

CAPITOLO 2

ANALISI TEORICA DEL MHOMS

In questo capitolo verrà fornita una descrizione generale del sistema utilizzato, del quale nel capitolo tre sarà ottimizzato lo stadio di conversione A/D (analogico digitale). Nel sistema viene utilizzato il filtro CIC descritto nel capitolo precedente nell'ottica di un progetto per realizzare schemi di modulazione e codifica capaci di consentire collegamenti satellitari digitali ad alta velocità.

Tale progetto ha come risultato finale la realizzazione del MHOMS (Modem for High Order Modulation Schemes).

Vista la complessità del sistema, ci si basa su un'architettura semplificata del MHOMS come quella riportata in figura 2.1.

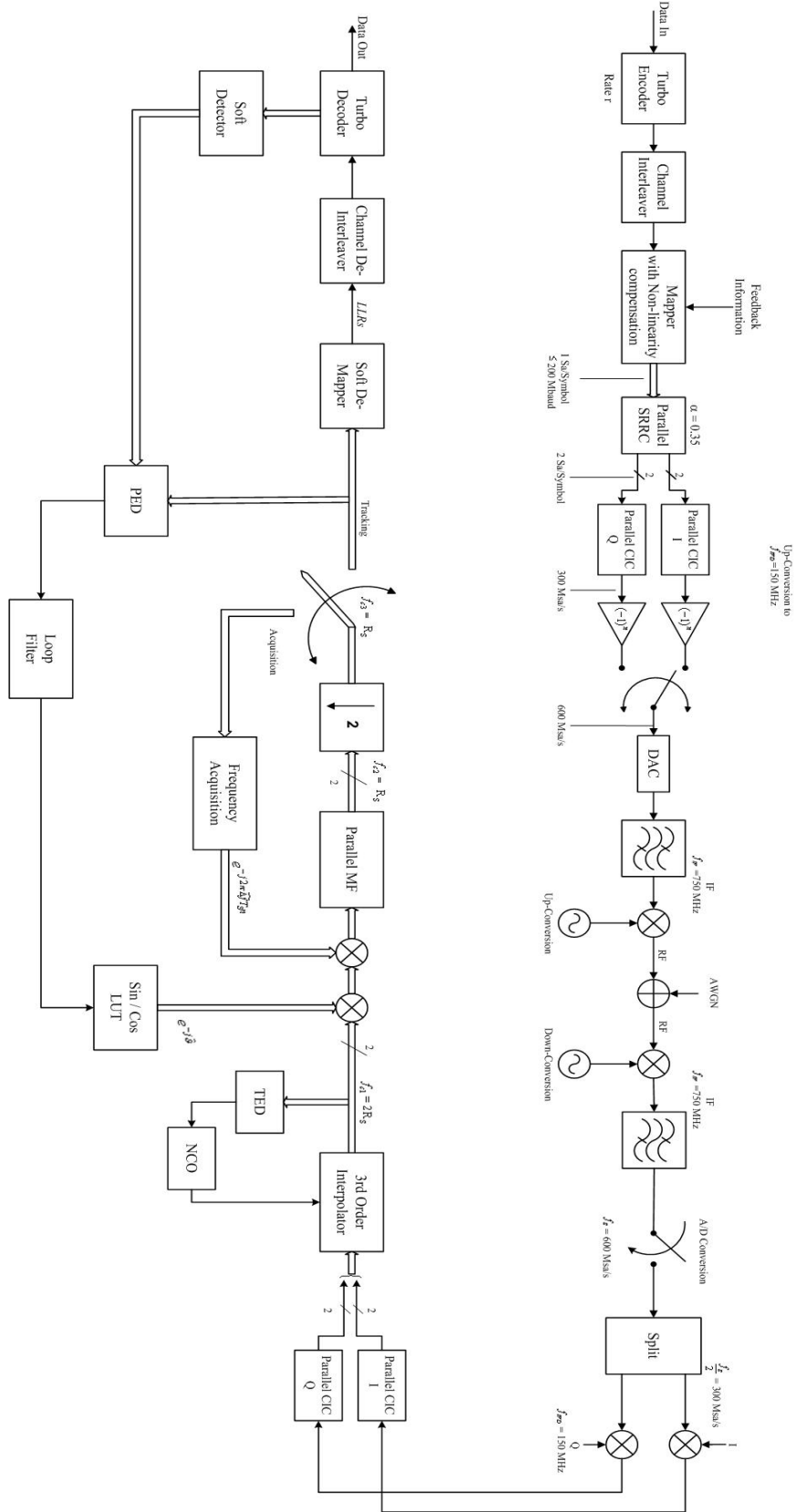


Figura 2.1-Architettura semplificata del modem MOHMS

Di tale architettura illustreremo gli elementi di maggiore interesse, descriveremo il modello di segnale utilizzato e ci soffermeremo sulla struttura del front-end del ricevitore con particolare attenzione alle problematiche dello stadio A/D, valutando come il sistema debba essere realizzato e dimensionato tenendo presente i limiti imposti dalla tecnologia utilizzata e le condizioni operative previste.

2.1 - TRASMISSIONE DEL SEGNALE

Al flusso di dati in ingresso al sistema viene applicata una codifica Turbo e dopo il rimescolamento dovuto all'interallacciatore (interleaver), il flusso di dati viene mappato su un' idonea costellazione (generalmente i dati vengono mappati attraverso una predistorsione per poter compensare le non linearità introdotte dall' amplificatore di potenza HPA a bordo del satellite).

La predistorsione adottata in trasmissione non modifica affatto le proprietà spettrali del segnale, mentre migliora le prestazioni del collegamento in termini di BER.

Il segnale ottenuto all'uscita del mappatore è un segnale digitale multilivello a valori complessi con un symbol rate $R_s \leq 200 \text{ Mbaud}$.

Tale segnale viene sagomato da un filtro digitale, che dovrebbe riuscire ad effettuare le necessarie operazioni ad una velocità pari a $2R_s$, ma ciò non è possibile poiché gli attuali limiti sulla velocità massima di elaborazione consentita nei circuiti DSP è pari a 300 MHz.

Per ovviare a tale problema è stata adottata un'architettura parallela del filtro sagomatore: questo fa sì che si possa elaborare i dati su ciascun ramo ad un rate compatibile con gli attuali standard dell' hardware utilizzato.

Per potere infine ottenere una frequenza di clock del segnale pari a 600 MSample/s i due rami in uscita dal filtro sagomatore vengono interpolati grazie a filtri CIC utilizzati in un'architettura parallela, in modo che ciascun filtro operi su un solo ramo e produca in uscita un segnale con frequenza di clock pari a 300 MSample/s .

Entrambi i rami vengono poi concatenati ed il flusso risultante, alla frequenza di 600 MSample/s , dopo essere stato campionato utilizzando una frequenza di campionamento pari a $f_{sa} = 600 \text{ MHz}$ nella conversione D/A (digitale analogico), viene portato alla frequenza

intermedia IF (pari a 750 MHz) attraverso una conversione di frequenza digitale che utilizza una frequenza intermedia digitale $f_{IFD}=150$ MHz.

Tale valore, è stato opportunamente scelto in modo da semplificare l'operazione di up-conversion necessaria per portare il segnale a radiofrequenza (RF) in modo da poterlo infine trasmettere.

Il segnale a radiofrequenza che viene così inviato sul canale risulta:

$$s_{RF}(t) = \text{Re}\{s(t) e^{j2p f_{RF} t}\} = s_I(t) \cos(2p f_{RF} t) - s_Q(t) \sin(2p f_{RF} t) \quad (2.1)$$

con:

$$s(t) = s_I(t) + j s_Q(t) = \sum_{i=-\infty}^{+\infty} c_i g(t - iT_s) \quad (2.2)$$

dove T_s rappresenta l'intervallo di simbolo, c_i il generico simbolo della costellazione impiegata e $g(t)$ rappresenta l'impulso usato in trasmissione, impulso che ha un sagomatura a radice di coseno rialzato (SRRC, Square Root Raised Cosine) con roll-off \mathbf{a} .

Sotto tali ipotesi la banda occupata a radio frequenza dal segnale $s_{RF}(t)$ risulterà quindi pari a:

$$B=(1+\mathbf{a})R_s. \quad (2.3)$$

Nel dominio del tempo gli impulsi a radice di coseno rialzato hanno come espressione analitica la seguente formula:

$$g(t) = \frac{1}{p} \frac{\sin\left[p(1-\mathbf{a})\frac{t}{T_s}\right] + 4\mathbf{a} \frac{t}{T_s} \cos\left[p(1+\mathbf{a})\frac{t}{T_s}\right]}{\frac{t}{T_s} \left[1 - \left(\frac{4\mathbf{a}t}{T_s}\right)^2\right]} \quad (2.4)$$

dove \mathbf{a} è il fattore di roll-off, che nel nostro caso si assume variabile tra 0.20 e 0.35.

Il grafico dell'espressione (2.4) è rappresentato nella figura 2.2:

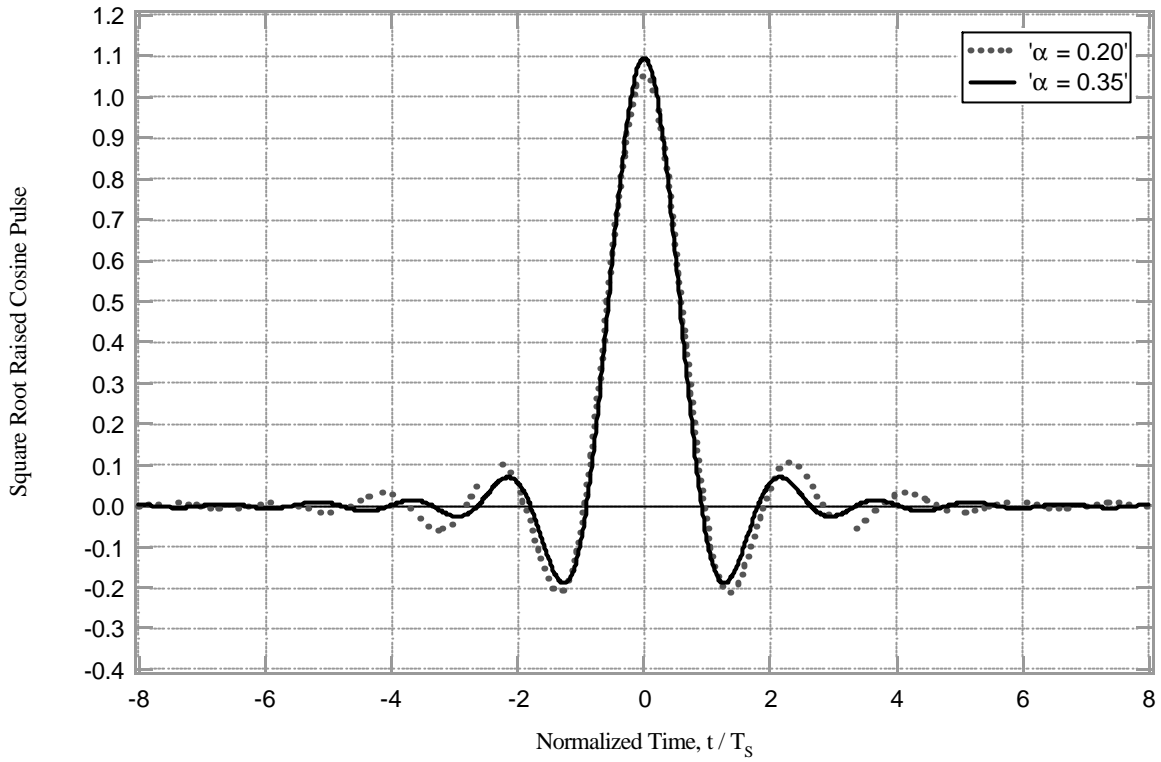


Figura 2.2-Impulso SRRC nel dominio del tempo

Invece nel dominio della frequenza gli impulsi a radice di coseno rialzato hanno come espressione analitica la seguente formula:

$$G(f) = \begin{cases} \sqrt{T_s} & |f| < \frac{1-a}{2T_s} \\ \sqrt{T_s} \cos \left[\frac{\pi}{4a} (2fT_s - 1 + a) \right] & \frac{1-a}{2T_s} \leq |f| \leq \frac{1+a}{2T_s} \\ 0 & |f| > \frac{1+a}{2T_s} \end{cases} \quad (2.5)$$

Il grafico dell'espressione (2.5) è rappresentato nella figura 2.3:

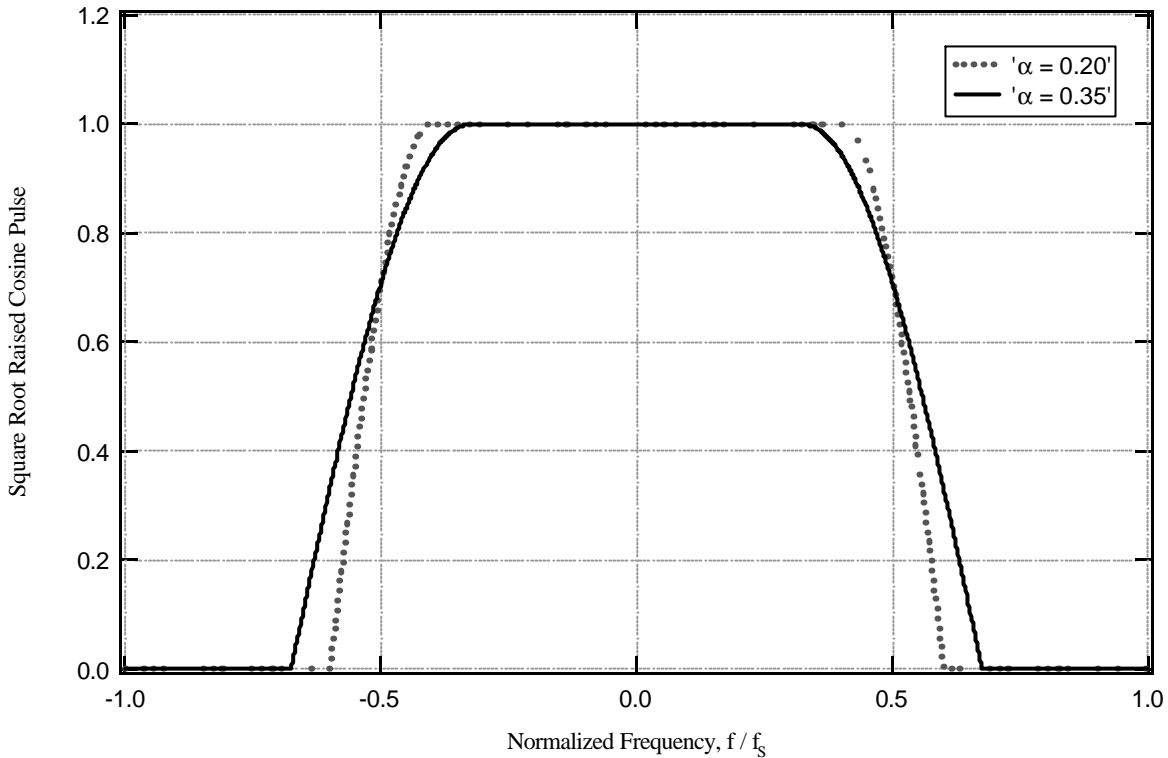


Figura 2.3-Impulso SRRC nel dominio della frequenza

Abbiamo finora descritto in termini generali il modello del segnale trasmesso; nei prossimi paragrafi si passerà alla descrizione del segnale ricevuto attraverso il canale, non tralasciando di descrivere l'architettura della parte in ricezione, con particolare attenzione allo stadio di conversione A/D ed alla struttura del front-end del ricevitore, tenendo conto della tecnologia impiegata e valutando le prestazioni del sistema.

2.2 - RICEZIONE DEL SEGNALE

2.2.1 – Stadio di down-conversion a IF e filtraggio passabasso

Il segnale a radiofrequenza che viene ricevuto risulta:

$$r_{RF}(t) = \text{Re}\{s(t-t)e^{j2\pi f_{RF}(t-t)}\} + w_{RF}(t) \quad (2.6)$$

dove f_{RF} è la frequenza della portante e t è il ritardo di propagazione che viene introdotto dal canale di trasmissione.

Assumendo come modello per il rumore che il canale introduca rumore AWGN (Additive White Gaussian Noise), allora $w_{RF}(t)$ rappresenta un processo di rumore additivo gaussiano bianco con densità spettrale di potenza bilatera pari a $N_0/2$.

Il segnale ricevuto $r_{RF}(t)$ viene inanzitutto convertito dalla radiofrequenza alla frequenza intermedia $f_{IF}=750$ MHz attraverso un'operazione di down-conversion, dopodichè viene elaborato da un filtro passabanda ideale $H_{FE}(f)$ necessario per rimuovere le componenti di rumore esterne alla banda utile, come si può vedere dalla figura 2.4.

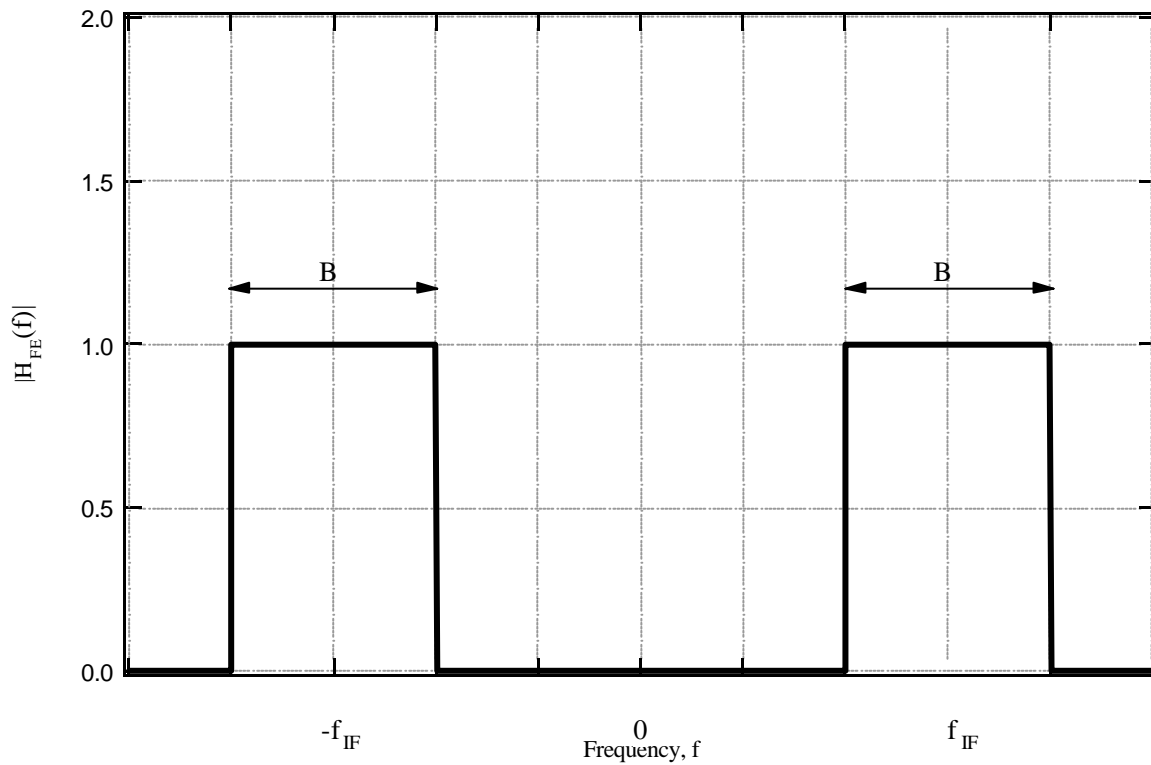


Figura 2.4 Risposta in frequenza del filtro di front-end ideale

Il segnale che si trova all'uscita del filtro $H_{FE}(f)$ risulta:

$$r_{IF}(t) = \text{Re}\{s(t-t)e^{j2\pi f_{IF}(t-t)}\} + w(t) \tag{2.7}$$

dove $w(t)$ rappresenta un processo gaussiano bianco sulla banda del filtro $H_{FE}(f)$.

2.2.2 – Stadio di conversione A/D

Lo stadio di conversione A/D (analogico digitale) deve convertire il segnale analogico $r_{IF}(t)$ espresso dalla (2.7) in un flusso di campioni, ossia deve digitalizzare il segnale in modo che questo possa poi essere elaborato in forma numerica, con notevoli vantaggi.

Allora per ottenere tale risultato lo stadio A/D deve effettuare contemporaneamente due distinte operazioni:

- il campionamento, ossia la discretizzazione dei tempi
- la quantizzazione, ossia la discretizzazione delle ampiezze del segnale

Possiamo per comodità teorica pensare le due operazioni separate, come mostrato in figura 2.5.

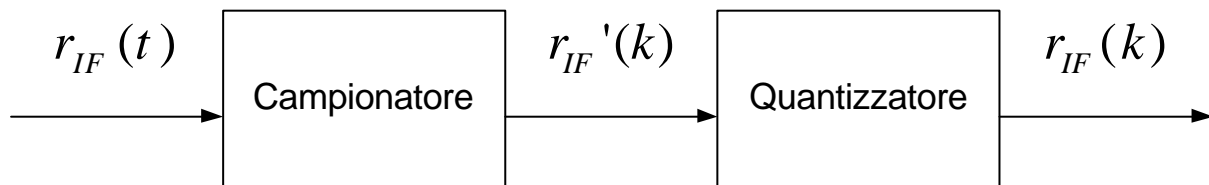


Figura 2.5-Lo stadio A/D

Il campionamento del segnale consiste nell'estrazione dei valori assunti dal segnale in istanti di tempo equispaziati e viene effettuato alla stessa frequenza di campionamento usata anche in trasmissione $f_{sa} = 600$ MHz, con conseguente replica spettrale del segnale come vedremo avanti.

La quantizzazione consiste nel dividere l'intervallo di ampiezze del segnale (dinamica del segnale) in intervalli di quantizzazione in modo che tutti i valori di ampiezza appartenenti allo stesso intervallo vengano rappresentati con un unico livello di quantizzazione, rappresentato dal valore medio dell'intervallo stesso.

La quantizzazione viene effettuata dal quantizzatore secondo la caratteristica rappresentata in figura 2.6, dove si possono riscontrare i due parametri fondamentali che descrivono il quantizzatore:

- il passo di quantizzazione Δ , ossia la larghezza dell'intervallo dei valori in ingresso che vengono quantizzati con lo stesso valore in uscita.
- il numero degli intervalli di quantizzazione M_Q .

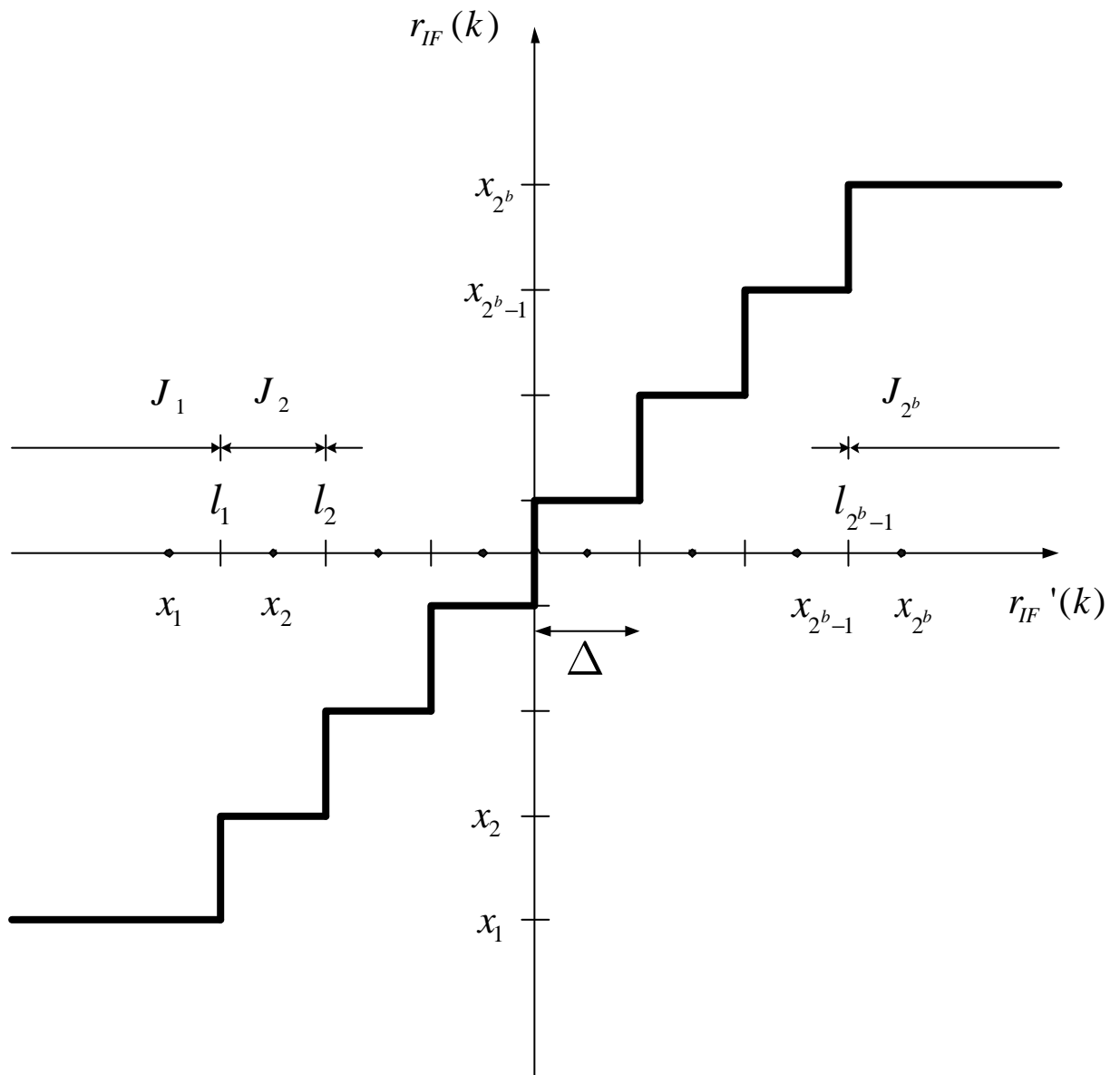


Figura 2.6-Caratteristica del quantizzatore

Per un convertitore A/D l'accuratezza è stimabile dalla differenza tra il valore dell'uscita (digitale) ed il valore dell'ingresso (analogico), differenza detta appunto *errore di quantizzazione*.

Il processo di quantizzazione comporta quindi un errore intrinseco per il sistema, come vedremo nel paragrafo 3.2.

Nel grafico si possono infine notare una regione granulare, quella centrale, dove l'ampiezza del segnale d'ingresso viene approssimata con un errore di quantizzazione non superiore a $\frac{\Delta}{2}$, ed una regione di saturazione o *overloading* dove l'ampiezza del segnale d'ingresso supera in valore assoluto il massimo livello di quantizzazione $x_{\max} = x_{2^b}$ e quindi si ottengono approssimazioni peggiori proprio a causa della saturazione.

Vedremo nel capitolo 3 come sia fondamentale minimizzare l'errore di quantizzazione.

Entrambi i parametri Δ e M_Q sono legati al numero di bit del registro del quantizzatore b , ossia il numero di bit su cui viene rappresentato il valore del campione.

Gli M_Q livelli di quantizzazione vengono rappresentati con stringhe di bit, per cui dati b bits è possibile rappresentare un numero di livelli dato da:

$$M_Q = 2^b \tag{2.8}$$

Al crescere del numero M_Q degli intervalli di quantizzazione aumenta quindi la precisione di un quantizzatore, fissata la dinamica del segnale, ma un aumento di M_Q comporta un aumento di b , ossia un utilizzo di registri molto lunghi con conseguente appesantimento della velocità di calcolo ed un notevole ingombro della memoria.

Si dovrà allora cercare il minimo valore di b che permetta di soddisfare alcuni requisiti in termini di prestazioni del sistema ed al tempo stesso minimizzi la degradazione introdotta dall'operazione di quantizzazione.

Se inoltre si indicano con x_{\min} e con x_{\max} rispettivamente i valori di ampiezza minimo e massimo assunti dalla soglia del quantizzatore, allora i livelli del quantizzatore possono essere definiti, attraverso semplici considerazioni geometriche, come:

$$x_i \triangleq x_{\min} + (i-1)\Delta \quad \text{per } 1 \leq i \leq 2^b \quad (2.9)$$

Le soglie degli intervalli di quantizzazione possono allora essere definite come:

$$l_i \triangleq x_{\min} + (i-1)\Delta + \frac{\Delta}{2} \quad \text{per } 1 \leq i \leq 2^b - 1 \quad (2.10)$$

e se inoltre vengono definite:

$$\begin{aligned} l_0 &\triangleq -\infty \\ l_{2^b} &\triangleq +\infty \end{aligned} \quad (2.11)$$

si ottiene che si possono definire gli intervalli di quantizzazione come:

$$J_i \triangleq (l_{i-1}, l_i] \quad \text{per } 1 \leq i \leq 2^b \quad (2.12)$$

ed allora la caratteristica del quantizzatore risulta completamente specificata da:

$$y(r) = \begin{cases} x_1, & r \in J_1 \\ x_2, & r \in J_2 \\ \dots & \dots \\ x_{2^b}, & r \in J_{2^b} \end{cases} \quad (2.13)$$

Ovviamente dalla (2.9) si ha che:

$$x_1 \equiv x_{\min} \quad (2.14)$$

e che:

$$x_{2^b} \equiv x_{\max} = x_{\min} + (2^b - 1)\Delta \quad (2.15)$$

per cui si ottiene facilmente che il passo di quantizzazione del quantizzatore si può calcolare come:

$$\Delta = \frac{x_{\max} - x_{\min}}{2^b - 1} \quad (2.16)$$

Nel nostro caso, inoltre, il quantizzatore è simmetrico, ossia

$$x_{\max} = -x_{\min} \quad (2.17)$$

per cui sostituendo la (2.17) nella (2.16) si ottiene:

$$\Delta = \frac{2x_{\max}}{2^b - 1} \quad (2.18)$$

Dalla (2.18) risulta evidente che il passo di quantizzazione risulta determinato esclusivamente dal numero di bit del registro del quantizzatore b e dal massimo valore assunto dalla soglia di quantizzazione x_{\max} .

Sarà su questi due parametri che si baseranno le simulazioni effettuate nel capitolo 3 per dimensionare il quantizzatore inserito nel sistema del MHOMS, dove andremo a valutare l'impatto della quantizzazione del segnale sui risultati ottenuti in termini di BER rispetto ad un sistema ideale.

Come esplicitato dalla (2.3), la banda del segnale ovviamente risulta variabile sia con \mathbf{a} che con R_s , per cui in figura 2.7 si riportano gli spettri del segnale all'uscita del filtro $H_{FE}(f)$ nei casi limite.

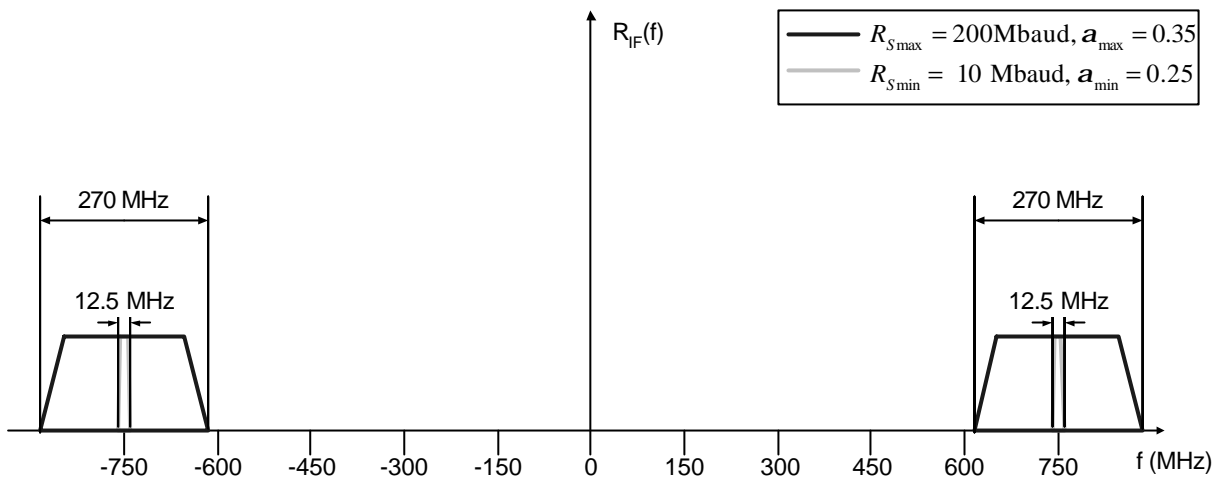


Figura 2.7-Spettro del segnale in uscita dal filtro di front-end ideale

Come accennato il segnale $r_{IF}(t)$, uscito dal filtro $H_{FE}(f)$, viene successivamente digitalizzato dal convertitore A/D che lo campiona con una frequenza di campionamento $f_{sa} = 600$ MHz. In uscita dal convertitore A/D si ottiene una sequenza di campioni $r_{IF}(kT_{sa})$ di cui si evidenzia lo spettro in figura 2.8.

La replica dello spettro originario è causata dall'operazione di campionamento, ed è proprio per questo che, come accennato nel paragrafo 2.1, la frequenza intermedia f_{IF} è stata scelta opportunamente in modo da poter ottenere un'operazione di down-conversion digitale molto semplice, come vedremo nel paragrafo successivo.

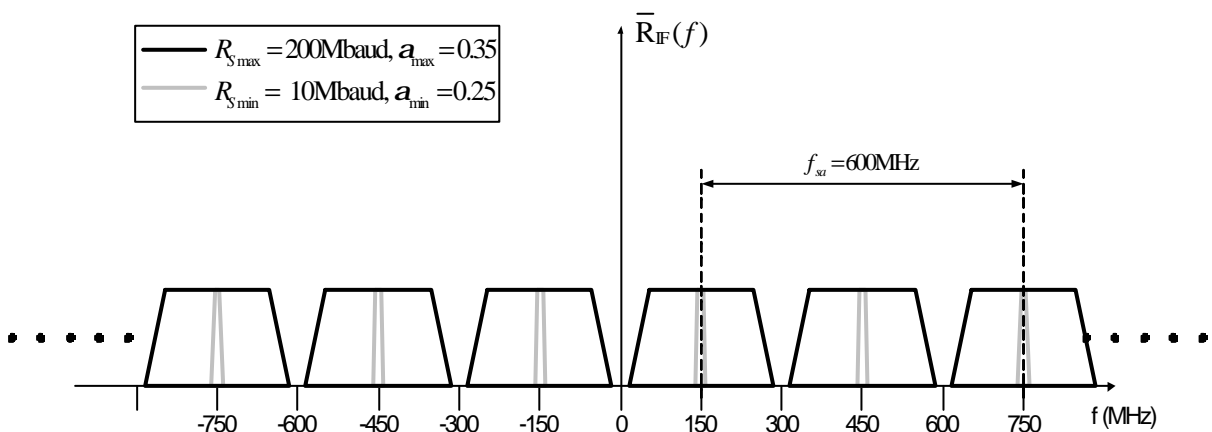


Figura 2.8-Spettro del segnale campionato in uscita dal convertitore A/D

2.2.3 – Stadio di down-conversion digitale

Come precedentemente accennato, per poter essere elaborato, il segnale viene convertito, dopo essere stato campionato, dalla frequenza intermedia alla banda base attraverso una down-conversion digitale con frequenza intermedia digitale $f_{IFD} = 150$ MHz.

Tale operazione, con i valori opportunamente scelti, risulta essere abbastanza semplice poiché il rapporto tra la frequenza intermedia digitale f_{IFD} e la frequenza di campionamento f_{sa} risulta essere:

$$f_{IFD} T_{sa} = \frac{f_{IFD}}{f_{sa}} = \frac{1}{4} \quad (2.19)$$

per cui se pensiamo di campionare una sinusoida di frequenza f_{IFD} con una frequenza di campionamento f_{sa} otteniamo una sequenza di valori che ciclicamente assumono i valori 1, 0, -1 e 0.

Allora l'operazione di conversione digitale consiste in questo caso nell'andare a moltiplicare i campioni in uscita dallo stadio A/D per suddetta sequenza, come illustrato nella figura 2.9.

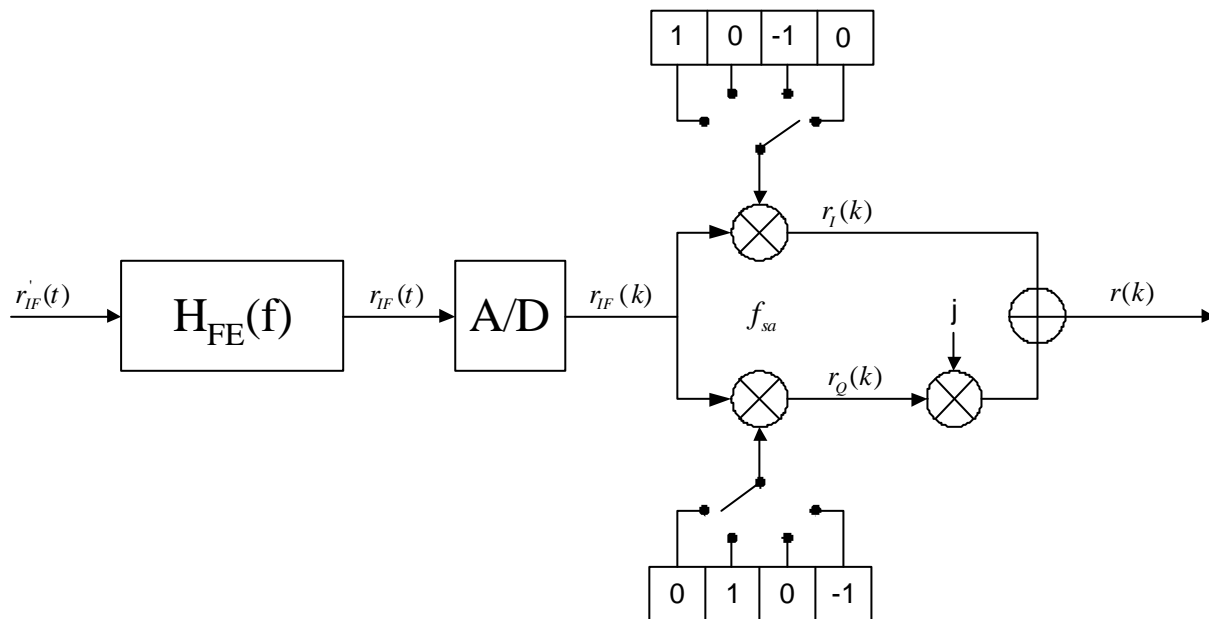


Figura 2.9-Architettura del down-converter digitale con frequenza di lavoro f_{sa}

Lo schema proposto tuttavia presenta notevoli difficoltà implementative, perché come già accennato le attuali tecnologie non permettono moltiplicazioni ad una frequenza di clock f_{sa} . Ma considerando che i campioni vengono alternativamente annullati dai valori nulli della sequenza, si sfrutta questa proprietà per ottenere uno schema del tutto equivalente a quello di figura 2.9 che viene illustrato in figura 2.10:

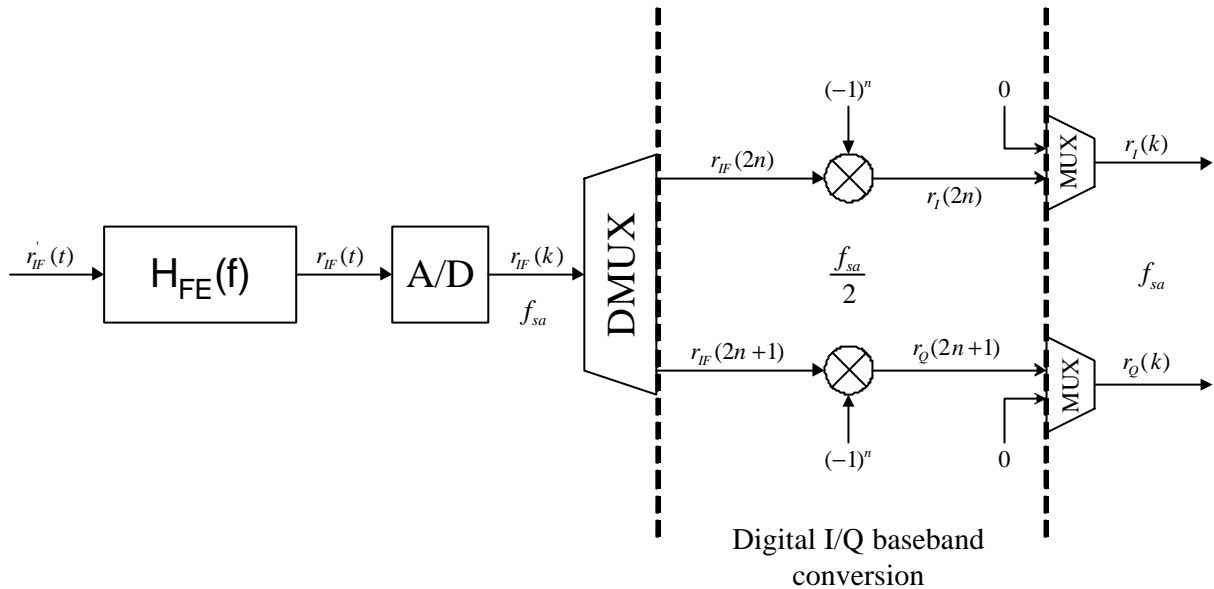


Figura 2.10-Architettura del down-converter digitale con frequenza di lavoro $f_{sa}/2$

Tale schema è realizzabile dalla tecnologia odierna, poiché il flusso di campioni viene suddiviso attraverso un demultiplexer in due sottoflussi con un rate dimezzato $\frac{f_{sa}}{2}$.

I campioni di entrambi i sottoflussi (il ramo in fase elabora i campioni di indice pari, quello in quadratura quelli di indice dispari) vengono moltiplicati per sequenze che assumono ciclicamente i valori 1 e -1, ottenendo quindi le successioni di campioni non nulle del convertitore di figura 2.9.

L'equivalenza dei due schemi è infine assicurata dall'operazione di zero-padding che introduce i campioni nulli tra due campioni consecutivi non nulli.

2.2.4 – Stadio di decimazione con filtri CIC parallelizzati

Dopo la down conversion digitale il segnale, portato in banda base, è pronto per poter essere elaborato, in modo che possa essere estratta dai campioni la maggiore quantità d'informazione possibile ad esso associata. Per tale elaborazione sono necessari e sufficienti solamente 2 campioni per intervallo di simbolo.

Ma le specifiche di progetto richiedono un symbol rate variabile tra il minimo $R_{S\min} = 10\text{Mbaud}$ ed il massimo $R_{S\max} = 200\text{Mbaud}$ e considerando come già anticipato il campionamento nello stadio A/D a $f_{sa} = 600\text{MHz}$, si ottiene un numero di campioni per intervallo di simbolo n_s :

$$n_s = \frac{T_S}{T_{sa}} = \frac{f_{sa}}{R_S} \quad (2.20)$$

ossia $n_s = 60\text{ samples/symbol}$ in corrispondenza di $R_{S\min}$ e $n_s = 3\text{ samples/symbol}$ in corrispondenza di $R_{S\max}$.

Come si vede il segnale risulta fortemente sovracampionato, soprattutto a bassi symbol rates. Da qui la necessità di utilizzare il filtro CIC, che come visto nel capitolo 1 permette di decimare il flusso di dati mantenendo un solo campione ogni r campioni.

In uscita al filtro CIC decimatore si ottiene così una frequenza decimata:

$$f_d = \frac{f_{sa}}{r} = 2R_S \quad (2.21)$$

per cui dalla (2.21) si ottiene che il fattore di decimazione r risulta:

$$r = \frac{f_{sa}}{2R_S} = \frac{n_s}{2} = \frac{300}{R_S[\text{Mbaud}]} \quad (2.22)$$

Nella tabella successiva vengono riportati alcuni valori relativi ad alcune specifiche di progetto:

R_s [Mbaud]	n_s	r	f_d [Mbaud]
10	60	30	20
30	20	10	60
75	8	4	150
150	4	2	300
200	3	-	400

Tabella 2.1-Valori di r al variare di R_s

dove nell'ultimo caso il fattore di decimazione r risulterebbe inferiore a 2 ed allora invece della decimazione si ottiene la frequenza decimata f_d voluta grazie ad una interpolazione polinomiale.

I filtri CIC dovrebbero ottenere al loro ingresso i campioni in uscita dalla down conversion digitale come mostrato in figura 2.10, solo che le attuali tecnologie, come già accennato, non permettono elaborazioni digitali ad un clock rate così alto, per cui si deve usare lo schema di figura 2.11 che risulta realizzabile grazie ad un'architettura "parallelizzata" del filtro CIC che opera solo sui campioni non nulli ad una frequenza dimezzata $\frac{f_{sa}}{2}$.

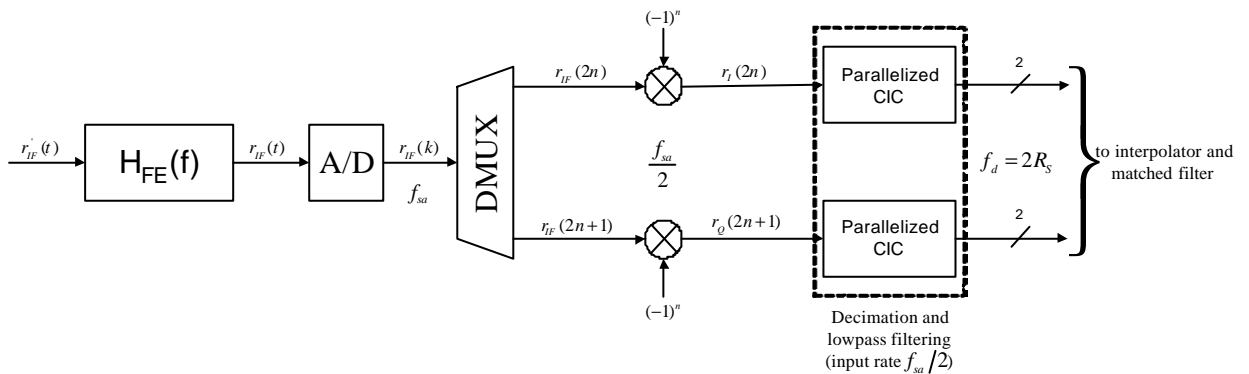


Figura 2.11-Architettura del front-end con filtri CIC operanti a frequenza $f_{sa}/2$

In appendice 2.A si riporta l'equivalenza dei due sistemi e la struttura del CIC parallelizzato, che differisce dal CIC convenzionale solo per quanto concerne la parte integratrice. Poiché il CIC parallelizzato lavora sul flusso di campioni non nulli, il suo fattore di decimazione :

$$\mathbf{r}_{//CIC} = \frac{f_{sa}/2}{2R_s} = \frac{\mathbf{n}_s}{4} = \frac{150}{R_s [Mbaud]} \quad (2.23)$$

risulta dimezzato rispetto a quello del CIC convenzionale, come si osserva dal confronto fra la (2.22) e la (2.23).

Nella tabella successiva vengono riportati alcuni valori relativi ad alcune specifiche di progetto:

$R_s [Mbaud]$	$\mathbf{r}_{//CIC}$	$f_d [Mbaud]$
10	15	20
30	5	60
75	2	150

Tabella 2.2-Valori di $\mathbf{r}_{//CIC}$ al variare di R_s

Dalla tabella 2.2 si nota che per symbol rate superiori a $R_s = 75$ Mbaud l'operazione di decimazione non risulta più necessaria, mentre si dovrà eventualmente correggere la spaziatura dei campioni attraverso uno stadio interpolatore che viene pilotato da una sequenza di campioni con frequenza f_{sa} , in modo da ottenere come richiesto campioni equispaziati tra loro di $T_D = T_s/2$ in ingresso al filtro adattato.

Per symbol rate inferiori a $R_s = 75$ Mbaud invece l'utilizzo del filtro CIC parallelizzato comporta anche l'utilizzo di un'equalizzatore in modo da compensare le distorsioni del segnale, causate da una risposta in frequenza del CIC non piatta sulla banda del segnale, come indicato nel paragrafo 1.3.

Infine osservando la tabella 2.3, che riporta tutti i parametri finora discussi in funzione dei vari baud-rate previsti dalle specifiche di progetto, si può notare (nei casi 2, 4 e 5) come la scelta di fattori di decimazione $\mathbf{r}_{//CIC}$ interi comporti una frequenza decimata f_d differente da quella desiderata specificata dalla (2.21).

Case	R_s	ρ'	$\rho //CIC$	fd	$2R_s$	Note
1	10	30	15	20	20	Equalizer needed
2	27.5	10.909	5	60	55	Equalizer needed, sample rate offset corrected by interpolator
3	30	10	5	60	60	Equalizer needed
4	32	9.375	4	75	64	Equalizer needed, sample rate offset corrected by interpolator
5	55	5.454	2	150	110	Equalizer needed, sample rate offset corrected by interpolator
6	75	4	2	150	150	Equalizer needed
7	90	3.333	no decimation	600	180	//CIC bypassed, zero-pad needed, sample rate offset corrected by interpolator
8	95	3.157	no decimation	600	190	//CIC bypassed, zero-pad needed, sample rate offset corrected by interpolator
9	127	2.362	no decimation	600	254	//CIC bypassed, zero-pad needed, sample rate offset corrected by interpolator
10	180	1.666	no decimation	600	360	//CIC bypassed, zero-pad needed, sample rate offset corrected by interpolator
11	190	1.578	no decimation	600	380	//CIC bypassed, zero-pad needed, sample rate offset corrected by interpolator
12	200	1.5	no decimation	600	400	//CIC bypassed, zero-pad needed, sample rate offset corrected by interpolator

Tabella 2.3 – Baud-rate e fattori di decimazione previsti

Anche in questi casi allora sar  necessario lo stadio interpolatore per garantire campioni equispaziati tra loro di $T_D = T_s/2$ in ingresso al filtro adattato.

Infine (nei casi 1, 3 e 6) poich  nel modem MHOMS lo stadio interpolatore risulta inserito, come si vede in figura 2.1, nell'anello di recupero del clock, allora verr  utilizzato anche quando la frequenza in uscita al filtro CIC parallelizzato coincide perfettamente con quella specificata dalla (2.21).

Tutto quanto finora illustrato risulta schematizzato in figura 2.12 dove viene riportata l'architettura generale dello stadio front-end del sistema:

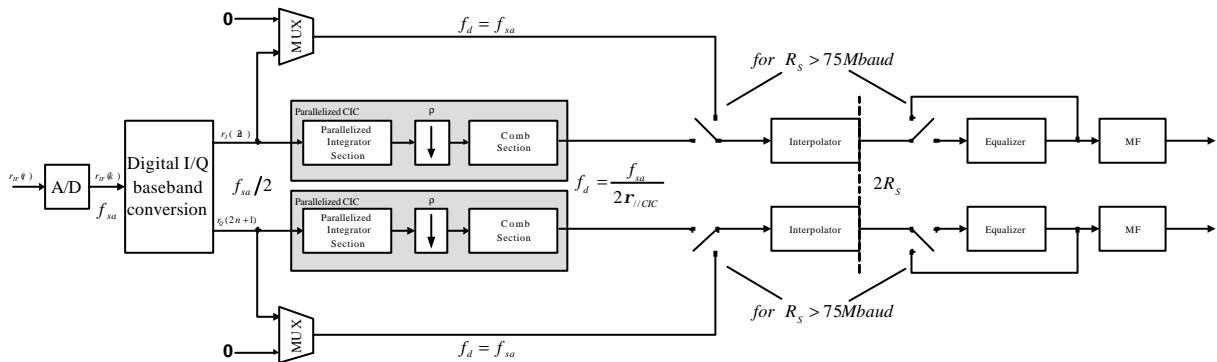


Figura 2.12 – Architettura generale del front-end del ricevitore

2.2.5 – Stadio interpolatore

L'interpolatore deve produrre, partendo dai campioni al suo ingresso, una sequenza di campioni interpolanti $\{y(t_n)\}$ in uscita equispaziati tra loro $T_I = T_S/2$.

Per fare questo si usa il sistema descritto in figura 2.13, dove all'interpolatore vengono forniti i valori degli istanti ottimi di campionamento ottenuti da un'anello di recupero del clock costituito dal TED (Timing Error Detector), dal filtro d'anello e dal controllore.

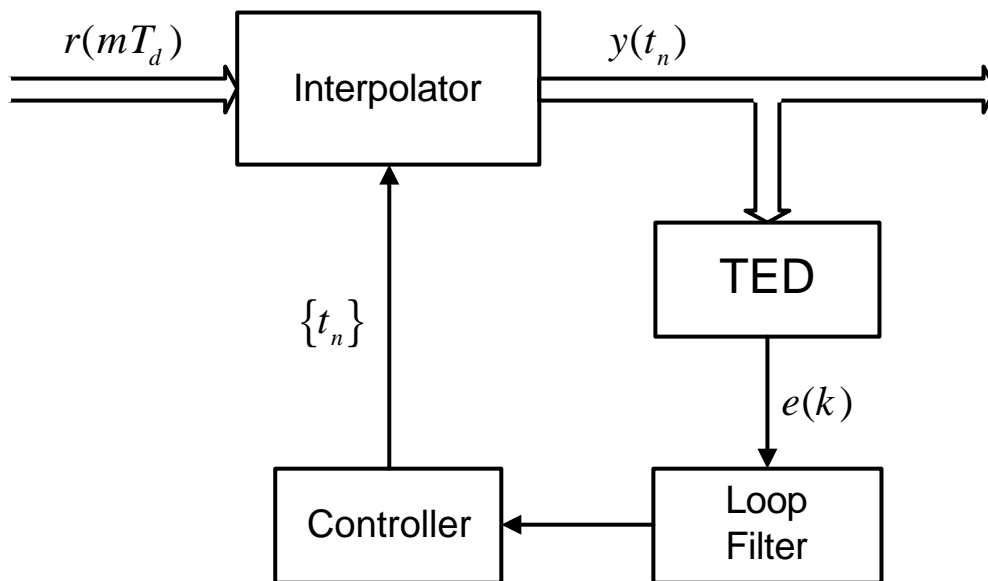


Figura 2.13 – Schema a blocchi dell'anello di recupero del clock

Il TED permette di stimare il ritardo t che viene introdotto dal canale, e genera in uscita un segnale errore $e(k)$ proporzionale alla differenza tra il ritardo effettivo t e la sua stima \hat{t}_k .

Il segnale errore $e(k)$ viene quindi utilizzato per aggiornare ricorsivamente le stime di ritardo.

Per implementare il controllore viene usato un NCO (Number Controlled Oscillator) in grado di fornire all'interpolatore i parametri necessari per il calcolo degli interpolanti secondo un algoritmo di controllo descritto in [2].

2.3 – IL FILTRO EQMF

2.3.1 – Il filtro equalizzatore

E' stato visto come il filtro CIC parallelizzato sia totalmente equivalente ad un filtro CIC convenzionale avente come ritardo differenziale lo stesso ritardo differenziale M , come ordine lo stesso ordine N , come fattore di decimazione un fattore di decimazione $r = 2r_{//CIC}$, e come risposta in frequenza (normalizzata):

$$H_{CIC}(f) = e^{-jp f T_{sa}(rM-1)N} \left[\frac{1}{rM} \frac{\sin(\mathbf{p}rMfT_{sa})}{\sin(\mathbf{p}fT_{sa})} \right]^N \quad (2.24)$$

Quindi, analogamente a quanto visto nel capitolo 1, anche la presenza del filtro CIC parallelizzato nel sistema del MHOMS causa una distorsione dello spettro del segnale trasmesso, poiché la risposta in frequenza del filtro non è piatta sulla banda del segnale.

Si utilizza allora un filtro equalizzatore in grado di poter compensare le distorsioni di ampiezza, subite dal segnale a frequenza f_d presente all'uscita del filtro CIC parallelizzato.

Nel caso ideale si vorrebbe che il filtro equalizzatore rappresenti il più accuratamente possibile la seguente risposta in ampiezza:

$$|H_{eq}(f)| = \frac{1}{|H_{CIC}(f)|} = \left| rM \frac{\sin\left(\mathbf{p} \frac{f}{f_{sa}}\right)}{\sin\left(\mathbf{p}rM \frac{f}{f_{sa}}\right)} \right|^N = \left| rM \frac{\sin\left(\mathbf{p} \frac{f}{rf_d}\right)}{\sin\left(\mathbf{p}M \frac{f}{f_d}\right)} \right|^N \quad (2.25)$$

con $|H_{eq}(0)| = 1$.

Tale risposta si ottiene implementando un filtro FIR con N_{eq} prese e con coefficienti $h_k^{(eq)}$ dove ($k = 0, 1, 2, \dots, N_{eq} - 1$).

Ovviamente la risposta impulsiva ideale verrà troncata, per cui si avrà $\hat{H}_{eq}(f) \cong H_{eq}(f)$ come risulta immediato verificare poiché

$$\hat{H}_{eq}(0) = \sum_{k=0}^{N_{eq}-1} h_k^{(eq)} \neq 1 \quad (2.26)$$

Maggiore è il numero di prese, migliore è l'approssimazione.

Ma un elevato numero di prese ha l'inconveniente di richiedere grandi memorie ed appesantire la capacità di calcolo, per cui si dovrà trovare un compromesso accettabile.

Dai risultati ottenuti in [3] si conclude che si ottengono prestazioni praticamente ideali del sistema con un filtro FIR a $N_{eq} = 13$ prese.

2.3.2 – Il filtro adattato

Come illustrato in figura 2.12, dopo che il filtro equalizzatore ha compensato le distorsioni eventualmente introdotte sul segnale dall'utilizzo del filtro CIC nei casi dove questo risulta necessario, il flusso di campioni viene mandato in ingresso ad un filtro adattato (MF).

Anche tale filtro viene implementato attraverso un filtro FIR con N_{MF} prese come mostrato in figura 2.14:

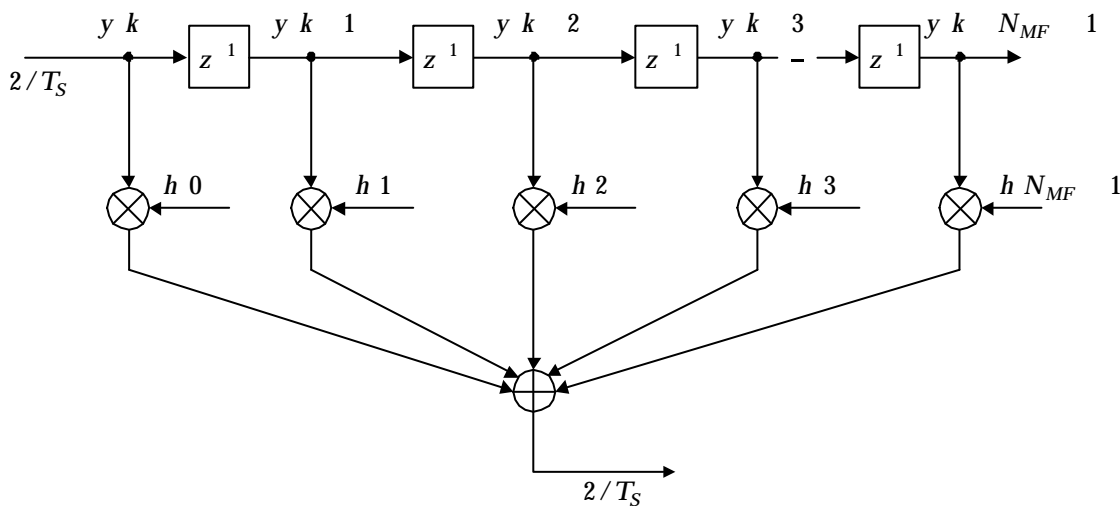


Figura 2.14 – Implementazione del filtro adattato

Analogamente al paragrafo precedente, il numero di prese del filtro FIR che implementa il filtro adattato deve essere scelto in maniera tale da poter approssimare il più possibile la

risposta in frequenza di un impulso a radice di coseno rialzato (SRRC) uguale a quello usato in trasmissione, ossia deve approssimare la funzione:

$$G_R(f) = \frac{G(f)}{\sqrt{T_s}} \quad (2.27)$$

con $G(f)$ espresso dalla (2.5) e $G_R(0) = 1$.

Anche in questo caso verrà trovato un compromesso accettabile.

Difatti dai risultati ottenuti in [3] si conclude che si ottengono prestazioni praticamente ideali del sistema con un filtro FIR a $N_{MF} = 21$ prese.

2.3.3 – Il filtro integrato EQMF

Come mostrato in [3], per ridurre fortemente la complessità di implementazione dello stadio front-end del modem MHOMS, i due filtri equalizzatore ed adattato vengono realizzati implementando un unico filtro FIR detto filtro EQMF in grado di correggere le distorsioni introdotte dal filtro CIC e contemporaneamente provvedere alle operazioni di filtraggio adattato.

I risultati effettuati hanno dimostrato che con un filtro EQMF avente un numero di $N_{EQMF} = 25$ prese si ottengono prestazioni praticamente ideali del sistema.

Tale risultato verrà sfruttato nel prossimo capitolo per il dimensionamento e l'ottimizzazione dello stadio di conversione A/D del MHOMS.

Appendice 2.A – Equivalenza dell'architettura parallelizzata del filtro CIC

Il sistema di figura 2.11 con i filtri CIC operanti alla frequenza $f_{sa}/2$ è del tutto equivalente al sistema di figura 2.A1, dove i filtri CIC dovrebbero operare alla frequenza f_{sa} , per cui praticamente irrealizzabile.

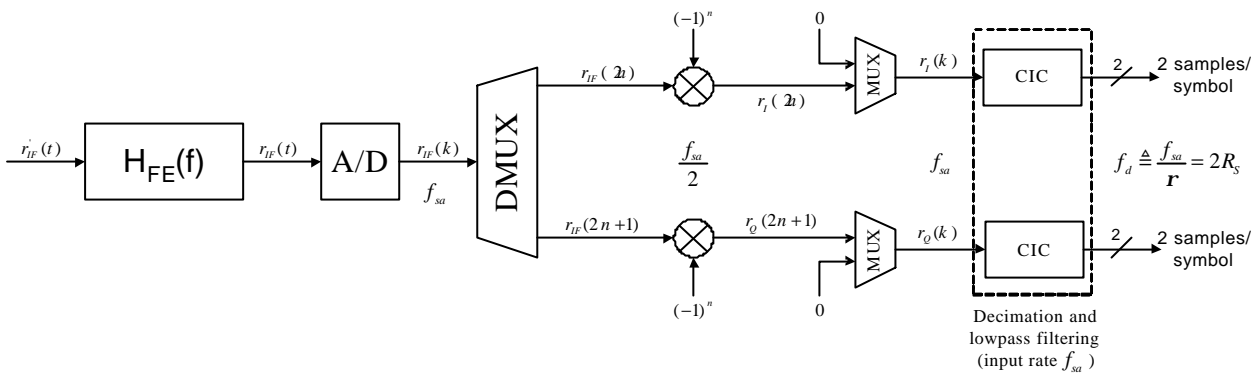


Figura 2.A1-Architettura del front-end con filtri CIC operanti a frequenza f_{sa}

Difatti partendo dalla struttura dello schema del filtro CIC presentato in figura 1.1, nel nostro caso in ingresso ai filtri si hanno i campioni alternativamente non nulli:

$$\begin{aligned}
 r_I(k) & \begin{cases} \neq 0 & \text{se } k = 2n \\ = 0 & \text{se } k = 2n + 1 \end{cases} \\
 r_Q(k) & \begin{cases} = 0 & \text{se } k = 2n \\ \neq 0 & \text{se } k = 2n + 1 \end{cases}
 \end{aligned} \tag{2.A.1}$$

dove intendiamo con n un generico valore intero.

Poiché i ragionamenti seguenti sono equivalenti per entrambi i rami, concentreremo la nostra attenzione sul ramo in fase I.

Le variabili di stato $S_i(k)$ del filtro CIC, ossia il contenuto dei registri del filtro, vengono aggiornate diversamente a seconda che si abbia in ingresso un campione nullo o non.

Allora negli istanti pari ($k = 2n$) si ha un campione non nullo in ingresso:

$$\begin{aligned}
 S_1(2n) &= S_1(2n-1) + r_1(2n) \\
 S_2(2n) &= S_2(2n-1) + S_1(2n) = S_2(2n-1) + S_1(2n-1) + r_1(2n) \\
 S_3(2n) &= S_3(2n-1) + S_2(2n) = S_3(2n-1) + S_2(2n-1) + S_1(2n-1) + r_1(2n) \\
 &\dots\dots\dots \\
 S_N(2n) &= S_N(2n-1) + S_{N-1}(2n-1) + \dots\dots + S_1(2n-1) + r_1(2n)
 \end{aligned}
 \tag{2.A.2}$$

che può essere generalizzata come:

$$S_i(2n) = r_i(2n) + \sum_{k=1}^i S_k(2n-1) \quad \text{per } 1 \leq i \leq N
 \tag{2.A.3}$$

Invece negli istanti dispari ($k = 2n+1$) si ha un campione nullo in ingresso:

$$\begin{aligned}
 S_1(2n+1) &= S_1(2n) \\
 S_2(2n+1) &= S_2(2n) + S_1(2n+1) = S_2(2n) + S_1(2n) \\
 S_3(2n+1) &= S_3(2n) + S_2(2n+1) = S_3(2n) + S_2(2n) + S_1(2n) \\
 &\dots\dots\dots \\
 S_N(2n+1) &= S_N(2n) + S_{N-1}(2n+1) = S_N(2n) + S_{N-1}(2n) + \dots\dots + S_1(2n)
 \end{aligned}
 \tag{2.A.4}$$

che può essere generalizzata come:

$$S_i(2n+1) = \sum_{m=1}^i S_m(2n) \quad \text{per } 1 \leq i \leq N
 \tag{2.A.5}$$

ed essere riscritta come:

$$S_i(2n-1) = \sum_{m=1}^i S_m(2n-2) \quad \text{per } 1 \leq i \leq N
 \tag{2.A.6}$$

Infine sostituendo la (2.A.6) nella (2.A.3) si ricava la seguente *equazione di aggiornamento delle variabili di stato* relative alla sezione integratrice del filtro CIC :

$$S_i(2n) = r_i(2n) + \sum_{k=1}^i \sum_{m=1}^k S_m(2n-2) \quad \text{per } 1 \leq i \leq N \quad (2.A.7)$$

La struttura che realizza l'aggiornamento delle variabili di stato espresso dalla (2.A.7) viene mostrata in figura 2.A.2:

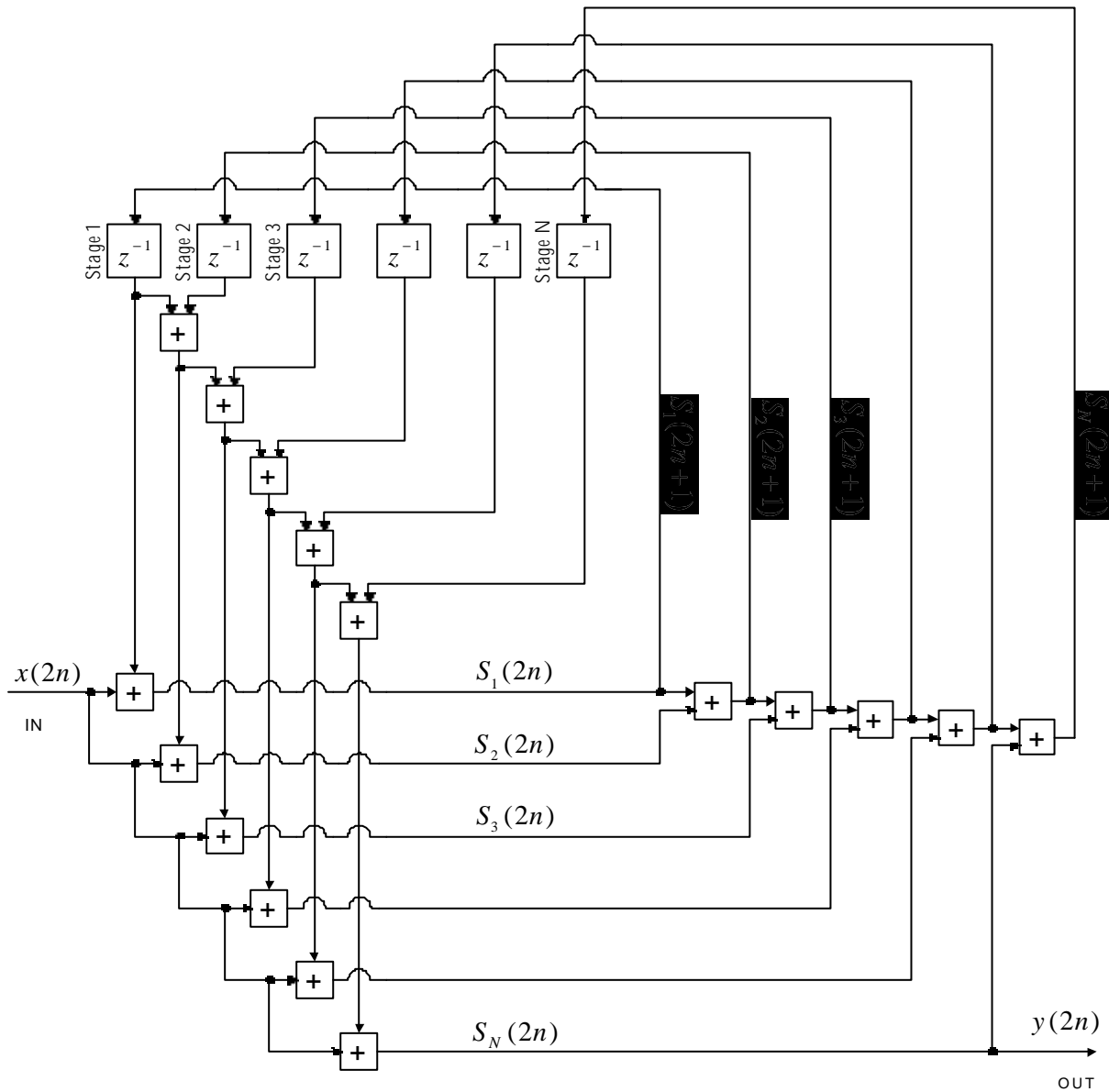


Figura 2.A.2-Architettura parallelizzata della sezione integratrice del CIC

Come mostrato dalla (2.A.7) le variabili di stato si aggiornano solo in corrispondenza di un campione non nullo all'ingresso del filtro, per cui con una frequenza $f_{sa}/2$, il che permette al filtro CIC che adotti per la sezione integratrice tale struttura di poter elaborare i campioni ad un clock rate accettabile dalla tecnologia odierna.

Un filtro CIC con tale struttura è denominato "Parallelized CIC" ed accetta al suo ingresso solamente i campioni non nulli, da cui il sistema equivalente rappresentato in figura 2.11.

