



Sviluppo di uno spettrometro digitale per misure radioastronomiche basato su scheda ROACH

*Questo lavoro è stato finanziato dall'ASI (Agenzia Spaziale italiana)
nell'ambito del progetto ITASEL*

Autore: Marco Bartolini, Salvatore Pluchino

RAPPORTO INTERNO IRA 456/12

referee: Stelio Montebugnoli



INDICE

1. Introduzione	2
2. Hardware	2
2.1 iADC	2
2.2 ROACH	2
2.3 Installazione	3
3. Spectra2	4
3.1 Sampling	5
3.2 Canalizzazione	6
3.3 Calcolo della potenza	8
3.4 Accumulazione	8
3.5 Pacchettizzazione e trasmissione	9
3.6 Registri software	9
3.6.1 Comandi	10
3.6.2 Parametri di configurazione	11
3.6.3 Punti di SNAP	12
3.7 Gestione del tempo	13
3.7 LEDs	14
4. Software di controllo	15
4.1 Prerequisiti	15
4.2 Moduli di interfacciamento con lo spettrometro	17
4.3 Telescopio e scheduler	17
4.4 Registrazione e formattazione	18
5. Risultati	21
BIBLIOGRAFIA	23



1. Introduzione

Nell'ambito del progetto ITASEL si è deciso di adottare hardware basato su FPGA per l'acquisizione e l'analisi realtime di segnali radioastronomici con particolare attenzione alle applicazioni di spettroscopia ad alta risoluzione. In questo rapporto andremo quindi a descrivere le scelte tecnologiche effettuate e le applicazioni che ne sono derivate, assieme alle loro caratteristiche.

2. Hardware

La stazione radioastronomica di Medicina è stata nel corso degli anni un centro di sviluppo tecnologico di riferimento per la comunità italiana, ed è grazie all'esperienza qui maturata che si è diffuso in Italia l'utilizzo di tecnologie derivanti dal consorzio internazionale CASPER [Paarsens, I] (<http://casper.berkeley.edu>), guidato dal Berkeley Wireless Research Center e dedicato allo sviluppo di hardware digitale per la radioastronomia. In questo contesto all'interno del progetto ITASEL è stato deciso di sviluppare una nuova generazione di spettrometri basati su hardware digitale ROACH (<https://casper.berkeley.edu/wiki/ROACH>).

2.1 iADC

Ogni progetto digitale di elaborazione dei segnali necessita di uno stadio di conversione del segnale da analogico a digitale. Le schede iADC (<https://casper.berkeley.edu/wiki/ADC2x1000-8>) basate su chip di conversione *Atmel/e2V AT84AD001B 8-bit Dual 1Gsps* ci permettono di convertire in digitale fino a due segnali analogici campionando una banda di 500MHz per ciascuno. Considerando che ad ogni scheda ROACH possono essere collegate due schede di conversione iADC, una configurazione massimale permetterebbe di elaborare 4 segnali analogici contemporaneamente, abilitando ad esempio una acquisizione dati da un ricevitore dotato di due feed separati in due polarizzazioni ciascuno, ideale per il progetto ITASEL che si basa su osservazioni in spectral line con tecnica on-off.

2.2 ROACH

Le schede ROACH sono in grado di accogliere l'output digitale generato da due schede iADC ed effettuare una elaborazione in tempo reale, sfruttando il potente motore di calcolo costituito da un FPGA (field programmable gate array) prodotto da XILINIX, modello Virtex-5 XC5VSX95T-1FF1136. Caratteristica peculiare degli FPGA è quella di essere hardware programmabile, in grado quindi di eseguire differenti istruzioni a seconda del firmware con cui vengono configurati prima del loro utilizzo di volta in volta. Questa caratteristica risulta essenziale al progetto permettendo ad esempio di progettare spettrometri con diverse risoluzioni frequenziali utilizzabili sullo stesso hardware senza costi aggiuntivi. Altra fondamentale caratteristica delle schede ROACH è quella di essere equipaggiate con

una buona quantità di memoria, essenziale per poter effettuare operazioni di accumulazione ed integrazione dei dati nel dominio del tempo con sufficiente elasticità, nonché per raggiungere buone risoluzioni frequenziali implementando trasformate di fourier a diverse migliaia di canali.

Altra importante caratteristica di queste schede è la alta performance dei sistemi di input/output e dei sistemi di controllo della scheda stessa. L' output può infatti essere realizzato utilizzando fino a quattro interfacce di rete ethernet a 10 gigabit per secondo, implementate con tecnologia in rame CX4, mentre le operazioni di controllo della scheda possono essere eseguite utilizzando una interfaccia di rete standard che mostra la scheda come un normale PC sulla rete interna dei laboratori, facilitando la gestione dei progetti e svincolando in questo modo lo sviluppatore da problemi legati alla comunicazione con lo strumento, che avviene attraverso protocolli standard.

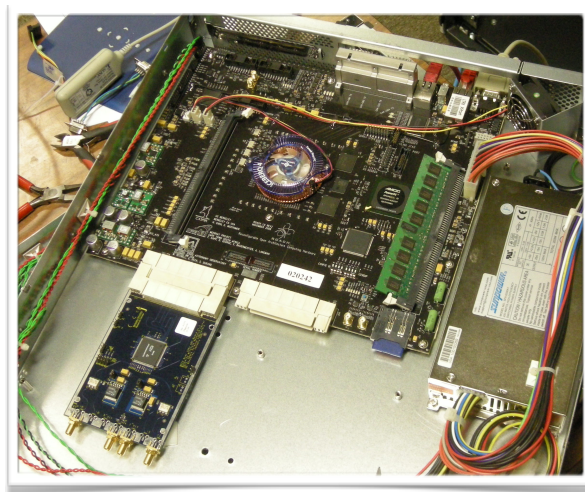


Fig. 1 Una scheda ROACH con una scheda iADC connessa

2.3 Installazione

Nell'ambito del progetto ITASEL l'hardware sopra descritto è stato acquistato ed installato presso le stazioni radioastronomiche IRA di Medicina e Noto. Per poter operare correttamente, l'hardware

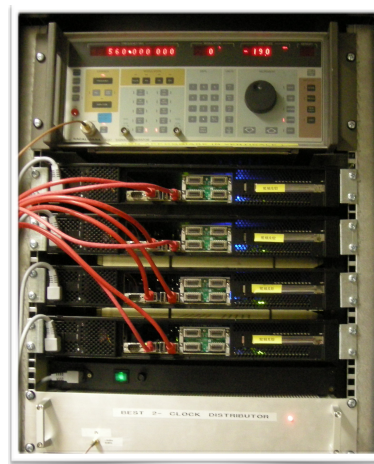


Fig. 2 Installazione di schede ROACH presso le stazioni di Noto (sinistra) e Medicina (destra)



necessita di una sorgente stabile di clock, che viene utilizzata per il campionamento, nonché di un segnale di Pulse Per Second, il più preciso possibile, utilizzato per sincronizzare le acquisizioni con gli orologi di stazione ed ottenere quindi dati coerenti con i risultati attesi. Le schede necessitano inoltre di una connessione ethernet alla rete locale in cui deve essere presente un server apposito che agisca da “controllore” delle schede, fornendo i servizi di NFS e DHCP necessari all’hardware per poter essere correttamente configurato. In entrambe le stazioni in cui si è operato il setup è stato completato e tutti i servizi accessori alle schede di elaborazioni sono stati attivati e testati.

3. Spectra2

Utilizzando l’hardware sopra descritto è stato implementato un primo spettrometro, denominato Spectra2, che rendesse possibile acquisire ed elaborare segnali radioastronomici nella stessa modalità degli spettrometri MSPEC0 e Spectra1 precedentemente utilizzati dal progetto ITASEL. La prima implementazione di Spectra2 è stata quindi quella di uno spettrometro che campionasse la banda in uscita da un videoconverter VLBI di vecchia generazione, in quanto al momento di inizio del progetto questo era l’unico modo per ottenere un segnale analogico nella sala ricevitori del radiotelescopio di Medicina, dove lo spettrometro è fisicamente posizionato. Al momento attuale (gennaio 2012) la sala ricevitori è stata collegata tramite link in fibra ottica ai ricevitori della parabola VLBI il che rende possibile lo sviluppo di spettrometri in grado di analizzare direttamente le IF prodotte dalla catena ricevente della parabola, aprendo di fatto la strada a molti possibili nuovi scenari.

Il firmware con cui la scheda ROACH viene configurata di volta in volta viene programmato tramite l’utilizzo del pacchetto simulink di Matlab®, utilizzando apposite librerie fornite da Xilinx® e dallo stesso consorzio CASPER. Nei paragrafi seguenti descriveremo l’implementazione dello spettrometro Spectra2, in grado di analizzare due segnali indipendenti, ciascuno con una banda di 16MHz, scomponendoli in 8192 canali frequenziali.

Il sistema di sviluppo utilizzato si basa su una interfaccia grafica, tipo CAD, in cui è possibile inserire i vari componenti dello spettrometro e configurarli nei loro parametri funzionali tramite apposite maschere. Ogni componente inserito in realtà costituisce un sottosistema molto complesso ed in questo modo è possibile sviluppare firmware anche molto complicati mantenendo un approccio grafico e user friendly basato sull’interconnessione di componenti indipendenti tra loro.

Per rendere un’idea del sistema di sviluppo utilizzato basti riferirsi alla figura seguente, dove i blocchi azzurri rappresentano degli operatori logici fondamentali, forniti da Xilinx® tra i tool di sviluppo per hardware FPGA, mentre i blocchi gialli e verdi rappresentano componenti di libreria sviluppati all’interno del consorzio CASPER che descrivono parti della scheda ROACH utilizzata e implementano

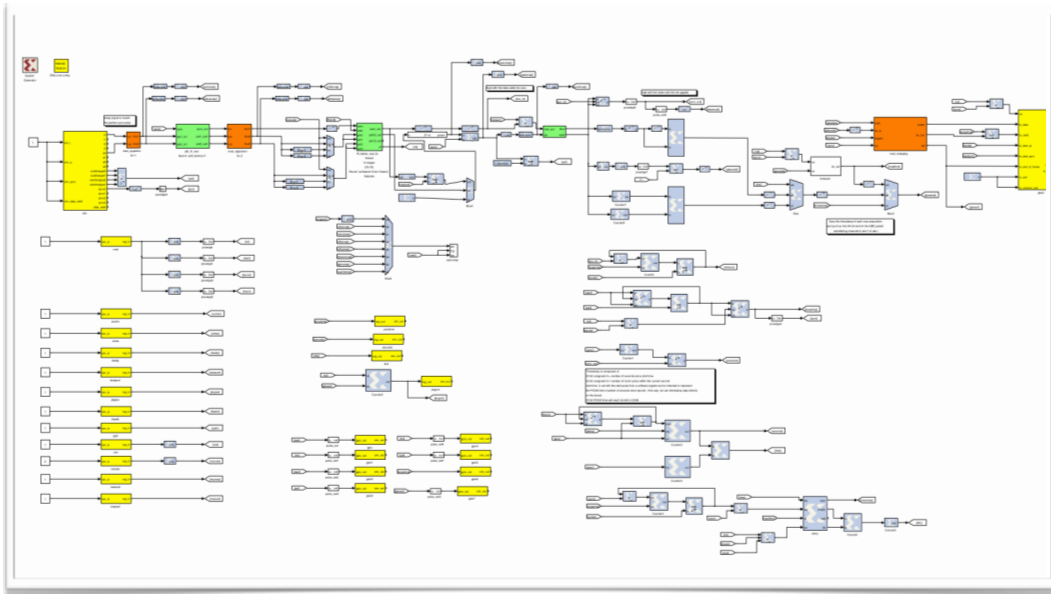


Fig. 3 Overview dello schematico dello spettrometro Spectra2

gli algoritmi fondamentali nell'elaborazione di segnali radioastronomici, come ad esempio le trasformate di Fourier. I componenti arancioni sono invece stati sviluppati presso la stazione di Medicina e svolgono compiti accessori.

3.1 Sampling

La configurazione dei parametri di sampling-rate avviene tramite il blocco giallo "adc", che rappresenta una scheda iADC collegata alla ROACH board.

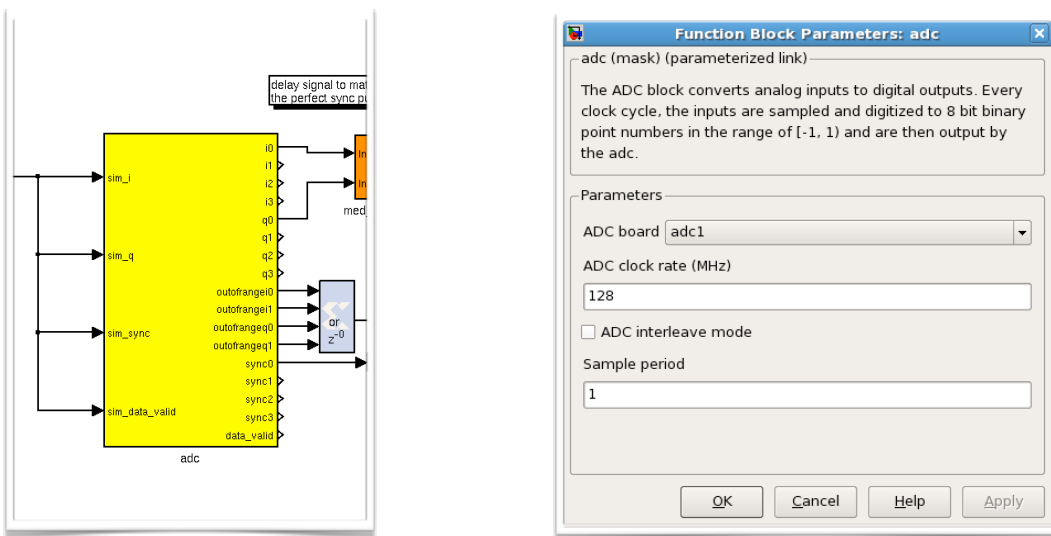


Fig. 4 ADC block parameters

Per ottenere la sampling rate desiderata di 32 Mega samples per second, il clock rate del chip di conversione è impostato a 128MHz, questo significa che i due segnali in ingresso sono sovracampionati, descrivendo una banda di 64MHz. Siccome la scheda di conversione fornisce però alla FPGA 4 samples consecutivi in parallelo, sapendo che la banda analizzata è analogicamente filtrata a 16MHz possiamo permetterci di effettuare una decimazione per quattro a valle della conversione in digitale prelevando uno solo dei quattro campioni paralleli per catena analogica. In questo modo otteniamo esattamente la sample rate desiderata di $128 / 4 = 32$ Msps.

3.2 Canalizzazione

Una volta campionato il segnale possiamo procedere quindi alla sua suddivisione in sottobande tramite l'applicazione di un filtro polifascico, costituito da un filtro digitale ed una trasformata di fourier in cascata tra loro. Il filtro *pfb_fir_real* è in grado di elaborare due segnali indipendenti contemporaneamente restituendo in uscita i segnali nel dominio del tempo dove sono state isolate le

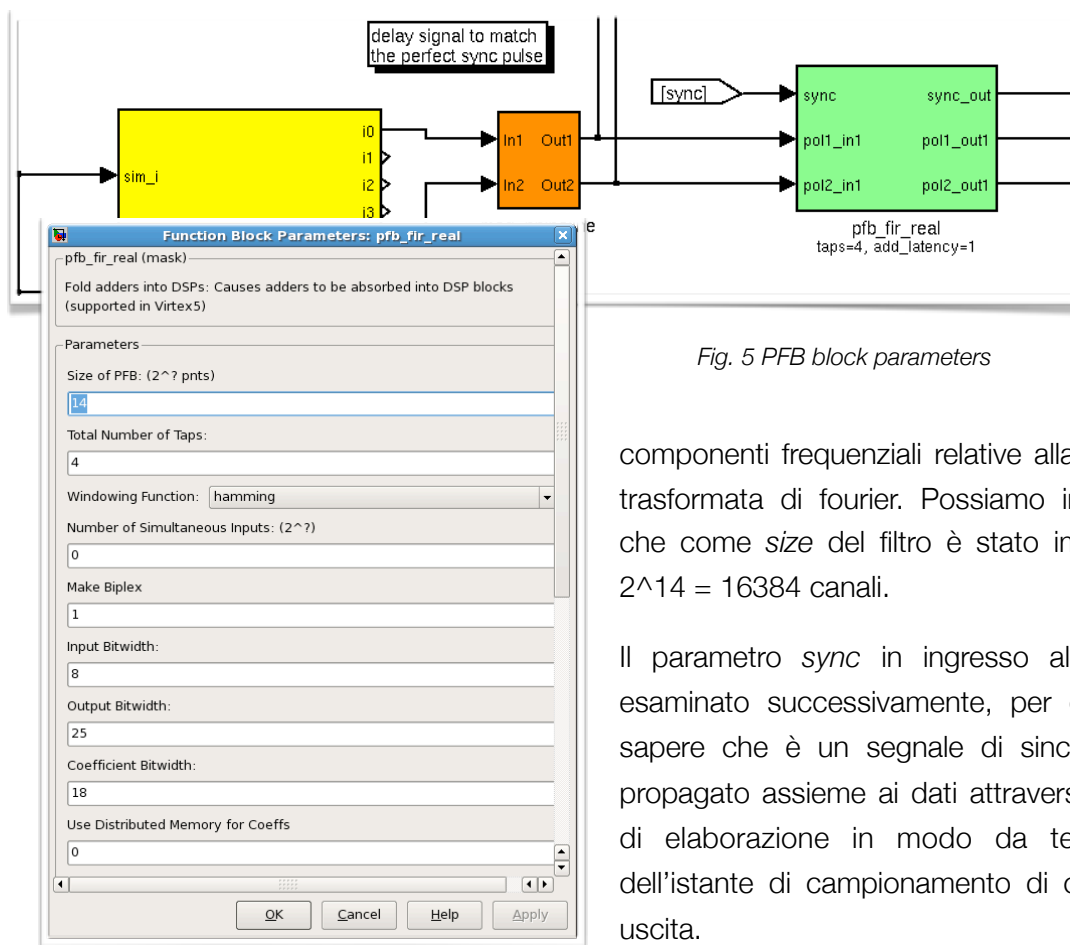


Fig. 5 PFB block parameters

componenti frequenziali relative alla successiva trasformata di fourier. Possiamo infatti notare che come *size* del filtro è stato impostato un $2^{14} = 16384$ canali.

Il parametro *sync* in ingresso al filtro verrà esaminato successivamente, per ora ci basti sapere che è un segnale di sincronizzazione propagato assieme ai dati attraverso la catena di elaborazione in modo da tener traccia dell'istante di campionamento di ogni dato in uscita.

In seguito al filtraggio, il segnale in ingresso viene elaborato tramite una trasformata di fourier realizzata dal blocco *fft_biplex_real_2x*, che porterà la successiva analisi nel dominio delle frequenze.

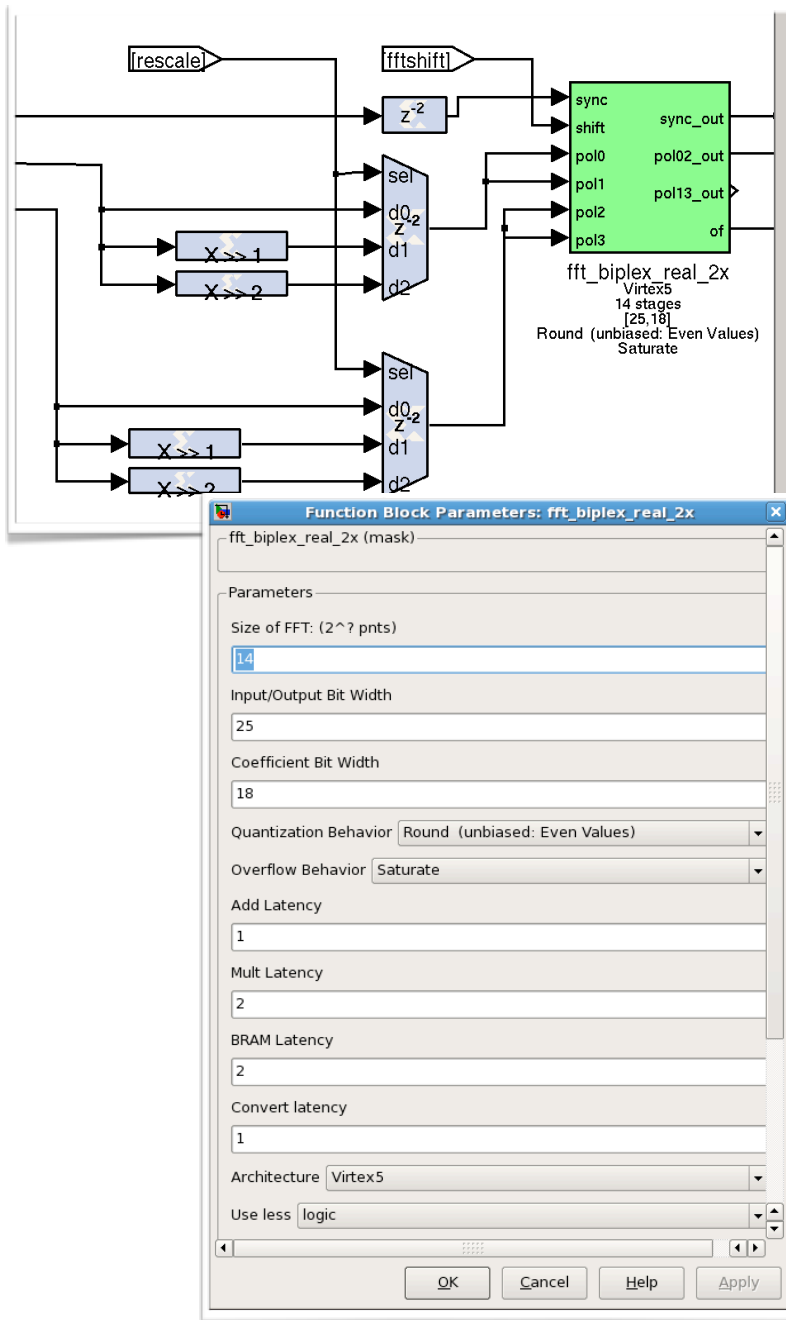


Fig. 6 FFT block parameters

I dati in ingresso alla FFT possono essere opzionalmente riscaldati mediante una operazione di shift di uno o due bit verso destra, questo per evitare successivamente pericolosi overflow numerici nel motore di calcolo della FFT. Anche il parametro *fft_shift* in ingresso al blocco che effettua la trasformata assolve la medesima funzione, comanda infatti lo shift a destra di un bit per ogni stadio di calcolo della trasformata stessa.

Il motore FFT è in grado di processare due stream di dati indipendenti, prendendo in ingresso dati reali e restituendo dati complessi rappresentanti le sole frequenze positive in uscita. Per questo la *size* della trasformata è definita come $2^{14}=16384$ ma produrrà in output unicamente 8192 canali frequenziali per segnale di ingresso.

3.3 Calcolo della potenza

In uscita dal blocco FFT, per ogni canale frequenziale viene calcolata la potenza del segnale ($a + ib$) come modulo quadro ($a^2 + b^2$). Al valore così ottenuto viene poi applicato un guadagno $gain$ che moltiplica il valore ottenuto, questo stadio è indispensabile per rendere efficace la successiva

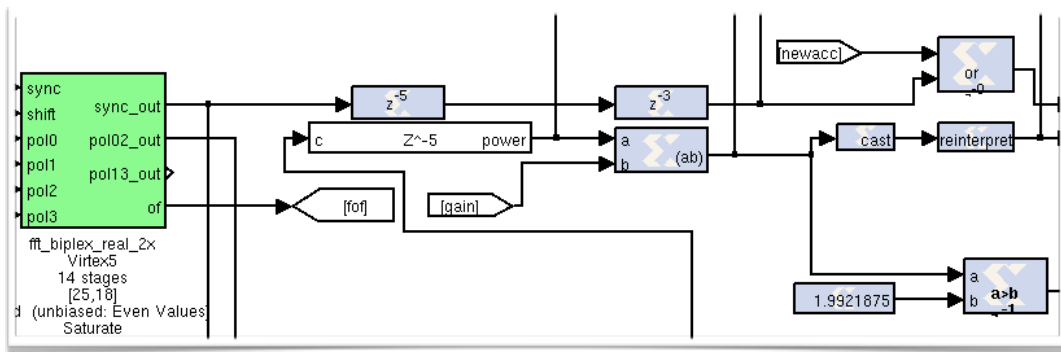


Fig. 7 calcolo della potenza e quantizzazione

operazione di quantizzazione ad 8 bit del segnale, effettuata dal blocco `cast`. Possiamo infatti pensare al `gain` come ad un sintonizzatore che ci permette di posizionare i bit rappresentativi del segnale ottenuto come i più significativi nella nuova rappresentazione binaria.

3.4 Accumulazione

Il segnale così ottenuto è accumulato utilizzando la memoria block RAM presente sulla FPGA per un numero di spettri programmabile.

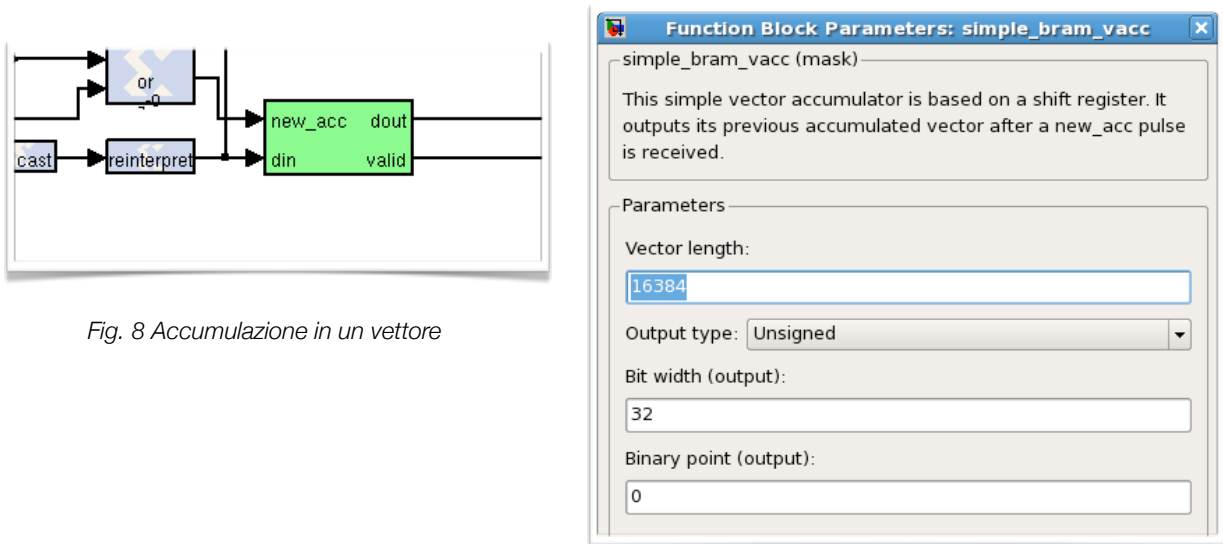


Fig. 8 Accumulazione in un vettore

Per realizzare l'accumulazione viene utilizzato il componente *vector accumulator* parametrizzato con lunghezza 16384 in modo da allocare lo spazio per entrambi i segnali in input concatenati uno di seguito all'altro come sequenze di 8192 canali frequenziali.

3.5 Pacchettizzazione e trasmissione

Al termine di ogni periodo di accumulazione il segnale salvato in memoria nella FPGA viene trasformato in una sequenza di parole a 64 bit precedute da un contatore di pacchetto, tramite il blocco arancione *med_makepkg*, e trasmesso sulla rete ethernet con protocollo UDP implementato dal blocco giallo *gbe0*.

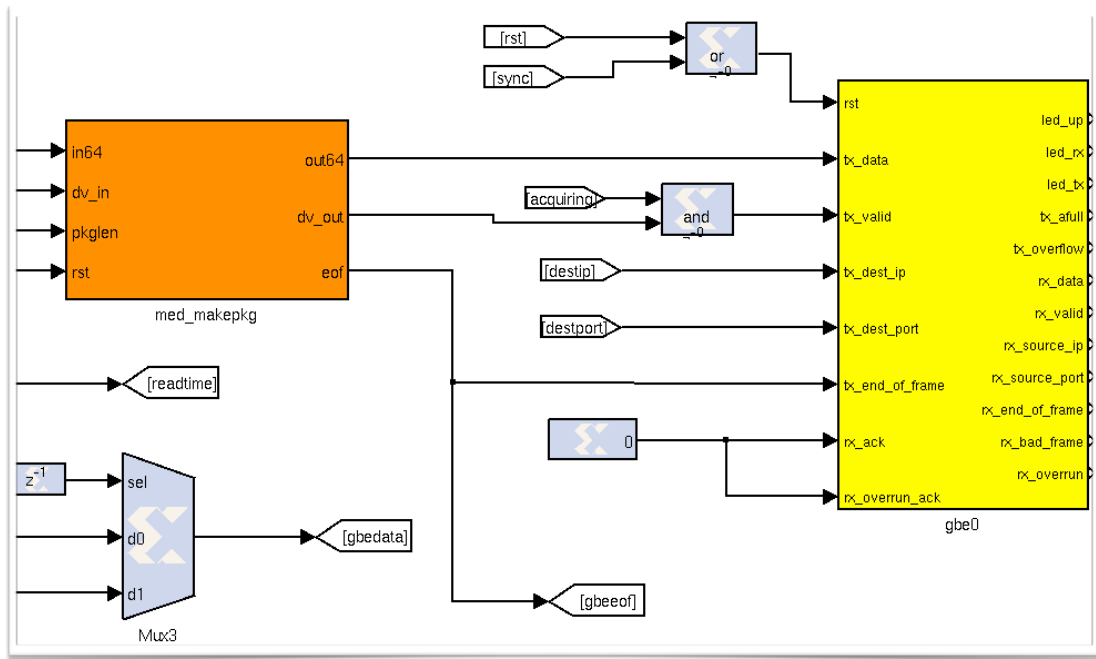


Fig. 9 Trasmissione del segnale in output

3.6 Registri software

Per comunicare dati ed inviare segnali di controllo alla FPGA su cui è implementato lo spettrometro, la scheda ROACH mette a disposizione dei registri software a 32 bit su cui è possibile scrivere valori arbitrari mediante un apposito pacchetto software. Tramite questi registri vengono perciò descritti tutti i parametri di cui lo spettrometro ha bisogno runtime e che possono essere forniti dall'utente, ovvero non fissati in fase di progettazione.

3.6.1 Comandi

Il registro principale che controlla il funzionamento dello spettrometro è il registro software *cmd* (command). I comandi implementati tengono conto di come lo spettrometro potrebbe essere in futuro reso compatibile con il sistema di controllo di antenna ESCS implementando le funzionalità richieste per l'integrazione.

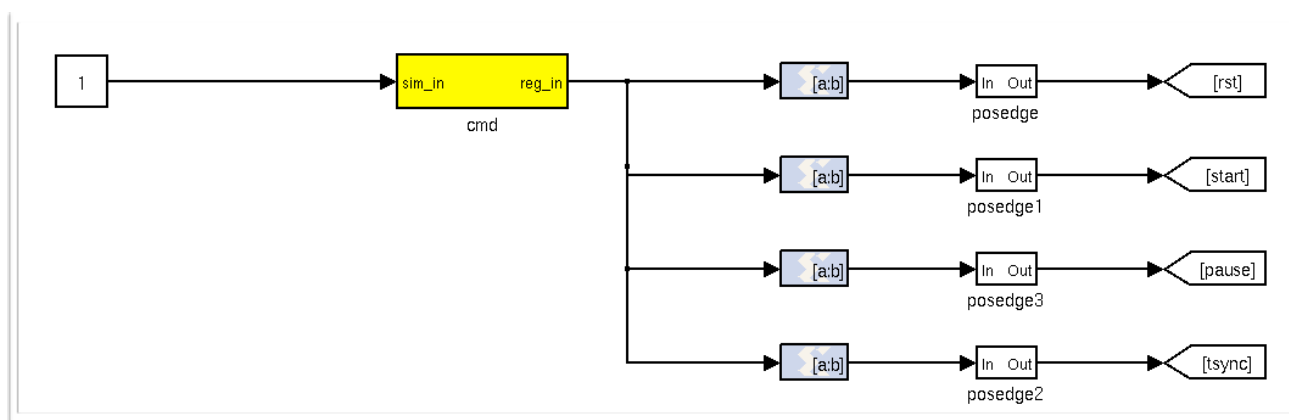


Fig. 10 il registro command

Scrivendo su questo registro, portando al valore 1 i quattro bit meno significativi otteniamo il seguente comportamento:

- **bit 0 : RESET** resetta tutte le logiche di calcolo dello spettrometro, svuotando eventuali buffer e azzerando i contatori e gli accumulatori. Tramite RESET vengono anche svuotati i buffer relativi alla trasmissione dei dati in rete 10Gbe. E' buona pratica effettuare un reset successivamente ad ogni cambio di configurazione della FPGA.
- **bit 1 : START** Abilita l'elaborazione dei dati da parte dello spettrometro e la loro trasmissione in output. Una volta dato lo start la logica interna allo spettrometro attende il successivo segnale di PPS, assieme al quale abilita tutte le acquisizione e immette un segnale di *sync* che si propaga attraverso il disegno.
- **bit 2 : PAUSE** Mette in standby lo spettrometro in maniera sincrona. Ogni operazione di accumulazione o trasmissione dati in corso viene interrotta, svuotando i buffer di calcolo intermedi e ripristinando una configurazione zero da cui poter poi riprendere tramite il comando START.
- **bit 3 : TSYNC** Avvisa lo spettrometro di sincronizzare il proprio tempo di sistema con quello scritto nel registro software *stime*.

3.6.2 Parametri di configurazione

Oltre al registro *cmd* sono presenti altri registri che impostano le funzionalità dello spettrometro. Il software di controllo realizzato andrà quindi a modificare il loro valore secondo i parametri impostati dall'utente, vediamo quali sono:

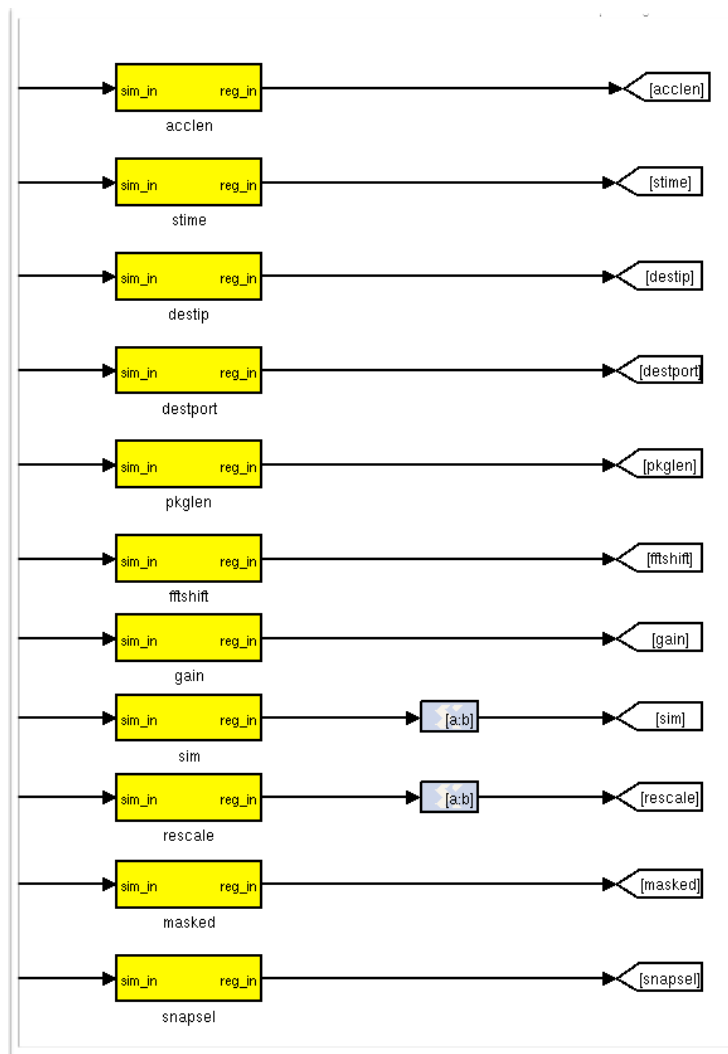


Fig. 11 registri di configurazione

- **ACCLEN** durata di un periodo di accumulazione, misurata in cicli di clock. Per ogni singolo spettro da accumulare in uscita avremo 16384 cicli di clock.
- **STIME** 32 bit posix time. Va settato prima di inviare il comando *tsync*.
- **DESTIP** rappresentazione binaria dell'indirizzo IP di destinazione della trasmissione dati via 10Gbe.



- **DESTPORT** Porta della socket UDP in ascolto della trasmissione dati via 10Gbe.
- **PKGLEN** Lunghezza del payload di un pacchetto dati 10Gbe espressa in parole da 64 bit. Solitamente settato a 256 che risulta sperimentalmente essere un buon valore. Il server in ricezione dei dati dovrà tenere conto di questo valore.
- **FFTSHIFT** Rappresentazione bit a bit degli stadi di calcolo della FFT. Se il valore corrispondente allo stadio n il motore di calcolo della FFT effettua uno shift di un bit a destra prima dello stadio successivo. Utile ad evitare saturazione durante il calcolo.
- **GAIN** Guadagno digitale applicato al segnale dopo il calcolo della potenza come sopra descritto. Solitamente viene utilizzato come uno shift a sinistra programmabile immettendo come valori delle potenze di due.
- **SIM** Se settato a 1 sostituisce ai dati in uscita dei valori sequenziali. Utile in fase di debugging e testing della trasmissione e ricezione dati.
- **RESCALE** 0, 1 o 2. Indica di quanti bit effettuare uno shift a sinistra dei dati tra il filtro *pfb_fir* e la FFT.
- **MASKED** Impostato a N pone uguali a 0 tutti i canali tra 0 e N in uscita dal motore FFT.
- **SNAPSEL** Selettore del punto di debugging in cui osservare i dati.

3.6.3 Punti di SNAP

Oltre alla ricezione dei dati in output è possibile ispezionare i risultati intermedi del calcolo dello spettrometro chiedendo alla FPGA di memorizzare in una zona dedicata di memoria i risultati di una computazione intermedia.

Questa pratica è molto utile sia in fase di sviluppo per verificare il corretto funzionamento dello spettrometro, sia in fase di calibrazione e setup dello strumento per poter individuare i corretti parametri da impostare per ottenere una buona descrizione del segnale in ingresso evitando errori di overflow nel calcolo e massimizzando l'utilizzo dei bit disponibili per descrivere il nostro dato.

Una volta selezionato il punto di snap tramite il registro di controllo *SNAPSEL* i dati saranno disponibili nella memoria interna *specsnap_bram* per essere letti tramite un apposito software. I punti di snap già configurati in SPECTRA2 sono i seguenti:

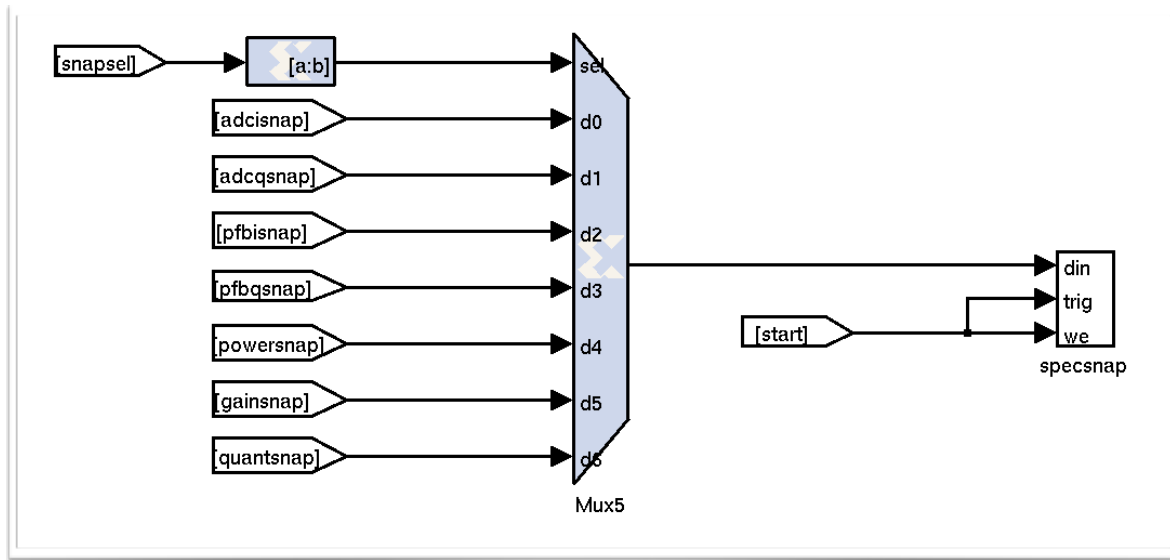


Fig. 12 snap points

- **ADCISNAP** dati in uscita direttamente dal convertitore AD, segnale I.
- **ADCQSNAP** dati in uscita direttamente dal convertitore AD, segnale Q.
- **PFBISNAP** dati in uscita dal filtro *pfb_fir_real*, segnale I.
- **PFBQSNAP** dati in uscita dal filtro *pfb_fir_real*, segnale Q.
- **POWERSNAP** dati in uscita dalla FFT dopo il calcolo della potenza.
- **GAINSNAP** dati in potenza a cui è stato applicato il gain digitale, ancora in full precision.
- **QUANTSNAP** dati quantizzati in ingresso all'accumulatore.

3.7 Gestione del tempo

Come già anticipato lo spettrometro mantiene all'interno delle proprie logiche le informazioni riguardanti il tempo assoluto con cui i dati vengono acquisiti. Questo è necessario perché comunicando i dati tramite rete ethernet i tempi di trasmissione sono imprevedibili e soggetti a latenze variabili, quindi non ci si può basare sulle informazioni ricevute per determinare l'esatto istante di campionamento di un dato.

La configurazione del tempo assoluto si basa sul registro software *stime* in cui deve essere salvato un valore di posix time a 32 bit (secondi dal 1 gennaio 1970). Una volta scritto questo valore, utilizzando il registro *cmd* bisognerà dare il comando di *tsync*. A questo punto lo spettrometro attenderà il prossimo segnale di PPS per inizializzare un contatore a 32 bit con il valore assegnato in *stime*. Il contatore sarà poi incrementato con ogni successivo segnale di PPS in modo tale da mantenere aggiornate le informazioni riguardanti i secondi. Per quanto riguarda le frazioni di secondo viene

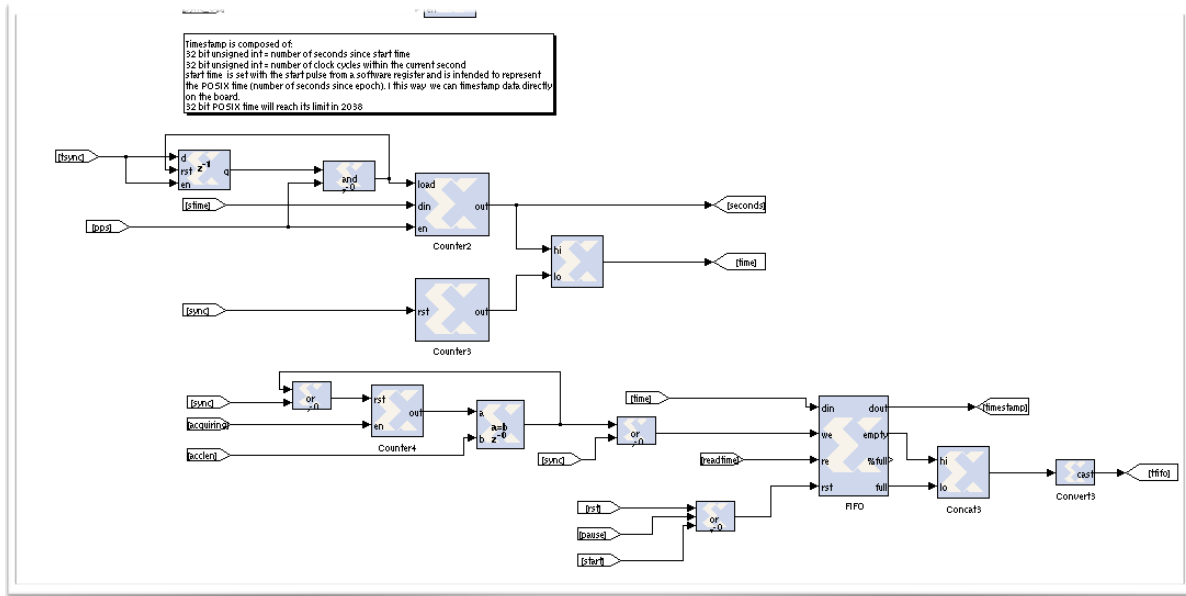


Fig. 13 gestione del tempo

invece utilizzato un secondo contatore a 32 bit, incrementato ad ogni colpo di clock e reinizializzato a 0 ad ogni segnale di PPS. I due contatori assieme descrivono il tempo assoluto con la precisione del clock di sistema.

Per mantenere le informazioni di tempo in linea con i dati acquisiti, all'inizio di ogni ciclo di accumulazione una parola a 64 bit costituita dalle due informazioni temporali viene salvata in una coda FIFO, da cui viene prelevata prima che il dato relativo all'acquisizione in corso venga trasmesso via rete, andando a sostituire con la marca temporale il canale frequenziale 0 dello spettro relativo all'input I.

3.7 LEDs

Le roach boards installate presso i radiotelescopi di Medicina e Noto sono state equipaggiate con una serie di LED per poter avere una diagnostica immediata e visibile del loro funzionamento o di eventuali errori.

Numerati in visione frontale da sinistra verso destra, vediamo come sono configurati i LED quando SPECTRA2 è attivo:

- **LED0 : PPS** segnale lampeggiante con periodo di 1 sec. collegato al PPS in ingresso alla scheda.
- **LED1 : RST** emette una luce di breve durata quando viene dato un comando di Reset.
- **LED2 : START** emette una luce di breve durata quando viene dato un comando di Start.
- **LED3 : AOF** emette una luce di breve durata quando il convertitore AD è in saturazione.
- **LED4 : FOF** emette una luce di breve durata quando il motore FFT causa saturazione.

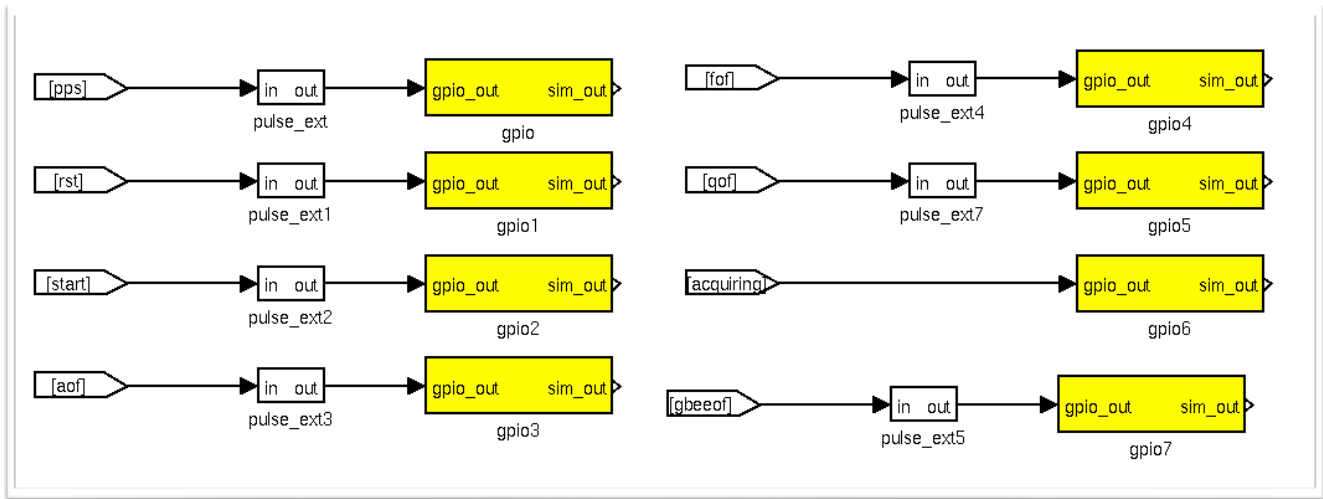


Fig. 14 i LED di Spectra2

- **LED5 : QOF** emette una luce di breve durata lo stage di quantizzazione precedente all'accumulazione causa saturazione.
- **LED6 : ACQUIRING** resta acceso quando l'acquisizione dei dati è in corso.
- **LED7 : GBEEOF** emette una luce di breve durata alla trasmissione di ogni frame ethernet.

4. Software di controllo

Per utilizzare Spectra2 in maniera agevole sono stati sviluppati alcuni moduli software in modo da automatizzare diversi aspetti del suo funzionamento quali la configurazione dello spettrometro, l'interfacciamento con l'antenna tramite fieldsystem, la schedulazione di una osservazione, la ricezione dei dati via rete e la loro formattazione secondo lo standard FITS.

Tutto il software ed il firmware sviluppato all'interno del progetto è conservato in un repository subversion, consultabile online all'indirizzo <http://svn.med.ira.inaf.it/repos/spectra/> . Per accedere al repository sono necessarie delle credenziali di accesso che possono essere ottenute contattando Marco Bartolini all'indirizzo mail mbartolini@med.ira.inaf.it .

4.1 Prerequisiti

Tutto il software sviluppato è stato scritto utilizzando il linguaggio di programmazione python (<http://www.python.org>) per il quale sono già utilizzabili alcune librerie messe a disposizione dal consorzio CASPER stesso per abilitare una efficiente comunicazione con le schede ROACH. Vediamo quindi quali sono le librerie necessarie per poter utilizzare spectra2:



1. **NUMPY** è un insieme di tool scientifici per il python, comprende un ottimo supporto ai tipi di dato numerici e agli array, nonché funzioni di utilità generica come ottimizzazione e interpolazione. Numpy può essere ottenuto tramite il sito internet <http://numpy.scipy.org> nelle sue versioni per windows, linux e Mac OSX.
2. **PYLAB e MATPLOTLIB** sono due pacchetti per la visualizzazione scientifica bidimensionale. Le librerie possono essere ottenute tramite il sito <http://matplotlib.sourceforge.net/>.

Tutti i tool scientifici relativi al mondo python possono essere ottenuti installando una di quelle che vengono definite come “distribuzioni di python” per utilizzo scientifico. In particolare nello sviluppo di Spectra2, è stata utilizzata la **enthought python distribution**, ottenibile in versione accademica dal sito www.enthought.com. Questa distribuzione rimpiazza l’interprete python di default sul sistema ed assicura un ambiente “sano” a chi inizia a sviluppare od utilizzare prodotti basati su python per utilizzo scientifico. La distribuzione include i tool scientifici e grafici di utilizzo generico come scipy, numpy, matplotlib, traits, chaco ecc... e consigliamo di utilizzarla, installandola precedentemente agli altri tool.

Una volta installata EPD consigliamo di installare tramite i python setuptools alcuni pacchetti di utilità, riportiamo le istruzioni da eseguire:

```
$ easy_install pip
$ pip install virtualenv
$ pip install virtualenvwrapper
```

PIP è un gestore di pacchetti per python in grado di automatizzare l’installazione di librerie, mentre **virtualenv** è in grado di creare un ambiente virtuale in cui installare pacchetti senza intaccare la distribuzione python di sistema, una volta installato ci basterà eseguire:

```
$ mkvirtualenv spectra
$ workon spectra
(spectra)$ pip install iniparse construct curses
(spectra)$ pip install speed
(spectra)$ pip install katcp
(spectra)$ pip install corr
```

In questo modo avremo a disposizione tutti i tool necessari in un ambiente protetto dal quale non sarà possibile danneggiare il sistema.



4.2 Moduli di interfacciamento con lo spettrometro

Per comunicare con lo spettrometro viene utilizzato il modulo **specclient.py** dove è implementata la classe **SpecClient** in grado di programmare lo spettrometro caricando un firmware sulla FPGA ed impostando i valori dei registri ad uno stato iniziale specificato. Tramite questa classe è inoltre possibile agire sul registro software *cmd* comandando il funzionamento dello spettrometro, sono inoltre qui implementati i meccanismi di sincronizzazione tra il tempo del PC e quello dello spettrometro.

Tutti i parametri con cui effettuare il setup dello strumento sono descritti nel modulo **specconf.py** che altro non è che un file di configurazione scritto direttamente in python. In questo file sono specificati sia alcuni parametri osservativi come ad esempio la frequenza dell'oscillatore locale utilizzato, la frequenza di riposo e la sampling rate, sia alcuni parametri strumentali, come ad esempio il numero di integrazioni per spettro ed il valore iniziale di ogni registro, oltre agli indirizzi ip della macchine a cui connettersi ed a cui trasmettere i dati. Utilizzare questo file dovrebbe essere banale anche per gli utenti meno esperti, basandosi sull'esempio fornito.

4.3 Telescopio e scheduler

I moduli **med_server.py** e **noto_server.py** implementano il main loop del software di controllo e vanno utilizzati alternativamente a seconda del telescopio in cui ci si trova. Entrambi i moduli utilizzano un'interfaccia di comunicazione con il fieldsystem, definita in **medfs.py** e **notofs.py**, oltre a rendere possibile lo scheduling di osservazioni con tecnica on-off, le cui logiche sono definite nel modulo **scheduler.py** in grado di utilizzare le schedule raccolte nella directory *schedules* e definite tramite file di testo seguendo la sintassi da sempre utilizzata dal progetto ITASEL, simile alle schedule utilizzate per MSPEC0, ad esempio:

```
# RA Dec Epoch R_off D_rec VCF VCB Tcal sam ac_
chs_n_f_c TStart TStop cut_vICRS_Dur
3c295 141120.65 +521209.0 2000.0 1.0 0.0 xxp 320.00 8.0 9.7 16.0
58590 4096 4 4 4 172_14:15:00 172_15:00:00 5.0 0.0 45
3c309 145907.6 +714019.9 2000.0 1.0 0.0 xxp 320.00 8.0 9.7 16.0
58590 4096 4 4 4 172_15:15:00 172_16:00:00 5.0 0.0 45
MO 035425.8 +200650.1 2000.0 1.0 0.0 xxp 322.96 8.0 9.7 16.0
58590 4096 4 4 4 173_08:14:20 173_08:30:00 5.0 0.0 45
MO 035428.2 +200658.1 2000.0 1.0 0.0 xxp 322.96 8.0 9.7 16.0
58590 4096 28 1 1 173_08:34:20 173_08:50:00 5.0 0.0 45
MRO 035434.9 +200720.0 2000.0 1.0 0.0 xxp 351.55 16.0 9.7 32.0
58590 8192 4 4 4 173_09:29:30 173_10:00:00 5.0 0.0 45
```



La maggiore differenza rispetto al sistema precedentemente utilizzato è che in questo è possibile specificare per ogni scan l'istante di inizio e quello di conclusione, senza lasciare adito ad errori derivanti dal calcolo della durata di un ciclo di accumulazione.

Per il radiotelescopio di Medicina, dove il sistema è stato sviluppato e testato, è stato implementato anche il modulo **slavesystemserver.py** in grado di comandare le acquisizioni dello spettrometro in *piggyback mode* senza interferire quindi con l'operatività del telescopio.

4.4 Registrazione e formattazione

Come anticipato, lo spettrometro trasmette i dati in output tramite una connessione 10gbit ethernet. Normalmente, se i tempi di integrazione non sono troppo bassi, lo stesso PC da cui si comanda lo spettrometro può essere utilizzato per ricevere i dati tramite la rete. Il modulo **recorder.py** implementa in ricezione il protocollo utilizzato dallo spettrometro, effettuando quindi un check sul contatore iniziale di ogni frame UDP, segnalando e gestendo gli eventuali pacchetti persi. Il modulo implementa inoltre le istruzioni per estrarre dal payload dei pacchetti le informazioni riguardanti lo spettro trasmesso ed il suo timestamp, decodificato secondo il protocollo a 64 bit utilizzato da Spectra2.

Al modulo di ricezione dei pacchetti si affianca il modulo **fitswriter.py** utilizzato per formattare i dati ricevuti secondo lo standard FITS. Ogni scan viene salvato in un differente file *.fits* nella cartella *data/* e contiene due tabelle **CONF** e **DATA** contenenti i parametri di setup osservativi ed i dati ottenuti, rispettivamente. Nella tabella conf sono infatti salvati i parametri:

- **pname** string describing the project name.
- **sname** string, the source object observed.
- **ra** string, right ascension.
- **dec** string, declination.
- **rxcode** string, receiver code.
- **lo** float32, local oscillator value in MHz.
- **rsf** float64, rest frequency value in MHz.
- **vcf** float32, video converter zero frequency in MHz.
- **vcb** float32, video converter bandwidth, in MHz.
- **sfreq** float32, sampling frequency, in MHz.



- **svel** float32, source velocity.
- **epoch** float32, coordinates epoch.
- **raoff** float32, right ascension offset.
- **decoff** float32, declination offset.
- **nbins** int32, number of frequency channels per polarization.
- **navr** int32, number of averages per output spectrum.
- **non** int32, number of on-source acquisitions in each on-off-cal cycle.
- **noff** int32, number of off-source acquisitions in each on-off-cal cycle.
- **ncal** int32, number of cal acquisitions in each on-off-cal cycle.
- **tcal** float32, calibration mark temperature, in Kelvin.

Mentre la tabella **DATA** contiene gli spettri acquisiti assieme allo status dell'antenna, le coordinate effettive, le informazioni sul tempo meteorologico e l'istante esatto di inizio campionamento per ogni spettro acquisito. Riportiamo di seguito l'esempio di un file ottenuto durante una campagna di monitoring RFI avvenuta nell'ambito delle osservazioni ITASEL.

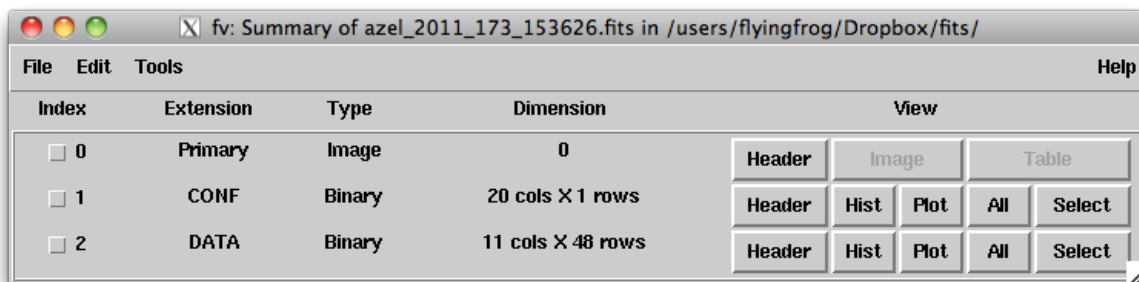


Fig. 15 le due tabelle nel FITS itasel



fv: Binary Table of azel_2011_173_153626.fits[2] in /users/flyingfrog/Dropbox/fits/

File Edit Tools Help

Select aaz 20A ael 20A ara 20A adec 20A press E mbar temp E C humid E

All

Invert	Modify	Modify	Modify	Modify	Modify	Modify	Modify
1	15.000	40.000	17:32:53	+78:02:10	1.013301E+03	3.080000E+01	5.040000E+01
2	15.000	40.000	17:32:58	+78:02:09	1.013301E+03	3.080000E+01	5.040000E+01
3	15.000	40.000	17:33:03	+78:02:09	1.013301E+03	3.080000E+01	5.040000E+01
4	15.000	40.000	17:33:08	+78:02:09	1.013301E+03	3.080000E+01	5.040000E+01
5	15.000	40.000	17:33:13	+78:02:10	1.013301E+03	3.080000E+01	5.040000E+01
6	15.000	40.000	17:33:18	+78:02:10	1.013301E+03	3.080000E+01	5.040000E+01
7	15.000	40.000	17:33:23	+78:02:10	1.013301E+03	3.080000E+01	5.040000E+01
8	15.000	40.000	17:33:28	+78:02:10	1.013301E+03	3.080000E+01	5.040000E+01
9	15.000	40.000	17:33:34	+78:02:09	1.013301E+03	3.080000E+01	5.040000E+01
10	15.000	40.000	17:33:39	+78:02:09	1.013301E+03	3.080000E+01	5.040000E+01
11	15.000	40.000	17:33:44	+78:02:09	1.013301E+03	3.080000E+01	5.040000E+01
12	15.000	40.000	17:33:49	+78:02:09	1.013301E+03	3.080000E+01	5.040000E+01
13	15.000	40.000	17:34:05	+78:02:09	1.013301E+03	3.080000E+01	5.040000E+01
14	15.000	40.000	17:34:10	+78:02:09	1.013301E+03	3.080000E+01	5.040000E+01
15	15.000	40.000	17:34:15	+78:02:09	1.013301E+03	3.080000E+01	5.040000E+01
16	15.000	40.000	17:34:20	+78:02:09	1.013301E+03	3.080000E+01	5.040000E+01
17	15.000	40.000	17:34:25	+78:02:09	1.013301E+03	3.080000E+01	5.040000E+01

Go to: Edit cell:

Fig. 16 Prima parte della tabella DATA del FITS itasel

fv: Binary Table of azel_2011_173_153626.fits[2]...

File Edit Tools Help

Select time J status 3A pol1 8192J pol2 8192J

All

Invert	Modify	Modify	Modify	Modify
1	1308760586	on	Plot	Plot
2	1308760590	on	Plot	Plot
3	1308760595	on	Plot	Plot
4	1308760600	on	Plot	Plot
5	1308760605	on	Plot	Plot
6	1308760610	on	Plot	Plot
7	1308760615	on	Plot	Plot
8	1308760620	on	Plot	Plot
9	1308760625	on	Plot	Plot
10	1308760630	on	Plot	Plot
11	1308760635	on	Plot	Plot
12	1308760640	on	Plot	Plot
13	1308760657	on	Plot	Plot
14	1308760661	on	Plot	Plot
15	1308760666	on	Plot	Plot

Go to: Edit cell:

Fig. 17 Seconda parte della tabella DATA del FITS itasel

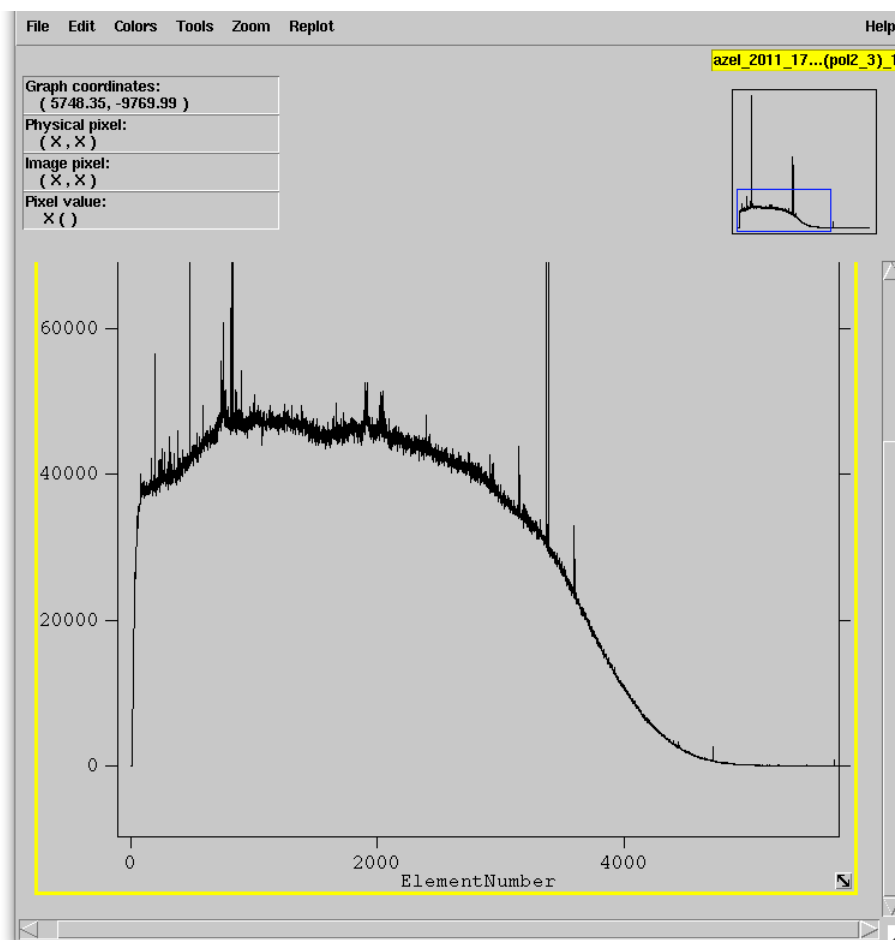


Fig. 18 Plot di una delle polarizzazioni registrata nel FITS itasel

5. Risultati

Lo spettrometro Spectra2 è stato fin'ora utilizzato in sole due sessioni osservative del progetto ITASEL nel corso dell'anno 2011, producendo i primi risultati utili e permettendo di effettuare una comparazione con gli strumenti utilizzati in precedenza.

Come mostrato in figura 19, sono stati osservati gli stessi target utilizzando simultaneamente lo spettrometro MSPEC0 ed il nuovo strumento SPECTRA-2 con setup strumentali pressoché identici. Come si può notare, SPECTRA-2 (in alto) detecta un segnale con un SNR maggiore di 4 mentre al contempo MSPEC-0 (in basso) non riesce a fare emergere il segnale dal rumore, pur mantenendo una similarità nella descrizione del noise pattern.

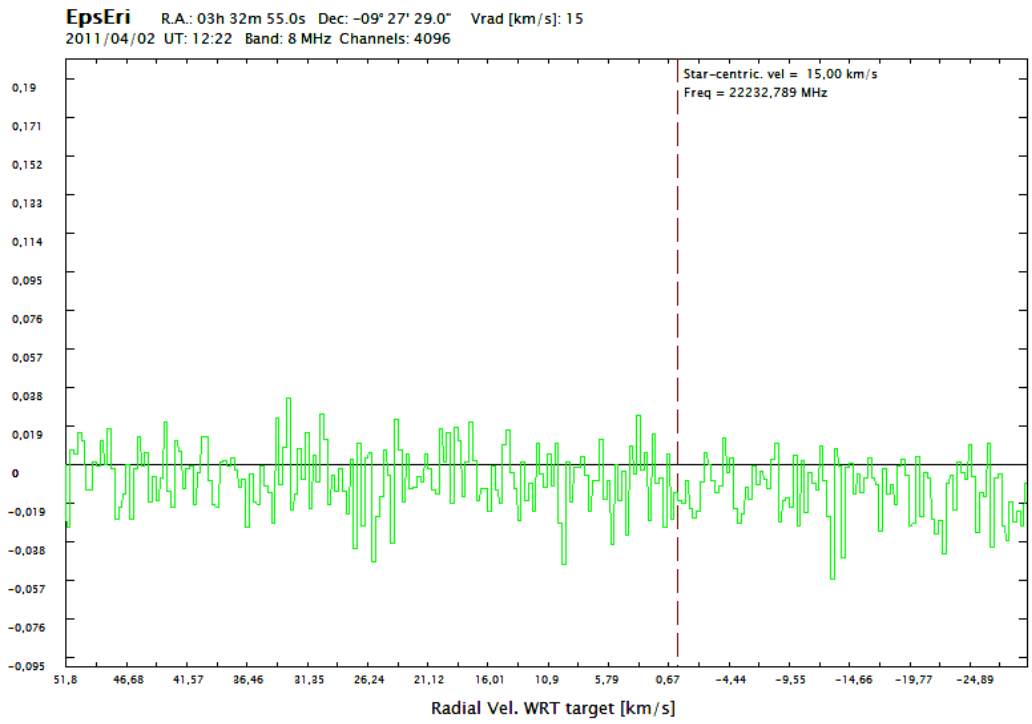
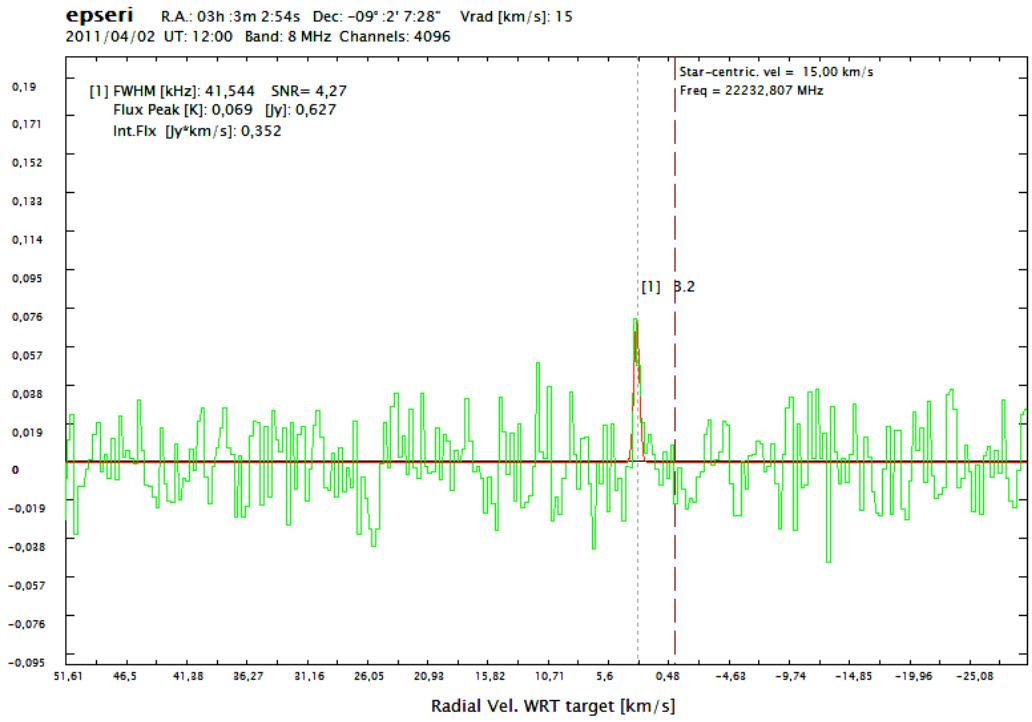


Fig. 19 Dati elaborati nel corso di una sessione ITASEL mostrano una possibile detezione di una riga di emissione a 22GHz



BIBLIOGRAFIA

- I. Paarsons, A. et al. *Digital Instrumentation for the Radio Astronomy Community*, arXiv: 0904.1181v1 [astro-ph.IM] 7 Apr 2009
- II. <http://www.med.ira.inaf.it/itasel/index.html> Home page ufficiale del progetto itasel.