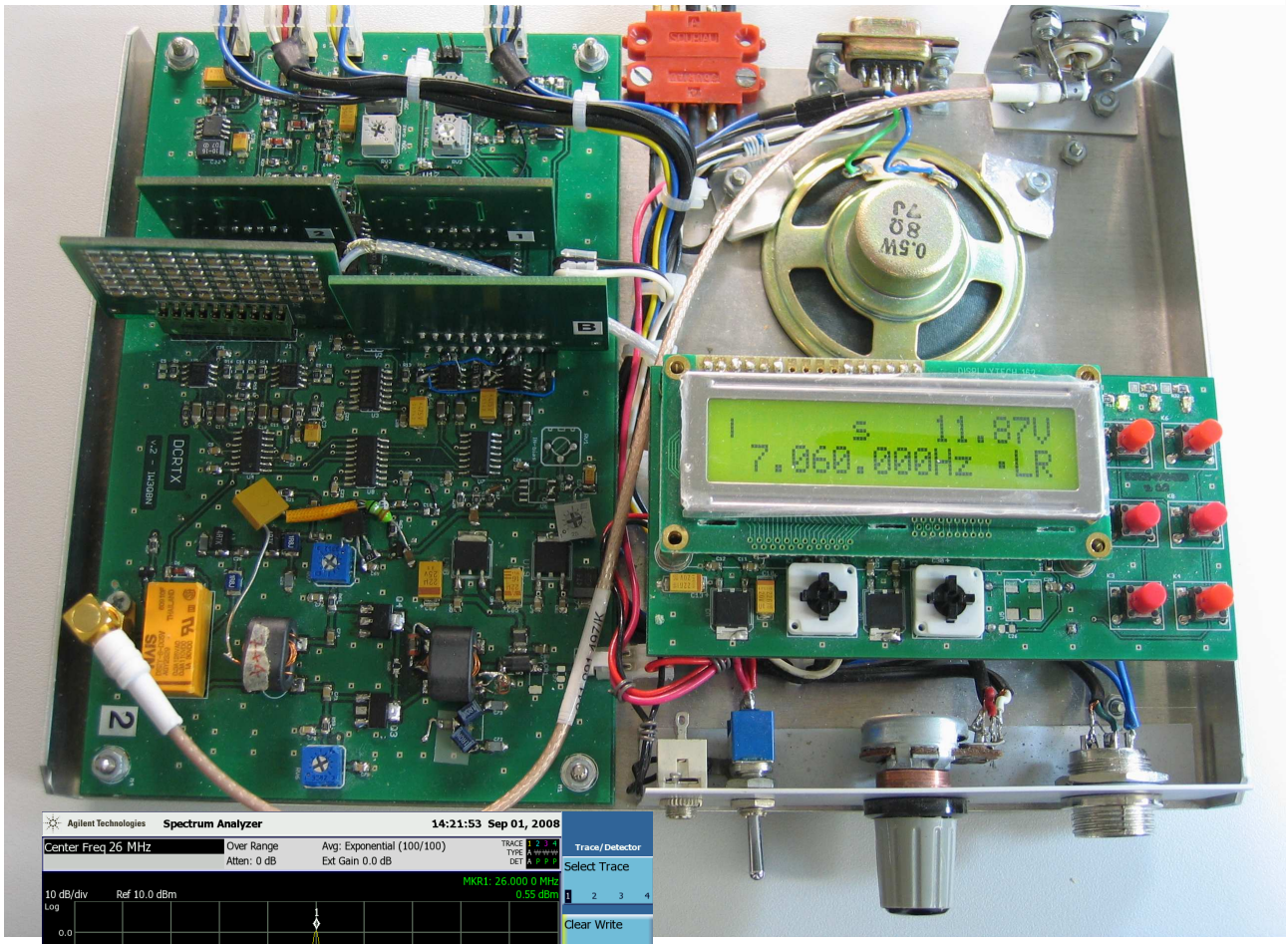
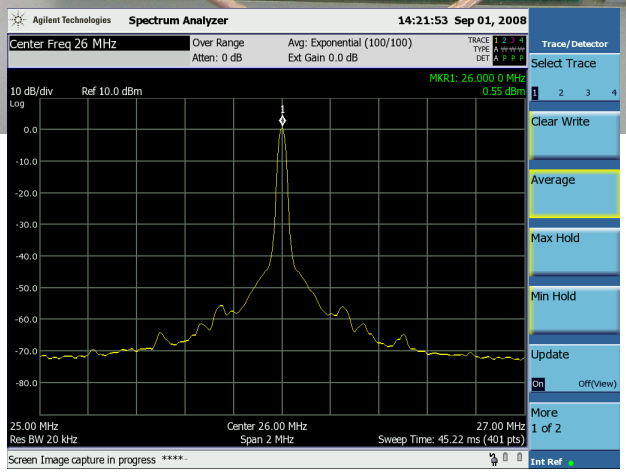
AMSAT-I news

Volume 17, Numero 5 Settembre/Ottobre 2009

Software Defined Radio





e DCRTX

(Simple Designed Radio)

Numero dedicato quasi interamente alla tecnica digitale; teoria e pratica

In questo numero:

ARISSsat-1	p2
Analisi di un ricevitore SSB SDR realizzato con GNURadio	
Prima parte: Front-end e USRP	p3
DCRTX “verso” la radio digitale #1	p8
Complessità computazionale	p12
Bombardamento lunare	p14
Notiziario Aerospaziale	p15

Una sintetica ma precisa descrizione del prossimo "satellite" che verrà lanciato dalla ISS.

ARISSsat-1

Francesco de Paolis - IK0WGF

ARISSat-1 è il nome assegnato all'insieme riconfigurato del progetto SuitSat-2 in una nuova struttura (più ampia). Grazie al fatto di essere stato pensato come sistema modulare, il team è riuscito facilmente ad adattare le varie parti per sistemarle in una nuova struttura.

Avendo perso la possibilità di usare la tuta spaziale russa Orlan (come inizialmente previsto) che doveva contenere SuitSat-2, il gruppo SuitSat-2/ARISSat-1 ha reagito e risposto alla situazione, e guidato da Bob Davis, KF4KSS, ha sviluppato una nuova struttura in cui alloggiare, con modifiche minime o addirittura senza modifiche, i moduli del satellite che erano già disponibili, realizzati e collaudati.

I moduli principali di ARISSat-1 sono:

⇒ Safety circuit and arming switches - sistema di sicurezza ed interruttori di avvio: ora è montato sulla parte superiore della struttura (+Z).

⇒ IHU enclosure - contenitore della IHU: reingenierizzato per permettere una migliore posizione dei cavi e loro posizione.

⇒ IHU (Integrated Housekeeping Unit) - computer principale: rimane lo stesso.

⇒ SDX (Software Defined Transponder) - transponder digitale basato sul sistema DSP: rimane lo stesso.

⇒ PSU (Power Supply Unit) - alimentatore: scheda modificata separando le funzioni in due parti (e due schede) diverse, 1) Max Power Point Tracker e 2) Power Supply.

⇒ ICB (Interconnect Board) - scheda di interconnessione: ridisegnata per adattarli ai connettori ed alla nuova disposizione dei moduli.

⇒ TX e RX - moduli TX ed RX: semplicemente separati dall'antenna e dal contenitore della parte video.

⇒ 2m/70cm antenna: - separata e ridisegnata come due monopoli.

⇒ Video cameras - telecamere: separate dalla struttura elettronica (era l'elmetto della tuta) e montate sulla nuova struttura.

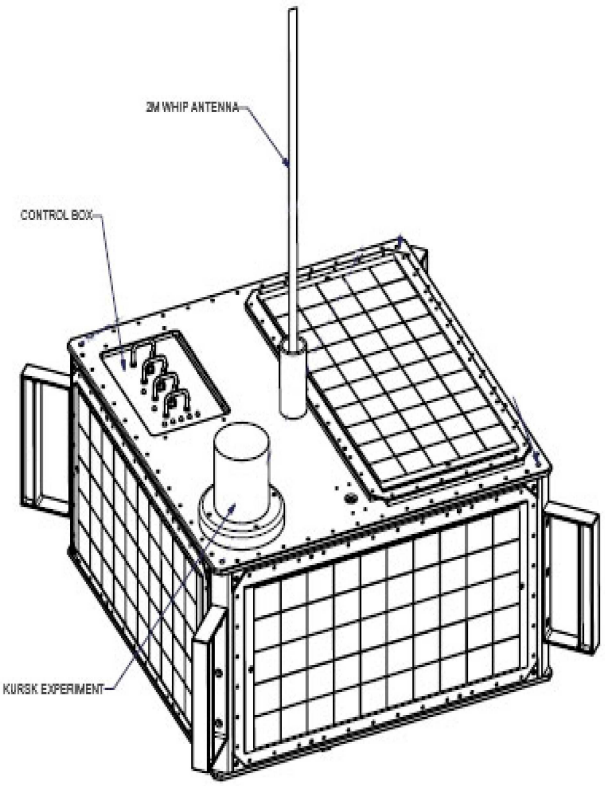
⇒ Russian Ag-Zn battery - batterie russe allo Zinco-Argento: ora montate su una piastra all'interno della struttura.

⇒ Kurst State University Experiment: ora montato direttamente sulla struttura del satellite.

⇒ Solar Panels - pannelli solari: ora montati su tutti i lati della struttura (inizialmente venivano fissati con del velcro agli stivali della tuta).

La struttura completa di ARISSat-1 è riportata nel disegno qui sotto.

Quella che vedete è stata sviluppata dal nulla da Bob Davis, KF4KSS, e dal suo gruppo, in appena 9 settimane!



AMSAT-I News, bollettino periodico di ***AMSAT Italia***, viene redatto, impaginato e riprodotto in proprio. Esso viene distribuito a tutti i Soci.
La Redazione di ***AMSAT-I News***, è costituita da:
Paolo Pitacco, IW3QBN
Segreteria
Francesco de Paolis, IK0WGF
Hanno collaborato a questo numero:
Francesco de Paolis, IK0WGF
Alberto Trentadue, IZ0CEZ
Fabio Azzarello, IW8QKU
Giovanni Lorusso, IK7ELN
copertina:
Foto grande: prototipo del ricetrasmettitore HF descritto in questo numero [IW3QBN]
Foto piccola: spettro del segnale generato (26MHz) dal DDS del ricetrasmettitore [IW3QBN]

Analisi di un ricevitore SSB SDR realizzato con GNURadio

Prima parte: Front-end e USRP

Alberto trentadue - IZ0CEZ

Distruttivo e Costruttivo

La maggior parte degli appassionati di Radio ed Elettronica in generale si sono avvicinati alla sperimentazione diretta di queste tecniche con un approccio distruttivo. La forte curiosità di sapere “cosa c'è dentro” ci armava di strumenti adatti allo “scasso” meccanico, al fine di poter vedere con i nostri occhi e toccare con mano la composizione di un circuito elettronico, per ritrovare nel concreto nozioni che erano rimaste un po' vaghe dalle lezioni di scuola o dalla lettura di manuali.
Molto spesso, soprattutto per quelli meno “delicati” dal punto di vista manuale, questo processo di analisi era senza ritorno... ovvero comportava la rottura di qualcosa che, nella migliore delle ipotesi, si limitava all'involucro esterno o a qualche manopola. Ma davvero poco importava, perchè l'appagamento del desiderio di vedere dentro e fare esperienza diretta compensava l'inconveniente.
E' indubbio che molti hanno imparato tanto da queste esplorazioni. Anzi, oserei dire che l'apprendimento più efficace è sempre venuto dall'aprire la scatola e scomprre il circuito nei suoi dettagli. Possiamo allora dire che la fase costruttiva della nostra esperienza in campo radio-elettronico, cioè l'essere in grado di costruire qualcosa a partire da un progetto, è resa possibile anche da una serie di esperienze distruttive precedenti, oltre che da una buona base teorica.
Se questo processo è così efficace, direi che potremmo provare a fare la stessa cosa anche con una nuova tecnologia del nostro tempo, con lo stesso spirito di allora e la stessa prospettiva di poter realizzare poi, autonomamente.
Ecco il motivo fondamentale di esplorare in dettaglio una realizzazione Software Defined Radio - SDR. In questo caso il “cacciavite” è sostituito dalla ricerca della documentazione e dalla scomposizione del sistema SDR in parti più elementari da cui è possibile isolare un comportamento di tipo numerico.
Inoltre, abbiamo ora in ambito SDR, una piattaforma che non ha bisogno di molti “cacciaviti” per aprirsi perchè è già “aperta”, o meglio “*open*”, di per sé e si presta naturalmente ad esplorazioni a diversi livelli di dettaglio, fino ad arrivare alla singola riga di codice.
In questo articolo “apriremo” un ricevitore SSB realizzato con GNURadio/Companion e ne analizzeremo in dettaglio i componenti principali, sia hardware che software. Per componenti software intendo i blocchi funzionali dell'elaborazione numerica del segnale (DSP); perciò questa esplorazione ci permetterà di parlare di soluzioni tipiche di questa tecnologia e, per quanto possibile, appropriarcene per usi futuri.

Struttura del ricevitore

Il ricevitore che analizzeremo è realizzato mediante l'architettura di GNURadio [1] che ho avuto modo già di introdurre su AMSAT-Italia News [2].
Questa architettura prevede il campionamento diretto nella banda delle HF ed eventualmente una prima conversione a monte del campionatore nel caso si voglia coprire bande superiori. Per focalizzarci sulla parte SDR, considereremo il ricevitore in HF.
Ricordo poi che l'architettura di GNURadio include l'hardware di acquisizione come parte integrante del sistema. Questo hardware è la Universal Software Radio Peripheral – USRP, una scheda molto flessibile in termini di configurabilità, dotata di diversi canali RX e TX e controllata da una FPGA di cui anche il firmware potrebbe essere oggetto di modifica.
Bisogna tenere presente che esistono anche altre architetture SDR, alcune molto più semplici, in cui per esempio l'hardware ha il solo compito di traslare una fetta di banda nella banda audio gestibile dal codec di una scheda audio (vedi per es. le schede SoftRock e il software WinRad, quest'ultimo utilizzabile con diverse schede di acquisizione esterne). Perciò, mentre l'analisi delle funzioni della USRP potrà essere utile soprattutto a coloro che sono interessati a GNURadio in particolare, l'analisi del software eseguito dal PC affronta problematiche comuni a tutte le implementazioni SDR.
Fatta questa premessa, possiamo iniziare con un rapido sguardo di insieme allo schema funzionale del ricevitore SSB HF realizzato con GNURadio, con maggiore dettaglio, per ora, al lato front-end/USRP.

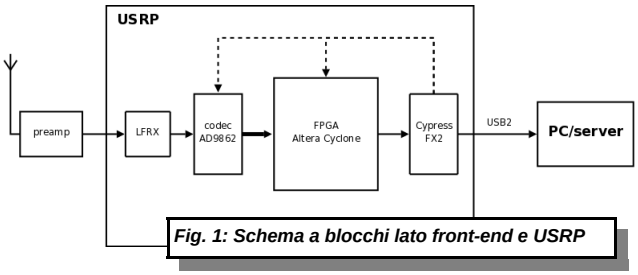


Fig. 1: Schema a blocchi lato front-end e USRP

La ***daughter board*** (scheda figlia) LFRX costituisce l'interfaccia verso la RF ed ha in generale il compito di presentare alla **USRP** un segnale avente specifiche caratteristiche. L'uscita della daughter board viene campionata da convertitori A/D ed acquisita numericamente dalla **FPGA**.
La logica della FPGA ha un duplice obiettivo: multiplare più canali RX e ridurre la banda del segnale campionato ad una frequenza tale da rendere possibile il trasferimento al PC via **USB2.**
I campioni così ottenuti vengono inviati all'elaboratore (PC o server) e costituiscono i dati su cui lavoreranno i diversi algoritmi (**blocchi**) DSP che realizzano le funzioni di demodulazione e trattamento del segnale, secondo il modo operativo prescelto. Nel caso del ricevitore SSB, oltre alla demodulazione, sarà necessario filtrare il segnale e presentarlo opportunamente al driver della **scheda audio** per la riproduzione sonora.
Entriamo ora nel dettaglio di questa architettura, cominciando dalla sezione RF e di acquisizione.

La sezione RF e di acquisizione (USRP)

Questa sezione del sistema è specifica dell'architettura GNURadio e viene implementata dalla USRP e dalle sue *daughter boards*.

La daughter board LFRX

Trattandosi di un ricevitore HF, la *daughter board* adatta allo scopo è la **LFRX**. La produttrice della USRP, la Ettus Research Ltd. offre una vasta scelta di *daughter boards* per diverse applicazioni [3]. La LFRX copre la banda 0-30 MHz e gestisce due canali in ricezione. Ricordando che la USRP campiona una banda ampia fino ai 64MHz, si capisce come la LFRX sia molto semplice perchè non deve operare alcuna conversione.
Il segnale proveniente da ciascuno dei due ingressi di antenna (50Ù sbilanciati) va in ingresso all'**AD8132**, che è un amplificatore differenziale con uscita differenziale, così come mostrato dalla fig.2, che descrive una delle due sezioni di front end della LFRX.

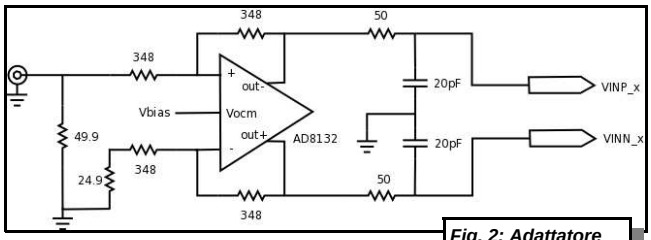


Fig. 2: Adattatore d'ingresso di LFRX.

L'AD8132 viene usato a guadagno unitario con funzione sia di disaccoppiatore dalla linea di antenna che come driver della linea differenziale d'ingresso al convertitore A/D ad alta velocità.
Da notare come questa sezione non può essere considerata come l'analogo dell'amplificatore di front-end dei ricevitori analogici. Questo ha due conseguenze importanti per la ricezione da antenna:

1 Un filtro analogico in ingresso è necessario, altrimenti il rumore a larga banda raccolto dal driver renderà il segnale di ingresso inutilizzabile; esso dovrà lasciar passare il segmento di banda di interesse. Nel caso in esame potrebbe essere tra i 100 e i 500 kHz

2 Servono almeno 30dB di guadagno a monte della USRP, da realizzare con amplificatore a basso rumore.

L'uscita differenziale del driver viene poi filtrata da una rete differenziale passabasso ad un polo che riduce ulteriormente il rumore e adatta il segnale alla sezione di conversione successiva.

La conversione A/D

I convertitori A/D ad alta velocità hanno normalmente ingresso differenziale in quanto il segnale differenziale è soggetto a minore distorsione e minore rumore. I due segnali bilanciati provenienti dalla LFRX sono presentati in ingresso al convertitore A/D montato sulla scheda madre della USRP. I convertitori A/D sono dual-channel e poichè la USRP dispone di due connettori a *daughter board* in RX, troviamo due di questi convertitori A/D sulla USRP.

La conversione A/D viene realizzata non già da un semplice convertitore A/D, ma piuttosto da un **Mixed Signal Front-End (MxFE)** della Analog Devices, l'**AD9862,** un codec programmabile con interfaccia seriale (SPI) che gestisce 2 canali RX e 2 canali TX ad alta velocità. Quindi ciascun MxFE gestisce anche la conversione D/A nel cammino di trasmissione.

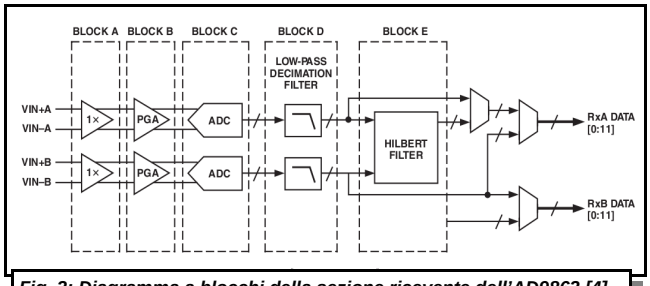


Fig. 3: Diagramma a blocchi della sezione ricevente dell'AD9862 [4] .

Concentrandoci sulle sue funzioni di RX (vedi fig.3) , possiamo riassumere così le sue funzionalità principali:

1 Ingressi differenziali con banda passante di 140MHz
2 Ingressi dotati di buffer e amplificatori programmabili fino a 20dB di guadagno.
3 Conversione A/D a 12 bit.
4 Una sezione programmabie di decimazione per 2 (half band) seguita da un filtro di Hilbert per ottenere on-chip una reiezione delle frequenze immagine fino a -38dB. In questo caso i due canali RX sono in quadratura. Nel caso del LFRX, questa funzione non viene attivata, per cui entrambi i canali sono disponibili per la ricezione.

Quando viene attivata la USRP dall'applicazione SDR, la procedura di inizializzazione permette di settare la velocità di campionamento fino ad un massimo di 64 megacampioni al secondo (MSPS): questa velocità gestisce agevolmente tutte le bande radio da LF fino ad HF. La programmazione degli AD9862 avviene attraverso il controllore USB2 FX2 Cypress, che vedremo in seguito.
Le uscite digitali dei due AD9862, composte ciascuna da 12 bit CMOS, sono direttamente collegate agli ingressi DA_x della FPGA, che riceve così i dati campionati.

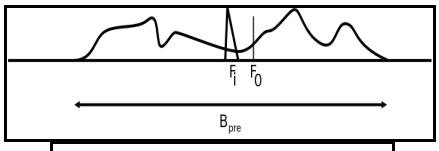


Fig. 4: Spettro in ingresso al codec.

Il segnale captato dall'antenna e campionato dal codec non è certo un segnale isolato nella banda di ricezione, per cui in generale dobbiamo considerare che lo spettro del segnale in ingresso, ed in uscita, dal codec sia riccamente popolato dal segnale di interesse e da una moltitudine di segnali indesiderati che sono comunque passati attraverso il front-end analogico a monte della USRP (il QRM che ben conosciamo...). Indichiamo con $\mathbf{B_{pre}}$ la banda passante del front-end analogico.
Ricordo brevemente che è possibile studiare nel dominio della frequenza anche i segnali campionati, considerando come spettro del segnale campionato quello che si otterrebbe realizzando il campionamento con opportune funzioni campionatrici analogiche ideali. Per esempio, si può moltiplicare il segnale analogico d'ingresso con un

treno di impulsi ideali spaziati del periodo di campionamento $\mathbf{T_s}$. Tale treno di impulsi ideali ha come spettro in frequenza ancora un treno di impulsi ideali. Ricordando la proprietà della convoluzione, si osserva che lo spettro di un segnale limitato in banda campionato ad una certa frequenza $\mathbf{F_s}$ è una sequenza di repliche dello spettro del segnale originario, spaziate di $\mathbf{F_s}$ (figura 5).

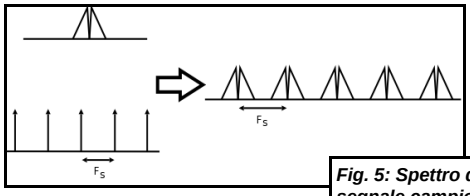


Fig. 5: Spettro di un segnale campionato.

Il ruolo della FPGA

La FPGA è il componente chiave della USRP perchè realizza la logica che rende la USRP così flessibile e ricca di possibili applicazioni. Si tratta di un dispositivo low-cost **Altera Cyclone EP1C12**, low voltage (1,5V), che dispone di oltre 12000 elementi logici, circa 240.000 RAM bit, 2 PLL onboard, 103 canali differenziali che possono operare fino a 640 Mbps.
Il compito della FPGA è quello di adattare le diverse configurazioni di flussi di dati, ottenibili dalle diverse *daughter boards* effettivamente collegate alla scheda madre, con la banda passante disponibile sulla linea USB2 verso l'host su cui gira GNURadio.
L'interfaccia USB2 supporta 480Mbit/s in full duplex, il che significa 60MB/s. Tuttavia i dati vengono trasferiti con un certo overhead dovuto all'header dei pacchetti di trasmissione e ai ritardi del firmware della FPGA. Perciò i progettisti della USRP hanno limitato la velocità complessiva massima dei dati via USB2 a **256Mbit/s**.
Ciascun canale RX invia sulla USB2 dati campionati in formato complesso (I-Q); ciascun dato è un numero a 16 bit, per cui ogni campione (complesso) utilizza 4 bytes per essere trasmesso. Per il criterio di Nyquist, la banda massima disponibile, nella migliore ipotesi di un solo canale RX, è di **8 MHz**. Questa banda deve essere ridotta in base ai canali RX effettivamente configurati (1, 2 o 4), oppure può essere ridotta per effettiva necessità, ad es. in applicazioni a banda più stretta. E questo è proprio il nostro caso della ricezione SSB. Vediamo ora in cosa consiste l'adattamento operato dalla FPGA.

Il Digital Down-Converter (DDC)

La banda dei campioni del canale RX può andare da un massimo di 8 MHz giù fino 250 kHz, in base a come viene impostata dal software GNURadio in esecuzione sull'host. Ricordando che la banda passante del codec è di 32 MHz, sarà quindi necessario selezionare una frequenza centrale ($\mathbf{F_0}$) e riportarla in banda base.

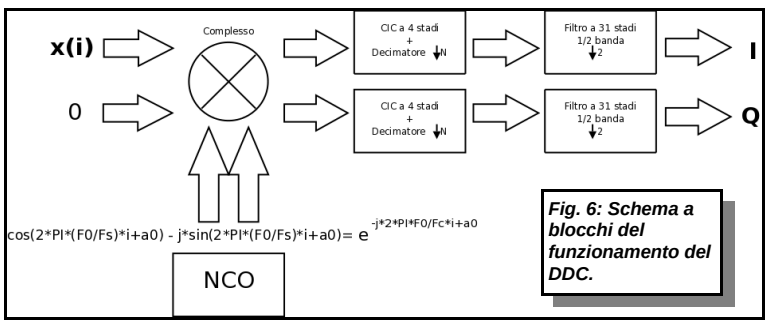


Fig. 6: Schema a blocchi del funzionamento del DDC.

Per far ciò, sul cammino di ricezione, la FPGA implementa un **Digital Down Converter (DDC)**. Il firmware può realizzare fino a 4 DDC, uno per canale RX disponibile dalle *daughter boards*.
Come si vede dallo schema a blocchi in fig.6 [5], ciascun DDC ha due ingressi e ciascun ingresso può essere connesso all'uscita di uno dei 4 convertitori A/D dei 2 codec, per realizzare multiplazioni complesse. Nel caso in esame del ricevitore SSB, un solo flusso, proveniente da un singolo canale RX, è utilizzato in ingresso al DDC.

L' NCO

Per far ciò, il flusso di campioni RX viene moltiplicato con le sinusoidi in quadratura generate dal **Numerically Controlled Oscillator (NCO)**. Di fatto si realizza un mixer, o se vogliamo, un'eterodina digitale a numeri complessi. L'NCO non è un vero e proprio DDS, ma è un sottosistema realizzato internamente alla FPGA.
Da notare, sempre in figura 6, come l'NCO calcola la variazione di fase per ogni campione con il fattore $\mathbf{F_0 / F_s}$, dove $\mathbf{F_0}$ è la frequenza centrale della banda di interesse e $\mathbf{F_s}$ è la frequenza di campionamento del segnale in ingresso **x(t)** (64 MHz nel caso più comune). $\mathbf{a_0}$ è una fase arbitraria.
L'NCO sintetizza gli andamenti di seno e coseno con l'algoritmo **CORDIC** (**Coordinate Rotation Digital Computer**). Questo algoritmo utilizza il concetto di rotazione di fase di un numero complesso così da ottenere i valori di funzioni trigonometriche senza moltiplicazioni hardware. Per questo motivo è particolarmente utile quando la realizzazione deve essere fatta su FPGA, dove i moltiplicatori costano caro ed è generalmente utile risparmiare gates il più possibile.
Nella figura 4 viene anche evidenziato come in generale la frequenza $\mathbf{F_0}$ su cui viene viene programmato l' NCO potrebbe non coincidere con la frequenza centrale $\mathbf{F_i}$ della banda di interesse. In effetti, si può scegliere di far eseguire la traslazione finale della banda di interesse sul segnale numerico dal software su PC, piuttosto che direttamente in USRP. Ed anzi, questo è spesso il caso.
L'eterodina digitale moltiplica il segnale di ingresso per $\mathbf{exp(-2j\eth i*F_0/F_s+a_0)}$**.** Questa moltiplicazione, nel dominio della frequenza, corrisponde ad una traslazione di banda verso il basso (o sinistra se preferite). Lo spettro risultante da questa moltiplicazione è quindi lo sdoppiamento dello spettro mostrato in figura 4, a causa della moltiplicazione per funzioni sinusoidali, e poi traslato di $\mathbf{F_0}$ verso sinistra.

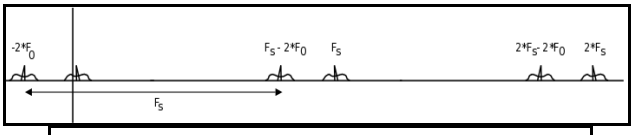


Fig. 7: Spettro del segnale campionato e mixato con l'NCO.

Dalla figura 7 si vede che tale spettro non ha più simmetria rispetto all'origine (**f=0**), perciò si tratta di uno spettro di un segnale non più reale ma complesso, che si ottiene sommando due spettri di segnali reali opportunamente sfasati. Tali segnali reali sono la componente in fase (**I**) e in quadratura (**Q**) del segnale complesso risultante.

La decimazione per N e il filtro CIC

Dopo aver traslato il segnale di interesse in banda base, è possibile ridurre la frequenza di campionamento secondo le esigenze della multiplazione sulla USB2 e/o le esigenze di banda stretta del software utilizzatore. Segnali a banda stretta sono comunque convenientemente campionati a frequenze più basse, per ridurre il carico computazionale a valle della USRP.
La riduzione della frequenza di campionamento di un fattore N (**decimazione per N**) richiede che lo spettro del segnale (nella sua versione non campionata) non abbia componenti significative al di fuori della banda di Nyquist $\mathbf{F_d / 2}$, essendo $\mathbf{F_d = F_s / N}$. Se ciò non fosse, in fase di ricostruzione, si avrebbe un effetto di **aliasing** (sovrapposizione delle immagini e dei segnali indesiderati) che renderebbe inutilizzabile il risultato della decimazione.
E' quindi necessario filtrare opportunamente il segnale campionato e farlo ad alta velocità, cioè alla frequenza di campionamento del convertitore A/D del coded ($\mathbf{F_s = 64}$ **MSPS o 64 MHz**), prima di procedere alla decimazione.
Estrarre una banda ridotta da un segnale campionato a 64 MSPS, però, non è affatto banale. La struttura dei filtri digitali classici, per ottenere la selettività richiesta, richiederebbe un numero molto grande di moltiplicatori, un bel carico di lavoro anche per un moderno processore DSP. Se ricordiamo che stiamo lavorando con una FPGA, la cosa diventa davvero troppo onerosa, per il costo in termini di elementi logici che richiederebbe ciascun moltiplicatore.
Ecco perchè viene usato un filtro **Cascaded Integator-Comb (CIC)** a 4 stadi per ciascuna componente I e Q. Il filtro CIC provvede anche alla rimozione di N-1 campioni su N dal flusso. N è un parametro passato alla FPGA in base alla larghezza di banda richiesta dal software.
I filtri **CIC** sono una soluzione efficiente per estrarre segnali a banda stretta da sorgenti a larga banda rese disponibili dai moderni campionatori ad alta velocità. Un filtro CIC a K stadi è composto da K integratori numerici (o accumulatori), un decimatore per N, e K celle di tipo *comb* (pettine o setaccio).
L'integratore numerico è a ben guardare un semplice accumulatore; il *comb* invece è un blocco che sottrae dalla sequenza i propri campioni ritardati di M passi. Si nota subito la sua estrema semplicità computazionale.

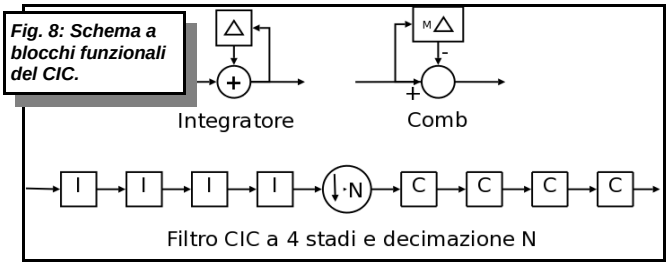


Fig. 8: Schema a blocchi funzionali del CIC.

Si può dimostrare che il modulo della risposta in frequenza di un CIC di ordine K e a decimazione N, nel dominio della frequenza campionata ridotta $\mathbf{F_d}$ (in uscita) è dato da

$$|H(f)| = \left| \frac{\sin(\frac{\pi M f}{F_d})}{\sin(\frac{\pi f}{N F_d})} \right|^K$$ che per N >>1 diventa $$\left| NM \frac{\sin(\frac{\pi M f}{F_d})}{\frac{\pi M f}{F_d}} \right|^K$$

Supponendo M=1, intorno allo 0, esso si presenta più o meno come in figura 9.

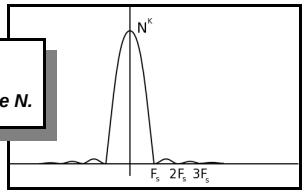


Fig. 9: La risposta in frequenza del CIC di ordine K e decimazione N.

Il lobo centrale ha un ampiezza pari a $\mathbf{N^K}$, con K l'ordine del CIC, ed il primo nullo proprio in corrispondenza di $\mathbf{F_d}$, cioè la frequenza decimata. L'ampiezza del primo lobo laterale è di circa $\mathbf{N^K * (2 / 3\eth)^K}$. Per K=4, significa un'attenuazione in ampiezza di circa 54 dB.
Risulta chiaro allora quali sono gli effetti di questo filtro sul segnale digitale in uscita dal mixer:

1 Assicurare una banda passante < $\mathbf{F_d / 2}$

2 Rimuovere tutto ciò che può generare aliasing per un segnale campionato a frequenza $\mathbf{F_d}$, è cioè posizionato nell'intorno dei multipli di $\mathbf{F_d}$.

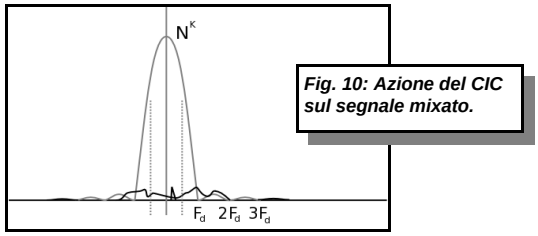


Fig. 10: Azione del CIC sul segnale mixato.

Nel caso del ricevitore SSB, conviene scegliere la minima banda possibile per il segnale RX e quindi la minima frequenza di campionamento (250 kSPS): perciò 64M / 250K = 256 e **N = 256 / 2 = 128**.
La divisione per due avviene a causa del filtro half-band successivo che, vedremo, dimezzerà ulteriormente la frequenza di campionamento. Perciò all'uscita dal CIC la frequenza di campionamento è di **500 kHz**.
I filtri CIC sono una soluzione efficiente perchè 1) sono composti solo da sommatori e ritardi e 2) perchè richiedono pochi elementi logici per la loro realizzazione. Abbiamo ottenuto quindi, con una struttura molto semplice per la realizzazione in FPGA, l'eliminazione delle immagini e, allo stesso tempo, la decimazione per N.

Il CIC ha anche l'importante pregio di avere struttura e forma della risposta in frequenza indipendenti dal fattore di decimazione N, per cui permette decimazioni **programmabili**, come avviene proprio in GNURadio, con la possibilità di controllare la banda dei dati in ingresso alla USB.
Un approfondimento sul CIC lo si può trovare in [6].

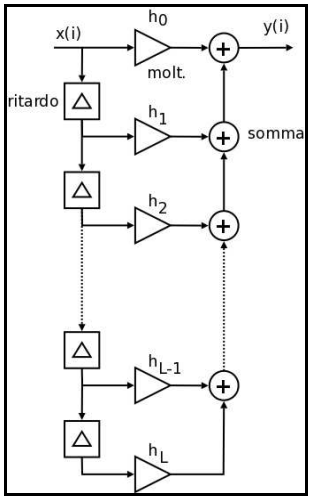


Fig. 11: Struttra a blocchi dell'algoritmo di un FIR.

L'Halfband Filter

Il CIC da solo, però non basta. L'andamento del modulo come

$$\mathbf{| \sin(x)/x |^K}$$

non è certo piatto nella banda di interesse: valori superiori

di K aumentano sì l'attenuazione a $\mathbf{F_d}$, ma accentuano pure la pendenza della risposta nella banda passante.
Da qui segue la necessità di filtrare ulteriormente il segnale in uscita dal CIC per equalizzare quanto più possibile questo comportamento. Questo risultato lo si può ottenere con tecniche classiche di sintesi di filtri FIR.
Ricordo brevemente che i filtri FIR (**Finite Impulse Response**) sono quei filtri digitali che rispondono con una sequenza limitata nel tempo all'**impulso unitario digitale**, la sequenza composta dal solo numero 1 al tempo 0. Partendo da questa definizione, si può dedurre che un generico FIR, in risposta all'impulso unitario, genera un'uscita che è la risultante di L di stadi (detti "**tap**") posti in cascata, o meglio in *pipeline*, ciascuno composto da un moltiplicatore e da un ritardo (vedi fig. 11)
I fattori **h(i)** per i=0..L sono proprio la sequenza di risposta all'impulso unitario (risposta impulsiva).
L'ordine **L** del filtro FIR determina la cosiddetta banda di transizione, cioè l'intervallo di frequenza in cui il filtro passa dal guadagno massimo allo zero: valori più grandi di L danno un filtro più ripido e selettivo, ma più costoso computazionalmente. Le tecniche di sintesi dei FIR permettono anche di definire i fattori h(i) in base alla forma della risposta in frequenza desiderata.
Nel caso del CIC, la forma dovrà essere inversa a $\mathbf{|sin(x)/x|^K}$.
In realtà più è ampio l'intervallo di frequenza nel quale vogliamo controllare tale risposta in frequenza, maggiore è l'ordine L del FIR. Si dimostra che per equalizzare una banda di $\mathbf{F_d / 2}$, servirebbero 63 stadi.
Ancora una volta il fatto di lavorare con una FPGA richiede di minimizzare quanto possibile le moltiplicazioni. Allora si ricorre a due accorgimenti.
Il primo è quello di associare a questo filtro una ulteriore decimazione per 2, così da dimezzare la banda da equalizzare – e dimezzare l'ordine del FIR a **31** stadi. Questo spiega perchè, in precedenza, si lascia che il CIC decimi di un fattore 2 in meno rispetto al necessario: il CIC non dipende affatto dal fattore di decimazione, il FIR sì.

Il secondo accorgimento è quello di utilizzare un FIR di tipo **halfband**. Un filtro halfband ha il modulo della risposta in frequenza antisimmetrico rispetto alla frequenza **F / 4** , essendo F la frequenza di campionamento a cui esso opera. Se H(2jðf) è la risposta in frequenza del FIR, questa proprietà si esprime con:

$H(2j\pi f) + H(2j\pi(F/2-f)) = H(0)$ per $0 = f = F/2$

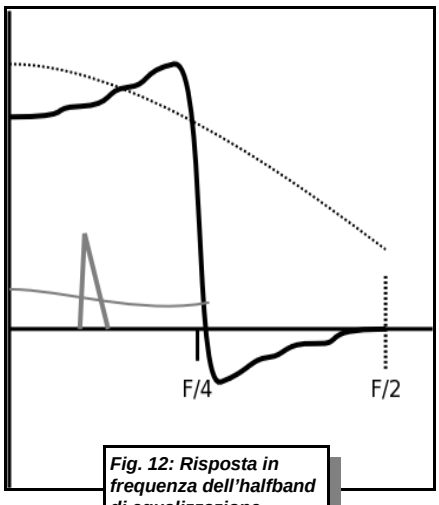


Fig. 12: Risposta in frequenza dell'halfband di equalizzazione.

Considerazioni di carattere analitico, che non riporto, fanno sì che un filtro halfband abbia tutti i fattori **h(i) = 0** per indici **i** pari. Cioè, solo metà delle L moltiplicazioni deve essere realizzata. Perciò questa struttura è quella che, a parità di ordine L, ha il minor numero di moltiplicatori. Nel caso in esame del DDC, solo 16 dei 31 moltiplicatori dovranno essere implementati per ciascuna componente del segnale complesso.
Quindi filtro FIR halfband realizza con il minor numero di moltiplicazioni la giusta equalizzazione a valle del CIC. La figura 12 mostra la risposta del filtro complementare alla risposta del CIC (tratteggiata) all'interno della banda $\mathbf{F_d/4}$, che è la banda utile dopo l'ulteriore decimazione per 2.

In uscita dalla USRP

Ricapitolando la nostra analisi del ricevitore SSB, programmando la USRP con una decimazione complessiva pari a 256, in uscita dal DDC abbiamo un segnale, campionato ad una frequenza $\mathbf{F_{out}}$ = 250 kHz, che è la traslazione in banda base del segmento di banda centrato dalla frequenza selezionata $\mathbf{F_0}$. Tale segmento di banda è ampio approssimativamente la metà di $\mathbf{F_{out}}$, cioè 125 kHz. Perciò la scelta di $\mathbf{F_0}$ deve essere tale da includere nel segmento il segnale di interesse alla frequenza $\mathbf{F_i}$ (quando $\mathbf{F_i}$ **?** $\mathbf{F_0}$).
I campioni in uscita dal DDC sono presentati nelle componenti in fase ed in quadratura secondo lo schema: $\mathbf{I_1\ Q_1\ I_2\ Q_2\ I_3\ Q_3}$....

La gestione dell'interfaccia USB2 è delegata al controllore **Cypress FX2** che gestisce tutta la comunicazione USB tra USRP e host.
I campioni sono resi disponibili al software sul PC dal driver della USRP che viene attivato da un opportune chiamate a librerie che sono parte integrante di GNURadio. Le chiamate astraggono la USRP, permettendo di considerare solo la sequenza numerica del segnale in arrivo da essa.

Da questo punto in poi siamo nel mondo del software. (*continua...*)

Riferimenti:

[1]: Sito di GNURadio http://gnuradio.org/trac

[2]: "Introduzione a GNURadio" su AMSAT-Italia News di Marzo/Aprile 2009

[3]: Sito della Ettus Research Ltd. http://www.ettus.com/

[4]: Datasheet Analog Devices AD9862 – www.analog.com

[5]: Firas Abbas - The USRP under 1.5X Magnifying Lens! - v1.0 – July 2008 – www.guradio.org

[6]: Richard Lyons – Understanding cascaded integrator-comb filters – Free published on www.design-reuse.com/

[7]: Haddad-Parsons – Digital Signal Processing – Computer Science Press - 1991

Avete mai pensato ad una radio "diversa"? Questo articolo vi farà sicuramente incuriosire.

DCRTX
"verso" la radio digitale - #1

Paolo Pitacco - IW3QBN

Premessa

Non mi considero un "digitale" anche se ci lavoro ogni giorno, ma penso che qualcosa da raccontare, di "digitale" ce l'ho ancora.
Da un pò di tempo si sente parlare di SDR (e avrete letto sicuramente gli articoli interessanti di Alberto, IZ0CEZ, pubblicati su questo Bollettino) e ormai ci sono già molti apparati in commercio, attorno cui aleggia ancora un pò di diffidenza e di mistero; questo è dovuto al fatto che ogni cosa "nuova" viene vista come un problema prima che un'occasione di miglioramento.
Nel campo delle radio digitali alla novità dei concetti si aggiunge la (tragica) necessità di avere conoscenze ben superiori a quelle dell'OM medio, sia per la programmazione che, soprattutto, per la matematica che sta alla base del nuovo approccio al mondo della radio e delle comunicazioni in generale.
Non bastano però le conoscenze, servono (di solito) componenti o strumenti moderni e ... costosi oppure di difficile reperibilità per l'OM normale.
Allora ho pensato di affrontare l'argomento al modo "antico" che è lo stesso dello scalatore di montagne; lentamente con passo misurato.
Vedrete come si può arrivare in cima lo stesso!

La mia idea

L'obiettivo che mi proponevo era la realizzazione di un sistema digitale che fosse "al passo con i tempi" ma ovviamente raggiungibile e, soprattutto, comprensibile.
Quest'ultima considerazione è stata preziosa in quanto ho deciso che per arrivare avrei *dovuto* e *voluto* comprendere le cose nuove che il mondo dell'elaborazione digitale porta in sè.
Ho iniziato il lavoro nella primavera del 2007, senza fretta, e dopo aver ottenuto i primi risultati, spinto dalle domande degli amici di AMSAT Italia, ho mandato delle foto a Francesco, IK0WGF, che le ha pubblicate (assime a dei semplici commenti) sul web AMSAT-Italia, nell'area tecnica, sezione denominata DCRTX (che probabilmente qualcuno avrà già visto).
Per un pò di tempo, a causa del lavoro e di altre cose che hanno attirato la mia attenzione realizzativa, l'ho abbandonato, riprendendolo e completando recentemente.
Il nome è, come al solito, un acronimo e sta per **D**irect **C**onversion **R**eceiver **T**ransmitter (la **X** l'ho aggiunta perchè stava bene ad indicare la funzione di ricetrasmettitore).
Ovviamente l'avete capito, si tratta di un ricetrasmettitore per onde corte (da 0 a 15MHz), a conversione diretta, con un paio di watt di potenza d'uscita, ma "senza" bobine, e senza computer per farlo funzionare.
In realtà, la prima idea era di poter dimostrare come avrebbe funzionato una IF digitale per il transponder lineare che proposi per l'uso sulla ISS alcuni anni fa, ma poi l'ho modificata perchè ritenevo più importante disporre di una base per eventuali convertitori per le bande superiori da usarsi con nuovi satelliti (ci spero sempre).

Il percorso

Sono sempre stato dell'idea che se si vuole usare al meglio una radio, uno strumento, un computer, si deve conoscere com'è fatto o perlomeno sapere come funziona. Nel caso della SDR (Software Defined Radio) siamo al massimo della complessità, quindi ho pensato di prenderla con calma, lentamente.
Girando in internet e leggendo varie riviste ci si può fare un sacco di idee sulla complessità delle cose, ma poche su come affrontare la "salita" (escludo da questo gli articoli di Alberto, IZ0CEZ, che hanno invece dimostrato il contrario).
L'ho fatto anch'io, e quindi ho pensato ad una mia strada (che non è detto sia buona o cattiva), suddividendola in queste tappe:

1 realizzare un sistema base per iniziare

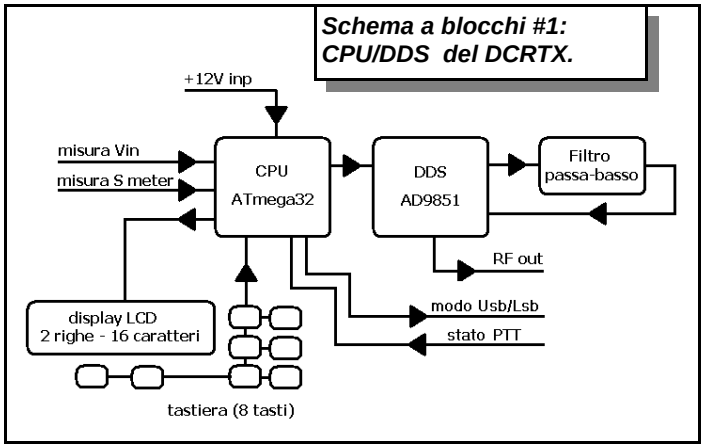
2 modificare alcune parti per migliorare

3 sostituire le parti possibili con sistemi attuali

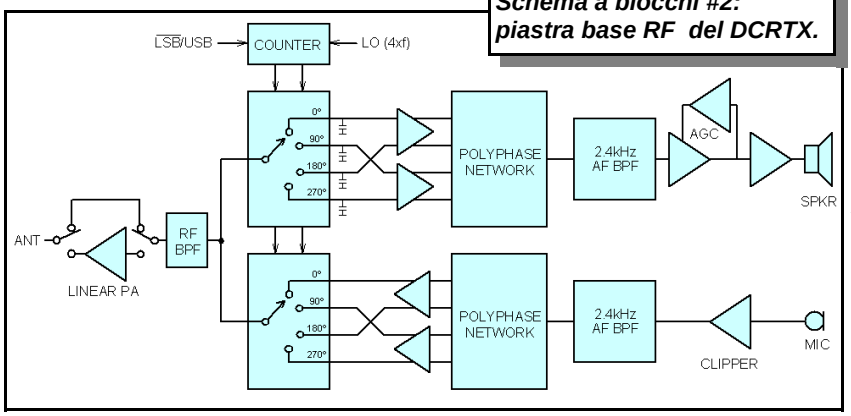
Per ora ho completato solo la prima delle tre, ma dispongo di una base "solida" su cui lavorare, ed è quella che descriverò in questa serie di articoli (è una serie perchè la lunghezza complessiva supererebbe le pagine del Bollettino, e poi è meglio andare con calma!).

Schema a blocchi del sistema base DCRTX

Cercando una via per iniziare, ed un modo poi per avanzare, ho scoperto un'interessante lavoro di due OM finlandesi, OH7SV e OH2NLT, ed ho pensato di rifarlo usando materiali e soluzioni che avevo disponibili.
Si trattava di un progetto in cui un microcontrollore (PIC nel loro caso) controllava un generatore digitale (DDS) che a sua volta pilotava dei mixer particolari (by Tayloe) per ottenere dei segnali che, mediante filtri polifase (by Gingell) venivano sommati opportunamente per ricostruire il segnale audio da demodulare (o, in senso inverso, per modulare). Nell'insieme si faceva quindi parzialmente uso di tecniche digitali, accompagnate da parti analogiche ma di notevole semplicità ed efficacia: un'ottimo inizio.
Si trattava di sistemi che non avevo mai preso in considerazione (anche se ne avevo letto su varie pubblicazioni) e quindi l'interesse è aumentato.
Ho suddiviso anch'io il sistema, come i due OM finlandesi, in due parti: una piastra di controllo/visualizzazione con generatore DDS (schema a blocchi #1) ed un'altra "analogica" con i circuiti RF (schema a blocchi #2).
In questo articolo descriverò solo la prima, rimandando al prossimo la descrizione della seconda. Per entrambe ho disegnato e realizzato i circuiti stampati, in modo da poter riprodurre, al caso, il lavoro (e per verificare che sia fattibile).



Schema a blocchi #1: CPU/DDS del DCRTX.



Schema a blocchi #2: piastra base RF del DCRTX.

Scheda CPU/DDS

E' stata la prima ad essere realizzata e provata, in quanto serve da comando e controllo dell'intero ricetrasmettitore.
La scheda contiene infatti un microcontrollore della Atmel ed un generatore DDS (Direct Digital Syntesis) della Analog Devices, funge da supporto sia al modulo LCD (un due righe, sedici caratteri, retroilluminato) che alla tastiera, composta da 8 tasti.
Dallo schema a blocchi #1 potrete notare quali siano i componenti principali della scheda e come siano connessi tra loro; analogamente, si vedoni quali segnali (pochi a dire il vero) vadano da questa alla scheda base RF.
La disposizione dei componenti, LCD e tastiera, è visibile in figura 1, mentre il circuito sottostante al display (cioé il micro ed il DDS), molto più "popolato", è visibile in figura 2.
Lo schema elettrico è riportato nelle figure 3 (micro) e 4 (DDS), e dovreste notare che il mio intento era di non fare una cosa "specifica" ma di avere un circuito per poter sperimentare con i microcontrollori Atmel, anche cose diverse dal DCRTX.
Questo circuito stampato ad esempio, può essere utilizzato così com'è per avere semplicemente un generatore RF in grado di produrre segnali sinusoidali da 0 (cc) a 25MHz (il limite è dato da alcune "regole" e dalla precisione che volete mantenere), oppure, senza il DDS, può essere usato per controllare il puntamento delle antenne (scrivendo il software adatto, tipo LVB traker), oppure come controllore di carica per batterie (era una cosa utile perchè mi serviva).
Vediamo un pò nel dettaglio lo schema elettrico della figura 3 (che è molto semplice).
Il micro (un ATmega32 della Atmel) occupa praticamente gran parte dello schema, ed è il "cervello", su di esso arrivano e partono tutti i fili di collegamento dal/al DDS (lato destro) e da/a LCD e tastiera (a sinistra in alto).

Il modulo LCD è del tipo standard Hitachi (esattamente è un Displaytech) e per risparmiare fili (ovvero bit del micro), ho usato l'interfacciamento a soli 4 bit, mentre un trimmer permette di regolare il contrasto del modulo LCD per la migliore visualizzazione.
La tastiera è a 8 tasti perchè ho utilizzato la tecnica di multiplexing tra i fili dati LCD e due fili aggiuntivi, risparmiando (4+2=6).
Alcune porte del micro le utilizzo per "leggere" dei segnali analogici (per ora solo Smeter e tensione d'alimentazione), mentre la parte di programmazione "in circuito" è visibile in centro nella parte inferiore dello schema.
Sotto al micro, vedrete anche due transistor che ho inserito per avere la possibilità di collegare il circuito ad un'interfaccia RS232 (non indicata nello schema a blocchi); ho pensato che così si può utilizzare "da remoto" l'apparato, se non altro per controllare il segnale ricevuto, la tensione d'alimentazione, il modo d'emissione e la frequenza (ovviamente).
Altra connessione che si vede sopra l'ingresso dell'alimentazione, è relativa ad un'interfaccia I^2C che ho predisposto per comunicare (lettura/scrittura) verso moduli di convertitori o transverter esterni al circuito (anche questa non è indicata nello schema a blocchi).

Nello schema di figura 4 è riportata la parte DDS, che fa uso del componente AD9851 della Analog Devices, praticamente tutto il generaore RF è racchiuso in pochi millimetri quadrati!
Tutti i controlli arrivano a questo componente dal micro (segnali DDSxx), che usa un clock del tipo da PC (modulo oscillatore).
Sullo stampato ho previsto l'uso di due diversi tipi di

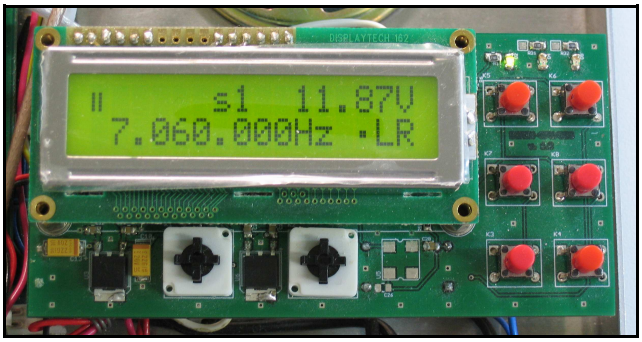
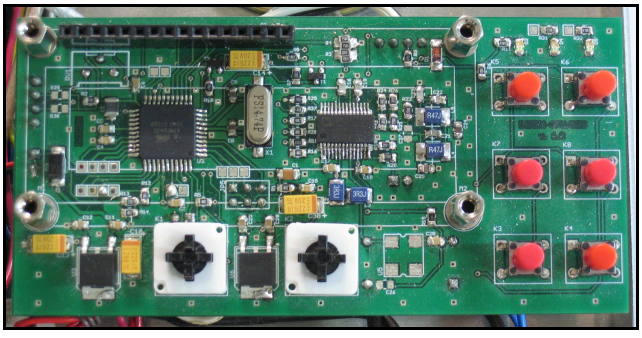


Fig 1: piastra CPU/DDS, vista tastiera e LCD.

Fig 2: piastra CPU/DDS, vista tastiera e componenti.



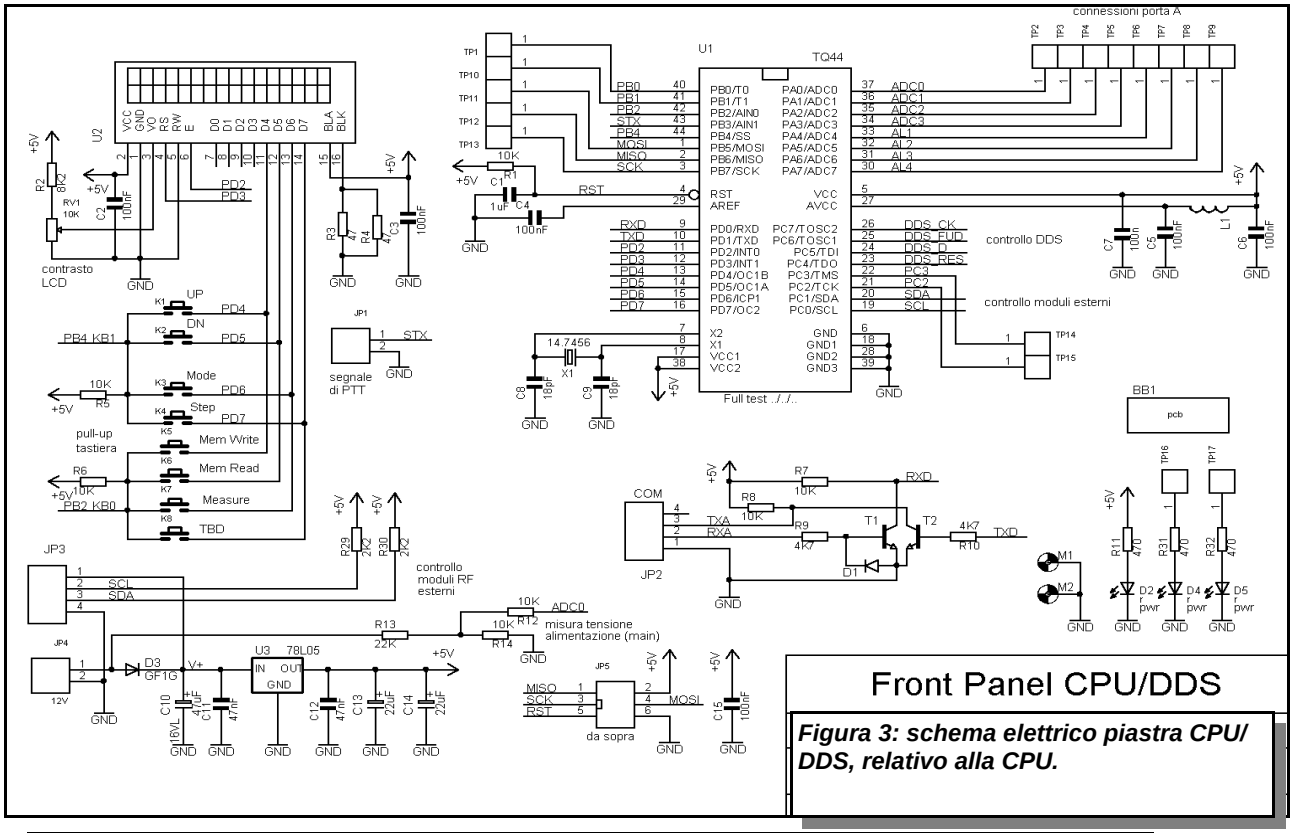


Figura 3: schema elettrico piastra CPU/ DDS, relativo alla CPU.

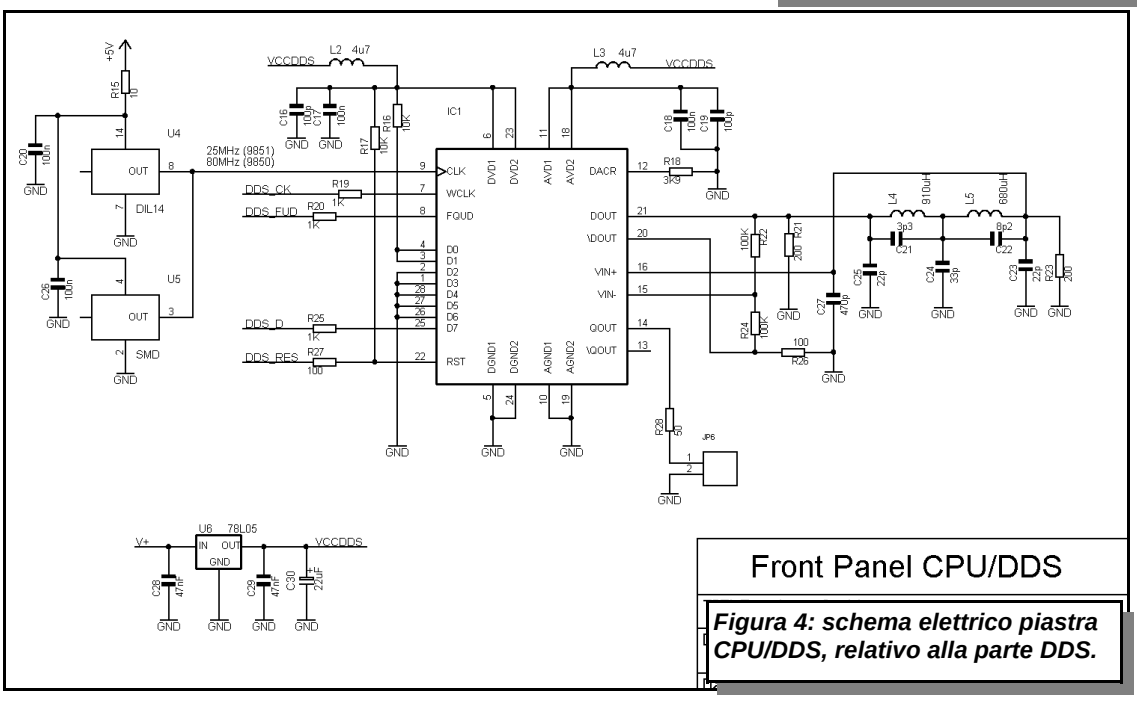


Figura 4: schema elettrico piastra CPU/DDS, relativo alla parte DDS.

oscillatori perchè è possibile usare il chip AD9850 al posto del 9851, essendo pin-compatibili, a parte la massima frequenza operativa.

Per il 9851 uso un oscillatore a 25MHz, mentre per il 9850 si deve usare uno da almeno 80MHz (max 125MHz, ma sono molto costosi!).

Questo perchè il primo ha un moltiplicatore di clock interno, che non ha invece il secondo; ovviamente il primo è più costoso del secondo ... entrambi i DDS si trovano a meno di 20 euro su Digikey.

Il principio di funzionamento del DDS non lo ripeto in quanto già descritto su un precedente numero del Bollettino, ma ricordo che in ogni caso, per il teorema di Nyquist, non si può ricostruire un segnale se non lo si campiona d almeno il doppio della sua frequenza; la pratica e l'esperianza, consigliano di non superare il limte dei almeno 1/3 del clock.

Nel caso del 9851, con clock a 25MHz e moltiplicatore x6, avremo un clock al DDS di 150MHz, quindi potremmo pensare di generare dei segnali fino a 150/3=50MHz.

L'uscita del DDS è un segnale digitale (di forma sinusoidale) e come tale dev'essere filtrato per evitare il problema della presenza di "alias" ovvero di segnali prodotti come somma o sottrazione tra la frequenza prodotta e quella dell'oscillatore, che potrebbero essere utili in alcuni casi e dannosi in tutti gli altri.

Figura 5: il primo prototipo (inscatolato) del DCRTX che ho realizzato.

Questo limita il valore massimo di funzionamento del generatore che qui è fissato a 50MHz (limite visto prima).
Potrebbe sembrare una cosa non importante, ma visto che il mixer utilizzato sulla piastra base RF richiede un clock pari a 4 volte la frequenza di ricezione, ecco che 50MHz del DDs rappresentano realmente un limite a 50/4=12,5MHz. Non ho voluto togliere il filtro o spostare in alto la frequenza di taglio, in quanto partivo sempre dall'idea di avere un ricetrans in IF (ovvero attorno ai 9MHz). Ultima parte, l'alimentazione, che ho voluto dividere in due regolatori separati, uno per la logica del micro ed uno per il DDS, entrambi sono dei semplici 7805.

Realizzazione

Essendo un circuito che ha la pretesa di andare verso il futuro, ovvero verso tecnologie moderne, l'ho disegnato utilizzando componentistica SMD (ovvero a montaggio superficiale), anche considerando che i dispositivi elettronici utilizzati (ATmega32 e AD9851) sono in contenitori SMD (il primo è un TQFP, Thin Quad Flat Package a 44 piedini passo 1mm, il secondo un SSOP, Shrink Small Outline a 28 piedini passo 0,5mm !).
Questo ha permesso di ridurre considerevolmente le dimensioni e come vedete nella figura 5, tutto riesce a stare "dietro" ad un pannello di una scatola commerciale, ottenendo un effetto estetico gradevole.
Il circuito stampato è stato fisicamente realizzato da una ditta secializzata ed ha il "solder resist" che facilita la saldatura dei componenti *smd* evitando di fare cortocircuiti (basta un pò d'attenzione). Basteranno delle normali attenzioni per saldare i chip, mano ferma ed un pò di pazienza. Una vista d'insieme senza scatola, è riportata nella figura 6, dove si vede il secondo prototipo che ho realizzato e che uso per fare i test (aprire e chiudere la scatola è fastidioso); si distingue la scheda CPU/DDS che ho appena descritto e la piastra base RF che descriverò nella seconda parte.

Software

Il programma di gestione del DCRTX è scritto in C usando i tools gratuiti della Atmel (avrStudio) e il WinAVR (avrGCC) e si presta sicuramente a miglioramenti ed ampliamenti. Ho utilizzato molte delle parti già ampiamente descritte su precedenti numeri del Bollettino, per cui non mi addentro nella loro descrizione, restando a disposizione per commenti o suggerimenti; per ora mi sono limitato alle cose essenziali, senza tanti fronzoli o cose complicate.

Sul tema DCRTX ho aperto una discussione sul forum AMSAT Italia, Area tecnica -> DCRTX, a cui invito gli interessati a partecipare.

Riferimenti:

Direct Conversion HF transceiver - www.nikkemedia.fi/hohtola/dc-trcvr/DCradio.htm
A technical tutorial on Digital Signal Synthesis - Analog Devices
Usare i microcontrollori - parti 1-6 - AMSAT-I news Vol14N3, Vol14N4, Vol14N5, Vol15N2, Vol15N4, Vol15N6

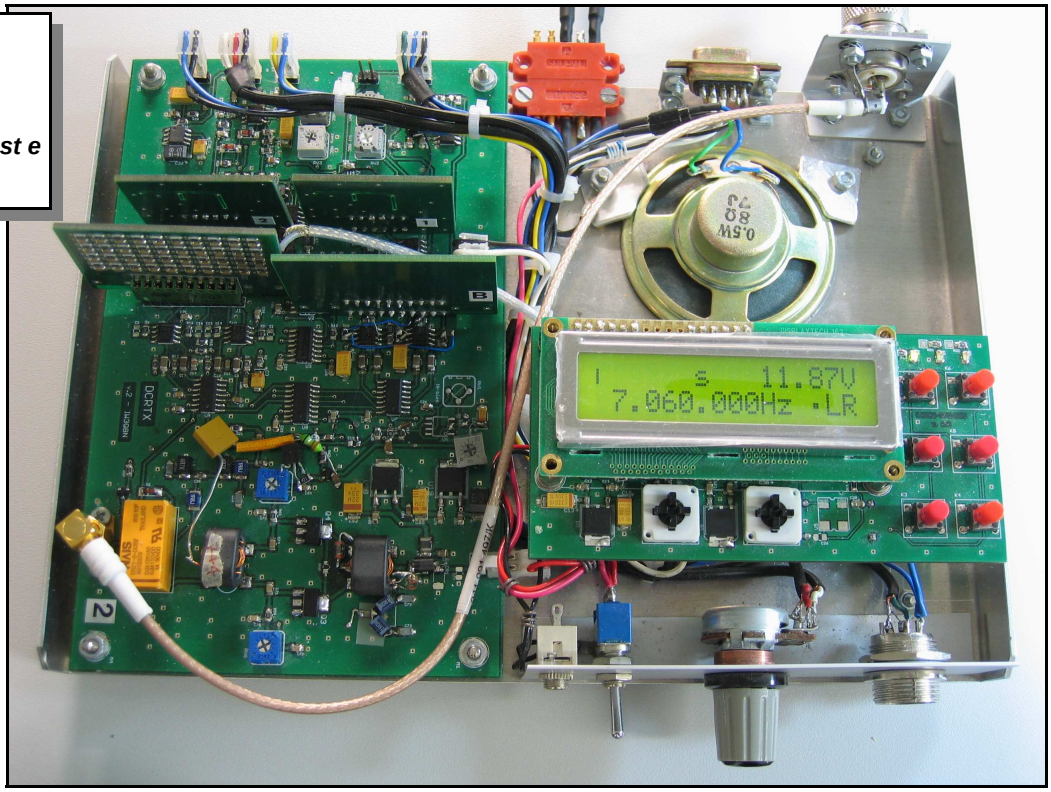


Figura 6: il secondo prototipo del DCRTX che utilizzo per i test e le modifiche.

Avete mai pensato ad una radio "diversa"? Oppure avete sentito parlare di radio "digitale" ma non avete idea di cosa (o come) sia fatta? Questo articolo vi farà sicuramente incuriosire.

Complessità computazionale

Fabio Azzarello - IW8QKU

Discutere delle qualità di un software forse è un po' off-topic per il nostro bollettino, tuttavia penso che chiunque sia appassionato di tecnica abbia provato l'esperienza della programmazione almeno un paio di volte.

A causa di un problema ho scoperto degli aspetti della programmazione che mi erano sconosciuti.
E' stato molto interessante imparare che cambiando tecnica di programmazione è possibile ottenere delle performance anche abbastanza diverse per lo stesso piccolo programma.

Da profano, ho trovato tutto ciò affascinante e quindi mi sono cimentato in una piccola prova che riporto di seguito.

Prima però, vorrei introdurre l'oggetto che mi ha permesso di sperimentare, cioè il calcolo del generico elemento della successione di Fibonacci.

http://it.wikipedia.org/wiki/Successione_di_Fibonacci

[su cui si trova anche un esempio di codice C]

Brevemente, con successione s'intende una sequenza di numeri che ubbidisca a delle regole.
In particolare, ogni elemento della successione di Fibonacci è dato dalla somma degli ultimi due elementi che lo precedono.

La successione è la seguente: 1,1,2,3,5,8,13,21,... e così via.

Più formalmente, si può definire la successione nel modo seguente:

$F(n) = F(n-1) + F(n-2)$ se $n>2$

con

$F(0) = 0$ se $n=0$
$F(1) = 1$ se $n=1$

Con F maiuscola si rappresenta la successione stessa, il numero tra parentesi indica il posto occupato nella serie, per esempio F(2) = 1 oppure F(0) = 0.

Per calcolare manualmente un elemento della successione si può procedere come segue:
ad esempio, se scelgo n = 5 ottengo dalla formula che F(n=5) = F(5-1)+F(5-2) = 5.

Ricordando che la sequenza è: 1,1,2,3,5,8,13,21,...

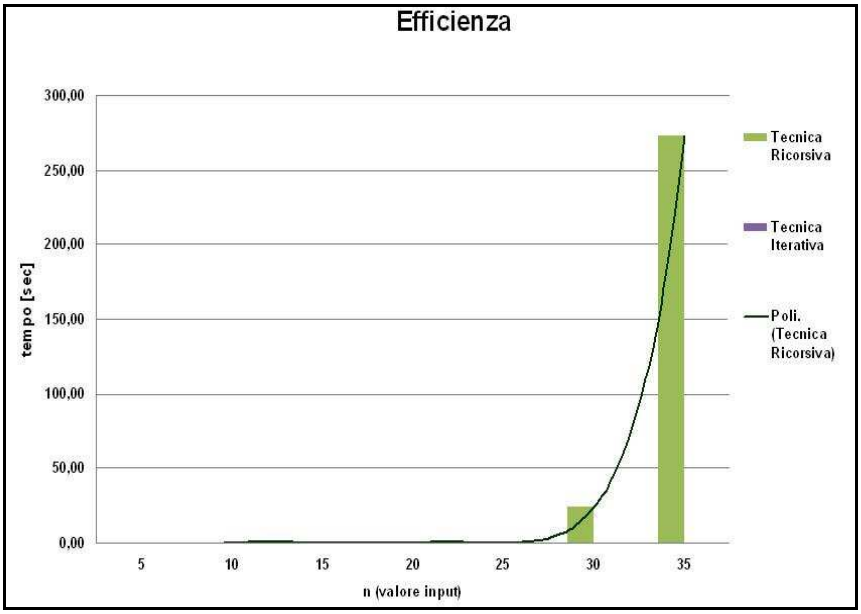
F(n-1) = F(5-1) = F(4) = 3
F(n-2) = F(5-2) = F(3) = 2

da cui:
F(5) = F(5-1) + F(5-2) = 3 + 2 = 5

Per tanto, l'elemento di posto 5 è F(5) = 5.

Visto che volevo mettere alla prova il calcolatore per fare gli stessi calcoli, ho usato le due funzioni riportate in appendice ed ho preso nota del tempo impiegato per il calcolo.

Eseguendo le funzioni con lo stesso input (cioè con lo stesso numero n della formula riportata prima) ed annotando in tabella il tempo di esecuzione si riesce a produrre il seguente grafico.



Tempo in Secondi			
Input n	Risultato	Tecnica Ricorsiva	Tecnica Iterativa
5	5	0,00	0,00
10	55	0,00	0,00
15	610	0,04	0,00
20	6765	0,10	0,00
25	75025	0,24	0,00
30	832040	24,12	0,00
35	9227465	273,21	0,00

Il risultato calcolato dalle due funzioni e quindi dai due diversi algoritmi è identico, quindi gli algoritmi dal punto di vista della correttezza sono interscambiabili, si può usare quello che si preferisce.

Solo per farne menzione, le tecniche usate per la realizzazione delle due funzioni sono la tecnica ricorsiva e la tecnica iterativa (ampiamente descritte in tutti i libri).
Osservando la tabella si vede subito che il tempo impiegato dall'algoritmo che sfrutta la tecnica iterativa è sempre zero.

In realtà è la risoluzione fornita dall'ambiente di sviluppo usato, nel mio caso MatLab, che restituisce un risultato pari a zero.
L'altra tecnica, invece, richiede tempi sempre maggiori, che aumentano esponenzialmente.

La tecnica ricorsiva è evidentemente svantaggiosa, almeno per questo tipo di calcolo.

Infatti, creare l'algoritmo sfruttando la definizione della sequenza di Fibonacci è sicuramente semplice ed immediato ma va a scapito della sua efficienza.

Si nota, osservando il codice della funzione ricorsiva, che molti passi dell'algoritmo vengono ripetuti per richiamare la stessa funzione, ma i risultati intermedi non vengono utilizzati per il calcolo finale.

Ciò si riflette nell'aumento esponenziale della complessità, l'algoritmo ripete molte volte le stesse operazioni senza trarne profitto.

Un simile effetto non si nota se il numero della successione che si vuole calcolare è piccolo (cioè in input piccolo), la cosa cambia radicalmente quando il numero cresce.

Come si vede dalla tabella, se l'input è minore di 25 gli algoritmi sono sostanzialmente uguali.
D'altra parte, la formulazione iterativa è meno intuitiva ma molto più efficiente.

In definitiva, entrambe le funzioni producono il risultato in modo corretto, quindi gli algoritmi che ho usato possono essere usati con tranquillità se i tempi di esecuzione non sono un problema.

L'efficienza degli algoritmi diventa molto importante nell'uso pratico, uno esempio su tutti è quello degli algoritmi usati per il calcolo delle FFT.

In un simile caso la velocità di esecuzione è d'obbligo, un algoritmo "lento" sarebbe sicuramente non produttivo se non inutile.

Faccio una breve considerazione finale per osservare che se anche usassimo computer velocissimi e potenti al crescere dell'input si ripresenterebbe sempre la stessa situazione, dunque la complessità è legata alla tecnica usata per la programmazione ed al problema che si vuole risolvere (esistono dei problemi intrinsecamente complessi).

Non è strettamente legata alla potenza della macchina che si vuole usare.

Appendice

Segue il codice Matlab che ho usato per l'esempio.

```
% Versione Iterativa – Complessità O(n)

function Fi = FibIT(n)
% f2=fib(n-1) - f1=fib(n-2) - f = f1+f2 per
definizione
f1 = 0;
f2 = 1;
Fi = 1;
 if (n==0)
      Fi = 0;
 else
      for i = 2:n
         Fi=f1+f2;
            f1=f2;
            f2=Fi;
      end
 end


% Versione Ricorsiva – Complessità O(2^n)

function Fr = FibRic(n)
 if (n==0)
      Fr = 0;
 elseif (n==1 | n==2)
      Fr = 1;
 else Fr = FibRic(n-1)+FibRic(n-2);  %
Sfrutto la definizione
 end


% Per il test ho usato i seguenti comandi
Matlab

n = 5,10,15… ecc.

tic, FibIT(n), toc

tic, FibRic(n), toc
```

Attenzione

Questo è il penultimo numero "cartaceo" del Bollettino. Come deciso dal CD AMSAT-Italia, da gennaio 2010 la pubblicazione sarà esclusivamente elettronica e quindi invitiamo tutti i Soci a prenderne nota Il Bollettino sarà quindi scaricabile da casa, in formato PDF, usando una password fornita dalla Segreteria.

Commento/descrizione di un avvenimento passato, per molti, quasi del tutto inosservato-

BOMBARDAMENTO LUNARE
Lunar Impact Oct. 10. 2009

Giovanni Lorusso - IK7ELN

Alle ore 07,55 (ora di Cape Canaveral) di Sabato, 10 Ottobre 2009, la sonda LCROSS (Lunar Crater Observation and Sensing Satellite) ha lanciato un "missile" ovvero il terzo stadio del vettore (motore Centaur) che l'ha portata in orbita lunare, con una potenza pari a una tonnellata e mezzo di dinamite, che ha colpito il cratere lunare Cebeus, al Polo Sud della Luna.

Disegno della separazione, e momento dell'impatto.

Il razzo, viaggiando alla velocità di circa 9000 Km/h, dopo l'esplosione, ha creato uno squarcio profondo 4 metri e largo 20 metri, alzando un enorme nuvola di polvere (silicio) e rocce lunari (basalto), utile agli esperti per poter analizzare, in maniera approfondita, la presenza di acqua presente nel sottosuolo (Permafrost) del Polo Sud lunare.

Dopo l'impatto del "missile", ha fatto seguito l'impatto della stessa sonda LCROSS, la quale, conficcandosi nel sottosuolo del cratere Cebeus, ha cominciato a raccogliere dati importanti di eventuali riserve idriche.
Una missione della NASA ben riuscita, se si considera che la sonda era stata lanciata nel mese di Giugno dalla base di Cape Canaveral (Florida) e, ancor primadell'impatto, ha inviato suggestive immagini del Polo Sud della Luna, fino ad un attimo prima di schiantarsi nel cratere Cebeus.

Ma, perchè proprio al Polo Sud lunare?

Ebbene, poiché l'irraggiamento solare non raggiunge mai questa latitudine lunare, è facile immaginare che, temperature estreme (circa -170°C), hanno permesso che il ghiaccio di acqua, forse generato dall'impatto di qualche cometa, si sia potuto conservare per oltre 4 miliardi di anni!

La sonda LCROSS ed il motore Centaur.

Infatti, poiché la Luna ci mostra sempre la stessa faccia, accade che i crateri del Polo Sud non ricevono mai i raggi solari e, quindi, le temperature estreme sono costanti di giorno e di notte, creando, così, una nicchia di ibernazione per il ghiaccio di acqua presente nel sottosuolo.
E, poiché sulla Luna non accadono fenomeni meteorologici, l'unica spiegazione logica della presenza di acqua lunare è l'impatto di una cometa di enormi proporzioni.
Questo è quanto gli Astrofisici vogliono scoprire!

Foto del Polo Sud della Luna.

NOTIZIARIO AEROSPAZIALE

aggiornato al 12 novembre

La nostra principale fonte di informazioni è l'autorevole rivista settimanale Flight International. Fonti addizionali di informazioni sono la rivista mensile Spaceflight, edita dalla British Interpanetary Society, ed alcuni notiziari elettronici, tra cui il Jonathan Space Report. Con questi siamo in grado di presentare una selezione di notizie sempre aggiornate con l'uscita del Bollettino.

Shuttle e ISS

La Soyuz TMA-16 è stata lanciata il 30 settembre in un'orbita di 195 x 240 km e 51.6° d'inclinazione.
Il 2 ottobre l'orbita era di 340 x 351 km a 51.6° d'inclinazione; il veicolo spaziale ha agganciato la ISS alle 08:35 UTC.
L'equipaggio era costituto da Maksim Suraev (comandante della TMA-16, ingegnere di volo dell'equipaggio 21, EX-21), Jeffrey Williams (ingegnere di volo della TMA-16 e dello Ex-21) e Guy Laliberte ("turista" su TMA-16 e "visitatore" come EP-17).
La Progress M-67, dopo alcuni giorni di volo libero, è stata fatta rientrare sull'Oceano Pacificoil 27 settembre alle 09:33 UTC.
Padalka, Barratt e Laliberte si sono trasferiti nella Soyuz TMA-14 il 10 ottobre; la capsula si è sganciata dal modulo Pirs della ISS alle 01:07 del giorno seguente, 11 ottobre.
L'atterraggio nella steppa del Kazakihstan è avvenuto alle 04:32 UTC dello stesso giorno.
Lo sgancio della Soyuz TMA-14 ha rappresentato l'inizio del cambio di equipagio Expedition 21 (EX-21), con il belga Frank De Winne a diventare il primo comandante europeo della ISS.
Successivamente, la Progress M-03M è stata lanciata il 15 ottobre, e si è agganciata al modulo Pirs alle 01:41 UTC del 18 ottobre.
Soyuz TMA-15 è agganciata al portello del modulo Zarya, mentre la Soyuz TMA-16 sul modulo Zvezda.
Il cargo giapponese HTV-1 è invece agganciato al modulo Harmony.
Dopo il cambio del 2 ottobre, Romanenko, De Winne e Thirsk sono stati designati equipaggio della TMA-15, mentre Suraev, Williams e Stott alla TMA-16.

ARES-1

Il razzo vettore Ares I-X, composto dal booster a propellente solido, modificato, dello Shuttle, RSRM-91A un carico fittizio che simulava il secondo stadio, ed un altro carico fittizio che simulava l'insieme CM/LAS (command module / launch abort system), è stato lanciato con successo dal complesso 39B del KSC lo scorso 28 ottobre.
Lo scopo del lancio era di verificare le caratteristiche aerodinamiche del volo nell'atmosfera; come previsto, esso ha raggiunto un apogeo di circa 46 km, all'interno della stratosfera.
A fianco, sulla rampa del complesso 39A, lo Shuttle Atlantis era in attesa del lancio per la missione STS-129 in programma per metà novembre.

LCROSS

LCROSS/AV-020 è entrato nellinflueza della sfera gravitazione della Luna alle 18:00 UTC del 8 ottobre, grazie ad un'orbita di 365700 x 644500 km e 48.6° d'inclinazione.
La sonda LCROSS si è separata dallo stadio Centaur (020) alle 01:50 UTC del 9 ottobre, mediante un'accensione del suo sistema di propulsione ha prodotto una "frenata" alle 02:30 UTC.
In questo modo, la sonda ha raggiunto la superficie lunare seguendo una traiettoria iperbolica.
L'impatto con la superficie è avvenuto nel cratere Cabeus alle 11:31 UTC (Centaur) e 11:35 UTC (LCROSS).
Nonostante le previsioni, non c'è stata la tanto attesa produzione di polvere visibile da terra, ma gli strumenti di bordo sia LRO che LCROSS hanno rivelato i segni dell'evento, confermando la presenza di notevoli quantità d'acqua sotto forma di ghiaccio, all'ombra del cratere.
(maggiori dettagli nell'articolo precedente).

Ariane L549 e L550

Il lancio Ariane numero V191, con il vettore L549 (modello 5ECA) è avvenuto dal poligono di Kourou il primo ottobre.
La missione aveva il compito di mettere in orbita due satelliti per comunicazioni, Amazonas 2 e COMSATBw-1.
Amazonas 2 è un satellite costruito da Astrium/Tolosa su bus Eurostar E3000 per conto della compagnia Hispasat (sede a Madrid), che lo userà per trasmettere programmi sulle americhe.
La massa al lancio era di 5465 kg.
COMSATBw-1 (ufficialmente conosciuto come SATCOMBw-1) è un satellite realizzato da Thales Alenia/ Cannes su bus Spacebus 3000B2 per comunicazioni militari della Germania.
In realtà il contratto era con EADS Astrium ma poi è stata la Thales a realizzare il satellite.
Il progetto è gestito dallo IT-AMTBw (Bundesamt fur Informationsmanagement und informationstechnik der Bundeswehr), ovvero il dipartimento tecnologico delle forze armate tedesche (Bundeswehr).
Il satellite aveva al lancio, una maassa di 2440 kg, di cui 1040 kg rappresentavano il carico.
E' conosciuto anche come GMS-2A (forse per indicare German Military Satellite?).
Esso è entrato in un'orbita di trasferimento di 265 x 35801 km e 2.9° d'inclinazione.
Il vettore Ariane L550 è stato lanciato il 29 ottobre con altri due satelliti per comunicazioni: New Skies NSS 12 e Telenor's Thor 6.

WorldView-2

Il satellite WorldView-2, della DigitalGlobe, dotato di una camera di ripresa con risoluzione a terra di 0,5 metri su 8 bande, è stato lanciato lo scorso 8 ottobre.
Il satellite è stato realizzato su un bus BCP-5000 della Ball Aerospace ed è in un'orbita di 766 x 768 km e 98.6° d'inclinazione.

DMSP 5D F-18

L'aviazione americana (US Air Force) ha lanciato dal poligono di Vandenberg, lo scorso 18 ottobre, un satellite meteorologico.
Si tratta del 5D-3, lancio F-18, facente parte del programma meteorologio della difesa (Defense Meteorological Satellite Program) ed è l'ultimo di questa serie iniziata con il lancio del 5D nel 1976.
Il programma era iniziato nel 1962 con il primo satellite meteorologico Program 35, usato dall'ufficio NRO per scoprire le zone libere da nubi ed indicarle agli altri satelliti spia, ritornando poi all'aviazione alla fine del 1960s quando si dimostrò importante per il supporto alle operazioni di guerra in Vietnam.
I primi satelliti DMSP erano lanciati da vettori molto più piccoli; la nuova generazione di Delta 4 (usata per il DMSP 5D F-17) e di Atlas V 401 (usata adesso) anno tutte molta più potenza disponibile rispetto al carico.
In questo lancio infatti, questo eccesso di potenza è stato sfruttato per fare un test sul sistema Centaur AV-017, inviandolo in un'orbita solare.
La traiettoria iperbolica terrestre dello AV-017 aveva una caratteristica C3 (energia specifica all'infinito) di 37.47 km**2/s**2, sufficente per mandare il Centaur sulla fascia degli asteroidi, se fosse stato nella giusta direzione.
Questo è stato il primo lancio in orbita solare effettuato da Vandenberg.

GOCE

Il satellite progettato per studiare il campo magnetico terrestre ha iniziato la sua missione di mappatura.
A forma di freccia, esso ha sensori molto precisi e sensibili per misurare il campo gravitazionale terreste, orbitando (o "volando" se preferite) a soli 250 Km d'altezza.
La mappatura permetterà agli scineziati di capire meglio come si muovono gli oceani e dare quindoi un riferimento universale per per le misure di qualunque punto della terra.

Lanciato nello scorso marzo dal cosmodromo di Plesetsk, ha dovuto passare un lungo periodo di tempo afficnhè i tecnici verificassero tutti gli strumenti di bordo.
L'orbita e l'altitudine fanno sì che GOCE abbia un solo lato illuminato dal sole mentre l'altra metà rimane sempre in ombra, irradiando così nello spazio il calore prodotto dal satellite.
Per la mappatura, il satellite dispone di un set di sei accelerometri di utlimissima generazione (stato dell'arte) e di uno strumento denominato gradiometro.
La vita prevista per il satellite è di 20 mesi.

La collaborazione al bollettino è aperta a tutti i Soci. Vengono accettati articoli tecnici, teorici, pratici, esperienze di prima mano, impressioni di neofiti, storie di bei tempi andati, opinioni, commenti, riferimenti e traduzioni da riviste straniere specializzate.

SCRIVERE E' UN'ESPERIENZA UTILE PER ENTRARE IN CONTATTO CON FUTURI AMICI E COLLEGHI. CHIUNQUE HA QUALCOSA DA RACCONTARE,

Il bollettino bimestrale ***AMSAT-I News*** viene inviato a tutti i Soci di ***AMSAT Italia***. E' possibile inviarne copie a chiunque ne faccia richiesta dietro rimborso delle spese di riproduzione e di spedizione.

Per maggiori informazioni sul bollettino, su AMSAT Italia e sulle nostre attività, non esitate a contattare la Segreteria.
segreteria@amsat.it

***AVVISO IMPORTANTE**:*
Se non altrimenti indicato, tutti gli articoli pubblicati in questo bollettino rimangono di proprietà degli autori che li sottoscrivono. La loro eventuale riproduzione deve essere preventivamente concordata con la Redazione di AMSAT-I News e con la Segreteria di AMSAT Italia. Gli articoli non firmati possono considerarsi riproducibili senza previa autorizzazione a patto che vengano mantenuti inalterati.

AMSAT Italia
GRUPPO DI VOLONTARIATO
Registrazione Serie III F. n. 10 del 7 maggio 1997 presso Ufficio del Registro, Sassuolo (MO)

Riferimenti:

Indirizzo postale:
Segeteria: segreteria@amsat.it
Internet WEB: http://www.amsat.it

Consiglio Direttivo: cd@amsat.it

Presidente iz0fno@amsat.org
Segretario ik0wgf@amsat.org
Consigliere i0kpt@amsat.org
Consigliere iw3qbn@amsat.org
Consigliere iw8qku@amsat.org

Pagamenti:

Tutti i pagamenti possono effettuarsi a mezzo:

Conto Corrente Postale: n° 14332340
Intestato a: AMSAT Italia

Codice IBAN: IT35 M076 0102 2000 0001 4332 340

Codice Fiscale: 930 1711 0367