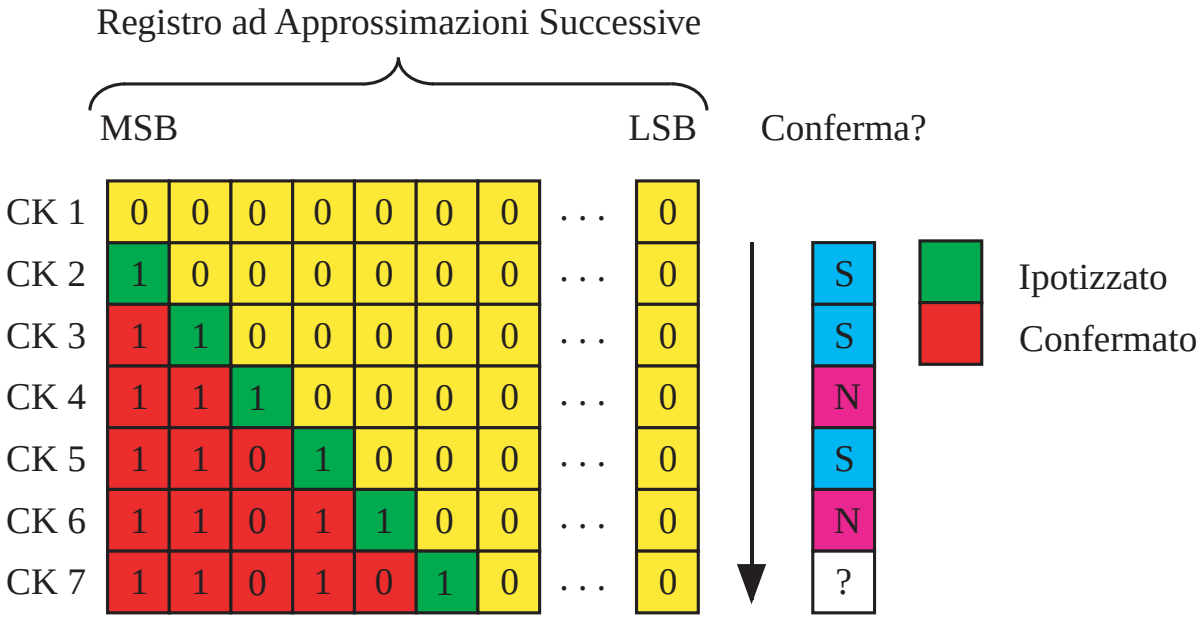
Se il ***segnale di ingresso*** è di ***ampiezza inferiore*** rispetto al ***segnale*** fornito dal ***DAC*** significa che il ***bit più significativo*** della ***parola digitale*** di uscita deve essere posto a “0”, altrimenti significa che esso deve essere posto a “1”

Una volta stabilito il valore del ***bit più significativo***, esso viene ***memorizzato*** dal ***registro ad approssimazioni successive*** e mantenuto

Si passa quindi al ***bit successivo***, ***confrontando*** la ***tensione*** fornita dal ***DAC*** con e il ***segnale d’ingresso***

In base alla ***decisione*** del ***comparatore*** si stabilisce se il ***bit*** in questione deve essere “0” o “1”, ***memorizzando*** poi il risultato nel ***registro ad approssimazioni successive***

Si procede in questo modo per *n* ***periodi di clock*** fino a che non vengono determinati tutti i ***bit***



I ***convertitori*** ad ***approssimazioni successive*** sono notevolmente ***più veloci*** rispetto ai ***convertitori*** a ***rampa lineare*** o ***incrementali***

Tuttavia, essi presentano un ***incertezza maggiore***, legata alla ***precisione*** con cui si riescono a realizzare le ***tensioni*** di uscita del ***convertitore D/A***

Con i ***convertitori*** ad ***approssimazioni successive*** è pertanto molto difficile superare i ***12 bit*** di ***precisione***

9.4.11. Convertitore A/D Flash

Il principio di ***funzionamento*** di un ***convertitore A/D flash*** è molto semplice, in quanto si basa sulla ***definizione*** di ***quantizzazione***

Il ***segnale di ingresso***, infatti, in un ***convertitore*** a n bit viene confrontato con 2^n ***tensioni di riferimento***, tipicamente generate con una ***stringa resistiva***, tramite 2^{n-1} ***comparatori***

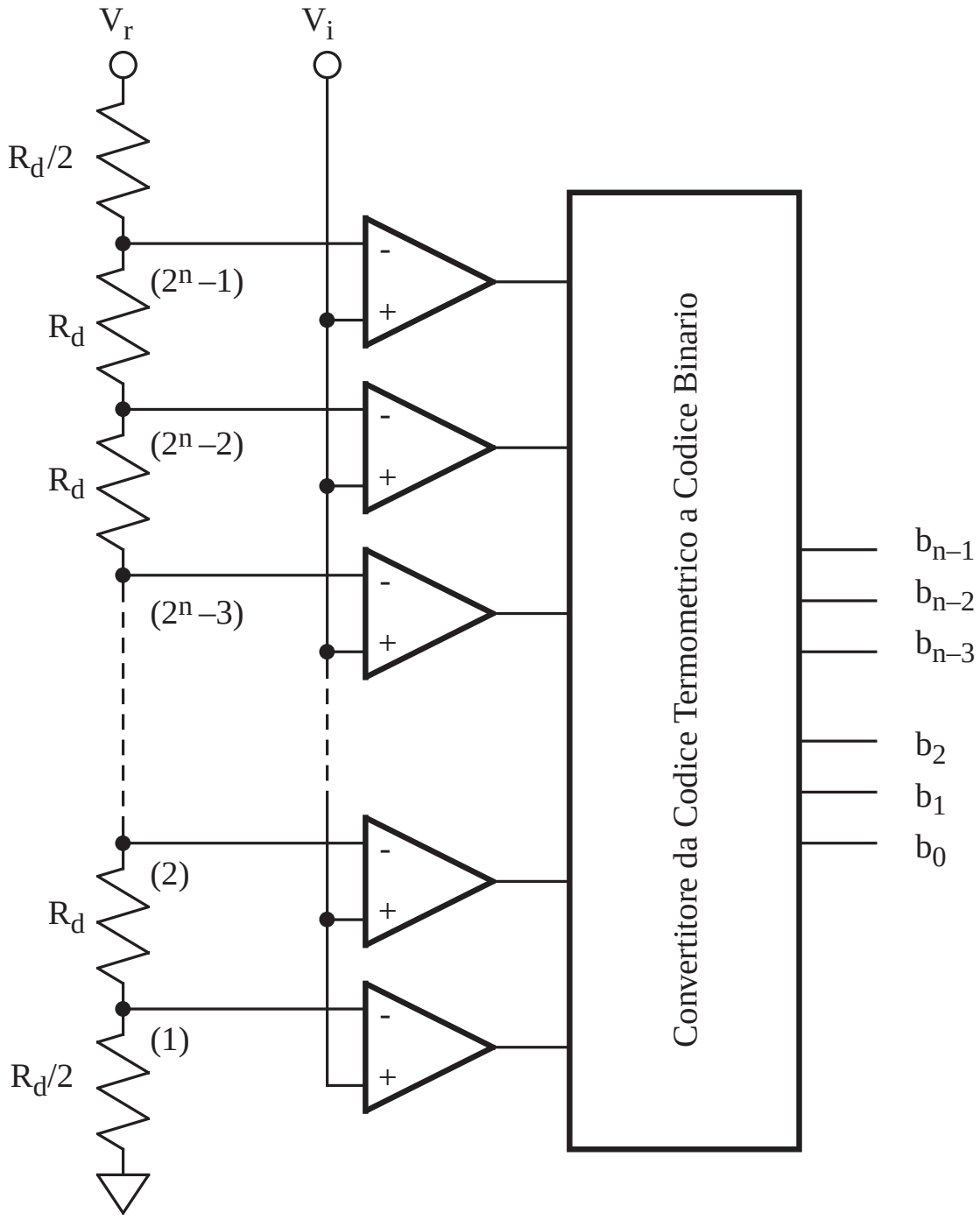
Le ***tensioni di riferimento*** corrispondono ai limiti dei singoli ***intervalli di quantizzazione***

In uscita ai ***comparatori*** si ottengono 2^{n-1} ***segnali digitali*** a 1 bit

Tutti i ***segnali digitali*** corrispondenti a ***comparatori*** la cui ***tensione di riferimento*** è ***inferiore*** al ***segnale di ingresso*** saranno a “0”, mentre gli altri saranno a “1”

Si ottiene quindi una ***rappresentazione digitale*** del ***segnale di ingresso*** secondo un ***codice*** detto "***termometrico***" (per l'ovvia analogia con un termometro a mercurio)

Il ***codice termometrico*** può poi essere ***convertito*** tramite un semplice ***circuito logico*** in un ***codice binario***, in modo da ottenere la ***parola digitale*** di uscita *N*



I ***convertitori A/D flash*** possono raggiungere ***velocità di conversione*** molto elevate in quanto richiedono un solo ***periodo di clock*** per fornire il risultato in uscita

Tuttavia, per via della presenza di un ***numero*** elevato di ***componenti*** (2^n ***resistori*** e 2^{n-1} ***comparatori***), ciascuno con le sue ***tolleranze*** e ***non-idealità***, l'***incertezza*** associata a questi ***convertitori*** risulta elevata

Inoltre, si può notare come la ***complessità*** del ***circuito*** cresca ***esponenzialmente*** con la ***risoluzione***

Pertanto, la ***risoluzione*** massima raggiungibile con ***convertitori A/D flash*** risulta dell'ordine dei ***6 bit***

9.4.12. Convertitore A/D Pipeline

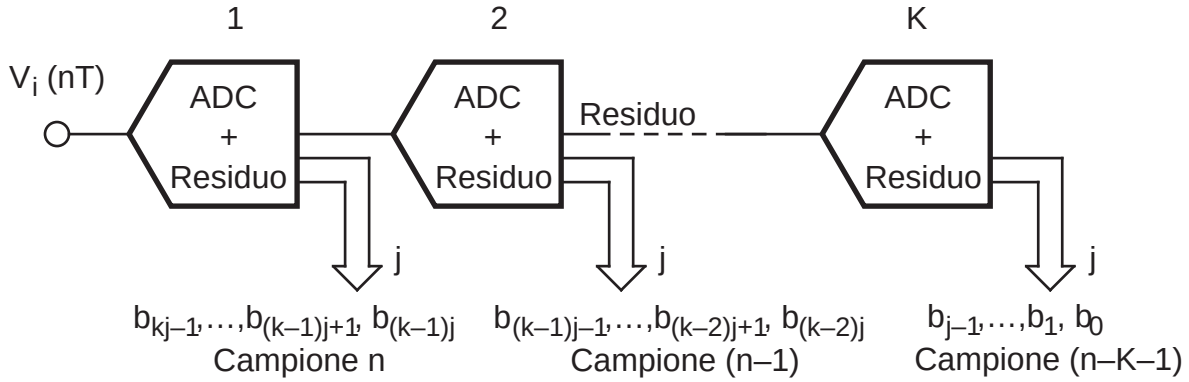
Un ***convertitore A/D pipeline***, invece di operare in modo completamente ***parallelo*** come il ***convertitore flash***, sfrutta il principio della ***catena di montaggio***

In pratica il ***segnale di ingresso*** viene ***convertito*** in ***passi successivi*** da *k* ***stadi*** posti in ***cascata***

Mentre il ***primo stadio*** elabora il ***campione corrente*** del ***segnale d'ingresso***, il ***secondo stadio*** elabora ulteriormente il ***campione*** già elaborato dal ***primo stadio*** nel ***periodo di clock*** precedente, e così via fino all'***ultimo stadio***

Ciascuno degli ***stadi*** produce un ***sottoinsieme*** dei ***bit*** che compongono la ***parola digitale*** di uscita

Il ***convertitore pipeline***, pertanto, a parte una ***latenza iniziale*** di *k* periodi di ***clock***, fornisce in uscita una nuova ***parola digitale*** per ogni ***periodo di clock***, come il ***convertitore flash***



Ciascuno degli ***stadi*** converte in ***digitale*** il proprio ***segnale di ingresso*** con una data ***risoluzione*** (*j*) e fornisce in uscita la corrispondente ***parola digitale*** a *j* bit, nonché un ***residuo*** che dovrà essere poi ***convertito*** dallo ***stadio successivo***

Il ***residuo*** si ottiene ***moltiplicando*** per 2^j la ***differenza*** tra il ***segnale di ingresso*** dello ***stadio*** e il ***segnale di uscita*** a *j* bit ***convertito*** in ***analogico*** da un ***convertitore D/A*** (errore di quantizzazione)

I ***bit*** in uscita ai diversi ***stadi*** vengono poi ***riallineati*** tramite opportuni ***registri*** in modo da costituire la ***parola digitale*** di uscita corrispondente a ogni campione del ***segnale di ingresso***

La ***risoluzione*** di un ***convertitore pipeline*** risulta limitata dall'***accuratezza*** con cui si riescono a realizzare i ***fattori*** 2^j necessari per generare il ***residuo***, nonché dalla ***precisione*** dei ***convertitori A/D*** e ***D/A*** presenti nei singoli ***stadi***

La massima ***risoluzione*** ottenibile si aggira intorno ai ***12 bit***