



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**



**DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE**

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA ELETTRONICA

CONVERTITORI A/D SIGMA-DELTA

Relatore: Prof. Roberto Corvaja

Laureando: Gava Francesco

ANNO ACCADEMICO 2022 – 2023

Data di laurea 17 Luglio 2023

Indice

| | |
|--|-----------|
| Introduzione..... | 5 |
| 1 Fondamenti della Conversione analogico-digitale..... | 7 |
| 1.1 Campionamento | 7 |
| 1.2 Quantizzazione e codifica | 9 |
| 2 Architettura e funzionamento degli ADC Sigma Delta | 13 |
| 2.1 Modulazione a densità di impulso | 13 |
| 2.2 Modulatore Sigma-Delta..... | 14 |
| 2.3 Sovracampionamento..... | 17 |
| 2.4 Filtro Digitale e Decimatore | 18 |
| 2.5 Noise shaping..... | 19 |
| 3 Esempio applicativo: il circuito integrato ADuM7703 | 21 |
| 3.1 Considerazioni sui parametri di performance | 22 |
| 3.2 Applicazioni di sensing della corrente e della tensione | 24 |
| 3.3 Filtro digitale sinc3 | 26 |
| Conclusioni..... | 31 |
| Bibliografia | 33 |

Introduzione

Nell'ambito dello sviluppo tecnologico, la conversione analogico digitale è un processo importante per l'elaborazione dei segnali elettronici. Permette di interfacciare le grandezze analogiche con un sistema digitale, rappresentandole con serie di numeri in formato binario. Si adotta questo approccio per semplificare la memorizzazione e la trasmissione dei dati.

In questa tesi verrà fornita una panoramica sulla conversione A/D in generale, analizzando le fasi principali del processo. Ci sarà una spiegazione sull'architettura sigma-delta, una tecnologia che può sembrare complessa, ma offre numerosi vantaggi per la manipolazione dei segnali. Infine è presente un capitolo che analizza il funzionamento di un ADC reale fabbricato dalla *Analog Devices*.

1 Fondamenti della Conversione analogico-digitale

Il processo di conversione da segnale analogico a segnale in formato digitale avviene in tre fasi principali:

- campionamento
- quantizzazione
- codifica



Figura 1: fasi della conversione

1.1 Campionamento

È il procedimento che permette di acquisire in forma digitale un segnale in ingresso $x(t)$ a tempo continuo. Come già presentato in precedenza, un segnale a tempo continuo può variare in un tempo infinitesimo, perciò è impossibile memorizzare un segnale in modo perfetto usando un approccio digitale e discreto. Si decide di leggere il segnale solo in alcuni istanti, a intervalli definiti dal periodo di campionamento T_s . Quest'ultimo può essere espresso come l'inverso della frequenza di campionamento f_s .

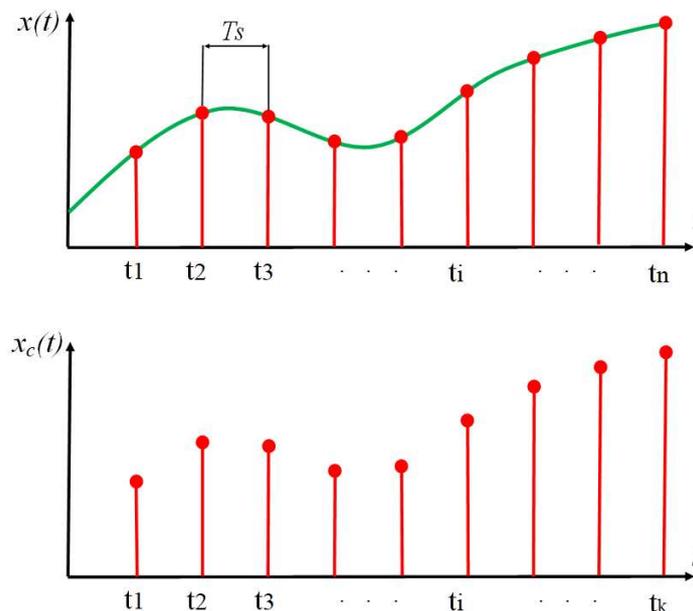


Figura 2: discretizzazione dei tempi

Il segnale $x_c(t)$ può essere scritto anche sotto forma di segnale discreto $x_c(kT_s)$. Si effettua una discretizzazione dei tempi e si definisce il segnale ottenuto come la moltiplicazione dell'ingresso con un treno di impulsi di periodo T_s , espresso come la ripetizione periodica della delta di Dirac.

$$x_c(t) = x(t) \cdot p(t) \quad \text{dove } p(t) = \sum_{k=-\infty}^{+\infty} \delta(t - kTs) \quad (1)$$

Ora l'obiettivo è calcolare la trasformata di Fourier di $x_c(t)$ per vedere come si comporta nel dominio della frequenza, allo scopo di decidere il criterio da seguire per la scelta della frequenza di campionamento.

Si scrive innanzitutto la serie di Fourier di $p(t)$:

$$p(t) = \frac{1}{Ts} \sum_{k=-\infty}^{+\infty} e^{i2\pi k \frac{t}{Ts}} = f_s \sum_{k=-\infty}^{+\infty} e^{i2\pi k t f_s}$$

$$P(f) = f_s \int_{-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} e^{i2\pi k t f_s} e^{-i2\pi k f t} dt = f_s \int_{-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} e^{i2\pi k t (f_s - f)} dt$$

Dato che la trasformata di un segnale modulato corrisponde alla traslazione in frequenza:

$$P(f) = f_s \sum_{k=-\infty}^{+\infty} \delta(f - k f_s) \quad (2)$$

Poiché la trasformata del prodotto $x(t) \cdot p(t)$ viene ottenuta mediante la convoluzione in frequenza $X(f) * P(f)$:

$$X_C(f) = f_s \sum_{k=-\infty}^{+\infty} X(f - k f_s) \quad (3)$$

Si ottiene la trasformata di Fourier del segnale campionato x_c che rappresenta il suo spettro in frequenza.

Dalla formula 3 si può dedurre che lo spettro X_C è composto dallo spettro X più altre repliche dello spettro di X centrati in f_s , come illustrato in figura 3. In questa trattazione si prendono in considerazione solo le frequenze positive, poiché le frequenze negative non esistono nella pratica.

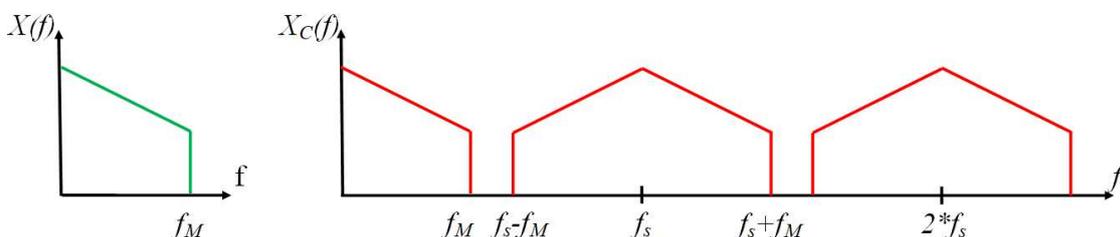


Figura 3: trasformata di fourier del segnale campionato

Bisogna assicurarsi che le repliche dello spettro iniziale non si sovrappongono perché questo effetto causerebbe l'*aliasing*, ovvero vengono prodotte frequenze che non appartengono al segnale di partenza e si verifica una distorsione di x .

Si stabilisce che f_M sia la frequenza massima di x . Guardando il grafico della figura 3, si può dedurre che per evitare l'*aliasing*, la frequenza di campionamento f_s deve essere almeno il doppio di f_M . Si enuncia nella formula 4 il teorema di Nyquist-Shannon.

$$f_s \geq 2 f_M \quad (4)$$

Tipicamente si aggiunge un filtro passa-basso, antecedente al campionatore, con una frequenza di taglio f_M . In questo modo si riesce a eliminare le componenti in alta frequenza indesiderate prima di effettuare il campionamento.

Il campionamento può essere rappresentato con un interruttore che viene chiuso per un istante relativamente breve, ogni T_s . Allo scopo di agevolare la conversione si può aggiungere un circuito che mantenga il segnale costante alla tensione campionata per tutto il periodo di campionamento, in modo da facilitare la lettura del valore operata dallo stadio successivo. Esso viene chiamato circuito di *sample & hold*, è composto da un condensatore che viene caricato quando l'interruttore è chiuso e da un buffer che garantisce semplicemente che il suo ingresso sia uguale alla sua uscita. Il buffer ha un'alta impedenza di ingresso e permette di isolare lo stadio del campionamento dallo stadio successivo senza che il segnale venga perturbato.

1.2 Quantizzazione e codifica

Dopo aver diviso il segnale in intervalli temporali, bisogna partizionare ogni campione ottenuto dal punto di vista del suo valore di tensione, compiendo una discretizzazione delle ampiezze. L'obiettivo della conversione ADC è quello di associare ad ogni campione di $x_c(t)$ un valore di tensione determinato $x_q(t)$, che poi viene riferito ad un numero digitale.

Tipicamente la tensione variabile dell'ingresso è delimitata in un range di tensione, chiamato ΔV_{FSR} che sta per *voltage full scale range*, può essere espresso anche definendo un massimo e un minimo e si ipotizza che il segnale sia sempre compreso tra questi due valori. La quantizzazione consiste nel suddividere il range in livelli detti livelli quantizzati, ogni campione fa riferimento al livello di tensione più vicino. Stabilendo che L indica il numero di livelli con cui si divide il segnale, conseguentemente si definisce anche il numero di bit n della parola prodotta dall'ADC, 2^n è il numero di parole binarie diverse che può assumere una parola di n bit, si avrà quindi:

$$L = 2^n \quad (5)$$

Utilizzando la seguente formula si può trovare quanto dista un livello quantizzato da un altro:

$$\frac{\Delta V_{FSR}}{L} = LSB \quad (6)$$

(*Least Significant Bit*) viene anche chiamato "quanto", e corrisponde al primo livello quantizzato di tensione; esso è rappresentabile con un numero binario che vale '1' e corrisponde appunto al bit meno significativo.

Dopo il processo di quantizzazione viene persa l'informazione di quanto distante si trovava il campione rispetto al suo livello quantizzato corrispondente, questo si chiama

errore di quantizzazione e_q e provoca una distorsione del segnale. L'errore di quantizzazione, detto anche rumore di quantizzazione, è un segnale aleatorio, un valore compreso tra $\pm \frac{LSB}{2}$, non dipende dalle caratteristiche dell'ingresso. Per ridurlo bisogna aumentare il numero di livelli; siccome la densità spettrale di potenza di questo rumore è costante nella banda di interesse, esso viene classificato come "rumore bianco".

In sostanza esso definisce quanto si differenziano il segnale campionato e il segnale quantizzato:

$$e_q = x_q(kTs) - x_c(kTs) \quad (7)$$

Tracciando la funzione di trasferimento ideale tra ingresso e codifica si ottiene l'errore di quantizzazione.

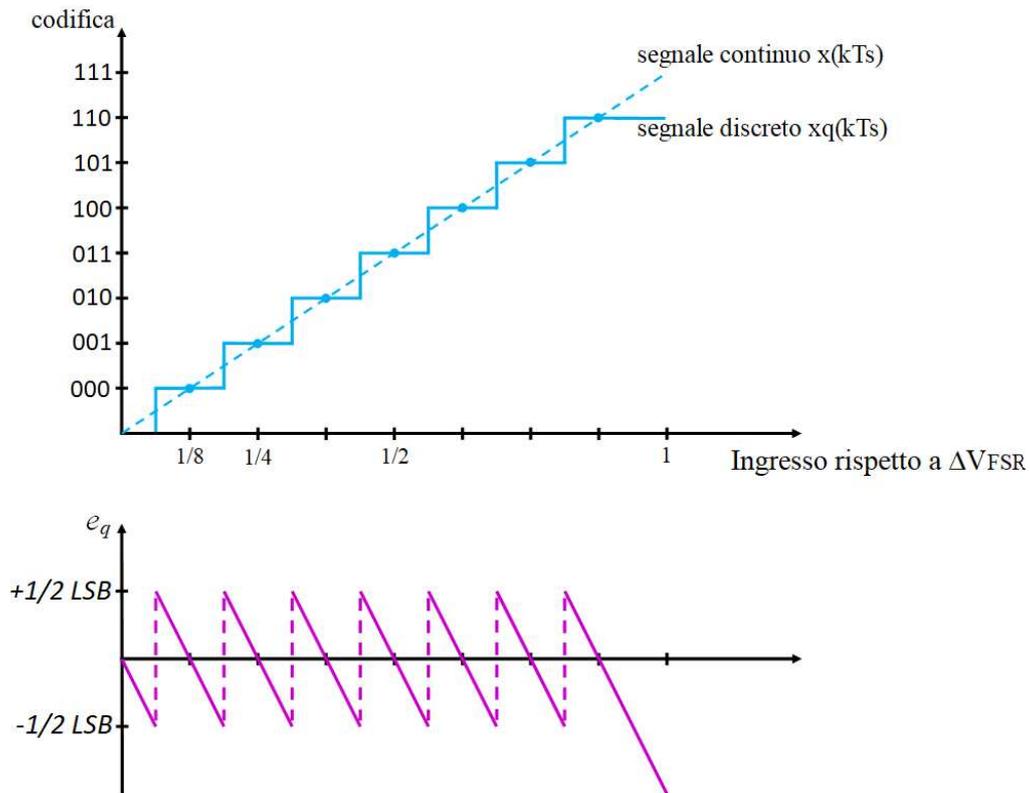


Figura 4: funzione di trasferimento ingresso/codifica ed errore di quantizzazione

Si introduce il rapporto segnale rumore SNR , un parametro quantitativo nell'elaborazione dei segnali che indica appunto il rapporto tra la potenza statistica del segnale utile e la potenza statistica del rumore indesiderato, cercando di massimizzare questo fattore.

$$SNR = \frac{Potenza[x]}{Potenza[e_q]} = \frac{M_x}{M_{e_q}} \quad (8)$$

Per semplicità si ipotizza un segnale x a media nulla, per questo si può scrivere che la potenza statistica è uguale alla varianza:

$$M_x = \sigma_x^2 \quad (9)$$

Si definisce il fattore di carico k_f che misura l'adattamento del segnale al quantizzatore:

$$k_f = \frac{\sigma_x}{\Delta v_{FSR}/2} \quad (10)$$

La potenza statistica diventa:

$$M_x = k_f^2 \cdot \frac{\Delta v_{FSR}^2}{4} \quad (11)$$

Per quanto riguarda l'errore di quantizzazione si deduce che la sua potenza statistica è distribuita uniformemente. Gli estremi sono $\pm \frac{LSB}{2}$ e l'ampiezza è $\frac{1}{LSB}$

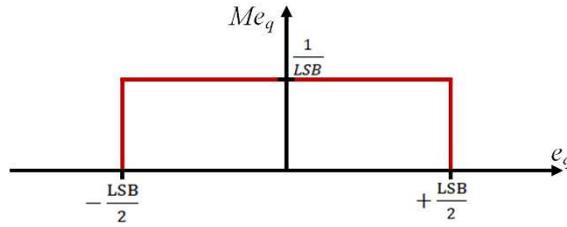


Figura 5: potenza statistica dell'errore di quantizzazione

$$M_{e_q} = \frac{1}{LSB} \int_{-\frac{LSB}{2}}^{+\frac{LSB}{2}} e_q^2 de_q = \frac{1}{LSB} \left[\frac{e_q^3}{3} \right]_{-\frac{LSB}{2}}^{+\frac{LSB}{2}} = \frac{LSB^2}{12} \quad (12)$$

Ora si può completare la formula 8:

$$SNR = \frac{M_x}{M_{e_q}} = \frac{k_f^2 \cdot \frac{\Delta v_{FSR}^2}{4}}{\frac{LSB^2}{12}} = \frac{k_f^2 \cdot \frac{L^2 LSB^2}{4}}{\frac{LSB^2}{12}} = 3 k_f^2 L^2 = 3 k_f^2 2^{2n} \quad (13)$$

Tipicamente si esprime il rapporto segnale-rumore in decibel come segue:

$$\begin{aligned} SNR[dB] &= 10 \log(SNR) = 10 \log(3) + 10 \log(k_f^2) + 10 \log(2^{2n}) \quad [dB] \\ &= 4,77 + 20 \log\left(\frac{\sigma_x}{\Delta v_{FSR}/2}\right) + 20 n \log(2) \quad [dB] \\ &= 4,77 + 20 \log\left(\frac{\sigma_x}{\Delta v_{FSR}/2}\right) + 6,02n \quad [dB] \end{aligned} \quad (14)$$

Questa formula si può semplificare, supponendo che il segnale si mantenga all'interno del range dell'ADC, si trascura quindi il fattore di carico:

$$SNR[dB] = 4,77 + 6,02n \quad [dB] \quad (15)$$

In ogni caso si nota che l' SNR aumenta, al crescere del numero di bit n , se si vuole migliorare l'accuratezza del convertitore analogico-digitale si deve massimizzare n .

Il rumore di quantizzazione non è l'unico difetto di un ADC. Esistono altri errori dovuti alla distorsione del segnale nel processo di elaborazione, tra questi possiamo citare gli errori causati dal clock, dalle interferenze elettromagnetiche esterne al circuito; inoltre

sono presenti errori di offset, di guadagno e di non linearità, che sono dovuti dal fatto che il processo di conversione non è ideale. Per quantificare la distorsione, si sceglie il parametro SINAD che è il rapporto tra la potenza del segnale ottenuto e la somma tra rumore e distorsione.

$$SINAD = \frac{Potenza[x] + Potenza[e_q] + Potenza[distorsione]}{Potenza[e_q] + Potenza[distorsione]} \quad (16)$$

Occorre precisare che è vantaggioso incrementare il numero di bit fino ad un certo limite. Potrebbe accadere che, se n è molto grande, genererà un rumore di quantizzazione molto basso, tale da essere sovrastato dal rumore e dalle distorsioni. Questo effetto causerebbe la perdita di informazione dei bit meno significativi, ed essi risulterebbero inutili. Conviene scegliere n pari a *ENOB* (*Effective Number Of Bits*). Questo parametro si ottiene ricavando n dalla formula 15, e sostituendo *SNR* con il parametro più completo *SINAD*.

$$ENOB = \frac{SINAD - 1,76dB}{6,02} \quad (17)$$

Aumentare i bit fa salire inevitabilmente i costi e la complessità del circuito, perciò in base alla tipologia dell'ADC, occorre scegliere accuratamente il numero di bit per trovare il giusto compromesso. Dopo aver campionato e quantizzato, c'è la fase di codifica, da cui si genera la parola binaria corrispondente al valore quantizzato. Questo processo avviene in modo differente, a seconda della tipologia del convertitore. Nei paragrafi a venire, si analizza nello specifico la conversione di tipo Sigma-Delta.

2 Architettura e funzionamento degli ADC Sigma Delta

Nell'ambito degli ADC, ha acquisito sempre più importanza l'architettura di tipo Sigma-Delta, inventata intorno agli anni '60, diffusa alla fine del secolo scorso grazie alle sue elevate prestazioni e ad una buona flessibilità di progettazione. Questi convertitori sono utilizzati in larga scala, ad esempio sono presenti in sistemi audio e in sensori di temperatura o pressione, permettono di ottenere un'eccellente risoluzione e precisione, senza eccedere in costi e spazi eccessivi per l'implementazione circuitale, a discapito di avere una frequenza di conversione relativamente bassa. Questa modalità di conversione si basa sui seguenti concetti che vanno analizzati con cura:

- Modulazione a densità di impulso
- Sovracampionamento
- Noise Shaping di quantizzazione
- Filtro Digitale
- Decimazione

Fanno parte dello schema elettrico dei semplici circuiti elettronici ottenuti con dei amplificatori operazionali:

- comparatori
- integratori
- sommatore

Come verrà spiegato nel dettaglio nel paragrafo del sovracampionamento, la frequenza di campionamento f_S non va fraintesa con la frequenza di lavoro di clock f_{CLK} , in quanto f_{CLK} è molto elevata ed è maggiore di f_S . In questa prima analisi non verrà presa in considerazione la frequenza f_S , supponendo che x sia un valore di tensione costante all'interno di T_S , invece si parlerà della frequenza del clock f_{CLK} .

2.1 Modulazione a densità di impulso

Prima di addentrarci nella spiegazione degli ADC sigma-delta, è importante comprendere il concetto teorico della PDM (*Pulse Density Modulation*). Supponiamo di avere come ingresso una tensione costante x che nel nostro caso è il segnale campionato, esso viene definito all'interno del range ΔV_{FSR} . Come output otteniamo y , un'onda quadra che possiamo trattare come una sequenza di bit sincronizzata mediante un clock ed è in funzione della tensione di ingresso x da convertire. La quantità di "1" del segnale seriale di uscita è direttamente proporzionale al valore della tensione di ingresso rispetto al suo ΔV_{FSR} .

Considerando M bits, che vengono prodotti in M periodi di clock, possiamo scrivere la relazione tra ingresso e uscita. A destra dell'uguale si esprime la posizione dell'ingresso rispetto al suo range e definendo il range d'ingresso tra x_{min} e x_{Max} , si ottiene un fattore compreso tra 0 e 1. A sinistra dell'uguale ci sarà il rapporto tra numero valori logici alti ($m1$) e i periodi di clock totali (M):

$$\frac{x - x_{min}}{x_{Max} - x_{min}} = \frac{m1}{M} \quad (18)$$

Si illustra la modulazione a densità di impulsi con un esempio. Si suppone che i limiti del range ΔV_{FSR} siano ± 4 V e l'ingresso sia 0,8 V. Si calcola in che posizione si trova l'ingresso rispetto a ΔV_{FSR} e si ottiene 0,6. Vuol dire che il 60% dei bit seriali che vengono generati sono degli "1" mentre i rimanenti 40% saranno "0". Se si considera $N = 20$ impulsi di clock, si ottengono 12 impulsi di valore logico alto, come si osserva in figura 6:

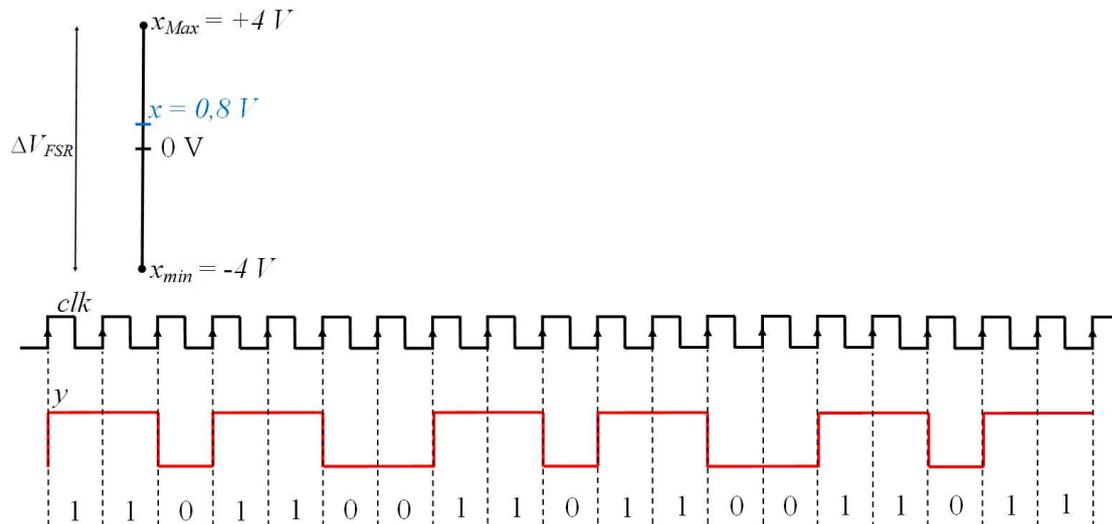


Figura 6: esempio modulazione sigma-delta

Finora è stato analizzato solo il comportamento dell'uscita in funzione dell'ingresso. Negli ADC sigma-delta, la modulazione a densità di impulsi viene eseguita dal modulatore sigma-delta. Spiegando il funzionamento di quest'ultimo si comprenderà il meccanismo per ottenere la PDM.

2.2 Modulatore Sigma-Delta

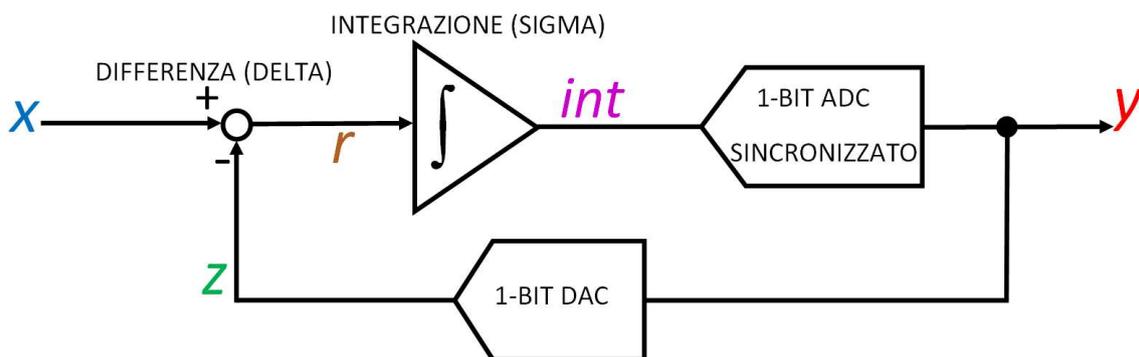


Figura 7: schema a blocchi del modulatore

Il modulatore sigma-delta rappresenta il componente fondamentale di un ADC sigma-delta. Il tempo per convertire un campione è T_s , si descrive cosa accade e come variano i segnali durante questo periodo. In figura 7 si descrive lo schema a blocchi, i segnali x e y corrispondono ai medesimi del paragrafo precedente. È un sistema ad anello chiuso, si comincia analizzando la retroazione.

Il segnale y digitale modulato a densità di impulsi va in ingresso ad un DAC (convertitore digitale analogico) a 1-bit; la sua uscita può assumere due valori diversi che coincidono con gli estremi del range di ingresso. Per semplicità si decide che essi siano uguali e opposti, si decide di chiamarli $\pm V_{ref}$. Viene generato z un altro segnale PDM, con l'unica differenza che gli "0" sono tradotti in $-V_{ref}$ e gli "1" diventano $+V_{ref}$.

Si può dire che y e z hanno lo stesso andamento, per questo anche il comportamento di z è strettamente legato al valore di tensione di x in ogni istante e siccome x e z hanno lo stesso range di tensione ci si può convincere che questi due segnali tendono ad avere lo stesso valore medio quando il sistema è in equilibrio. Successivamente viene eseguita una somma algebrica, l'onda quadra z viene sottratta al segnale di ingresso x , e sapendo che essi sono mediamente uguali, il risultato di questa differenza (r) è nullo in dc. Con il termine "media" si intende la media mobile di segnale calcolata in un periodo di conversione T_S mediante l'integrale della formula 19:

$$\bar{r}(t) = \bar{x}(t) - \bar{z}(t) = 0 \quad \text{dove in generale} \quad \bar{a}(t) = \int_{t-T_S}^t a(\tau) d\tau \quad (19)$$

Per capire come si ottiene questa stabilità occorre esaminare attentamente la circuiteria che parte dal nodo sommatore fino a produrre y . E' necessario prima illustrare i componenti rimanenti da cui è composto lo schema e poi verrà spiegato il meccanismo. Il nodo sommatore è incluso in un operazionale che funge da integratore, la sua uscita *int* può avere due pendenze diverse, dipende da quanto vale " $r = x - z$ " in ogni ciclo di clock. In base ai due casi che si possono verificare in ingresso per ogni istante generico t_1 o t_2 , si avrà:

1. $z(t_1) = -V_{ref} \quad r(t_2) = x(t_1) - z(t_1) > 0 \quad \Rightarrow \quad \text{crescente}$
2. $z(t_2) = +V_{ref} \quad r(t_2) = x(t_2) - z(t_2) < 0 \quad \Rightarrow \quad \text{discendente}$

Il segno di r stabilisce se il segnale *int* è crescente o decrescente ed è indipendente dal valore di x , perché x è sempre compreso tra $\pm V_{ref}$, perciò è z che definisce se r sia positivo o negativo. Il modulo di r invece, esprime la ripidità del segnale *int*. Si spiega così l'origine del nome sigma-delta: la lettera greca sigma fa intendere una somma e quindi l'integrazione mentre delta è un simbolo che ricorda l'operazione di differenza.

L'integratore viene collegato a un ADC a 1-bit, per realizzarlo è sufficiente adoperare un comparatore non invertente con soglia 0 V che produce un flusso di bit seriale. Quando rileva una tensione positiva genera un "1" e viceversa. In serie ci sarà un flip flop di tipo D che permette di sincronizzare quel flusso binario, ad ogni fronte di salita del clock. Il flip flop legge il bit e lo trasferisce in uscita, si ha ottenuto il segnale in uscita al modulatore sigma-delta chiamato y .

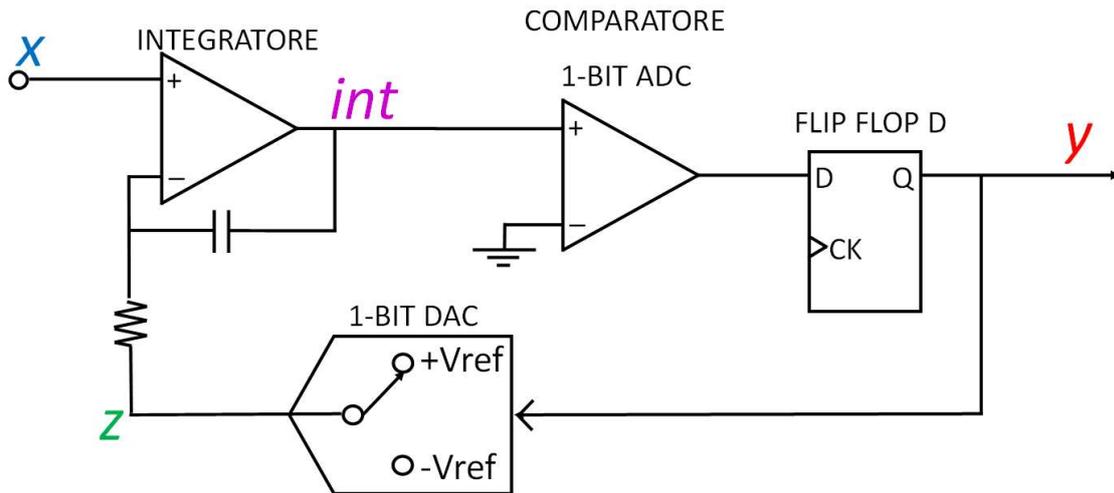


Figura 8: implementazione del modulatore sigma-delta

In figura 8 si mostra l'implementazione dello schema a blocchi della figura precedente. Si descrive il meccanismo che mantiene il sistema in equilibrio. Si suppone che la tensione in uscita all'integratore sia minore di 0, e il bit del segnale y sia 0, troveremo di conseguenza che z vale $-V_{ref}$. Si verifica il primo caso relativo all'integratore elencato nella formula 20 ovvero che r è positivo e la tensione in uscita all'integratore sarà una rampa positiva. In un istante a venire, questa tensione sale sopra gli 0 V della tensione di soglia del comparatore, quando questo accade, il comparatore e il flip flop D fanno in modo che al successivo ciclo di clock il bit di y sia a "1". A questo punto verrà retroazionato al nodo sommatore $z = +V_{ref}$, questa volta r sarà negativo e la pendenza del segnale int è negativa. In seguito avverrà che la tensione int ritorna ad essere inferiore alla soglia di massa e il bit di y diventa 0 come nella condizione iniziale. Si crea così un funzionamento ciclico. Osservando la forma d'onda int si può constatare che è un'onda triangolare, che oscilla attorno allo 0 V.

Si visualizza l'andamento dei segnali mediante degli esempi. Si considera inizialmente il caso di $x = 0$ V, ciò equivale a dire che l'ingresso si trova a metà fra del suo range ΔV_{FSR} e l'onda quadra y commuta il suo stato ad ogni ciclo di clock. Ne consegue quindi che $r = -V_{ref}$ ed $r = +V_{ref}$ alternativamente e il segnale int è caratterizzato da due pendenze uguali e opposte.

Se si ipotizza $x = +V_{ref}/2$, vuol dire che x si posiziona al 75% del range di ingresso ΔV_{FSR} e quindi $3/4$ dei bit sono a "1". Si calcola r nei due casi, la pendenza positiva è determinata da $r = +\frac{3}{2}V_{ref}$ mentre la pendenza negativa è decisa da $r = -\frac{1}{2}V_{ref}$. Si può notare che in modulo la pendenza è una il triplo dell'altra e poiché il segnale y alterna tre cicli alti e un ciclo basso, il sistema rimane in equilibrio.

Il comparatore controlla se il segnale int ha varcato la soglia. Si nota inoltre che il segnale int può commutare esclusivamente al fronte di salita del clock, questo si deve al flip flop D. Si mostrano gli esempi in figura 9:

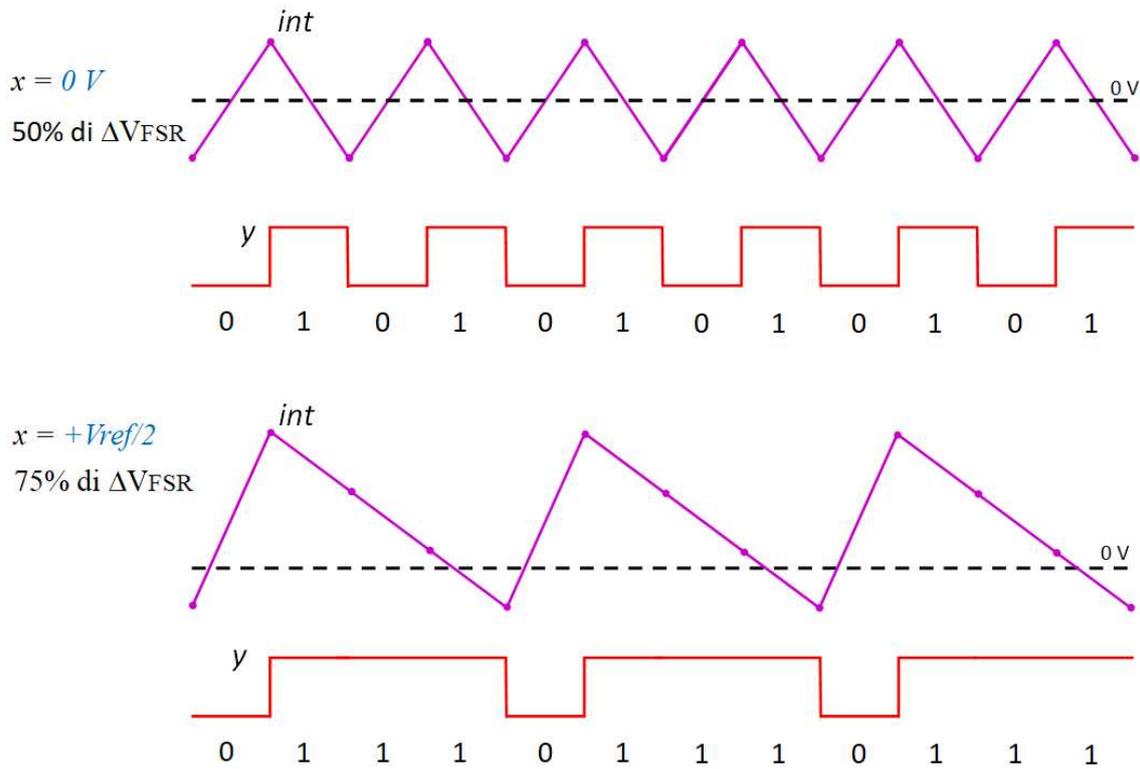


Figura 9: esempi di forme d'onda

2.3 Sovracampionamento

Precedentemente si è supposto il segnale ingresso x costante, tuttavia si deve tener conto che si tratta del segnale campionato $x_c(kT_s)$, e solo all'interno del periodo T_s si può considerare che l'ingresso sia una tensione costante. L'ADC Sigma-Delta sfrutta il principio di sovracampionamento, significa che la frequenza di lavoro del modulatore è molto più alta della frequenza f_s , esse sono correlate tra loro mediante il fattore K :

$$f_{CLK} = K \cdot f_s \quad \text{oppure} \quad T_s = K \cdot T_{CLK} \quad (21)$$

K viene detto fattore di sovracampionamento.

K è quindi il numero di cicli di clock T_{CLK} all'interno di un periodo di campionamento T_s . Ricordando che la quantità di bit logici alti di y definisce il valore di x , ora si può perfezionare la formula 22:

$$\frac{x(nT_s)}{\Delta V_{FSR}} = \frac{T_{CLK}}{T_s} \sum_{k=1}^K y(kT_{CLK}) \quad (22)$$

dove la sommatoria conta la quantità di "1" presenti in T_s .

Il rumore di quantizzazione è bianco, perciò non dipende dalla frequenza di campionamento f_s . La densità spettrale di potenza di un ADC comune (che non sia di tipo sigma-delta) calcolata per il rumore è illustrata in figura 10, dove si può notare che essa è distribuita equamente tra 0 e $\frac{f_s}{2}$. L'area sottesa dal grafico rappresenta la potenza

statistica di e_q che è $M_{e_q} = \frac{LSB^2}{12}$, come si è già visto nella formula 12. Quest'ultima equazione vale per qualsiasi tipologia di convertitore ed è indipendente dalla frequenza di campionamento. La potenza del rumore si calcola di conseguenza ed è pari a:

$$P_{e_q}(0) = \frac{M_{e_q}}{f_s/2} \quad (23)$$

Negli ADC sigma-delta invece si effettua il sovracampionamento, in questo caso il convertitore ADC a singolo bit legge dati alla frequenza di clock f_{CLK} che è K volte f_s . In quanto M_{e_q} rimane invariato, l'area rimane invariata, sarà il valore della potenza del rumore che diminuisce e vale :

$$P_{e_q}(0) = \frac{M_{e_q}}{f_{CLK}/2} \quad (24)$$

Si spiega così perché questi convertitori hanno una buona reiezione al rumore.

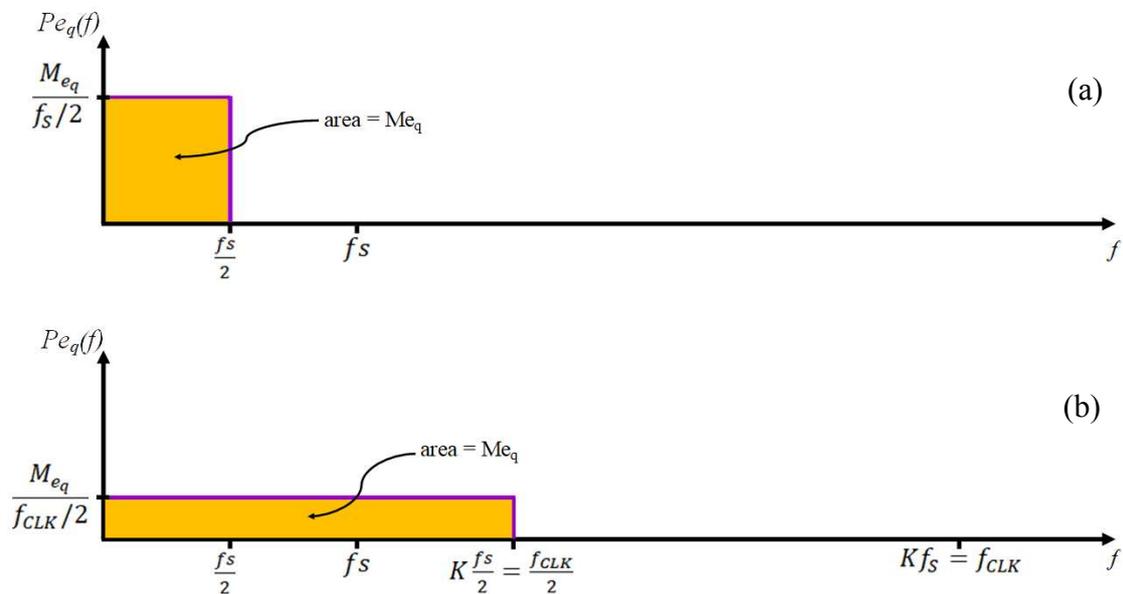


Figura 10: densità spettrale di rumore per ADC comune (a) e per ADC sigma-delta (b)

2.4 Filtro Digitale e Decimatore

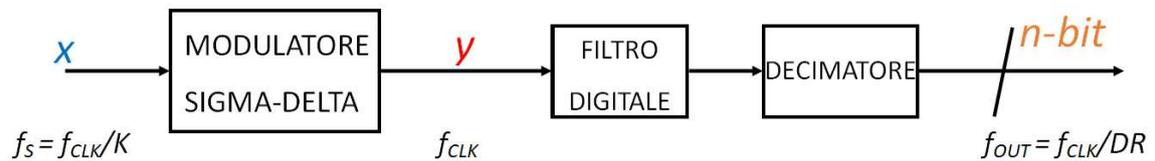


Figura 11: Schema a blocchi di un ADC sigma-delta

Si esaminano le due unità che seguono il modulatore sigma-delta, le quali consentono di estrarre l'informazione dal segnale PDM y che esprime i bit in modo seriale, in maniera tale da ottenere i bit in una parola binaria in parallelo, corrispondente all'uscita dell'ADC.

Si esamina ora il filtro digitale. L'informazione del segnale di partenza x è contenuta nel valor medio di y . Per estrarlo si impiega un filtro di tipo passa-basso a risposta impulsiva finita. Quest'ultimo elabora il suo ingresso y , in modo tale da mantenere il segnale utile ed eliminare le componenti frequenziali più alte.

È importante precisare che il sovracampionamento non contribuisce ad aumentare la velocità di trasmissione dei dati in uscita al sistema complessivo. Attraverso il decimatore verrà abbassata la frequenza. La velocità di trasmissione dei dati in uscita è decisa dal fattore DR *Decimation Rate*.

$$f_{OUT} = \frac{f_{CLK}}{DR} \quad (25)$$

DR = K è il massimo fattore per cui si può decimare il segnale y , per assicurarsi che il data rate dell'uscita sia almeno il doppio della banda del segnale di partenza.

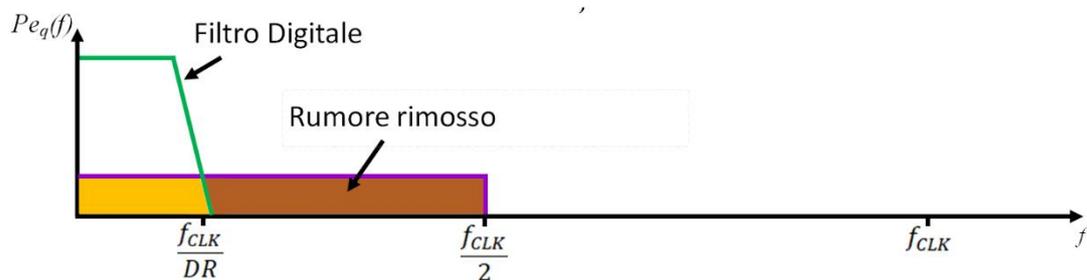


Figura 12: densità spettrale di rumore aggiungendo il filtro digitale

2.5 Noise shaping

Il *noise shaping* è una tecnica utilizzata dai convertitori sigma-delta che permette di ridurre il rumore, che finora nelle spiegazioni è stata omessa. È molto difficile spiegare come si comporta il rumore nel dominio del tempo, per questo si analizza il comportamento del rumore di quantizzazione nel dominio della frequenza per descrivere l'effetto del noise shaping. Verranno trattate le trasformate di Fourier dei segnali: x , y , ed e_q .

La parola “shaping” in inglese significa modellare. Il principio del *noise shaping* consiste nel modellare la risposta in frequenza del rumore di quantizzazione in modo favorevole.

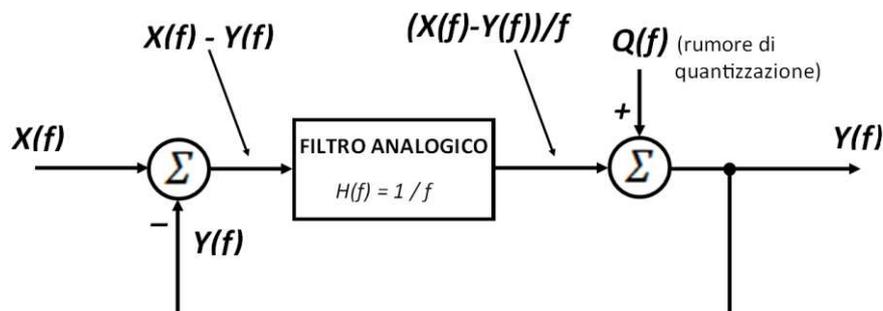


Figura 13: schema a blocchi del modulatore semplificato con trasformate dei segnali

Si semplifica lo schema a blocchi del modulatore, supponendo che il segnale y venga retroazionato, direttamente sul nodo di somma algebrica. Il comparatore che funge da ADC a 1 bit introduce rumore di quantizzazione, il quale viene iniettato tramite il feedback sul nodo sommatore. Si trova $X(f)-Y(f)$ in ingresso all'integratore che ha la caratteristica di comportarsi da filtro analogico passa basso per l'ingresso e quindi è inversamente proporzionale alla frequenza. Si esprime l'equazione osservando lo schema a blocchi e ricaviamo l'output:

$$Y(f) = \frac{1}{f} [X(f) - Y(f)] + Q(f)$$

$$Y(f) = \frac{X(f)}{f + 1} + \frac{Q(f) \cdot f}{f + 1} \quad (26)$$

È intuibile che se la frequenza tende a zero, il valore di $Y(f)$ si avvicina a $X(f)$, senza avere la componente del rumore di quantizzazione. Al contrario, all'aumentare della frequenza, $Y(f)$ è composto per la maggior parte da rumore:

$$\lim_{f \rightarrow 0} Y(f) = X(f) \quad \text{ad esempio} \quad Y(0) = X(0)$$

$$\lim_{f \rightarrow \infty} Y(f) = Q(f) \quad (27)$$

Il *noise shaping* è quindi operato dall'integratore, è un filtro analogico passa-basso per sua natura, rimane tale per il segnale utile, mentre agisce da filtro passa-alto per il rumore. Il filtro passa-basso digitale, che segue il modulatore, svolge un ulteriore filtraggio passa basso, come già visto in precedenza. Si ottiene così un segnale minimizzando il contributo del rumore di quantizzazione, come si può osservare guardando la densità spettrale di rumore nel grafico di figura 14.

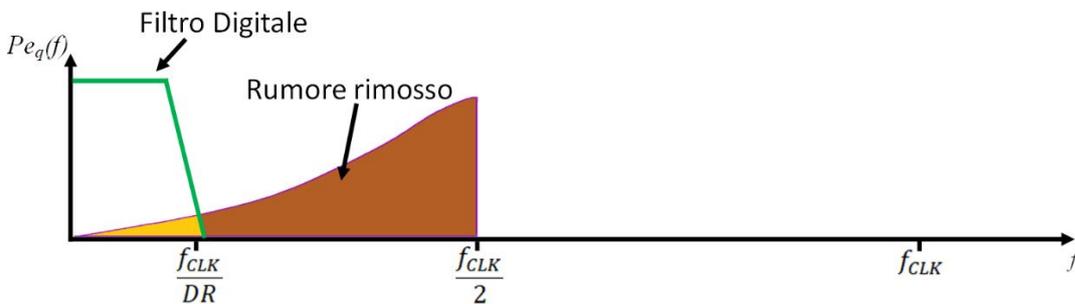


Figura 14: densità spettrale di rumore considerando anche il noise shaping

3 Esempio applicativo: il circuito integrato ADuM7703

In questo capitolo si descrive il funzionamento di un esempio applicativo: il circuito integrato ADuM7703 prodotto dalla multinazionale “Analog Devices”, in modo da capire anche dal punto di vista pratico l’argomento di questa tesina. Nella trattazione si enunciano le caratteristiche, accompagnate da una spiegazione teorica. Si analizzano i parametri di qualità, le imperfezioni e un’applicazione. Infine viene spiegato il funzionamento di un filtro digitale da aggiungere a questo integrato per ricostruire il segnale digitale convertito.

ADuM7703 è un modulatore A/D sigma-delta a 16 bit del secondo ordine ad alte prestazioni. L’integrato è fornito di un regolatore lineare di tensione *low dropout*, che permette all’uscita di avvicinarsi molto all’alimentazione. Possiede un ingresso pseudo differenziale (pin V_{IN+} e V_{IN-}), significa che un pin dell’ingresso ha una tensione fissa e l’altro pin può variare in quel range. Questo tipo di ingresso si adatta alle applicazioni di misura della corrente e tensione di shunt. L’ingresso viene campionato ad una frequenza di lavoro di 21 MHz, la sequenza di bit viene propagata ad ogni fronte di salita del clock.

Il pin del clock (MCLKIN) e il pin di uscita dell’ADC (MDAT) sono isolati galvanicamente grazie alla nuova tecnologia *iCoupler*, che integra un microtrasformatore a semiconduttore all’interno del chip. Questa tecnologia è più efficiente e affidabile del componente optoisolatore ormai obsoleto. Di conseguenza ci saranno due pin per l’alimentazione (pin V_{DD1} e V_{DD2}) dato che il circuito integrato è composto da due sezioni isolate tra di loro.

Un modulatore del secondo ordine ha 2 integratori in serie, l’uscita del dac viene retroazionata sia all’ingresso dell’adc, sia all’uscita del primo integratore. Permette di ottenere una risoluzione maggiorata rispetto ad un adc del primo ordine.

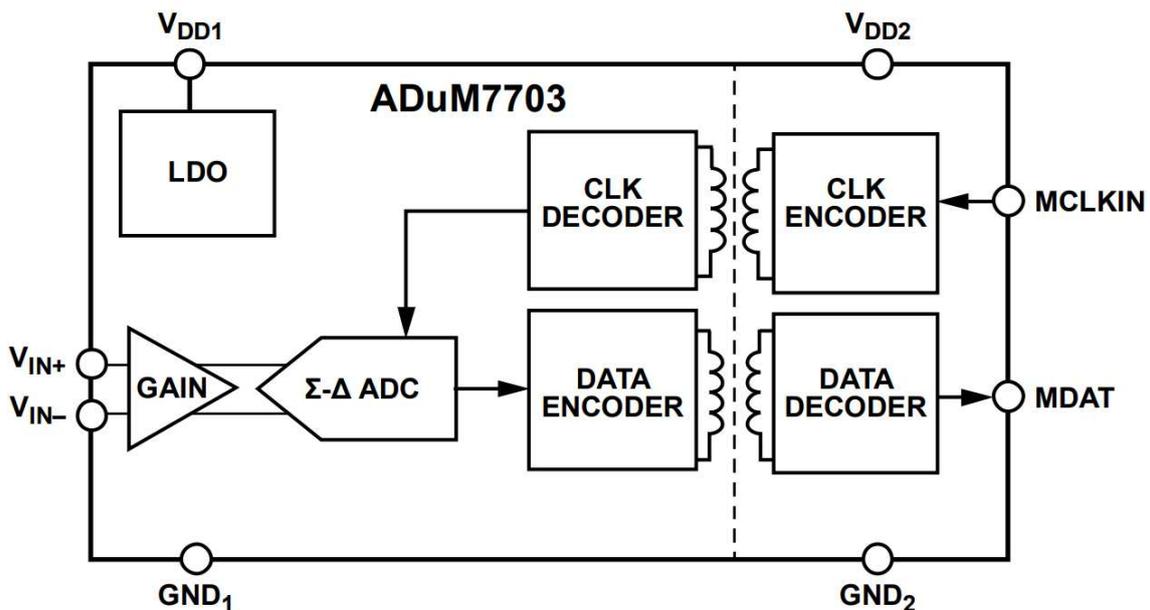


Figura 15: schema e piedinatura di ADuM7703

Il ΔV_{FSR} *voltage full scale range* è delimitato da ± 320 mV, si ha $\Delta V_{FSR} = 640$ mV e si può calcolare il least significant bit: $\frac{\Delta V_{FSR}}{2^{16}} = 9,8$ μ V, a -320 mV corrisponde alla codifica 0 e per $+320$ mV associa la codifica $2^{16} = 65536$. Tuttavia, l'ingresso è ammesso solo nel range specificato che viene limitato all'intervallo ± 250 mV, il minimo e il massimo vengono associati rispettivamente alle codifiche 7168 e 58368.

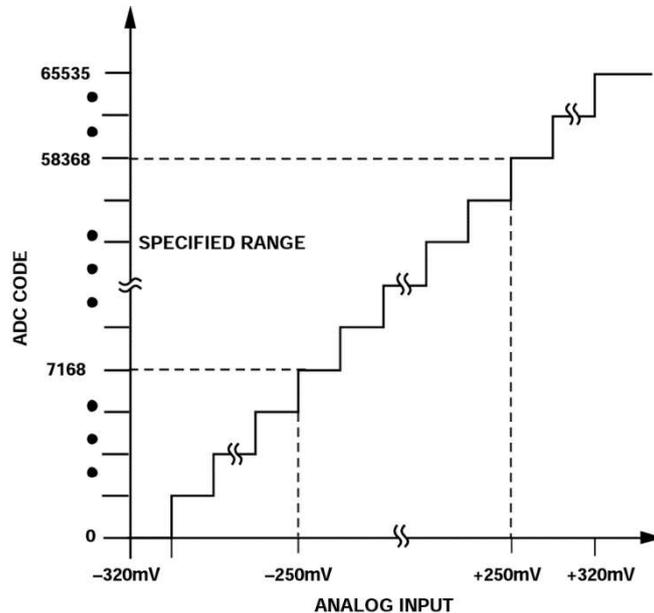


Figura 16: funzione di trasferimento dell'ADC a 16 bit

3.1 Considerazioni sui parametri di performance

Oltre all'errore di quantizzazione, si manifestano altri tipi di errore, che verranno discussi in questo paragrafo per capire i difetti di ADuM7703. Si inizia citando gli errori di non linearità, ne esistono di due tipi:

- non linearità differenziale (DNL)
- non linearità integrale (INL)

Come funzione di trasferimento ideale tra ingresso e uscita si fa riferimento alla retta ideale e teorica di figura 4. Per un ADC ideale, DNL e INL sono nulli.

L'errore DNL indica la discrepanza tra la larghezza reale dell' n -esimo intervallo e la larghezza teorica di 1 LSB. Per "larghezza reale" si intende la distanza tra due codifiche adiacenti. Si comprende facilmente questo tipo di errore guardando la larghezza di ogni step nel grafico di figura 17. Esso può essere positivo o negativo e si misura in "LSB". Il ADuM7703 garantisce che l'errore DNL massimo sia $\pm 0,99$ LSB. Significa che non verranno perse codifiche, infatti un errore superiore a 1 in modulo avrebbe causato la mancanza di un dato.

Si descrive ora l'errore INL. Si misura tracciando due curve che passano per i centri degli intervalli di quantizzazione considerando il caso reale e il caso ideale di figura 4.

L'errore integrale è la deviazione massima tra le due curve, come mostrato in figura 17. L'integrato in esame ha un INL massimo di ± 5 LSB.

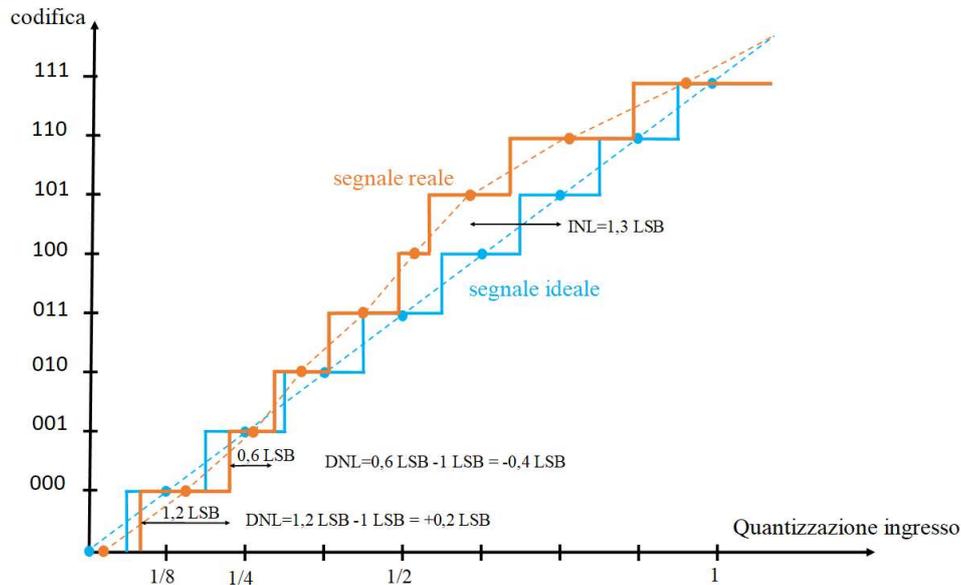


Figura 17: funzione di trasferimento reale ed errori DNL e INL

Sono presenti inoltre, altri due tipi di errore, i quali possono essere compensati mediante taratura:

- errore di offset
- errore di guadagno

L'input analogico corrispondente alla codifica intermedia $2^{15}=32768$ idealmente corrispondere a un input analogico di 0 V, la tensione dell'input che si ha per quella codifica è l'errore di offset.

Supponendo che l'errore di offset sia compensato, si può analizzare gli errori di guadagno positivo e negativo. Quello positivo misura di quanto discosta la tensione di ingresso dal valore $+V_{FSR}=+250\text{mV}$, previsto per la codifica massima (58368). Analogamente, l'errore di offset negativo è calcolato per la codifica minima (7168).

I parametri di offset e di guadagno dipendono dalla temperatura. Sperimentalmente si misura come variano nel range di temperatura stabilita. Otterremo un parametro che indica la deviazione di offset/guadagno con l'unità di misura ppm/°C (parti per milione per ogni grado centigrado).

Un altro parametro importante di un ADC è l'SNR descritto nel paragrafo 2.1. Tramite la formula 15 del paragrafo 1.2, si può affermare che l'SNR ideale è 98 dB. L'SNR reale è calcolato considerando vari tipi di rumore e assume un valore compreso tra 83 e 86 dB.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|-----|-------|-------|--------|---------------------------------------|
| STATIC PERFORMANCE | | | | | |
| Resolution | 16 | | | Bits | Filter output truncated to 16 bits |
| Integral Nonlinearity (INL) ¹ | | ±2 | ±5 | LSB | |
| Differential Nonlinearity (DNL) ¹ | | | ±0.99 | LSB | Guaranteed no missed codes to 16 bits |
| Offset Error ¹ | | ±0.05 | ±0.13 | mV | Initial at T _A = 25°C |
| | | ±0.1 | ±0.18 | mV | |
| Offset Drift vs. Temperature ¹ | | ±0.1 | ±0.6 | µV/°C | |
| Offset Drift vs. V _{DD1} | | ±0.1 | | µV/V | |
| Gain Error ¹ | | | ±0.2 | % FSR | Initial at T _A = 25°C |
| Gain Error Drift vs. Temperature ¹ | | ±12.5 | ±28 | ppm/°C | |
| | | ±8 | ±18 | µV/°C | |
| Gain Error Drift vs. V _{DD1} | | ±5 | | ppm/V | |

Figura 18: parametri di performance

3.2 Applicazioni di sensing della corrente e della tensione

L'ADC ADuM7703 viene utilizzato per misurare la corrente di un motore trifase a induzione. L'ingresso del convertitore rileva la tensione ai capi di una resistenza di shunt, che è attraversata dalla corrente che si desidera tenere sotto controllo. Tipicamente la resistenza di shunt (R_{SHUNT}) è una resistenza di valore molto basso, per non alterare la sua corrente.

Si conosce il picco di corrente I_{SENSE} del motore e si fa in modo che la tensione ai suoi capi sia adattata all'ingresso dell'ADC che è $\pm 250\text{mV}$. Si calcola la resistenza R_{SHUNT} tramite la legge di Ohm. Dato che la corrente è costante, la potenza dissipata dipende esclusivamente dal valore di resistenza. Una resistenza troppo alta potrebbe far sì che la potenza non rispetta il limite del resistore.

$$P_{DIS} = I_{RMS} \cdot R_{SHUNT} \quad (28)$$

Una resistenza troppo bassa invece, potrebbe compromettere l'accuratezza della misurazione. Bisogna trovare il giusto compromesso per una resistenza appropriata.

Occorre considerare anche i parametri dei motori, per la scelta di R_{SHUNT} :

$$I_{RMS} = \frac{P_W}{1.73 \times V \times EF \times PF} \quad (29)$$

I_{RMS} = corrente di fase del motore

P_W = potenza del motore

V = alimentazione del motore

EF = efficienza del motore

PF = efficienza di potenza

Nel grafico di figura 19 si evidenzia come variano gli $ENOB$ che stabiliscono la qualità del segnale digitalizzato; essi diminuiscono quando il segnale non occupa tutto il range dell'ADC.

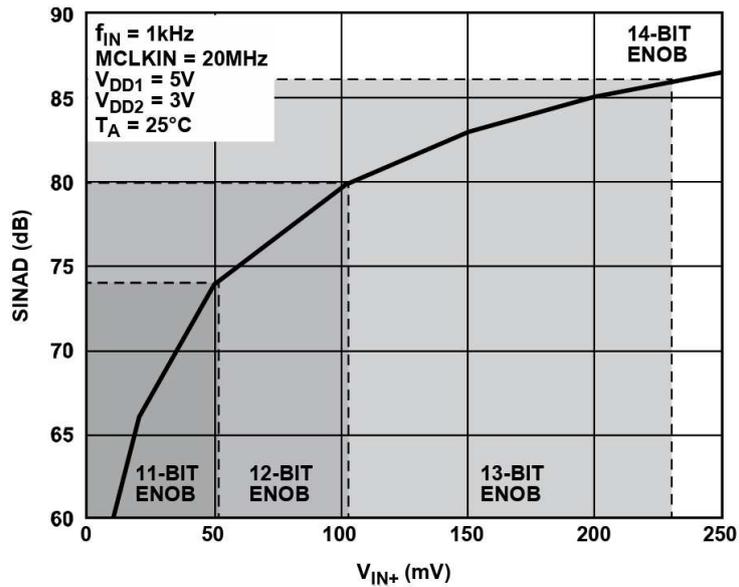


Figura 19: SINAD vs ingresso V_{in+}

Questo ADC può essere utilizzato anche per monitorare delle tensioni isolate. In applicazioni di controllo di motori, viene utilizzato per leggere il bus di tensione. Se la tensione supera il range dell'ingresso, è sufficiente aggiungere un partitore per dividere la tensione.

Tipicamente è presente un filtro analogico in ingresso. È appropriato inserire due semplici filtri passa-basso RC. Come valori si possono scegliere 10Ω e 220pF , per eliminare solo le frequenze elevate. Un'altra opzione è quella di scegliere un filtro RC differenziale.

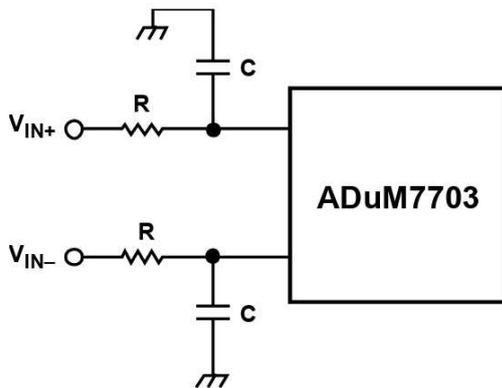


Figura 20: filtro passa-basso per ingresso

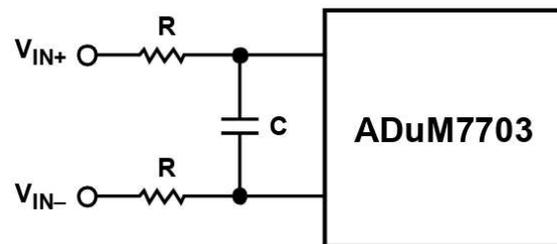


Figura 21: filtro differenziale per ingresso

3.3 Filtro digitale sinc3

L'integrato svolge solo la funzione di modulatore sigma-delta, generando il segnale y . Come descritto nel capitolo 2.4, per ottenere il segnale digitale convertito, il segnale y va elaborato eseguendo due processi:

- filtraggio passa basso, per eliminare le frequenze del rumore
- decimazione, per abbassare la frequenza, estraendo il valor medio

Nel caso di ADuM7703 si sceglie un filtro "sinc" o "CIC" che sta per *Cascade Integrator Comb*. Il sinc3 si implementa facilmente mediante un FPGA (*field programmable array processor*) o un DSP (*digital signal processor*) che sono dispositivi per l'elaborazione di segnali digitali. È necessario conoscere il funzionamento dello schema a blocchi del filtro integratore e del filtro comb nel dominio discreto Z .

L'integratore è un filtro che impiega un feedback. Viene sommato l'ingresso e l'uscita ritardata. Nel dominio Z , per indicare il ritardo di un campione si scrive z^{-1} .

$$b(n) = a(n) + b(n - 1)$$

$$H_i(z) = \frac{B(z)}{A(z)} = \frac{1}{1 - z^{-1}} \quad (30)$$

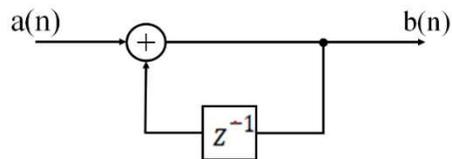


Figura 20: integratore nel dominio Z

Come filtro comb, si considererà la versione feedforward. Nel nodo sommatore, all'ingresso si sottrae l'ingresso ritardato di DR campioni.

$$b(n) = a(n) - a(n - DR)$$

$$H_c(z) = \frac{B(z)}{A(z)} = 1 - z^{-DR} \quad (31)$$

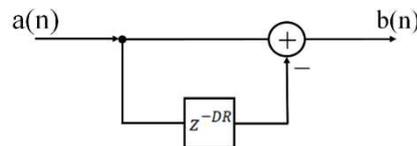


Figura 21: comb nel dominio Z

Il filtro sinc è un filtro FIR (*Finite Impulse Response*), ovvero ha una risposta finita all'impulso. Se in ingresso si immette un gradino, l'uscita impiega un tempo finito per assestarsi. È composto da una serie di tre componenti: un integratore, un sottocampionatore (che riduce la frequenza di un fattore DR *decimation rate*) e un comb, che vengono disposti in questo ordine.

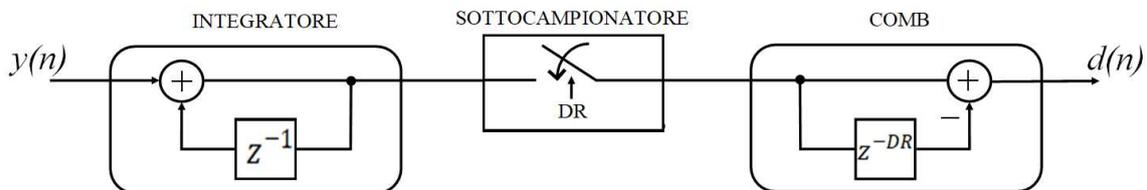


Figura 22: schema a blocchi del filtro sinc

La funzione di trasferimento del sistema discreto è:

$$H_s(z) = \frac{D(z)}{Y(z)} = \frac{1}{DR} H_i(z) H_c(z) = \frac{1}{DR} \frac{(1 - z^{-DR})}{(1 - z^{-1})} \quad (32)$$

Si tratta di un filtro a media mobile e si può dimostrare che questa configurazione riesce a ricavare il valore medio. Ricordando che $y(n)$ è il flusso di bit seriale in uscita al modulatore, si definisce $d(n)$ il segnale dei campioni filtrati.

La media mobile è la seguente:

$$d(n) = \frac{1}{DR} \sum_{m=0}^{DR-1} y(n-m) = \frac{1}{DR} [y(n) + y(n-1) + \dots + y(n-DR+1)] \quad (33)$$

Si scrive anche $d(n-1)$ per poter rielaborare $d(n)$:

$$d(n-1) = \frac{1}{DR} \sum_{m=1}^{DR} y(n-m) = \frac{1}{DR} [y(n-1) + y(n-2) + \dots + y(n-DR)]$$

Ora si può riscrivere $d(n)$ nel seguente modo:

$$d(n) = \frac{1}{DR} \cdot [y(n) - y(n-DR)] + d(n-1) \quad (34)$$

Si dimostra ora che la formula 34, si riferisce al filtro sinc:

$$d(n) - d(n-1) = \frac{1}{DR} \cdot [y(n) - y(n-DR)]$$

$$D(z) - D(z)z^{-1} = \frac{1}{DR} \cdot [Y(z) - Y(z)z^{-DR}]$$

$$\frac{D(z)}{Y(z)} = \frac{1}{DR} \frac{(1 - z^{-DR})}{(1 - z^{-1})}$$

Si è ottenuto la formula della funzione di trasferimento del sinc, enunciata nella formula 32. Si è dimostrato che un filtro sinc del primo ordine, permette di calcolare la media mobile di DR campioni del modulatore sigma-delta, dove DR è il fattore di sovracampionamento. Per via della decimazione, la velocità di trasmissione dei dati in uscita è:

$$f_{OUT} = \frac{f_{CLK}}{DR} \quad (35)$$

Siccome il modulatore di ADuM7703 è del secondo ordine, è necessario un filtro di ordine superiore a due. Viene scelto un sinc3, che è un filtro sinc del terzo ordine, il quale corrisponde a $sinc^3$. È composto da una serie di tre integratori, un decimatore e da tre comb. Gli integratori sono sincronizzati alla frequenza f_{CLK} , mentre i comb lavorano alla frequenza f_{OUT} .

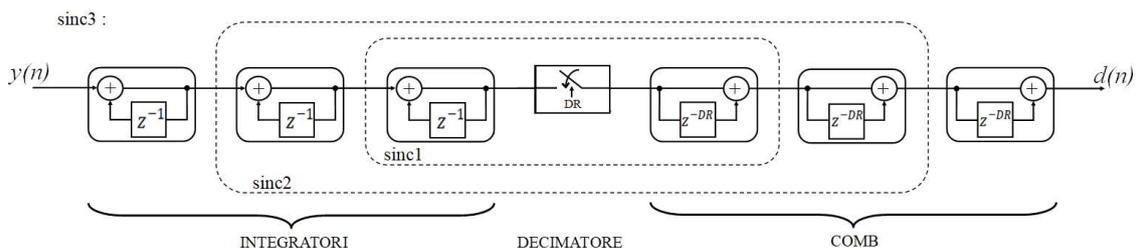


Figura 23: schema a blocchi di sinc3

Essendo un filtro, è opportuno calcolare la risposta in frequenza $H_s(f)$. Nel grafico si identifica la banda passante nella quale $H_s(f)$ vale 1 (0dB) e la banda oscura dove $H_s(f)$ tende ad annullarsi. Il filtro prende il nome di “sinc”, perché la sua risposta in frequenza $H_s(f)$ ricorda la forma del segnale sinc. $H_s(f)$ vale 1 per frequenze prossime a 0 Hz e si annulla per le frequenze a f_{OUT} e ai suoi multipli. La frequenza di taglio è la prima frequenza in cui $H_s(f)$ si annulla, ovvero f_{OUT} .

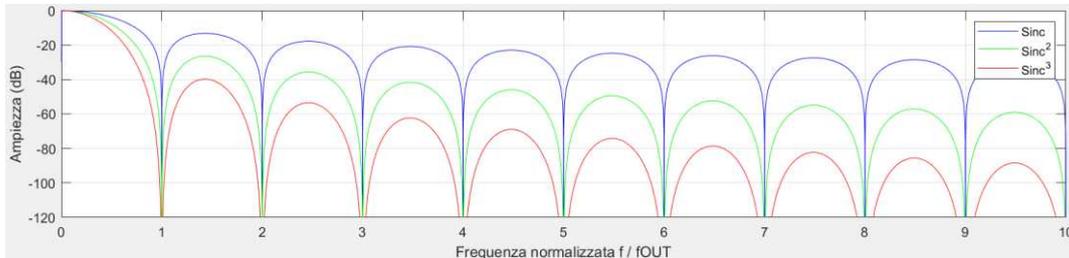


Figura 24: risposte in frequenza $H_s(f)$ di $sinc1$, $sinc2$, $sinc3$

Si può dire che il $sinc3$ ha al suo interno un filtro $sinc1$ e un filtro $sinc2$ come mostrato in figura 23.

$Sinc1$ fa la media di DR campioni, prodotti dal modulatore sigma-delta.

$Sinc2$ fa la media di DR campioni, prodotti dal $sinc1$.

$Sinc3$ fa la media di DR campioni, prodotti dal $sinc2$.

Il $sinc3$ esegue la media iterativamente per tre volte e questa operazione assicura una maggiore attenuazione del rumore nella banda oscura rispetto ad un $sinc1$, come si può notare in figura 24. Tale processo consente di ottenere una maggiore risoluzione (aumentano gli ENOB).

La risoluzione è in funzione anche del fattore di decimazione DR, in quanto aumentandola si abbassa f_{OUT} (che è la frequenza di taglio), come è raffigurato nella figura 25 si rende il filtro più selettivo, riducendo il rumore.

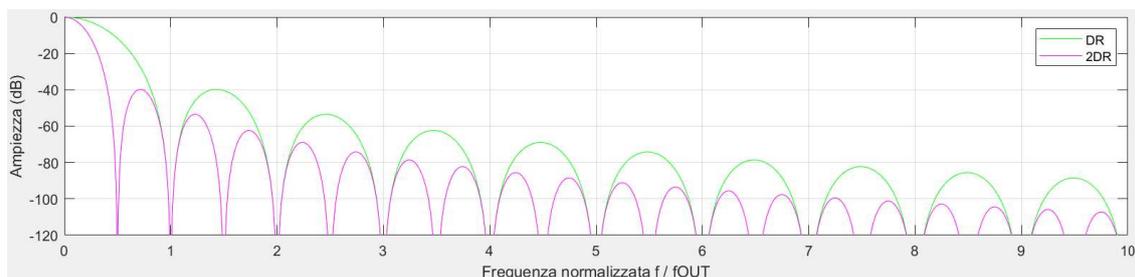


Figura 25: Risposta in frequenza $H_s(f)$ di $sinc3$ al variare di DR

La risoluzione corrisponde al un numero di bit in uscita, chiamato *data size* :

$$data\ size = N \cdot \log_2 DR \quad (36)$$

dove N è l'ordine del filtro sinc, in questo caso $N=3$.

L'aumento dell'ordine del sinc oppure l'aumento di DR ha anche un effetto negativo, prolunga il tempo di assestamento, anche detto tempo di latenza. Tipicamente si

preferisce tollerare questo ritardo, in quanto si sceglie il convertitore sigma-delta per via della sua alta risoluzione.

$$t_{latenza} = \frac{N}{f_{OUT}} = \frac{N \cdot DR}{f_{CLK}} \quad (37)$$

Nel dominio Z, la funzione di trasferimento di un sinc di ordine generico è la formula 32 elevata alla N:

$$H_s(z) = \left(\frac{1}{DR} \frac{(1 - z^{-DR})}{(1 - z^{-1})} \right)^N \quad (38)$$

Si può notare il contributo degli integratori al denominatore e il contributo dei comb al numeratore.

L'aumento di DR, che provoca la diminuzione del rumore, migliora l'SNR, come si può notare in figura 26.

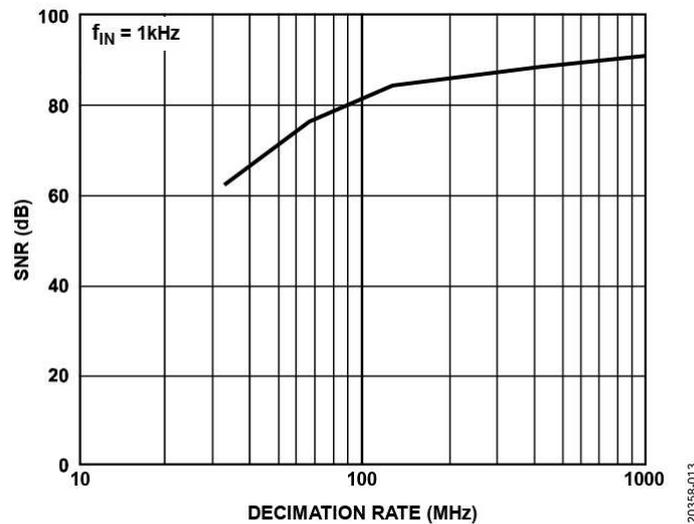


Figure 31. SNR vs. Decimation Rate of Sinc3 Filter Order

Figura 26: SNR al variare del Decimation Rate di un sinc3

Si descrive ora un esempio numerico, sapendo che la frequenza di clock è 20 MHz. Assumiamo DR=32, perciò verranno eseguite delle medie considerando 32 campioni.

Come frequenza di uscita si ottiene $f_{OUT} = \frac{f_{CLK}}{DR} = \frac{20 \text{ MHz}}{32} = 625 \text{ kHz}$.

I dati in uscita sono composti da $data \ size = N \cdot \log_2 DR = 3 \cdot \log_2 32 = 15 \text{ bit}$. Infine si può calcolare la risposta del filtro, il datasheet dice che è 0,262 volte f_{OUT} . Nella tabella di figura 27 si illustrano anche i casi con un DR più alto.

Table 14. Sinc3 Filter Characteristics for 20 MHz MCLKIN

| Decimation Ratio (DR) | Throughput Rate (kHz) | Output Data Size (Bits) | Filter Response (kHz) |
|-----------------------|-----------------------|-------------------------|-----------------------|
| 32 | 625 | 15 | 163.7 |
| 64 | 312.5 | 18 | 81.8 |
| 128 | 156.2 | 21 | 40.9 |
| 256 | 78.1 | 24 | 20.4 |
| 512 | 39.1 | 27 | 10.2 |

Figura 27: caratteristiche del filtro sinc3 per 20 MHz di clock

Conclusioni

Lo scopo di questa ricerca è stato quello di approfondire l'architettura sigma-delta, fornendo una descrizione dettagliata del principio di funzionamento. È importante avere chiari gli vantaggi e gli svantaggi, in modo da poter scegliere con competenza se è la soluzione più efficace per il caso specifico.

In questa analisi, si è evidenziato che l'applicazione dell'architettura sigma-delta, è conveniente quando risulta necessario avere un'alta risoluzione, una buona immunità al rumore e una discreta linearità.

Va tollerato che adottando questa tecnica si va incontro ad una limitazione di velocità di trasmissione che risulta più bassa per via del sovracampionamento rispetto ad altri tipi di convertitori.

Gli ADC sigma-delta sono contraddistinti dalla flessibilità dell'implementazione, essendo facili da modificare. Rendono possibile correggere la frequenza di clock e il fattore di decimazione, trovando il giusto compromesso tra velocità e risoluzione.

Bibliografia

- [1] <https://www.analog.com/media/en/training-seminars/tutorials/MT-022.pdf>
- [2] <https://www.analog.com/media/en/training-seminars/tutorials/MT-023.pdf>
- [3] <https://www.digikey.it/it/articles/analog-basics-part-2-delta-sigma-adcs-how-to-exploit-digital-capabilities>
- [4] https://it.wikipedia.org/wiki/Noise_shaping
- [5] <https://www.unisalento.it/documents/20152/1868258/Lezione+9+Teorema+del+campionamento.pdf>
- [6] https://ccrma.stanford.edu/~apinto/FST_6_campionamento.pdf
- [7] https://it.wikipedia.org/wiki/Pettine_di_Dirac
- [8] http://www3.deis.unibo.it/Staff/Research/CCaini/PDF/DispenseCE_quant.pdf
- [9] <https://tubaro.faculty.polimi.it/FENS/sigma-delta.pdf>
- [10] <https://classes.engr.oregonstate.edu/eecs/spring2017/ece627/Lecture%20Notes/First-Order%20D-S%20ADC.pdf>
- [11] http://www.ece.mcgill.ca/~grober4/TESTOSCILLATORS/DS_Modulation/Intro_DS_Modulation.html
- [12] https://digilander.libero.it/karmaproductions/unipa/el_n_digit_2/ErroriADC.pdf
- [13] https://it.wikipedia.org/wiki/Filtro_sinc
- [14] https://en.wikipedia.org/wiki/Cascaded_integrator%E2%80%93comb_filter
- [15] https://www.microst.it/Tutorial/sigmadelta_adc_4.htm
- [16] https://www.ti.com/lit/an/sbaa230a/sbaa230a.pdf?ts=1687065998112&ref_url=http%253A%252F%252Fwww.ti.com%252Fproduct%252FADS1220
- [17] <https://www.dsprelated.com/showarticle/1337.php>
- [18] Simone Buso, 2018, Introduzione alle applicazioni industriali di microprocessori e DSP
- [19] Datasheet Analog Devices: ADuM7703

