by Alberto I2PHD

ARM Radio, ovvero la radio a braccio (di ferro…)

by Alberto I2PHD

La SDR, pur presentando innegabili vantaggi prestazionali rispetto alle più tradizionali radio realizzate convenzionalmente, non sempre ha avuto quella “acceptance” che inizialmente ci si era illusi che potesse avere….

PERCHÉ ???

Eppure i suoi vantaggi sono innegabili, sia dal punto di vista delle prestazioni che anche , se vogliamo , di quello economico.

Si può partire da un Softrock da 15 dollari, trovando poi un range di prezzi adatto a tutte le borse.

Senza dimenticare i vantaggi operativi che può dare la visualizzazione istantanea e in tempo reale, con rinfresco dello schermo anche di 20 – 25 volte al secondo… tutta un'altra minestra rispetto ai patetici "panoramici" degli Icom o degli Yaesu, che altra funzione non hanno se non quella di fare scena…

Però poi, parlando con molti OM, alla fine la verità viene fuori, in tutta la sua crudezza…

La necessità di usare un PC….

by Alberto I2PHD

Il PC non voglio usarlo perché ….

- Il PC è rumoroso…

- Il PC emette disturbi radio…

- Il PC è ingombrante…

- Il PC a volte si pianta nel bel mezzo di un DX…

- **Manca la Manopola della sintonia !!!!**

- Usare un PC vuol dire avere cavi fastidiosi in giro…

- Per usare un PC bisogna sapere quando schiacciare il tasto destro e quando quello sinistro del mouse… (oddio, questo è un problema che la Apple risolse al momento della introduzione del Mac… conoscendo la levatura mentale del target a cui il Mac era destinato, fece un mouse con un tasto solo, così i poveretti non potevano sbagliarsi… ☺

by Alberto I2PHD

Ci sono, a dire il vero, dei pregevoli tentativi di superare il problema mascherando il PC dentro il contenitore della SDR, ottenendo dei risultati più che accettabili…

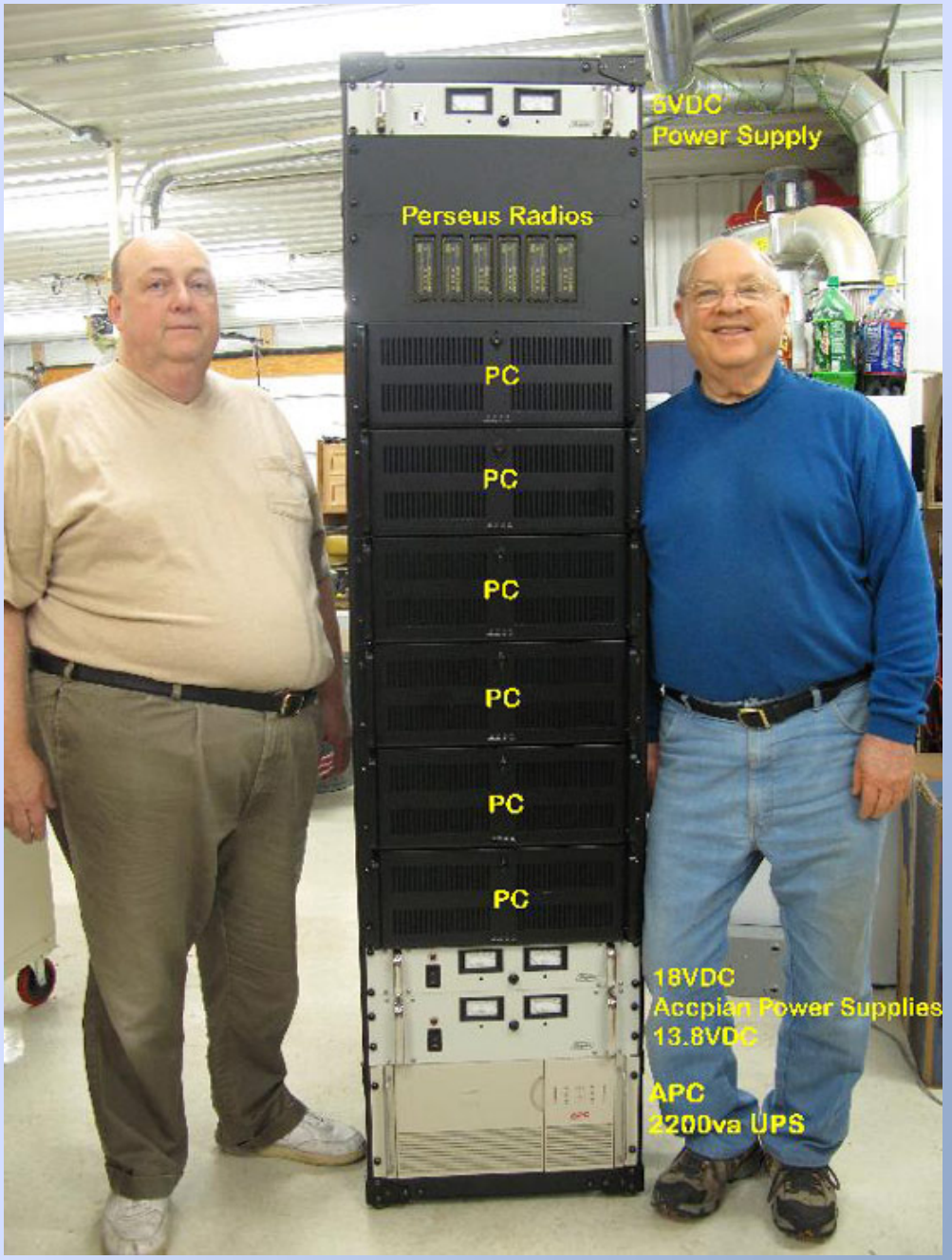
Ma non tutti si pongono quei problemi…. C'è chi non si accontenta di un solo PC, magari perché ha 6 SDR, su 6 bande diverse, e allora necessita di 6 PC, con il CW skimmer che gira su ciascuno…

C'è chi può…

La postazione DX di Tim K3LR…

6 Perseus e 6 PC in rack ….

Per la gioia di Beppe IK3VIG…



by Alberto I2PHD

Comunque oggettivamente il problema sussiste, anche se sembra che alcuni dei maggiori produttori di SDR non abbiano ben capito la cosa… Flex, tanto per non fare nomi…

Un RTX da 7000 dollari, con tutto il processing DSP fatto on-board, su un DSP Texas dedicato, ma che comunque ha bisogno del PC per la GUI e per far vedere lo spettro…

by Alberto I2PHD

Invece a qualcuno è venuta l'idea di fare un RTX completamente autonomo, con tutta la elaborazione DSP fatta dentro la SDR stessa, senza appoggiarsi ad un PC.

Questo vuol dire però abbandonare la comodità di usare la potenza di calcolo di un Pentium, che oggi ha raggiunto, con le ultime serie I5 e I7, dei livelli veramente considerevoli.

Per fortuna il consumismo ci viene in soccorso… la massa di decerebrati che non riesce a vivere se non è attaccata con un cordone ombelicale al proprio cellulare ha fatto sì che i volumi di produzione dei cosiddetti "telefonini" abbia raggiunto cifre di centinaia di milioni se non miliardi di pezzi.

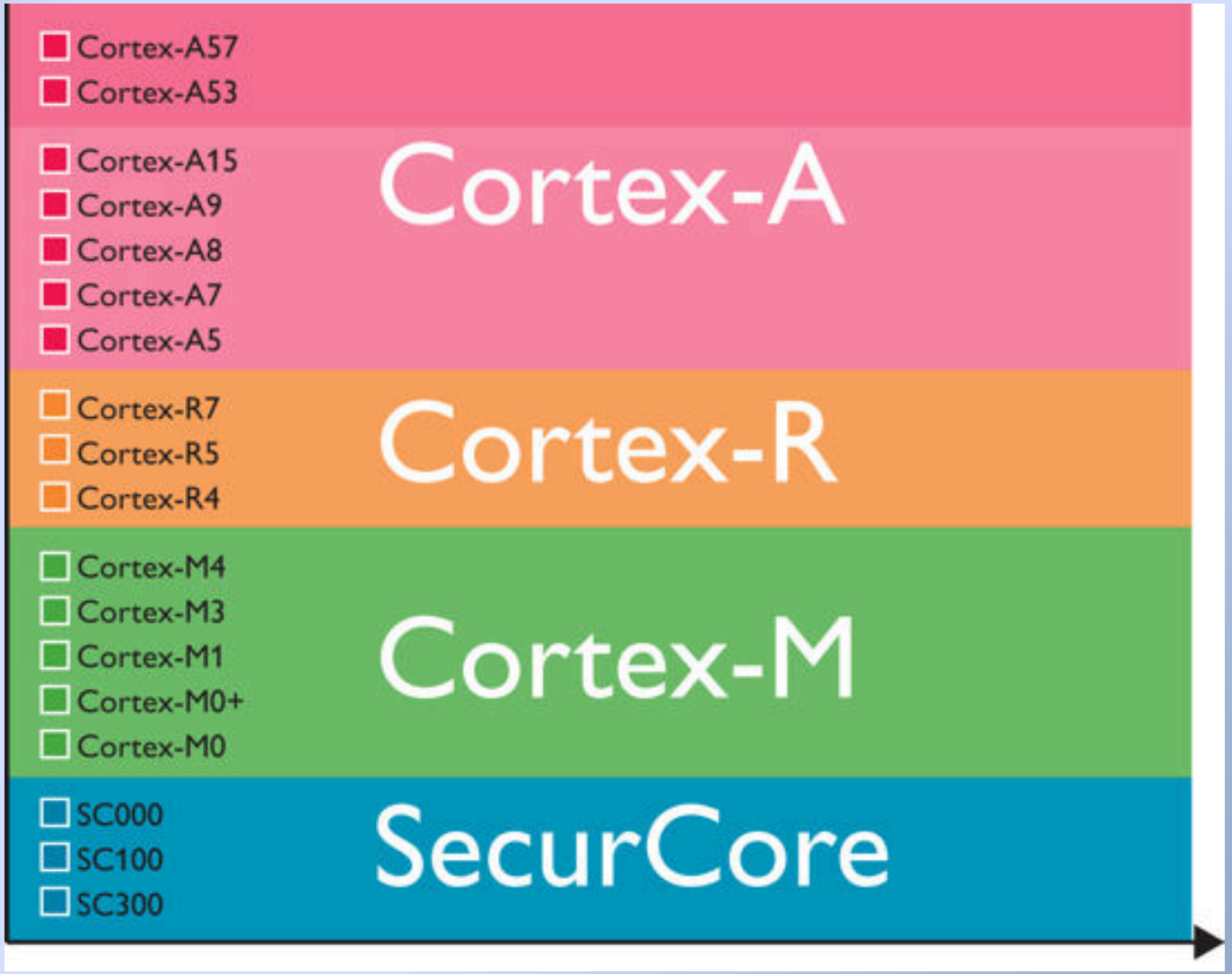
E qualunque economista della mutua sa che alti volumi = bassi prezzi. La cosiddetta economia di scala. E i cellulari hanno bisogno di alte potenze di calcolo per far divertire il bambino cretino che gode quando allargando due dita sullo schermo l'immagine si ingrandisce…

Advanced RISC Machine

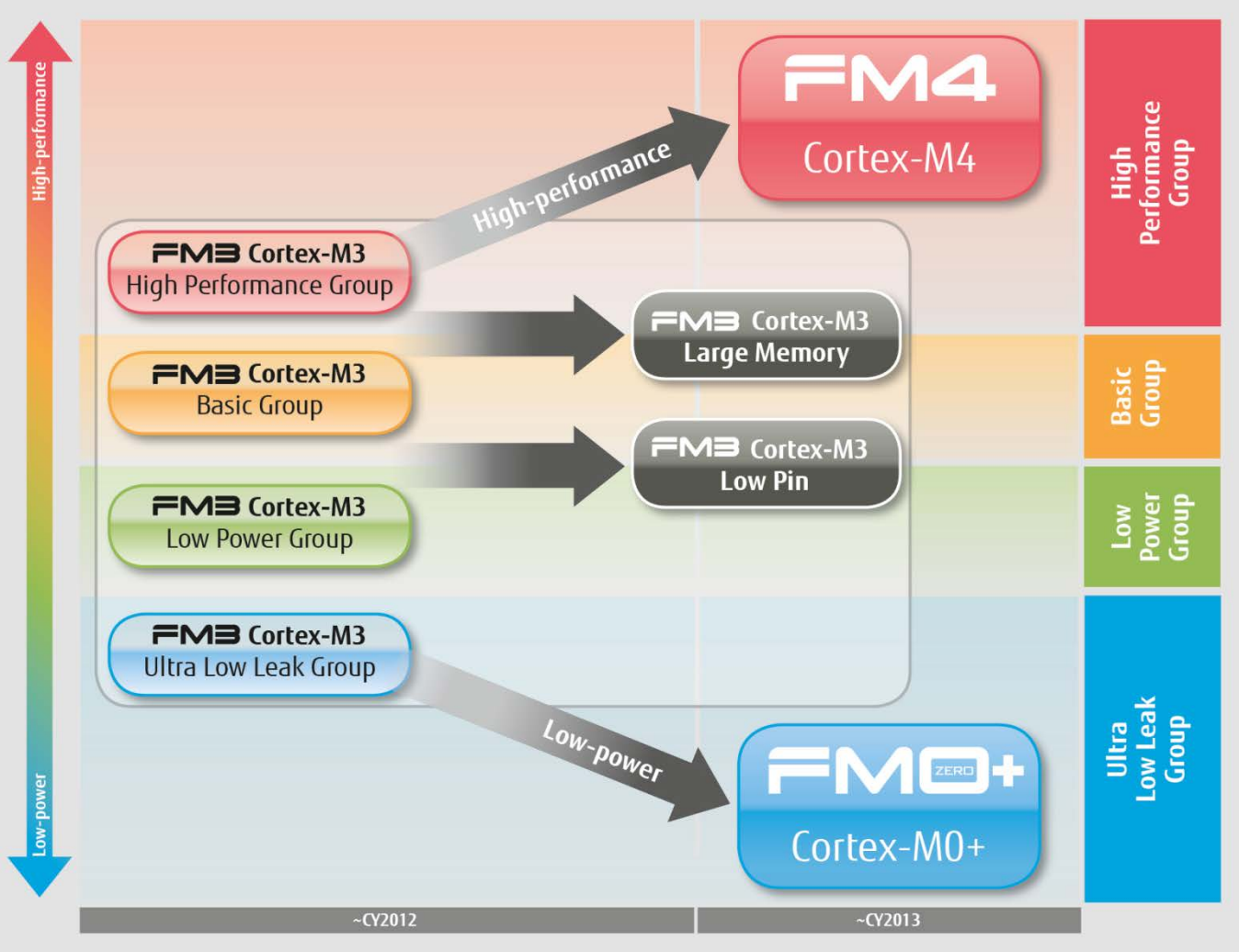
Il consorzio ARM nacque nel 1984 come Acorn RISC Machine, poi cambiò nome diventando ARM Holdings, e cedendo in licenza nel 1990 la architettura base del processore a varie aziende nel mondo. Nel solo 2010 sono stati prodotti oltre 6.1 miliardi di processori ARM… È oggi, 2013, la architettura a 32 bit numericamente più diffusa, distanziando alla larga l'x32 della Intel.

by Alberto I2PHD

La famiglia dei processori ARM.

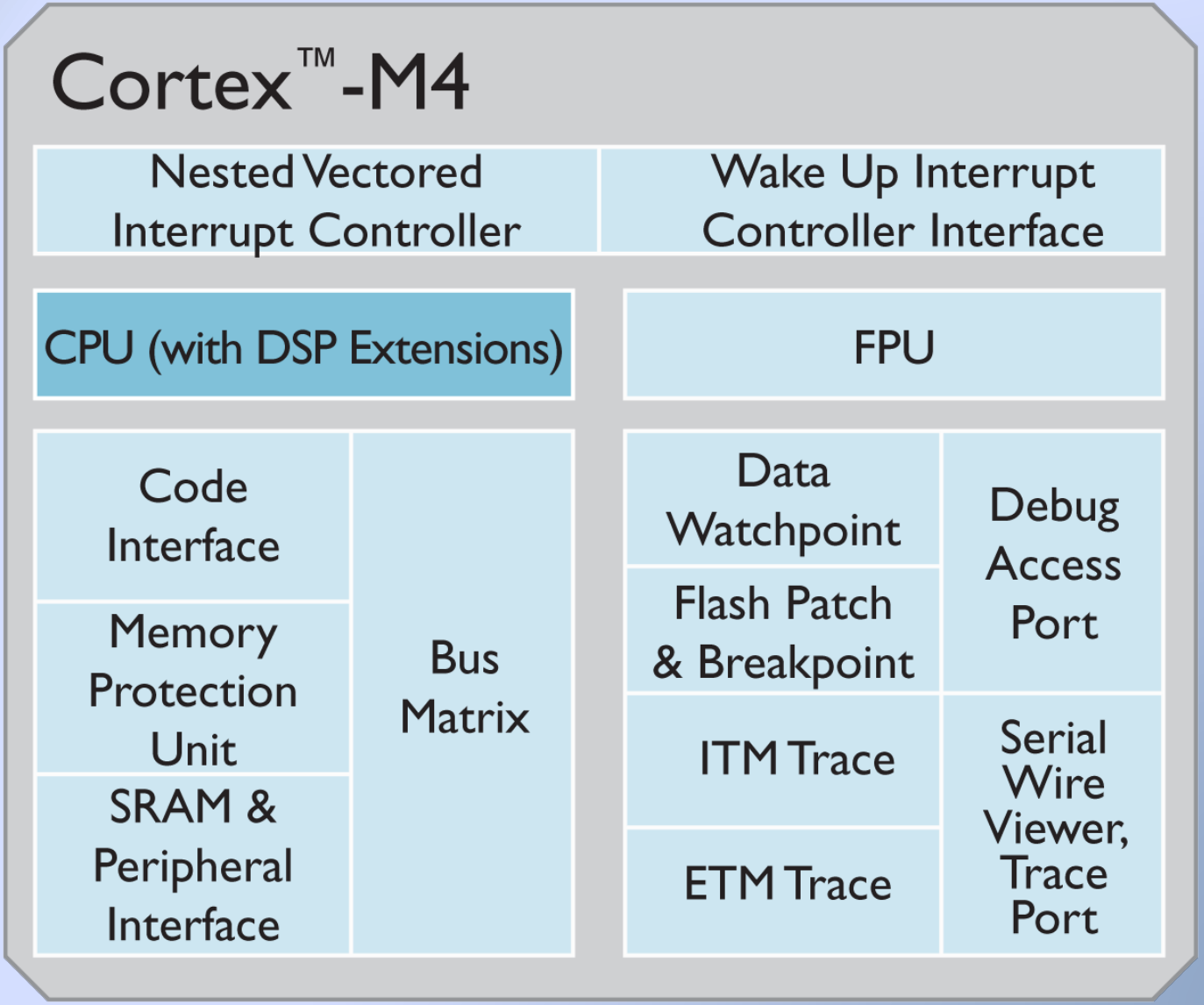


La sottofamiglia Cortex-M



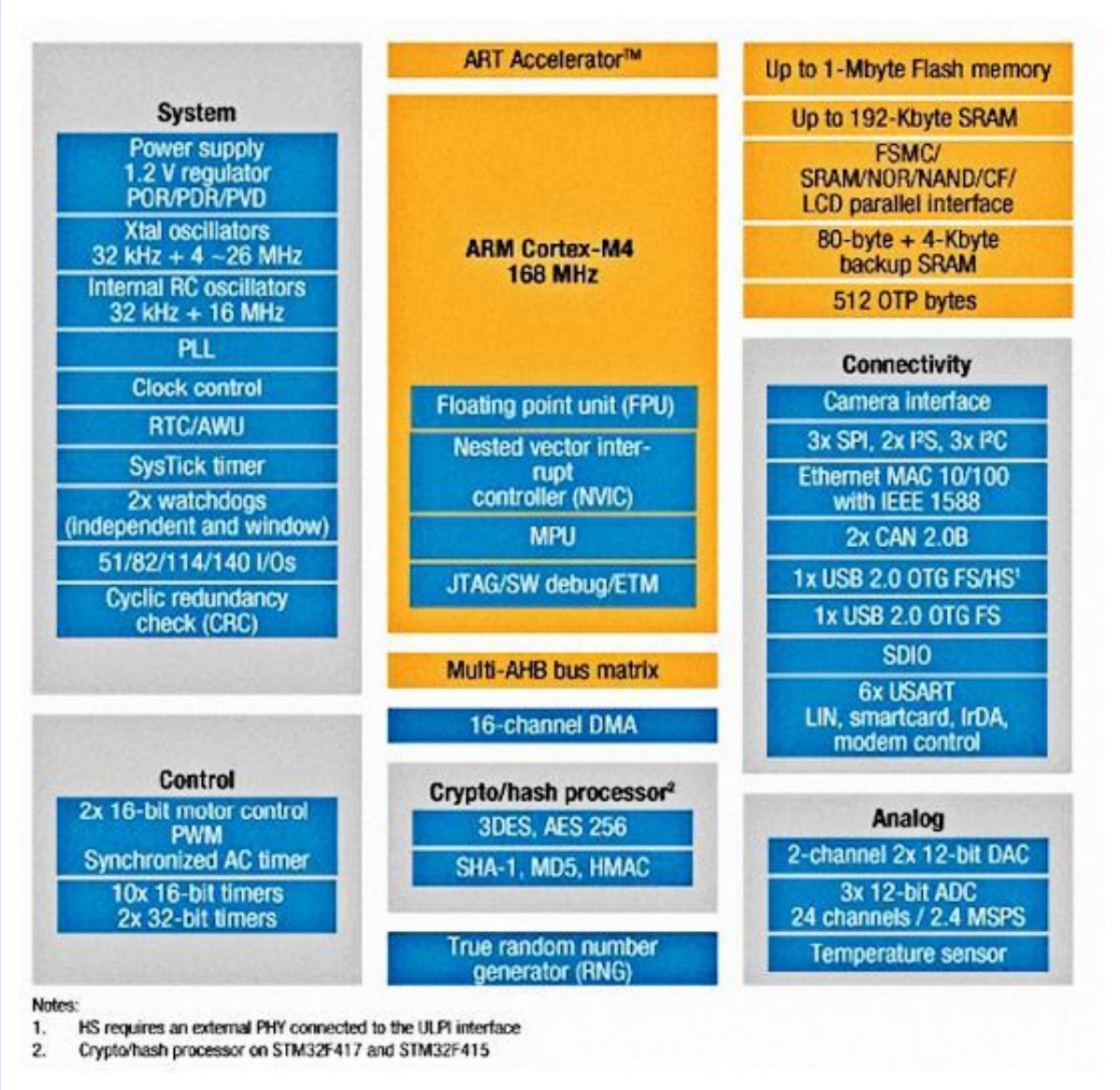
by Alberto I2PHD

Il membro della famiglia Cortex-M che sembra più adatto per applicazioni SDR embedded low-cost.



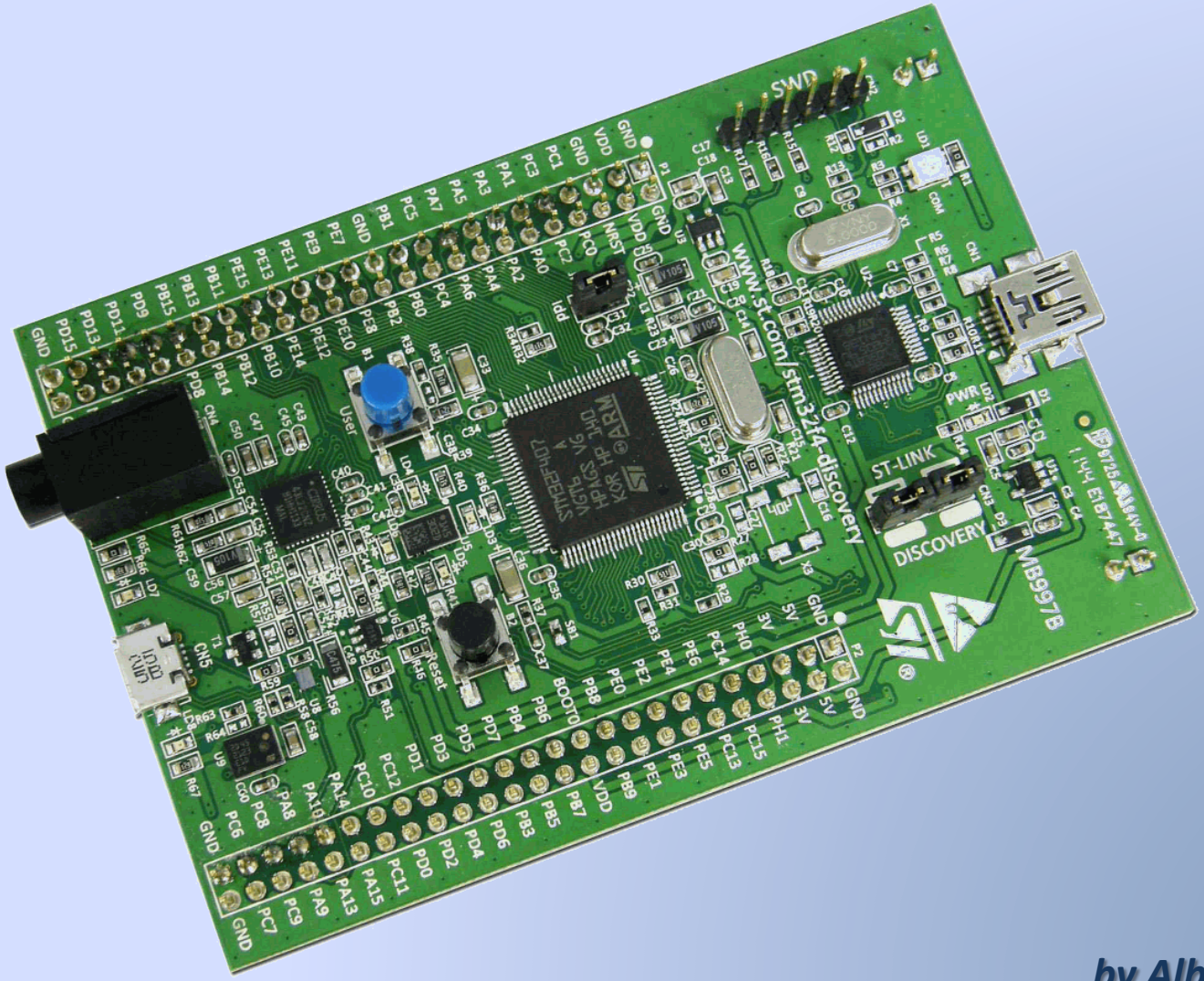
by Alberto I2PHD

E nella fattispecie l'STM32F4M della STM



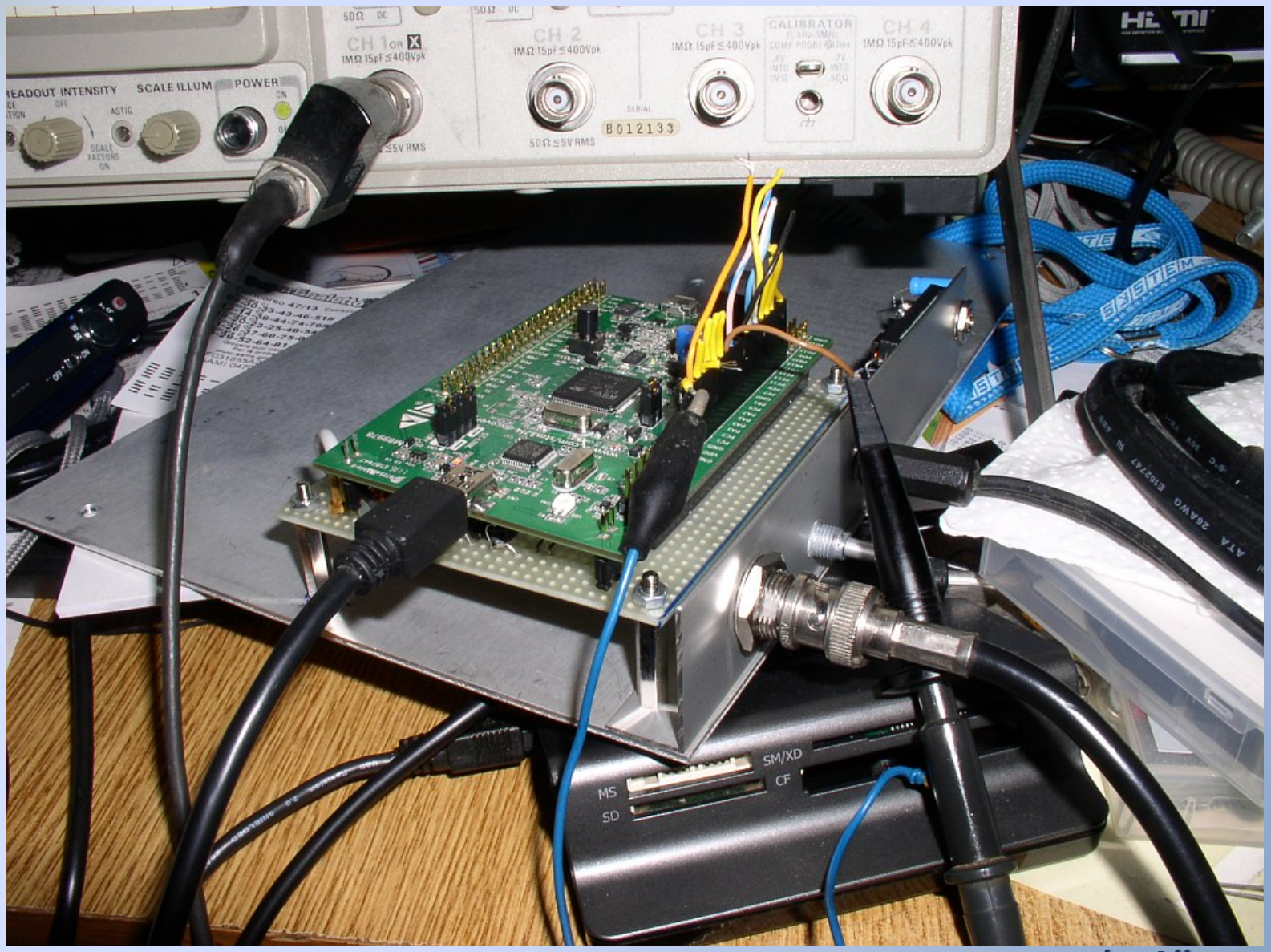
by Alberto I2PHD

Per questo processore esiste una Evaluation board della STM a basso costo, circa 15 Euro



by Alberto I2PHD

Fatta la scelta, incomincia il periodo della sperimentazione, delle prove, dei grattamenti di capo, delle battute della testa contro il muro…

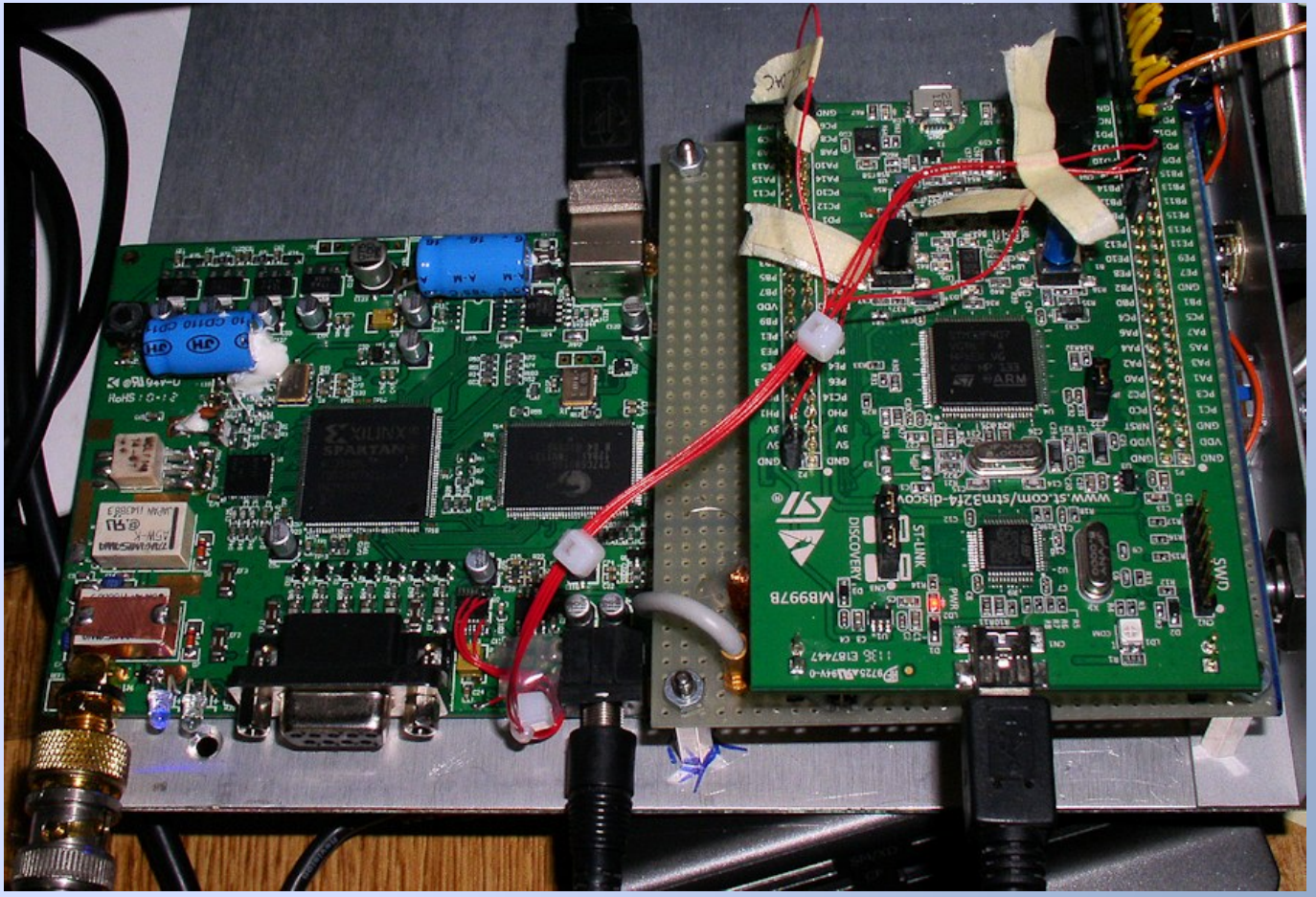


by Alberto I2PHD

Però l'M4F, pur avendo tre ADC on-board, non può essere usato come campionatore RF per tutte le HF, perché al massimo, in interleaved mode, i suoi tre ADC possono arrivare a poco più di 8 Ms/s, e poi sono a 12 bit…

Serviva qualcosa da essere usato come front-end…

Parlando con Franco Milan, il responsabile della ELAD, in occasione della edizione primaverile del Mercatino di Marzaglia, venne fuori l'idea di usare il campionatore dell'FDM-S1, seguito da un ARM M4F come back-end SDR.



by Alberto I2PHD

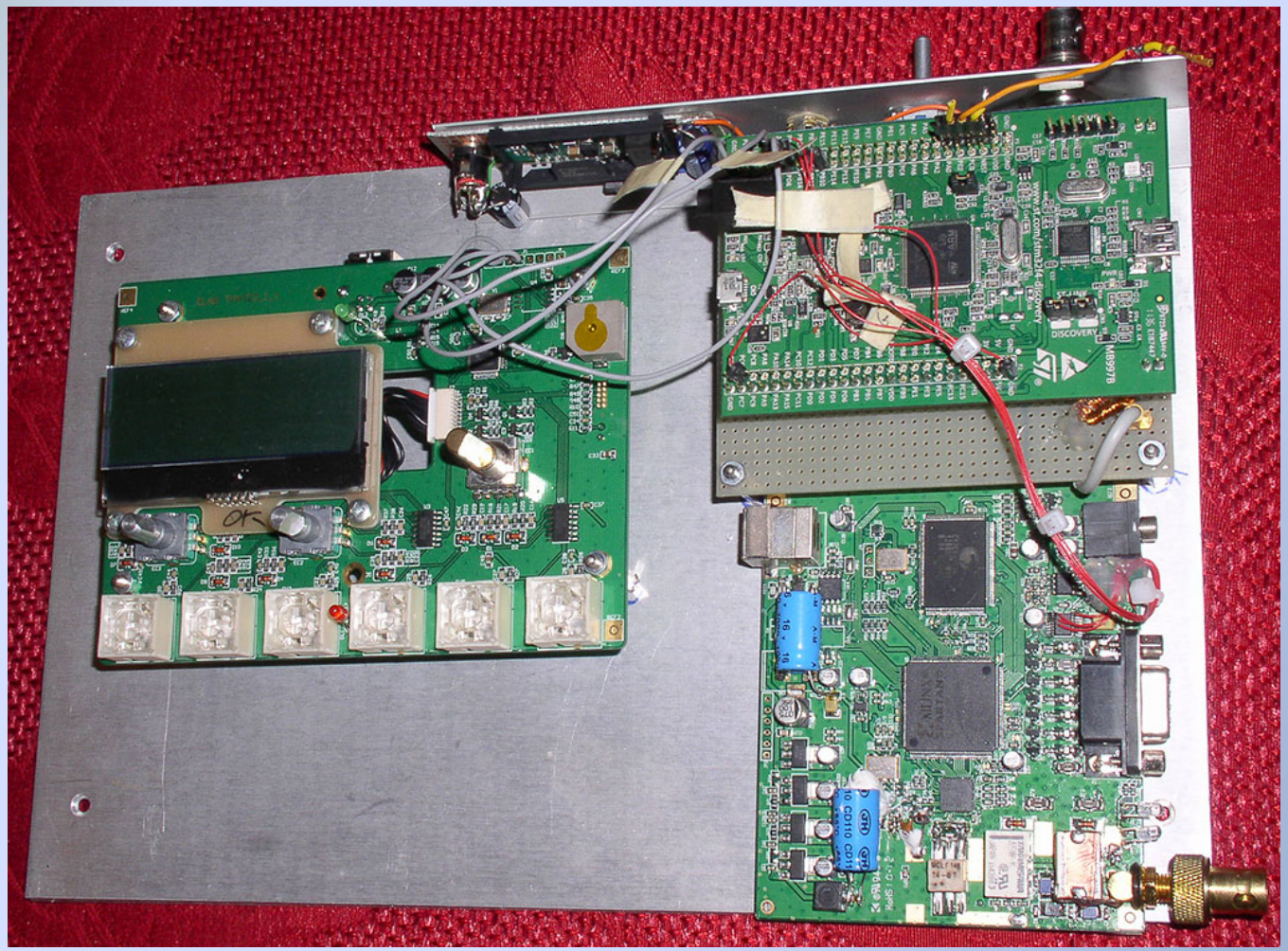
Ecco che era nata, in nuce, l'idea di una SDR PC-less, stand-alone, che potesse funzionare in modo autonomo, avendo on-board tutta la potenza di calcolo necessaria per i filtraggi, la demodulazione, etc.

Restava comunque il problema della interfaccia utente… la sintonia, il volume, la banda, la larghezza di banda, eventuali filtri anti-noise, etc.

In prima battuta fu scelto di usare ciò che già esisteva, e cioè il controllore Tmate 2, sempre di produzione ELAD.

by Alberto I2PHD

Ecco quindi approntata una versione ad-hoc sia dell'FDM-S1 che del TMate 2, entrambi con firmware modificato secondo le necessità del progetto.



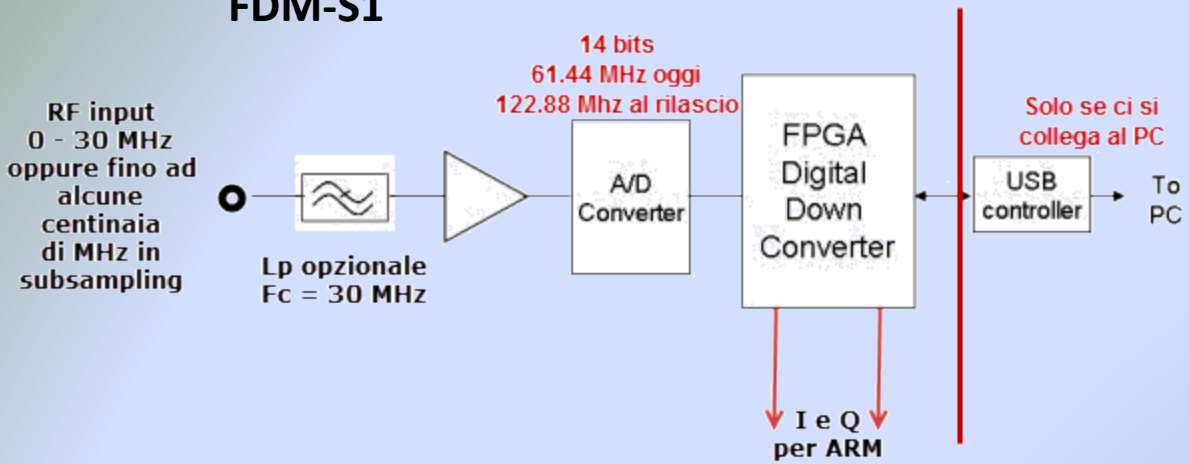
Non è molto elegante…

Ma funziona… ☺

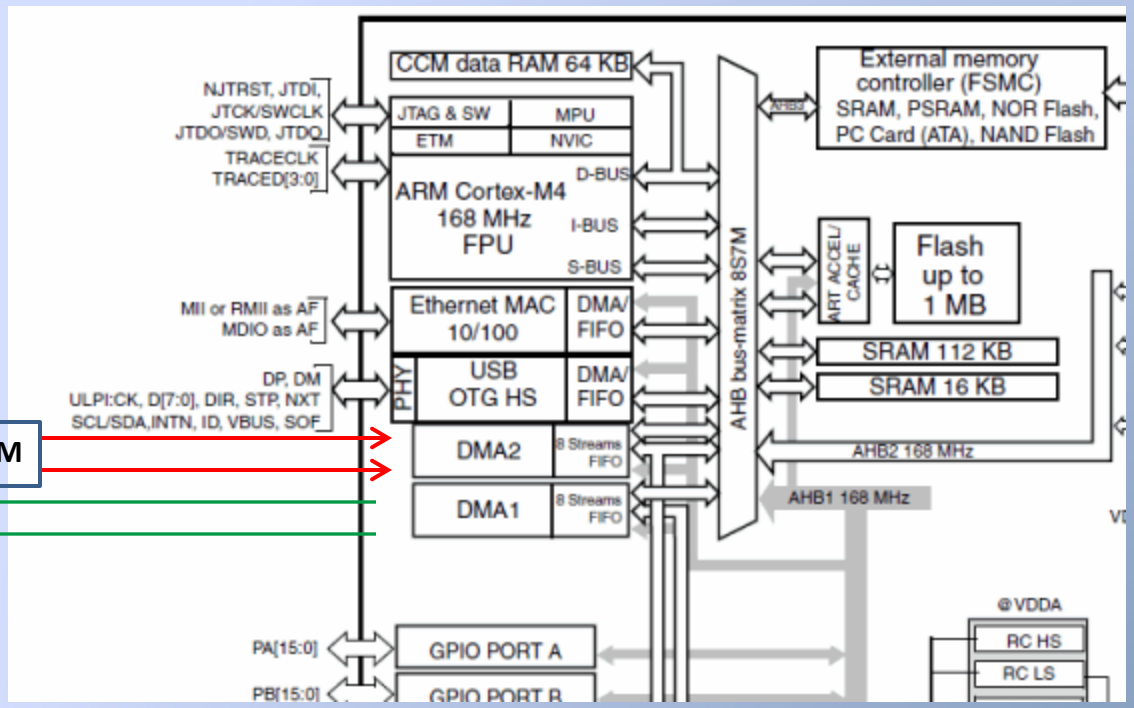
by Alberto I2PHD

Architettura HW

FDM-S1



Cortex ARM M4F



I e Q da FDM

Colloquio seriale verso il TMate 2 e verso il modulo display TFT

by Alberto I2PHD

Architettura SW su Cortex ARM

Idle priority thread

Main program :

• Inizializzazione FFT, DMA e Vectored Interrupt Controller

• Gestione colloquio seriale verso il display TFT e verso il Tmate 2, tramite il DMA 1

• Calcolo FFT per spettro e waterfall (circa 15 al secondo)

• Gestione GUI

Highest priority thread

DMA 2 interrupt service routine :

• Acquisizione dati con trasformazione da int32 a float32

• Calcolo NCO con oscillatore IIR in quadratura di fase con stabilizzazione di livello

• Preparazione del buffer WOLA per lo spettro

• Calcolo di un doppio CIC con decomposizione polifase, con divisione totale per 16

• Preparazione del buffer per il thread di filtraggio e demodulazione, e suo triggering

Medium priority thread

Filtraggio e demodulazione:

• FIR e decimazione per 4 (kernel = 68 taps)
• Filtraggio passabanda con la tecnica della fast convolution (kernel = 769 taps)
• Demodulazione CW, SSB, AM, FM
• Filtraggi noise blanker e denoiser

by Alberto I2PHD

Architettura SW su Cortex ARM

Nella prima versione del progetto, il data rate tra FDM e ARM era di oltre 1.5 Ms/sec
Nella ipotesi di un filtraggio passabasso con banda di transizione pari a 100 Hz (ad esempio 3000 Hz passband e 3100 Hz stopband) , un calcoletto con una formuletta empirica, ma abbastanza accurata, ci dice che servirebbe in teoria un FIR da :

Nt = Att(dB)/(22*(fsn – fpn)) dove :

Att attenuazione richiesta in dB
fsn frequenza normalizzata dell'inizio della banda oscura.
fpn frequenza normalizzata della fine della banda passante.

Mettendo i numeri, supponendo 50 dB di attenuazione, 3000 Hz di fine banda passante e 3100 Hz di inizio banda oscura, il risultato è 50/(22*(100/1536.E3)) = 34909 taps….. !!!

È ovvio che non è pensabile neppure dalla persona con la fantasia più malata l'ipotesi di implementare un FIR con 34909 taps che viaggi ad un sampling rate di oltre 1.5 Ms/s …….

Occorre quindi fare un downsampling fino ad arrivare ad un sampling rate adeguato alle bande passanti in gioco. Ad esempio 24 kHz potrebbero andar bene. Quindi il fattore di decimazione risulta essere pari a 1536/24 = 64, che può essere conveniente spezzare in due fattori, 16 e 4.

Il fattore 16, visto l'alto sampling rate, conviene implementarlo con una decimazione preceduta da un filtro anti-alias a media mobile. In prima battuta viene subito in mente il classico CIC.

Il CIC, questo sconosciuto

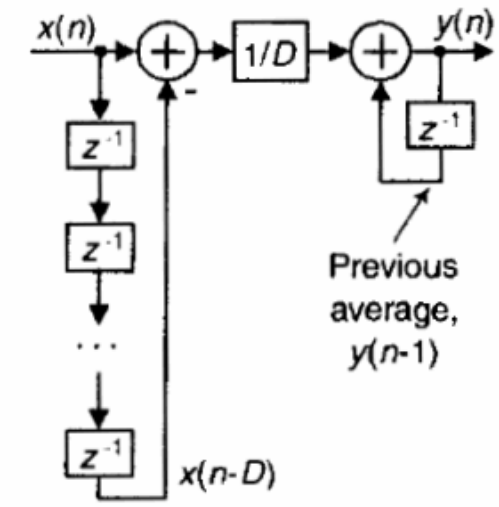
Per implementare un filtro passabasso anti-alias prima di una decimazione, si può usare il metodo della media mobile :

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j]$$

dove il calcolo viene ripetuto spostandosi di un campione in avanti ogni volta. Ovviamente se calcolato effettivamente in questo modo sarebbe altamente inefficiente. Un metodo molto migliore è quello della somma ricorsiva :

$$y[i] = y[i-1] + x[i+p] - x[i-q]$$

dove si somma l'ultimo campione arrivato alla somma precedente, e si sottrae il campione più vecchio



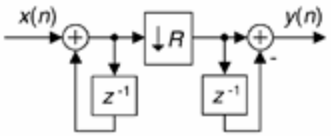
Un filtro a media mobile di questo tipo viene detto "Boxcar filter" perché la sua risposta all'impulso è rettangolare, come un garage per auto…

by Alberto I2PHD

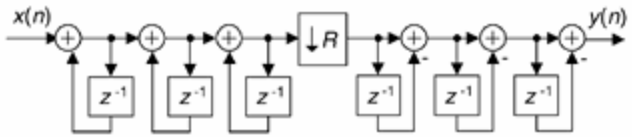
Il CIC, questo sconosciuto

Avendo a che fare con operatori LTI (Linear Time Invariant), è possibile invertire l'ordine di esecuzione delle operazioni, ed inserire nel mezzo la decimazione per il fattore voluto R.
Se questo fattore è uguale al numero di termini della media mobile, ecco che le cose si semplificano… e poi, volendo un filtraggio più spinto, è possibile aumentare il numero di operatori, ovvero incrementare il cosiddetto "ordine" del filtro.

Ordine 1



Ordine 3



Cos'è che ho ottenuto in pratica ? Uno stadio di integrazione, e poi, dopo il decimatore, un comb filter con un solo termine di delay. Quindi il tutto viene detto

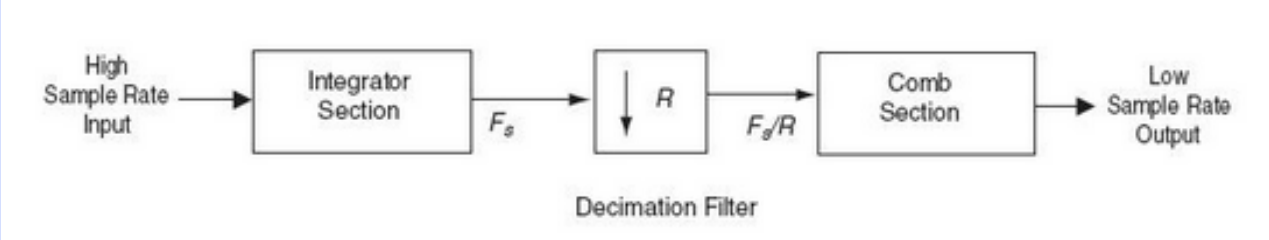
Cascaded Integrator Comb

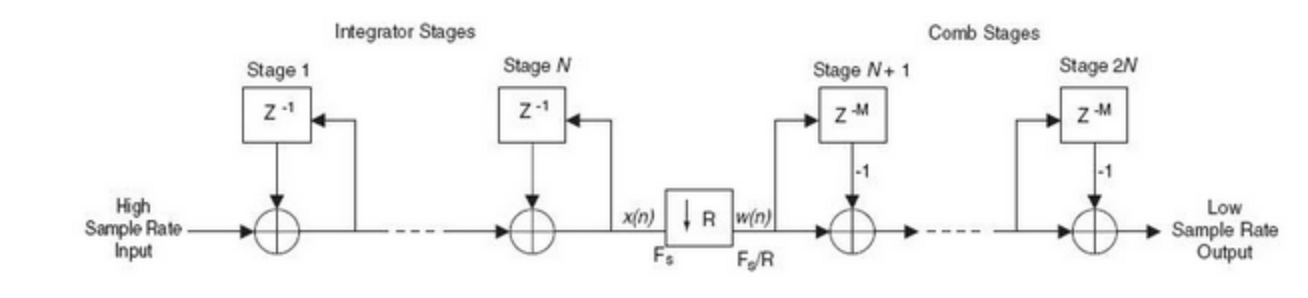
Ovvero un **CIC** ….

by Alberto I2PHD

Il CIC, questo sconosciuto

Abbiamo visto che uno dei modi di implementare un Boxcar filter seguito da decimazione è il cosiddetto CIC, cioè Cascaded Integrator Comb. Non bisogna attribuire al termine CIC quello più generale di filtro Boxcar, perché il CIC è solo uno dei metodi per implementarlo.





by Alberto I2PHD

Il CIC, questo sconosciuto

Il CIC ha il grande vantaggio di poter essere implementato senza moltiplicazioni, servono solo somme e sottrazioni. Questo lo fa essere il filtro preferito dagli scrittori di firmware per le FPGA, che notoriamente sono sempre un po' a corto di moltiplicatori.

Però ha un problemino… il primo stadio di integrazione è a tutti gli effetti un IIR, cioè un filtro a risposta all'impulso infinita, che può non essere stabile. Basta considerare ad esempio come si comporta l'integratore con un input di valore costante… il suo output diverge verso l'infinito…

Se però lavoriamo in aritmetica modulo 2, quella solita usata nei processori con registri interi di lunghezza fissa, il problema diventa un non problema… infatti, anche se si verifica un overflow, poi questo viene compensato nel successivo stadio di comb (a patto che la base di rappresentazione numerica abbia un numero sufficiente di bit…). La dimostrazione si può trovare su molti testi di Digital Signal Processing, e non sto certo qui a ripeterla…

Sì… però… bisogna lavorare in aritmetica fissa, cosa certo possibile con l'ARM Cortex M4, però ormai è stata fatta la bocca al sapido gusto di poter lavorare in floating point…

Come si può fare ?

by Alberto I2PHD

Il CIC a decomposizione polifase

Riesaminiamo bene cosa stiamo facendo… vogliamo filtrare con un boxcar prima di una decimazione. Il CIC è solo un particolare modo, molto efficiente, di implementare un boxcar e una decimazione combinati. Ma ce ne possono essere altri…

Un boxcar, se lo scrivo con la notazione "z" tipica degli algoritmi DSP, e tralasciando inessenziali fattori di scala, lo si può rappresentare in questo modo :

$$H(z) = 1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} + \ldots\ldots Z^{-R+1}$$

Le cose si fanno interessanti se R è una potenza di 2, cioè se $R = 2^j$

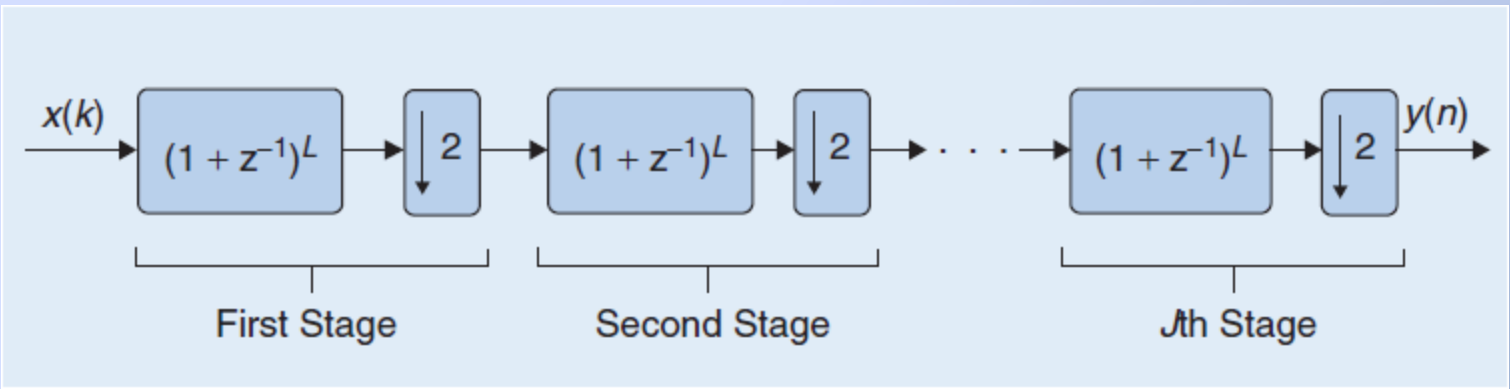
Supponendo che l'ordine del filtro sia L, allora si può fare una fattorizzazione della funzione di trasferimento come mostrato nella slide successiva.

by Alberto I2PHD

Il CIC a decomposizione polifase

Fattorizzazione della funzione di trasferimento di un filtro Boxcar, quando la decimazione è per un fattore potenza di 2

$$H(z) = (1 + z^{-1})^L(1 + z^{-2})^L(1 + z^{-4})^L(1 + z^{-8})^L \ldots. (1 + z^{-2^{j-1}})$$



"L" è l'ordine del filtro, e il fattore di decimazione è 2^J

Il CIC a decomposizione polifase

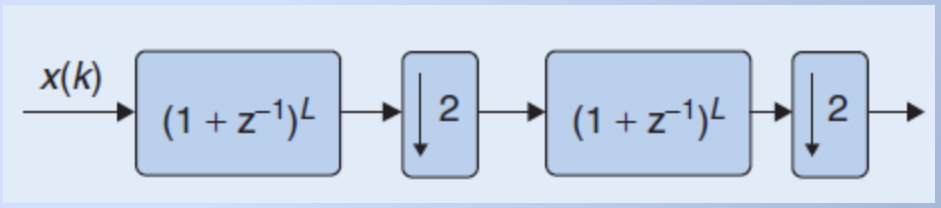
Nel caso specifico, ci sono due Boxcar, implementati con decomposizione polifase. Esaminiamo il secondo, una decimazione per 4, con filtro del 4° ordine.

La funzione di trasferimento è semplicemente :

$H(z) = (1 + z^{-1} + z^{-2} + z^{-3})^4$ che si può scrivere così :

$H(z) = (1 + z^{-1})^4(1 + z^{-2})^4$ seguita da una decimazione per 4

A questo punto possiamo scomporre la decimazione per 4 in due decimazioni per 2, e usare le cosiddette “noble identities” per invertire una delle due con una sezione del Boxcar :



La seconda sezione non è più funzione di z^{-2}, bensì di z^{-1} perché è preceduta dalla decimazione

Il CIC a decomposizione polifase

Vediamo ora come si può decomporre ciascuna delle sezioni di tipo $(1 + z^{-1})^4$

$$\mathbf{(1 + z^{-1})^4 = 1 + 4z^{-1} + 6z^{-2} + 4z^{-3} + z^{-4}}$$

Fattorizzando z^{-1} sulle potenze dispari di z abbiamo :

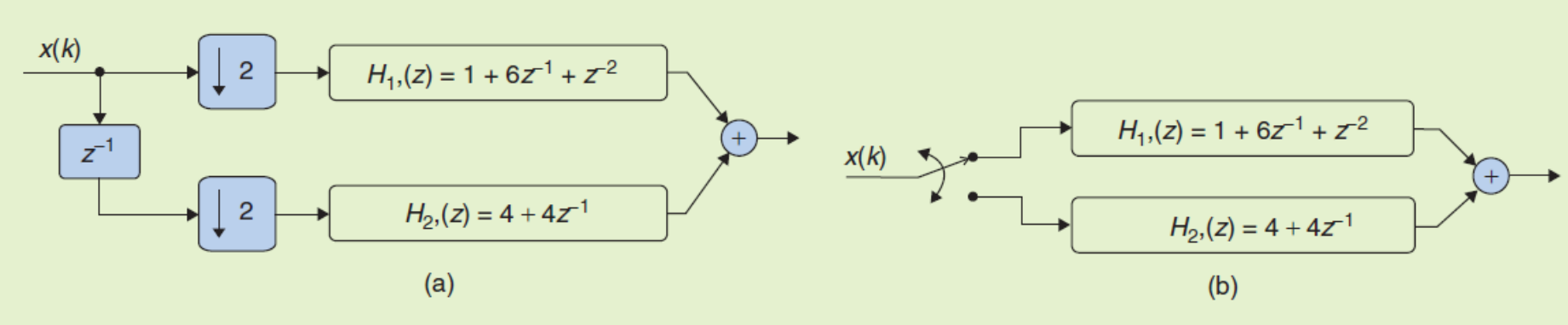
$\mathbf{1 + 6z^{-2} + z^{-4} + z^{-1}(4 + 4z^{-2})}$, cioè $\mathbf{H1(z) + H2(z)z^{-1}}$ dove

$\mathbf{H1(z) = 1 + 6z^{-2} + z^{-4}}$ e $\mathbf{H2(z) = 4 + 4z^{-2}}$

A questo punto possiamo spostare, sempre grazie alle “noble identities”, la decimazione per 2 in fronte al filtro, quindi le potenze di z vengono divise per 2, e tenendo anche conto del ritardo z^{-1} applicato a una delle due funzioni di trasferimento, si arriva alla struttura mostrata nella schermata successiva…

Il CIC a decomposizione polifase

Decomposizione polifase di un filtro Boxcar (che non si può più chiamare CIC…)



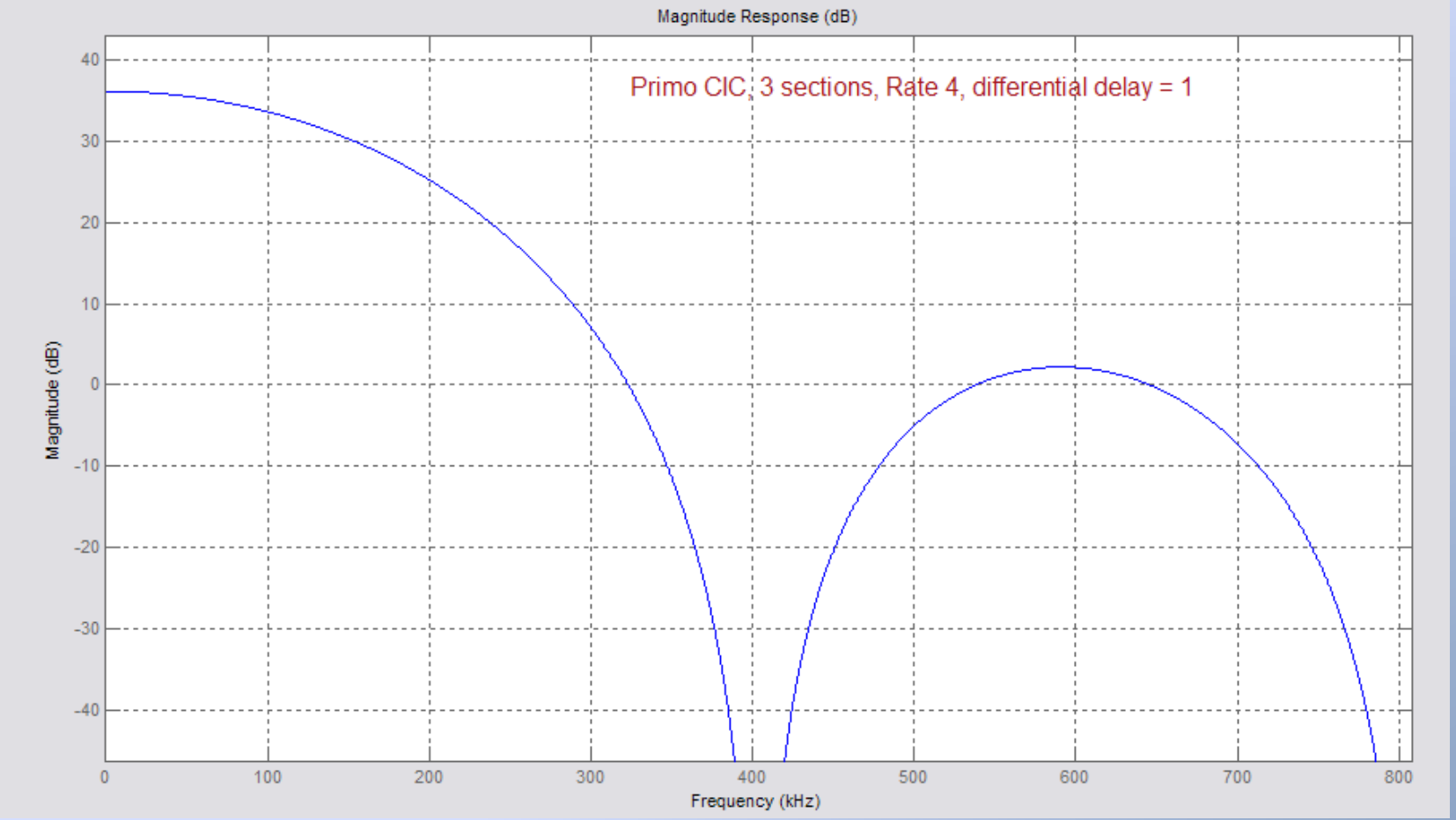
Non ha elementi ricorsivi, nessun integratore, quindi non ci sono rischi di wraparound, e questo consente di implementarlo in aritmetica floating point, cosa impossibile da farsi con una implementazione CIC di tipo classico.

In pratica si dirottano i campioni dispari alla sezione H1 e quelli pari alla sezione H2

by Alberto I2PHD

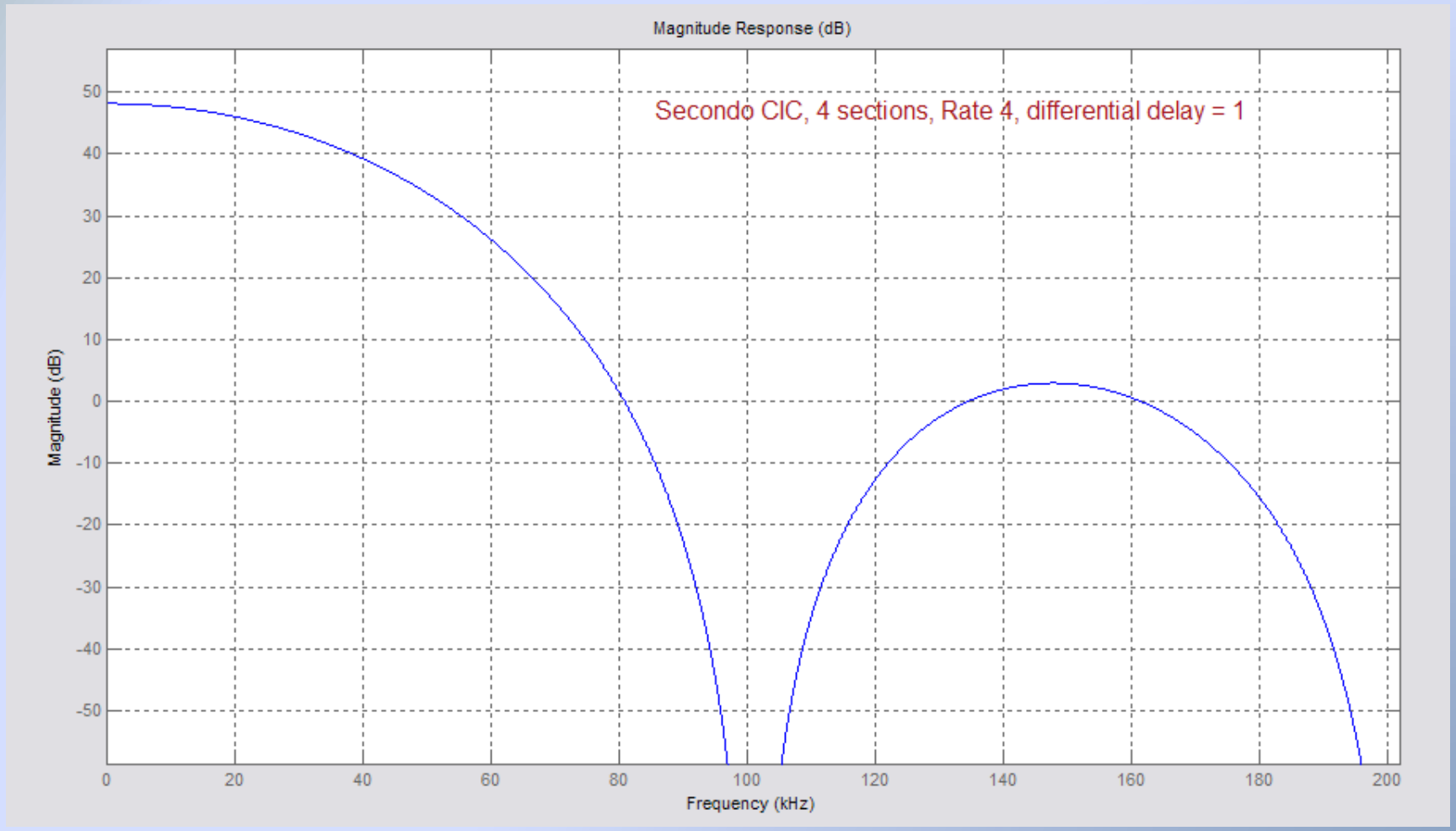
Il CIC a decomposizione polifase

Risposta del primo Boxcar (ho commesso anche io l'errore di chiamarlo CIC nella didascalia…). Il primo null è ad un quarto della frequenza di campionamento, visto che il Boxcar ha rate 4 (e ordine 3)



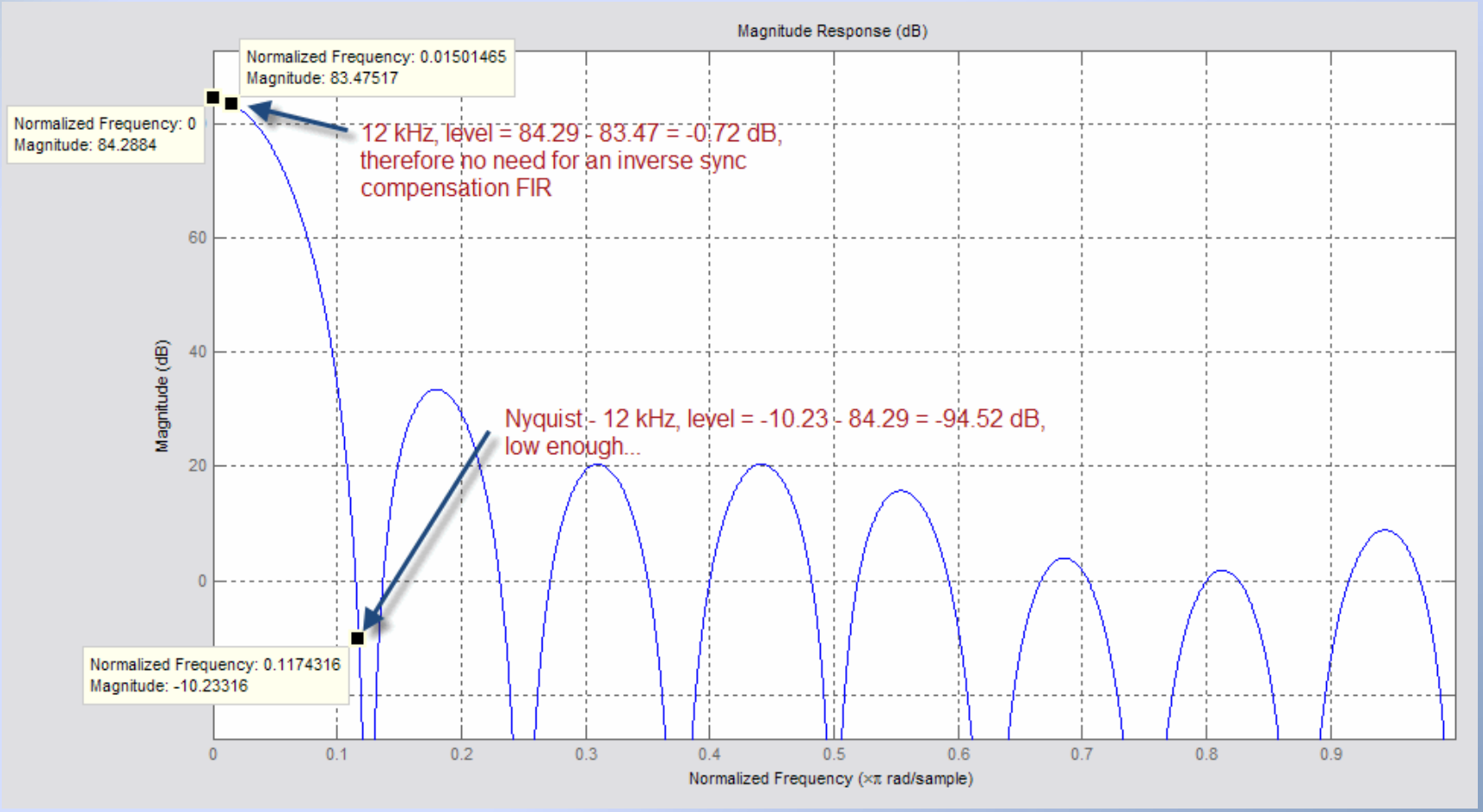
Il CIC a decomposizione polifase

Il secondo Boxcar, anche lui con rate 4, del quarto ordine



Il CIC a decomposizione polifase

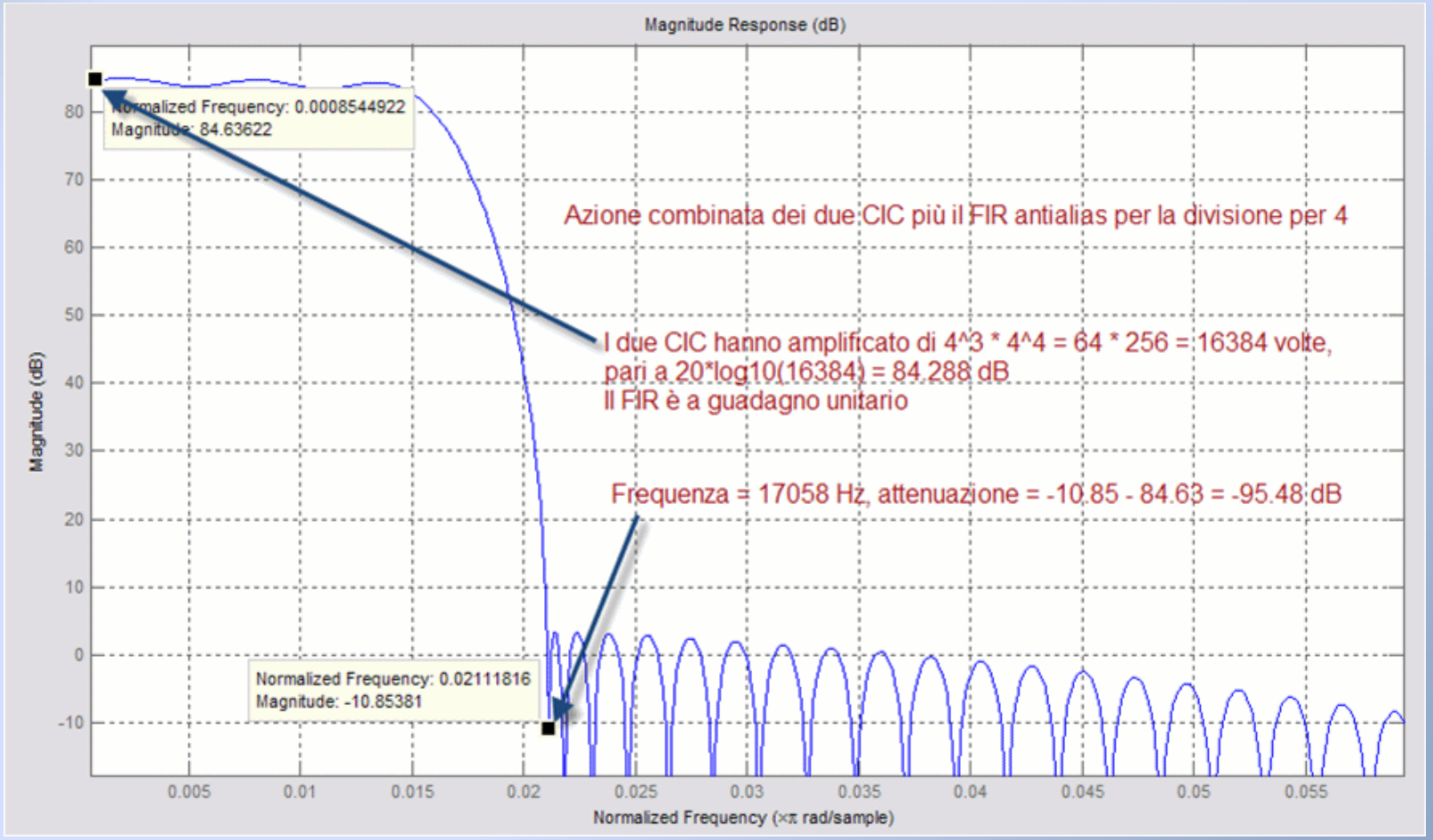
Risposta dei due Boxcar in cascata.
Vista la ridotta banda di interesse (12 kHz) il droop di tipo sinc è così ridotto che non serve neppure un FIR di compensazione



by Alberto I2PHD

Il CIC a decomposizione polifase

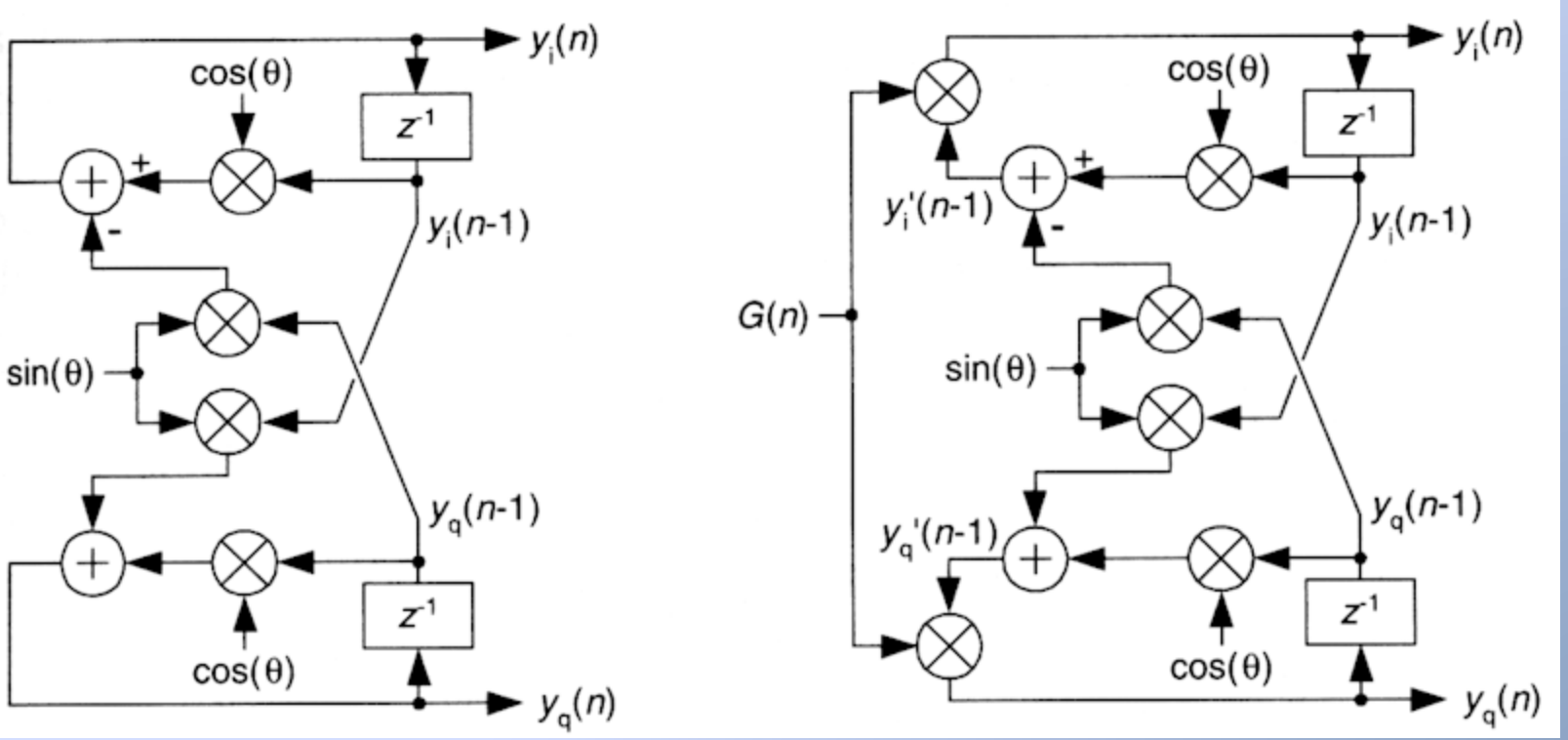
Comunque poi un FIR anti alias serve, visto che divideremo successivamente per 4.
E male non fa neppure per migliorare ulteriormente la reiezione del CIC… pardon… del Boxcar



by Alberto I2PHD

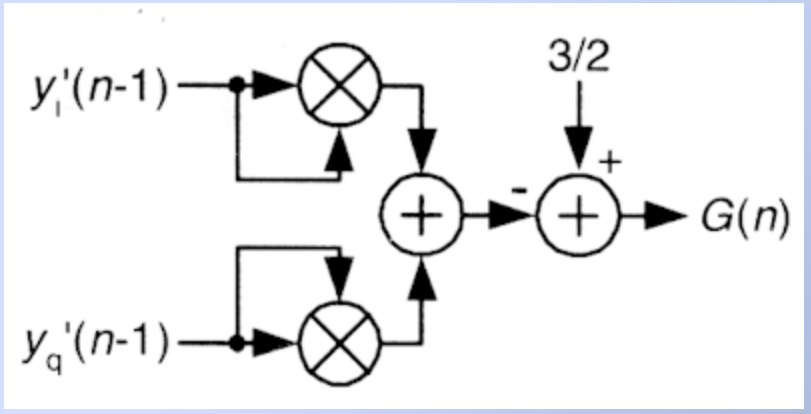
L'oscillatore locale numerico (NCO)

L'NCO è stato implementato con tecnica IIR, facendo cioè oscillare un filtro a risposta all'impulso infinita, in modo controllato. Però all'atto pratico serve un controllo del livello.. a me è successo che dopo un quarto d'ora il segnale lentamente si abbassava… sono diventato matto per capire perché … ☺



L’oscillatore locale numerico (NCO)

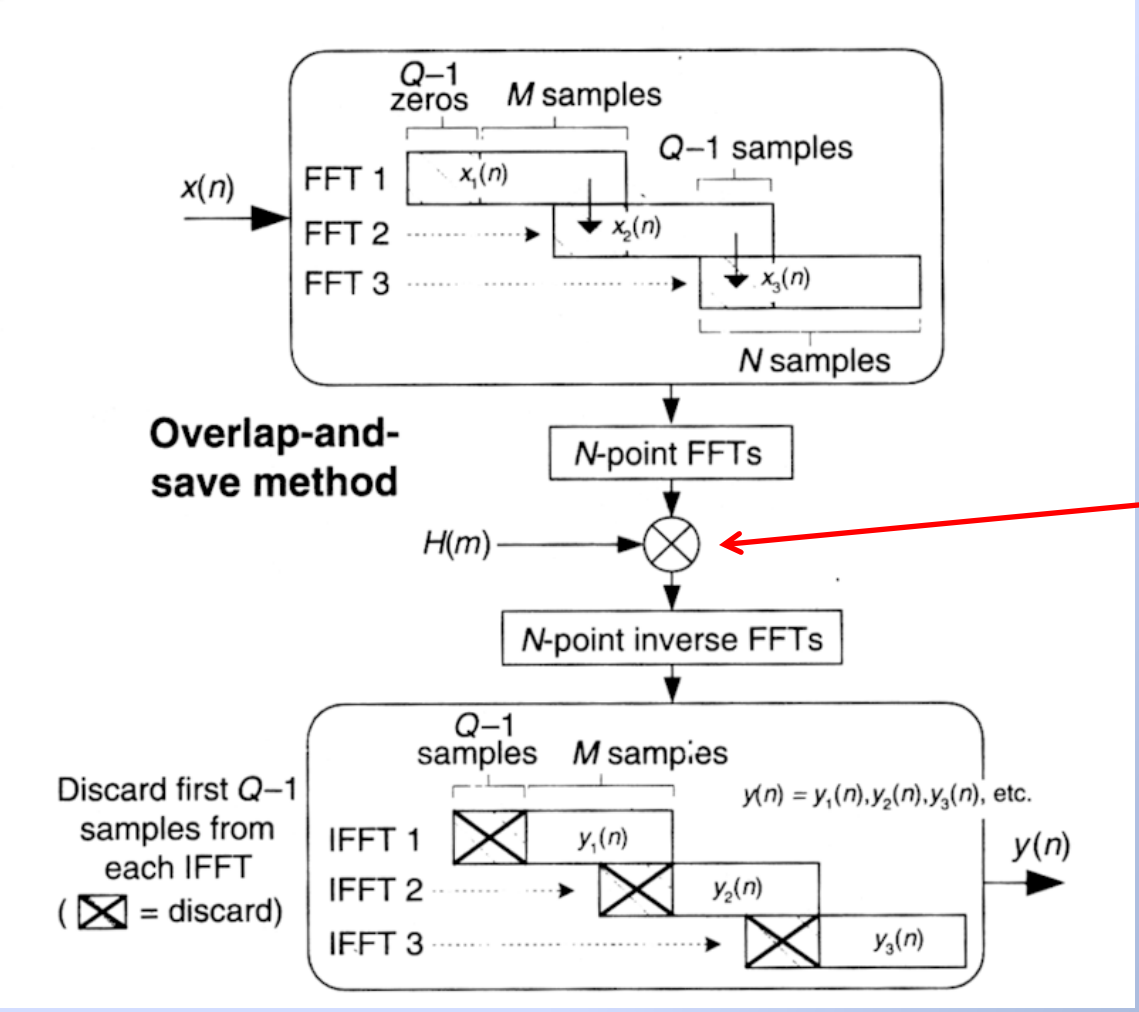
Il parametro “G” della slide precedente lo si può calcolare in modo semplificato nella seguente maniera. In pratica si sommano i quadrati di I e Q, trovando così il modulo del fasore ruotante, e si sottrae la somma da 1.5 (o da un valore simile). Questo genera un valore di G che stabilizza l’ampiezza delle due componenti I e Q.



Poi, anche se non è mostrato nella figura, il valore G(n) viene smussato tramite un IIR con costante di tempo molto lunga, nell’ordine dei secondi.

Filtraggio passabanda tramite fast convolution

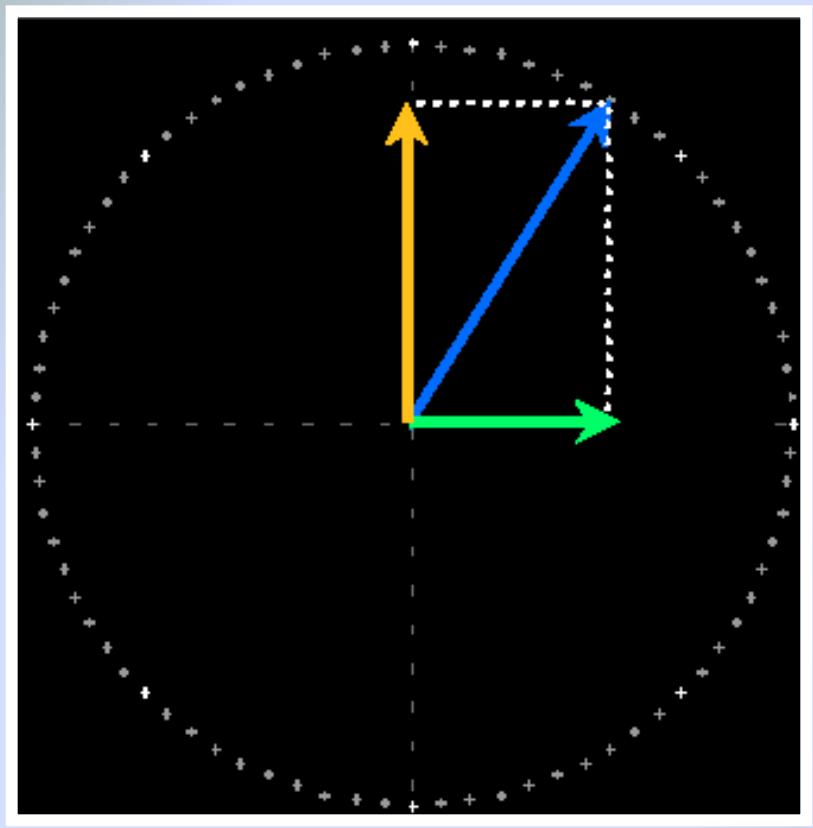
Un po' complicato da spiegarsi in poche parole… fidatevi se vi dico che funziona !



Il kernel della risposta all'impulso usata per la generazione della maschera H(m) che moltiplica la FFT è di 769 punti, che sono una ottima garanzia per la ripidità dei fianchi

by Alberto I2PHD

Il demodulatore AM



La modulazione di ampiezza altro non e' che la variazione istantanea del modulo del vettore descritto dalle due componenti I e Q

Quindi basta applicare il Teorema di Pitagora…

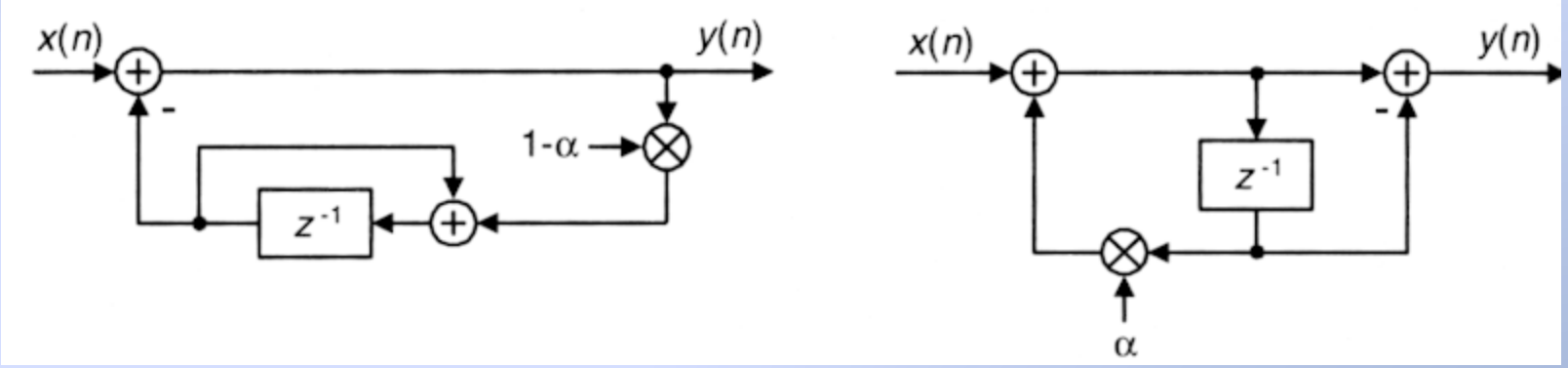
$$A = (I^2 + Q^2)^{0.5}$$

(e poi magari mettere un bel passa-alto per eliminare la DC…)

by Alberto I2PHD

Rimozione della DC con un notch a zero Hz

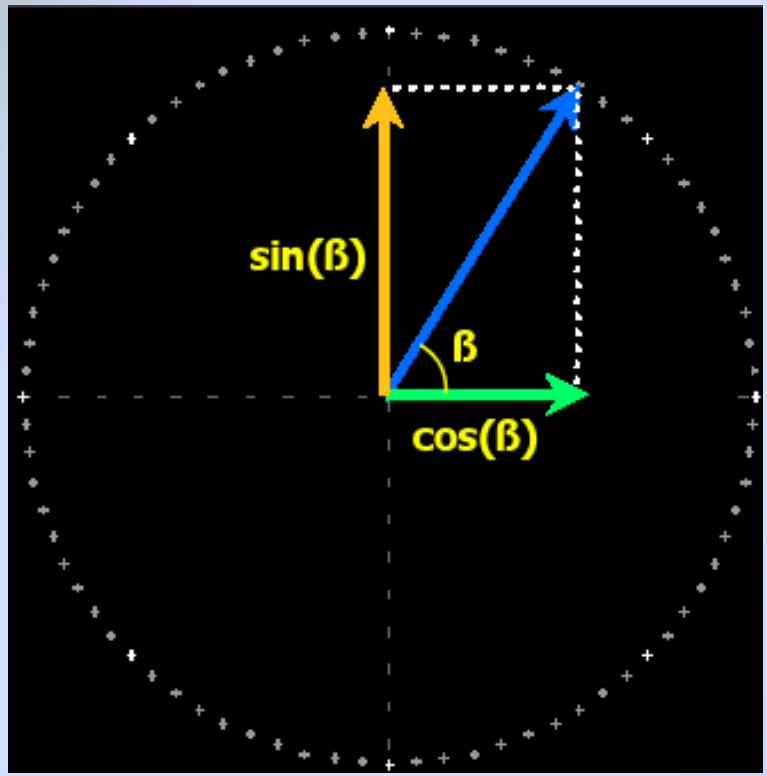
I due schemi sono equivalenti …



Entrambi hanno questa funzione di trasferimento

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 - z^{-1}}{1 - \alpha z^{-1}}$$

Il demodulatore FM



La modulazione di frequenza altro non e' che la variazione istantanea della velocita' di incremento dell'anomalia del vettore descritto dalle due componenti I e Q

Quindi basta applicare una formuletta trigonometrica

ß = arctan(Q/I)

E poi differenziare rispetto al tempo, per passare dalla fase alla frequenza (e poi anche qui serve un bel passa-alto)

by Alberto I2PHD

Il demodulatore SSB

La SSB altro non e' che banda fonica traslata nello spettro, quindi basta rimetterla al suo posto. Pero', come tutti i segnali pertinenti alla fisica e non all'astrazione matematica, e' un segnale reale, con parte immaginaria nulla.

Quindi bisogna ritrasformare il segnale da analitico (I e Q) a segnale reale

Il demodulatore SSB

Si possono usare due metodi :

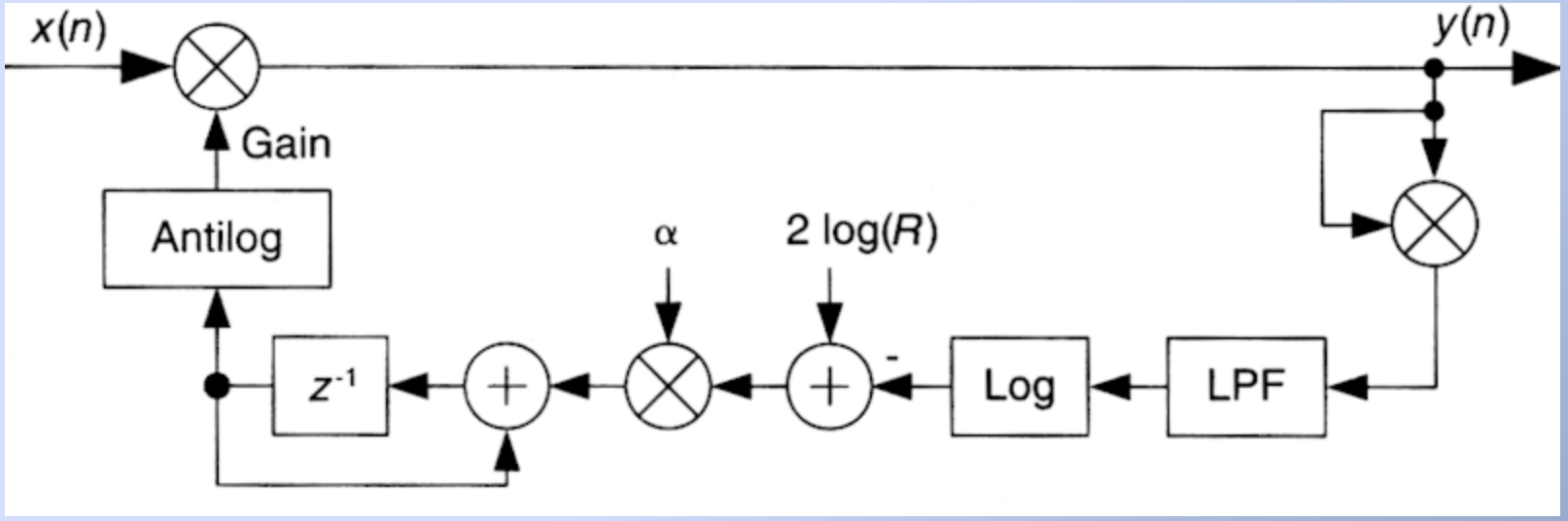
- **Operando nel dominio del tempo, ovvero un filtro FIR accoppiato a un trasformatore di Hilbert (in analogia a quello che si faceva ai tempi dell'hardware…)**

- **Operando nel dominio della frequenza, ovvero lo specchiamento coniugato in frequenza, seguito da una IFT reale**
  (questo è il metodo da me preferito e implementato…)

by Alberto I2PHD

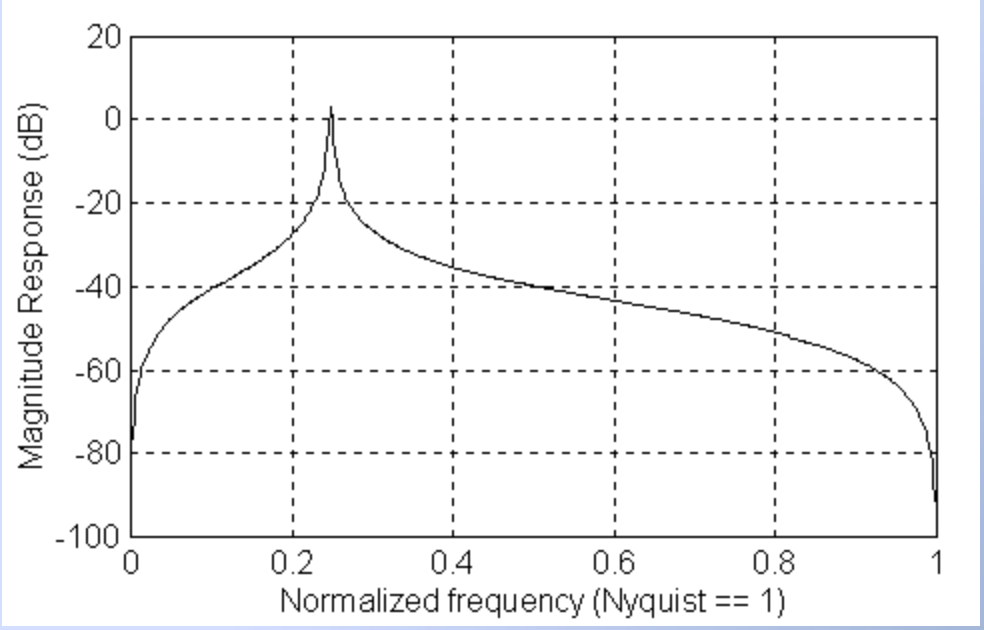
In conclusione …..

Quelli descritti sono i blocchi principali della radio SDR “a braccio”, ovvero implementata nel processore ARM. Ci sono altri blocchi ancora da completare, come ad esempio un AGC decente :



In conclusione …..

Così come vanno completati i varii algoritmi di denoising (il buon vecchio Widrow – Hoff completo di leakage sembra essere ancora il migliore), e di noise blanking. Anche un buon CW peaking rimane sempre un preferito da chi pratica il CW, ed è già stato implementato :



Che altro non è che un paio di poli avvicinabili a piacere al cerchio unitario sul piano Z, e un paio di zeri sull'asse reale. Ma l'effetto è veramente buono…

E il risultato finale è ……

by Alberto I2PHD

FDM-DUO, presto sui vostri schermi !

by Alberto I2PHD