Software di comunicazione con il correlatore Altera

G. Comoretto[1], A. Russo[2]

[1]INAF - Osservatorio Astrofisico di Arcetri
[2]Università di Firenze - Dip. di Astronomia

Arcetri Technical Report N° 2/2007

Sommario

Nell'ambito del progetto ALMA, sono state realizzate alcune schede che implementano un banco di filtri digitale. Queste schede utilizzano componenti programmabili, e quindi sono utilizzabili per realizzare uno spettrometro.

Una di queste schede è attualmente disponibile presso il laboratorio di radioastronomia di Arcetri, dove viene impiegata per sperimentare le diverse soluzioni circuitali realizzabili. In particolare sono stati realizzati alcuni componenti (ricevitore digitale, modulo FFT) che possono venir impiegati come elementi per costruire uno spettrometro modulare multicanale.

In questo rapporto si descrive l'interfaccia di programmazione di questi componenti. Per ciascun componente viene fornita l'interfaccia hardware a livello dei singoli registri di ciascun componente e un modello software definito in una struttura di programmazione ad oggetti. Gli oggetti descritti sono utilizzati in due programmi di esempio, in vista della realizzazione di programmi di controllo per lo spettrometro completo.

1 Introduzione

Nell'ambito del progetto ALMA sono state realizzate alcune schede che implementano un banco di filtri digitale. Le schede sono basate su componenti programmabili (FPGA o Field Programmable Gate Array)[1]. La stessa scheda può essere pertanto utilizzate per realizzare uno strumento completamente differente, semplicemente modificando la configurazione dei componenti programmabili utilizzati, ed in particolare è possibile impiegare una di queste schede per realizzare uno spettrometro completo. Si intende adoperare a questo scopo la versione di prova del banco di filtri per ALMA, che include un convertitore ADC con banda passante di 60 MHz, e frequenza massima di esercizio di oltre 300 MHz. Sfruttando questa scheda, è possibile realizzare alcuni semplici spettrometri, sia ad autocorrelazione che a trasformata di Fourier, con banda passante selezionabile da 0.5 a 62 MHz, capacità di tuning della banda in una finestra di 60 MHz (es. 240-300 MHz), e una risoluzione spettrale di circa 4000 punti nella banda esaminata. Sono in fase di realizzazione sistemi più complessi, comprendenti fino a 32 canali di ingresso indipendenti con una banda passante di 62.5 MHz ciascuno [2], basati sulla scheda finale del banco di filtri ALMA, insieme ad una scheda di sampler esterna.

Per quanto utilizzato essenzialmente come dimostratore tecnologico, le prestazioni di questo strumento sono confrontabili con quelle del correlatore ARCOS, e consentono di eseguire osservazioni astronomiche in condizioni reali. È pertanto necessario realizzare un programma di controllo, che consenta di eseguire osservazioni astronomiche standard, in modo simile a quanto avviene con il correlatore ARCOS.

I componenti implementati sulla scheda comunicano con un computer di controllo tramite un'interfaccia standard sviluppata per il progetto ALMA. Questa interfaccia è descritta nei relativi memo del progetto [1], e richiamata nel capitolo 2 di questo rapporto. Come microprocessore di controllo si impiega un PC single chip sotto sistema operativo Linux, che a sua volta comunica con il mondo esterno attraverso un'interfaccia Ethernet standard. Pertanto lo strumento è visto come una socket ad un particolare indirizzo IP.

In questo rapporto si descrive l'insieme di comandi utilizzato, lo schema concettuale dello spettrometro (cap. 3), il modello di programmazione dei componenti del correlatore, a livello di singoli registri hardware (cap. 4) e di struttura ad oggetti (cap. 5), e la struttura di un semplice programma di controllo (cap. 6).

1.1 Elenco delle abbreviazioni

ADC Analog to Digital Converter. Convertitore che trasforma un segnale analogico in una sequenza di campioni numerici.

ALMA Atacama Large Millimeter Array.

ARCOS Arcetri Correlation Spectrometer. Spettrometro a correlazione attualmente in uso presso i radiotelescopi di Medicina e Noto.

BBC Baseband Converter. Componente di un sistema radio (analogico o digitale) impiegato per convertire una porzione di una banda radio in una banda di frequenze che inizia a frequenza (quasi) zero. È anche indicato con il nome **convertitore SSB**.

CPLD Complex Programmable Logic Device: tipo di componente digitale programmabile, caratterizzato da una complessità logica non elevatissima, e dall'avere una configurazione non volatile, cioè che non viene cancellata quando l'alimentazione viene tolta (al contrario delle FPGA).

CPLD2 Una particolare interfaccia, utilizzata nel correlatore di ALMA, e basata su alcuni componenti CPLD. È descritta nel cap. 2.1.

DAC Convertitore Digitale-Analogico. Trasforma un segnale digitale (sequenza di campioni numerici) in un segnale analogico.

[1] Una FPGA è un componente elettronico programmabile costituito da un gran numero di elementi (celle) che possono essere configurate per realizzare funzioni arbitrarie, e che possono essere interconnesse in modo arbitrario tra di loro. La funzione che una FPGA può svolgere è quindi completamente programmabile: lo stesso componente può implementare un microprocessore, un filtro digitale, uno spettrometro, ecc. La programmazione (configurazione) di una FPGA viene specificata scaricando nel componente un file opportuno, e viene persa quando viene tolta l'alimentazione.

FFT Fast Fourier Transform.

FIR Finite Impulse Response (filter): Una classe di filtri in cui il segnale in ingresso viene direttamente convoluto con un una opportuna sequenza di lunghezza finita.

FPGA Field Programmable Gate Array: tipo di componente digitale programmabile, caratterizzato da una grossa complessità circuitale, in cui la configurazione viene memorizzata in una memoria volatile. Il componente deve quindi essere riprogrammato ad ogni accensione.

TFB Tunable Filter Board: Scheda di filtraggio sviluppata per il correlatore di ALMA. Contiene 16 grosse FPGA, impiegate originariamente per implementare un banco di 32 filtri digitali veloci.

1.2 Glossario

Backplane La scheda posta sul retro dello strumento, che interconnette tra di loro le rimanenti schede.

Broadcast Operazione in cui un comando identico viene inviato a più destinatari. È utile dove sia necessario sincronizzare tra di loro i destinatari, o per operazioni intrinsecamente lunghe, come la configurazione di chip programmabili.

Canale La parte di spettrometro che tratta un segnale radio indipendente.

Chip Un circuito integrato fisico (es. una FPGA). Il protocollo di interfaccia CPLD2 indirizza registri, posti all'interno di chip, all'interno di schede. Nel sistema sono definiti inoltre *chip logici*, cioé un insieme di più chip fisici indirizzati come una singola entità, e *sottochip*, porzioni di un chip che eseguono funzioni logicamente distinte. Tutti i chip fisici che compongono un chip logico devono condividere la stessa configurazione.

Complesso (segnale). Segnale rappresentato come sequenza di campioni complessi. La parte reale ed immaginaria di ciascun campione vengono anche indicate con le lettere I e Q, rispettivamente (in fase e in quadratura). Un segnale complesso rappresenta una banda di frequenza pari alla frequenza di campionamento, in cui frequenze positive e negative rappresentano porzioni di segnale fisicamente indipendenti. L'algoritmo FFT opera su segnali complessi.

Configurazione di un componente programmabile. L'insieme di dati che, scaricati in modo opportuno all'interno del componente, ne determinano le funzionalità. Può essere immaginato come uno schema elettrico, che può essere implementato dal componente.

Personalità di un componente programmabile. Sinonimo di configurazione.

Polifase (filtro) Un filtro in cui la risposta cambia nel tempo, in modo ciclico (quindi a seconda della *fase* del segnale, da cui il nome). Un filtro di questo tipo viene utilizzato per modificare la risposta di un processore FFT, riducendo la contaminazione tra punti spettrali adiacenti.

2 Interfaccia CPLD2, ed adattatore Ethernet

Il sistema di controllo dello strumento è costituito da un microprocessore industriale interfacciato ad una porta TCP-IP Ethernet, che a sua volta controlla, tramite un'interfaccia parallela, un bus proprietario sviluppato per il correlatore di ALMA.

Su questo processore gira un semplice programma di controllo, che accetta connessioni su di una porta TCP prefissata, ed implementa un semplice protocollo di comunicazione, traducendo comandi testuali inviti alla socket TCP in una serie di transazione sull'interfaccia standard del correlatore ALMA.

2.1 Descrizione del protocollo CPLD2

L'interfaccia per ALMA è basata su un bus dati a 8 bit, un bus di controllo a 4 bit, e un segnale di strobe per ciascuna delle schede indirizzate (fig. 1). Il protocollo è descritto in Broadwell [1]. L'architettura è stata scelta in modo da ridurne al massimo la complessità, a spese di velocità e flessibilità. Le sue caratteristiche fondamentali sono:

- La programmazione dello strumento avviene scrivendo valori in opportuni registri all'interno dei chip che lo compongono. I risultati vengono letti da registri di risultati. Non esistono meccanismi (tipo DMA) per scrivere direttamente i risultati nella memoria del processore.
- Ogni registro è identificato da 3 livelli di indirizzamento: scheda all'interno di un rack (fino a 14 schede), chip entro una scheda (fino a 16 chip, eventualmente più chip fisici possono essere raggruppati in una singola entità logica), registro entro un chip (fino a 256 registri). Gli indirizzi sono determinati dalla posizione fisica della scheda nel rack, o del chip nella scheda. Non esistono ponticelli o meccanismi di modifica dell'indirizzo.
- È possibile indirizzare simultaneamente un numero arbitrario di chip e di schede, scrivendo in parallelo lo stesso valore in registri analoghi di chip identici (broadcast)
- I registri hanno una lunghezza di uno o più bytes, con la convenzione che un valore a più bytes viene scritto indirizzando più volte lo stesso registro, e scrivendo per primo il byte più significativo. In lettura, i registri sono sempre da 1 byte.
- L'interfaccia è pensata per essere utilizzata in sistemi con chip programmabili (FPGA), per cui ha una serie di istruzioni specifiche che consentono il download della configurazione (il disegno elettrico) in questi chip.

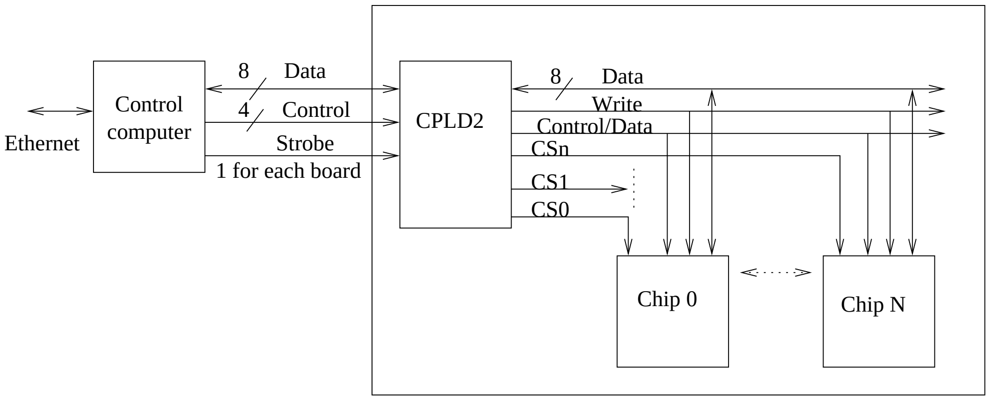

Figure 1: Struttura dell'interfaccia standard del correlatore di ALMA

Ogni scheda contiene un chip di interfaccia (CPLD2), indirizzabile direttamente, mentre per indirizzare i rimanenti componenti (chip) sulla scheda occorre prima scrivere il relativo indirizzo in un registro dell'interfaccia. Ogni chip a sua volta contiene un registro di controllo (indirizzo), e fino a 256 registri di dati, di cui in ogni istante uno solo è accessibile. Anche qui, il registro di controllo è accessibile direttamente, quello di dati solo dopo aver scritto nel registro di controllo il relativo indirizzo.

Dal punto di vista del programmatore il sistema è quindi visto come un insieme di registri, posti ai vari livelli della gerarchia di indirizzamento.

Per esempio, per scrivere il valore 0x10ff nel registro 7 del chip 5 della scheda 2 occorre:

- scrivere il valore 5 nel registro di indirizzo del chip CPLD2 della scheda 2
- scrivere il valore 7 nel registro FPGA Control della scheda 2. Questo fa sì che il valore venga scritto nel registro di controllo del chip 5

- scrivere il valore 0x10 nel registro FPGA Data della scheda 2
- scrivere il valore 0xff nel registro FPGA Data della scheda 2

L'interfaccia è semplice da implementare all'interno di una FPGA, ma non è particolarmente veloce. Velocità tipiche ottenibili sono di alcuni MByte/s, ma, per le limitazioni dell'interfaccia parallela del computer di controllo adottato, la velocità massima di lettura/scrittura nel nostro sistema è compresa tra 100 e 200 kByte/s.

2.2 Chip di interfaccia CPLD2

L'interfaccia CPLD2 ha uno spazio di indirizzamento complessivo di 8 registri, elencati in tabella 1.

Indir.	Scrittura	Lettura
0	Controllo	Stato
1	Broadcast reg1	Broadcast reg. 1
2	Broadcast reg2	Broadcast reg. 2
3	Test point select	FPGA data
4	FPGA control	
5	FPGA control broadcast	
6	FPGA data	
7	FPGA data broadcast	

Table 1: Elenco dei registri del chip di interfaccia CPLD2

Il registro di controllo/stato serve per varie funzionalità dell'interfaccia (es. iniziare il download della personalità delle FPGA). I registri di broadcast servono per selezionare che FPGA vengono indirizzate nelle operazioni di lettura/scrittura. E' possibile indirizzare una FPGA alla volta (unico modo sensato per operazioni di lettura) o un numero arbitrario di FPGA simultaneamente (operazioni di scrittura in broadcast).

Il chip pilota tre LED presenti sul lato frontale di ogni scheda. Questi LED segnalano, rispettivamente:

- LED 1 (rosso): segnala una condizione di out-of-lock nella circuiteria di clock (PLL) dei chip presenti nella scheda. Viene acceso anche impostando ad 1 il bit 2 del registro di controllo del chip.
- LED 2 (verde) Si accende ad ogni transazione sul bus, e si spegne 250 ms dopo che il chip indirizzato ha riconosciuto la transazione. Rimane quindi acceso fisso se il chip indirizzato non risponde correttamente.
- LED 3 (verde) Riflette lo stato del segnale `DONE` di programmazione dei chip programmabili. È quindi acceso quando questi sono programmati correttamente.

2.3 Protocollo di comunicazione su Ethernet

Il programma di comunicazione riconosce ed esegue comandi testuali, in modo da poter essere utilizzato tramite un qualunque programma di comunicazione (ad es. `telnet`).

Il programma accetta connessioni sulla socket 1234, e chiude automaticamente la connessione in presenza di errori. In ogni istante può essere attiva al massimo una connessione.

Il programma riconosce i seguenti comandi:

- `wl <board> <reg> <data>`: Scrive il dato `<data>` nel registro `<reg>` dell'interfaccia posta sulla scheda `<board>`
- `rl <board> <reg>`: Legge il registro `<reg>` dell'interfaccia posta sulla scheda `<board>`
- `wc <board> <chip> <data>`: Scrive il dato `<data>` nel registro di controllo del chip `<chip>` posto sulla scheda `<board>`

- `wd <board> <chip> <reg> <data>`: Scrive il dato `<data>` nel registro di dati `<reg>` del chip `<chip>` posto sulla scheda `<board>`
- `wm <board> <chip> <reg> <num> <data> <data> <data>`...: Scrive una sequenza di `<num>` dati nel registro di dati `<reg>` del chip `<chip>` posto sulla scheda `<board>`
- `rl <board> <reg>`: Legge il registro `<reg>` dell'interfaccia posta sulla scheda `<board>`
- `rd <board> <chip> <reg>`: Legge il registro `<reg>` del chip `<chip>` posto sulla scheda `<board>`
- `rm <board> <chip> <reg> <num>`: Legge `<num>` bytes dal registro `<reg>` del chip `<chip>` posto sulla scheda `<board>`
- `dl <board> <mask>`: Scrive nei chip selezionati dalla maschera `<mask>` posti sulla scheda `<board>` il file in formato MCS che segue. Il programma interpreta tutte le righe che seguono come record MCS (che cominciano con un ":" e contengono solo caratteri esadecimali) fino a quando non viene ricevuto un record di terminazione (di tipo 1, secondo lo standard MCS) o non viene rilevato un errore. La maschera è un intero di 16 bit, in cui ciascun bit posto ad 1 abilita un chip. Il bit meno significativo abilita il chip 0, quello più significativo il chip 15.
- `df <board> <mask> <file>`: Scrive nei chip selezionati dalla maschera `<mask>` (di 16 bit) posti sulla scheda `<board>` il file in formato binario specificato. Il file deve essere stato in precedenza caricato nel file system del computer di controllo dello strumento.

Questo protocollo è relativamente lento, per la necessità di decodificare i comandi testuali, ma consente di realizzare praticamente qualsiasi funzionalità in modo semplice e con il massimo controllo.

3 Descrizione delle schede ALMA

Nell'ambito del progetto ALMA, sono state realizzate tre schede Tunable Filterboard:

- Una scheda *pre-proptotipo*, contenente 3 chip Altera Stratix 1S40, un chip Altera Stratix 1S10, un convertitore ADC Analog Devices AD9433BSQ-125, e un convertitore DAC904 (fig. 2)
- Una scheda prototipo, contenente 16 chip Stratix 1S40
- Una scheda finale, contenente 16 chip Stratix2 2S40

Ogni scheda inoltre contiene un'interfaccia standard CPLD2, 3 chip Xilinx SpartanII 50, utilizzati nel correlatore per implementare un ritardo variabile, e un convertitore DC-DC per alimentare il core degli FPGA. Le schede si interfacciano ad un backplane in standard Hard-metric, da cui provengono le tensioni di alimentazione, il bus CPLD2, il clock e 96 linee di dati a 125 MHz.

Nell'uso normale, all'interno del correlatore di ALMA, il segnale radio viene rappresentato come 32 campioni consecutivi a 3 bit, alla frequenza di 125 MS/s, per un totale di 4 GS/s o una banda passante di 2 GHz. I campioni vengono forniti attraverso il connettore sul backplane, e i 3 bit vengono processati prima separatamente dai tre chip Xilinx, e poi insieme dai chip Altera. Una rete di distribuzione collega tra di loro le 16 FPGA e consente di portare ad ogni chip tutti i 96 bit di dati.

Nello spettrometro, i chip Xilinx posti all'ingresso della scheda non hanno una funzione specifica. Pertanto per questi chip si è utilizzata la personalità sviluppata per il Tunable Filter Board, in cui questi chip hanno essenzialmente funzioni di buffering del segnale. Per la programmazione di queste FPGA si rimanda al relativo manuale nella documentazione di ALMA [3].

Il riferimento di tempo, e di frequenza, sulle schede è garantito da un segnale di clock, di frequenza nominale di 125 MHz (segnale `CK125`), e da un riferimento di tempo, costituito da un'onda quadra con periodo di 1 millisecondo. Quest'ultimo segnale può essere prodotto sia internamente a ciascuna scheda, in uno dei tre chip Xilinx, che provenire dal backplane (segnale `STB`). Il segnale viene quindi distribuito all'interno della scheda, con un ritardo crescente allontanandosi dal backplane. Il ritardo esatto dipende dai dettagli di implementazione della logica nelle FPGA.

3.1 Scheda di prototipo ALMA Tunable Filterbank

La scheda di prova, contenendo al suo interno un convertitore ADC e un DAC, consente di realizzare un sistema *standalone*, e, per la sua semplicità concettuale, di provare in modo semplice i componenti di uno strumento completo. Lo schema a blocchi della scheda è mostrato in fig. 2.

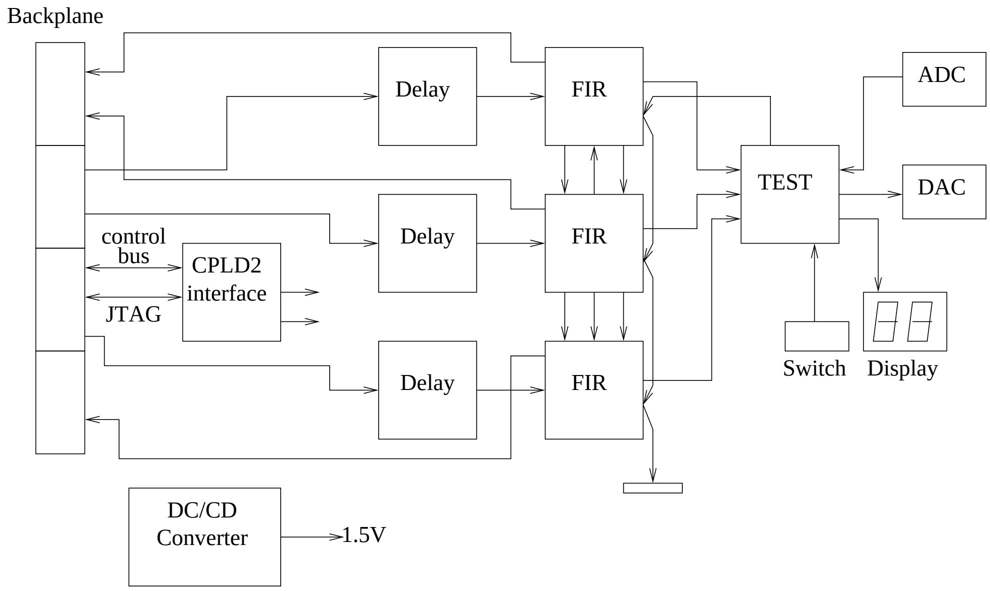

Figure 2: Schema a blocchi della scheda di prova per l'ALMA Tunable Fiterboard.

Ad esempio, nella scheda pre-prototipo è possibile utilizzare le linee di ingresso del backplane per fornire alla scheda segnali campionati da convertitori esterni, e realizzare fino a 6 canali spettroscopici nei 3 chip 1S40 (figura 3a).

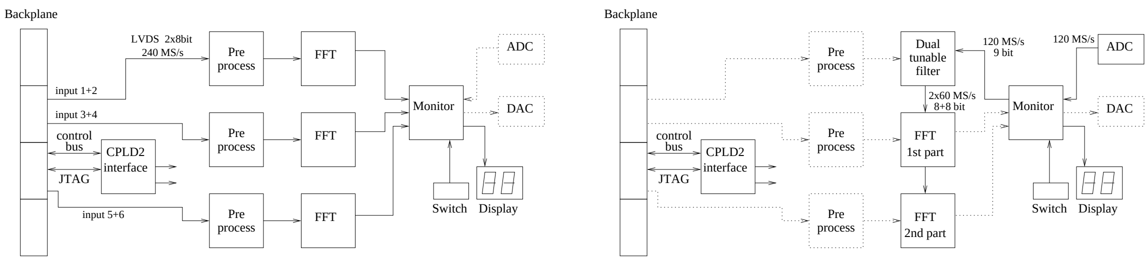

Figure 3: Esempi di utilizzo della scheda per realizzare spettrometri: (sinistra) 6 canali ad alta velocità, con campionatori esterni (destra) due canali nella banda di 60 MHz del campionatore interno.

Per far funzionare questa scheda in modo autosufficente, in una prima fase del progetto si impiegerà il convertitore ADC incluso. Da test di laboratorio si è verificato che il convertitore funziona correttamente fino alla frequenza di campionamento di 125 MHz, e quindi è possibile convertire la banda compresa tra 250 e 312 MHz, in uscita dalla media frequenza del terminale di acquisizione VLBI.

Il chip Altera 1S10 viene impiegato per generare segnali di test di vario tipo (rumore bianco, linea spettrale di frequenza nota), accanto al segnale proveniente dall'ADC. Il primo dei due rimanenti chip implementa una coppia di ricevitori digitali programmabili, ciascuno dei quali seleziona una porzione della banda passante di dimensione compresa tra 62.5 MHz (nel qual caso viene presa per intero), e 488 kHz. Questa porzione, campionata a 8 bit, viene inviata ai rimanenti rimanenti chip 1S40, che implementano uno spettrometro ed un filtro polifase per aumentare l'isolamento dei canali spettrali. (fig. 3b)

3.2 Scheda prototipo TFB

Sistemi più complessi sono implementabili utilizzando la scheda ALMA Tunable Filterboard completa, che comprende 16 FPGA Statix 1S40, organizzate secondo una matrice 4x4. Il relativo schema a blocchi è mostrato in fig. 4.

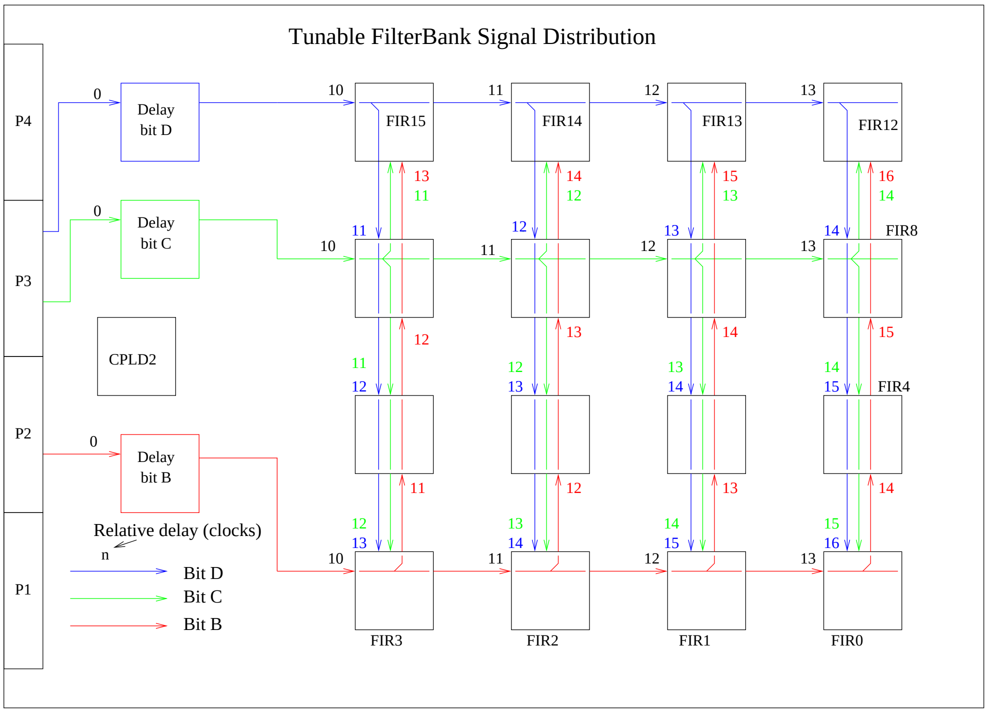

Figure 4: Schema a blocchi della scheda prototipo per l'ALMA Tunable Fiterboard. Sono indicati i percorsi della struttura di distribuzione dei segnali in ingresso

Utilizzando 8 convertitori ADC Analog Devices AD9480, con frequenza di campionamento di 250 MHz, montati su di una scheda realizzata ad hoc, è possibile realizzare uno spettrometro a 8 canali indipendenti, ciascuno con una banda passante compresa tra 0.5 e 62 MHz, sintonizzabile all'interno della banda 135-240 MHz. I blocchi funzionali sono tuttavia sempre i medesimi, e l'interfaccia di programmazione è stata scritta in modo da poter essere utilizzata in modo completamente modulare.

3.3 Meccanismo di indirizzamento delle schede ALMA

Le schede ALMA utilizzate possono contenere un numero elevato di chip programmabili, maggiore del massimo di 16 indirizzabili indipendentemente. Pertanto i 16 chip 1S40 e 2S40 sono raggruppati a due a due, occupando i primi 8 indirizzi dello spazio di indirizzamento della scheda. Inoltre ciascun chip ha la possibilità di realizzare due moduli indipendenti (due ricevitori digitali completi, nel caso dell'uso originale della scheda, o due canali spettroscopici, quando la scheda venga impiegata come spettrometro), per cui il blocco di due chip viene visto come un insieme di 4 entità logiche indipendenti. I rimanenti chip nelle schede sono indirizzati in modo convenzionale, utilizzando i rimanenti 8 indirizzi dello spazio di indirizzamento.

I 4 *sottochip* all'interno di un gruppo sono selezionati utilizzando i 4 bit più significativi del registro di controllo del gruppo (fig. 5. Il primo *sottochip* è selezionato dal bit 4, e l'ultimo dal bit 7 (MSB). In un'operazione di scrittura di un registro, il registro è quindi selezionato dai 4 bit meno significativi del registro di controllo, e pertanto lo spazio di indirizzamento per ogni sottochip è limitato a 16 registri. Risulta possibile scrivere un valore nello stesso registro di un insieme arbitrario di sottochip posti sulla scheda, utilizzando il modo *broadcast* dell' interfaccia. Ad esempio, scrivendo all'indirizzo esadecimale

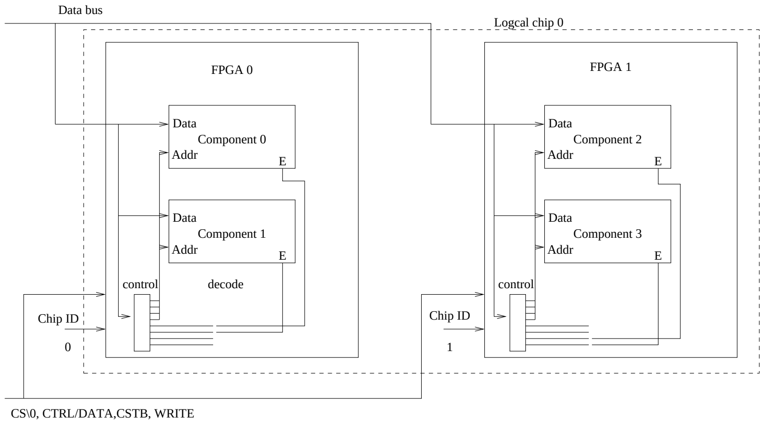

Figure 5: Schema di indirizzamento di 4 "sottochip logici" all'interno di un indirizzo dello spazio di indirizzamento dell'interfaccia

`a2` di un gruppo di chip, si scrive il valore specificato nel registro 2 del secondo e quarto sottochip (`a` esadecimale = 1010 binario).

Nella scheda pre-prototipo, contenente 3 chip 1S40, questi costituiscono il secondo chip (sottochip 2 e 3) dei gruppi indirizzati dagli indirizzi 5, 6 e 7. L'indirizzo 7 corrisponde al chip posto più in alto (primo nella catena di propagazione del segnale), l'indirizzo 5 a quello più in basso. La FPGA 1S10 è posta all'indirizzo 11. Nella scheda prototipo, l'indirizzo n corrisponde alla coppia di chip $\{2n, (2n+1)\}$, con la numerazione dei chip indicata in fig. 4. In entrambe le schede, i tre chip Xilinx sono posti agli indirizzi 8, 9 e 10 (dal basso verso l'alto).

Tutti i chip fisici che compongono un chip logico devono condividere la stessa configurazione, in quanto questa viene caricata in parallelo nei vari chip fisici. Questo comporta che la funzionalità implementata debba essere (quasi) la stessa, con piccole differenze impostabili ad es. programmando opportunamente dei registri.

4 Interfaccia hardware dei componenti

Dato il carattere di *banco di lavoro* di questo progetto, ogni componente (filtro digitale, integratore, processore FFT) è descritto in modo autonomo, con un set di registri indipendenti. Ogni componente può essere implementato all'interno di un arbitrario FPGA, che può contenere più di un componente (es. due ricevitori digitali). L'insieme di registri può quindi essere posto in un qualunque chip, in una posizione arbitraria del suo spazio di indirizzamento.

4.1 Filtro digitale

Il filtro digitale accordabile è composto da un mixer complesso, un oscillatore locale programmabile, e un filtro passa basso (fig. 6). A seconda delle applicazioni sia l'ingresso che l'uscita possono essere costituiti da uno stream di dati reale, a frequenza pari al doppio della banda passante selezionata, o uno stream complesso a frequenza uguale alla banda passante. Per uno spettrometro intrinsecamente complesso, come nel nostro caso di uno spettrometro a FFT, si utilizza un'uscita complessa. La programmazione del filtro é molto simile nei due casi. La maggior differenza consiste nel significato che viene dato alla frequenza dell'oscillatore locale: se l'ingresso e l'uscita sono reali, una frequenza negativa dell'oscillatore locale ha il solo risultato di rovesciare la scala delle frequenze, mentre in un convertitore complesso sia la banda di ingresso che quella di uscita comprendono frequenze negative.

Il filtro seleziona una banda passante prefissata B, centrata sulla frequenza dell'oscillatore locale ν_l (fig. 7. La banda si estende in ingresso da $\nu_l - B/2$ a $\nu_l + B/2$, e in uscita da $-B/2$ a $B/2$ (segnale complesso) o da 0 a $B/2$ (segnale reale). Sia ν_l che B sono espressi in funzione della frequenza ν_0 del clock

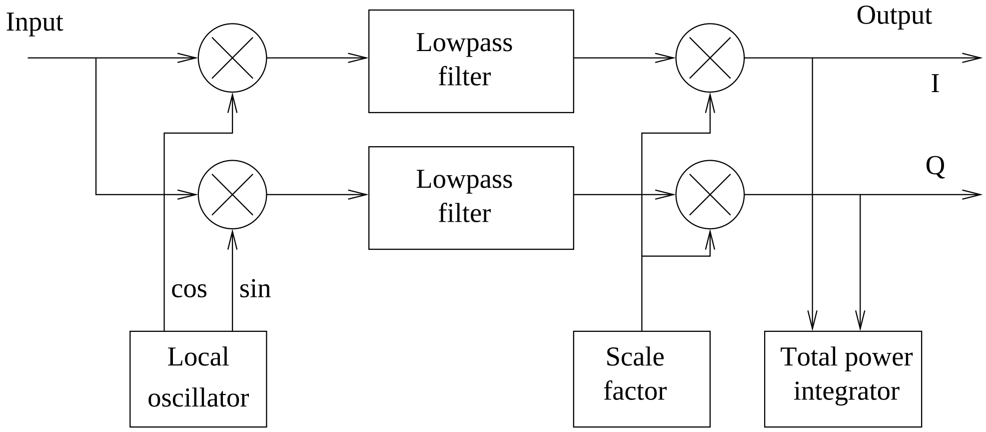

Figure 6: Schema concettuale del filtro passa basso accordabile.

interno, generato a partire da un clock esterno fornito alla scheda dal backplane. Nel nostro strumento il clock è fissato a 125 MHz.

L'interfaccia di programmazione del filtro è composta da otto registri, di cui solo i primi 5 sono utilizzati. Se in un chip sono presenti più filtri, i registri sono ripetuti, e ogni filtro viene programmato in modo del tutto autonomo.

Il segnale campionato viene convertito da un mixer complesso, pilotato da un segnale di oscillatore locale. La frequenza dell'oscillatore é espressa come un intero con segno a 24 bit. Il range di accordabilità della frequenza corrisponde a $\pm\nu_0/2$; se L è il valore programmato, la frequenza dell'oscillatore vale $\nu_l = L\nu_0/2^{24}$. Valori negativi di L selezionano le frequenze negative della banda in ingresso (nel caso di ingresso complesso), o rovesciano la scala delle frequenze (nel caso di segnale di ingresso reale).

La fase dell'oscillatore locale non è determinata, non essendo il sistema agganciato ad un riferimento assoluto di tempo, ma viene comunque azzerata quando viene inserito un nuovo valore di frequenza, in modo sincrono con un segnale di temporizzazione (1ms pulse) generato da uno dei tre chip Xilinx. Nel caso in cui sia importante mantenere una coerenza di fase tra segnali differenti, come ad esempio osservazioni polarimetriche, occorre impostare la frequenza dei ricevitori digitali interessati con un singolo comando di *broadcast*. In questo modo, la fase relativa tra gli oscillatori locali implementati in chip sulla stessa scheda è identica. Per garantire la coerenza di fase tra chip posti in schede differenti, occorre distribuire in modo controllato sia il segnale di clock che il riferimento di 1ms pulse.

Per semplificare la sincronizzazione della frequenza, questa viene specificata senza vincoli temporali, e memorizzata in un registro temporaneo. Il valore viene trasferito all'oscillatore locale e reso valido effettuando un'operazione di scrittura nel registro 4. Il valore scritto non ha importanza, ma l'operazione di scrittura avvia in modo sincrono al segnale di riferimento di tempo (quindi al successivo scoccare di un millisecondo) l'aggiornamento dei registri hardware. In questo modo, effettuando quest'ultima operazione in modo *broadcast* tra tutti gli oscillatori locali, si garantisce la loro coerenza di fase.

La banda passante è selezionabile in step binari tra $\nu_0/2$ e $\nu_0/256$, cioè tra 62.5 MHz e 488 kHz. Il segnale in uscita viene ricampionato ad una frequenza pari alla banda passante impostata. Il clock di ricampionamento è fornito in uscita assieme ai dati, come un impulso positivo valido durante il primo ciclo di clock di ciascun campione. Questo segnale è utilizzato per sincronizzare lo spettrometro. Pertanto cambiando la banda del filtro si seleziona automaticamente anche la banda dello spettrometro.

Il registro di test point è usato per inviare un segnale interno verso un pin nel connettore frontale della scheda. La funzione è usata solamente per debug, e l'elenco dei segnali interni dipende dalla specifica implementazione.

Il registro di controllo é usato per specificare la banda passante, azzerare e sincronizzare il misuratore di total power, e controllare alcuni bit di stato. In scrittura, i bit 2-0 specificano la banda passante in step binari. I bit possono assumere valori da 0, corrispondente a $B = f_c/256$, fino a 7 ($B = f_c/2$). Il bit 3 azzera il misuratore di total power, e può essere utilizzato per sincronizzare la misura, che parte

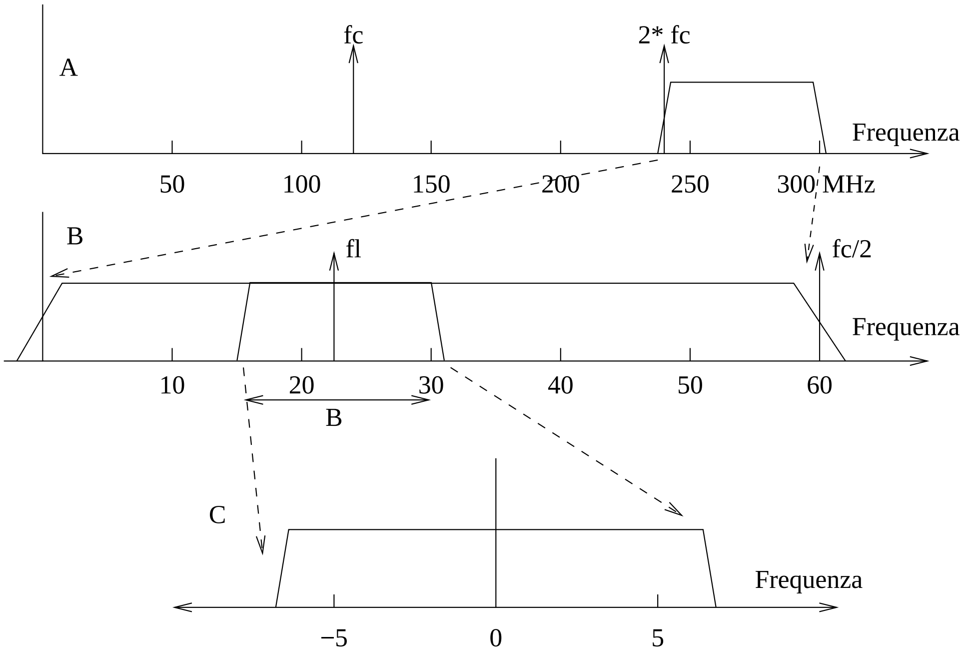

Figure 7: Processamento del segnale nel fitro accordabile: (A) Segnale in ingresso al campionatore, fc è la frequenza di campionamento. (B) segnale campionato, fl è la frequenza dell'oscillatore locale. (C) Segnale complesso convertito.

Indir.	Bytes	Funzione scrittura	Funzione lettura
0	1	Controllo	Stato
1	3	Frequenza	Total power LSB
2	1	Guadagno	Total power
3	1	Test Point select	Total power MSB
4	0	Impost. frequenza	

Bit	Funzione Scrittura	Funzione Lettura
0-2	Banda	
3	Reset T.P.	
4	Res. TP sync	T.P. sync
5	Res. ovfl	Overflow
6	Bypass mode	
7	Reset PLL	

Table 2: Elenco dei registri del filtro digitale (sinistra) e registro di controllo/stato (destra)

quando il bit viene rilasciato. Il bit 7 va pulsato all'accensione del sistema, per inizializzare il generatore di clock interno.

In lettura tutti i bit eccetto il 4 e il 5 ritornano il corrispondente bit scritto nel registro di controllo. Il bit 4 diventa vero quando un valore è stato scritto nel registro di total power (vedi sotto), e il bit 5 quando l'ampiezza del segnale filtrato è maggiore del range disponibile per rappresentarlo. Entrambi i bit sono azzerati pulsando (scrivendo un "1" e quindi uno "0") il corrispondente bit in scrittura.

Il bit 6 pone il filtro in *Bypass mode*: gli ingresso sono direttamente collegati all'uscita, e gli unici elemento che rimangono attivi sono il controllo di guadagno ed il misuratore di total power.

Il filtro contiene un misuratore di potenza del segnale filtrato, utile per funzioni di calibrazione. Il misuratore è composto da un rivelatore quadratico (un moltiplicatore), e un integratore, che accumula il segnale per un tempo pari a 0.2 secondi. Al termine dell'integrazione, i bit più significativi dell'integratore sono caricati in un registro di transito, accessibile tramite l'interfaccia CPLD2. Il bit 4 del registro di stato viene posto alto dopo che il valore è stato trasferito, e rimane alto finché non viene esplicitamente resettato. Il computer di controllo ha quindi 200 ms per leggere i tre byte del registro di transito, e resettare il bit 4.

Il bit 3 del registro di controllo può essere utilizzato per sincronizzare l'integratore dopo che la programmazione del filtro è stata cambiata. Quando il bit è posto ad 1, il circuito di total power è bloccato. Una transizione 1-0 di questo bit fa partire l'integrazione, garantendo che il successivo valore di total power sia valido, e non una commistione di valori presi con impostazioni differenti.

L'ampiezza del segnale filtrato varia notevolmente a seconda della banda passante e della forma del segnale in ingresso, ad esempio per la presenza di forti righe spettrali. Il filtro aggiusta in modo automatico il suo guadagno in funzione della banda, assumendo un una distribuzione spettrale piatta del segnale di ingresso (rumore bianco), ma non può compensare differenze di ampiezza dovute a variazioni dell'ampiezza in ingresso, o a distribuzione spettrale non bianca. È quindi utile disporre di un sistema di aggiustamento del guadagno. Questo può essere impostato tramite il registro 2 dell'interfaccia, che agisce come valore moltiplicativo all'interno del filtro e prima del misuratore di total power. Il valore piú adatto per questo fattore può essere scelto utilizzando il misuratore di total power, all'inizio dell'osservazione o in un loop di controllo automatico di guadagno.

4.2 Spettrometro FFT

Lo spettrometro FFT è realizzato utilizzando il kit di progettazione *jfft* sviluppato da J. Mock [4].

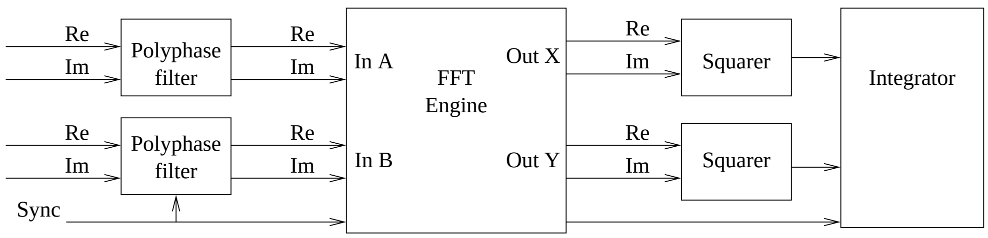

Figure 8: Schema concettuale dello spettrometro FFT

Il *motore* FFT é in grado di calcolare due trasformate complesse a 4096 punti in tempo reale. Il motore analizza due segnali complessi, *InA e InB*, e produce i relativi spettri, sempre complessi. La fase degli spettri è riferita a un segnale di sincronismo, che è generato internamente. Quindi ha significato solo la fase relativa dei due spettri, e non le fasi assolute di ciascuno dei due. È comunque possibile sincronizzare la FFT ad un particolare impulso del segnale di sincronismo di tempo, in modo da sincronizzare tra di loro più spettrometri.

Il numero di punti della trasformata è fissato dal disegno, e limitato dall'ammontare di memoria disponibile sul chip programmabile. La frequenza di clock é invece modificabile, e permette di ottenere risoluzioni più elevate al diminuire della banda passante analizzata. La massima frequenza di clock é pari a ν_0, cioé alla massima frequenza di campionamento del segnale in uscita dal filtro digitale.

Per limitare la necessità di memoria sul chip, il disegno non prevede un riordinamento dello spettro, che risulta disponibile in uscita in ordine *bit reversed*. Inoltre gli spettri dei due segnali sono sfalsati temporalmente. L'operazione di riordino viene effettuata durante la fase di lettura degli spettri da parte del computer esterno, dopo l'integrazione. In questo modo è possibile leggere uno solo dei due spettri, o anche solo una porzione dello stesso.

Il motore produce uno spettro complesso per ognuno dei due segnali di ingresso ogni 4096 cicli di clock, corrispondenti ad un *ciclo FFT*. Viene quindi calcolato il modulo quadro di ciascuno dei due spettri, e il relativo cross spettro (complesso). Il risultato viene integrato per un numero programmabile di cicli FFT. Il tempo di integrazione dipende quindi dalla banda passante esaminata B, edè necessariamente pari ad un multiplo di $4096/B$. Siccome B può variare tra 125 MHz (filtro in modo bypass) e 0.5 MHz, il tempo di integrazione elementare è compreso tra 33μs e 8 ms.

L'integratore è composto da due memorie indipendenti, la memoria di integrazione e quella di buffer. Durante il primo ciclo, la memoria di integrazione viene azzerata e il suo precedente contenuto trasferito in una memoria di buffer. Durante i cicli seguenti, le successive FFT vengono accumulate nella memoria di integrazione, mentre la memoria di buffer viene letta, in modo asincrono, dal computer di controllo.

Per ogni spettrometro devono essere letti 4 spettri (2 reali ed uno complesso, composto da parte reale ed immaginaria) di 4096 punti ciascuno, con ogni punto rappresentato da un intero a 4 bytes. Risulta quindi necessario leggere 64 Kbytes per ognuno degli 8 canali (a doppia polarizzazione) che compongono lo spettrometro. Ad una velocità di lettura di 100 Kbytes/s, per leggere i risultati di un'integrazione sull'intero strumento sono quindi necessari fino a 5-6 secondi. L'integratore deve poter quindi consentire tempi di integrazione di quest'ordine. Alla massima banda passante istantanea, un'integrazione elementare dura 33μs, e quindi l'integratore deve poter accumulare almeno 200.000 cicli elementari.

Per osservazioni su pochi canali d'ingresso, a bassa risoluzione spettrale e/o a banda stretta, il numero di integrazioni elementari può essere molto minore. Ad esempio, utilizzando lo spettrometro con banda di 0.5 MHz, e limitando il numero di canali e la risoluzione spettrale per avere tempi di integrazione di 1 secondo, l'integratore deve accumulare solamente 125 spettri elementari. Può essere utile in questo caso ridurre il numero di bit scartati, per aumentare il range dinamico.

In fase di lettura dei risultati può essere utile disporre di una funzione di media dei canali adiacenti. Ad esempio, impostando una media su 8 canali, si ottiene una risoluzione effettiva di 512 punti spettrali, riducendo in un fattore 8 il tempo di lettura. Questa funzionalità è utile quindi per ridurre i tempi di integrazione, ad es. per mappe on the fly.

La memoria di buffer contiene solo i 32 bit più significativi del risultato di ciascun canale, ed è leggibile a bytes. Viene indirizzata impostando un registro di indirizzo, a 15 bit, ed effettuando un numero arbitrario di letture nel registro zero dell'interfaccia. Scrivendo nella locazione 3, inoltre, si può incrementare il contatore di indirizzo anche senza leggere nessun dato.

Indir.	Bytes	Funzione scrittura	Funzione lettura
0	1	Test point	Test point
1	1	Controllo	Stato
2	2	Indirizzo lettura	Indirizzo
3	0	Incrementa indirizzo	Dato
4	3	Durata integrazione	

Table 3: Elenco dei registri dello spettrometro FFT

La tabella 3 descrive i registri presenti nello spettrometro FFT. Il registro di controllo (tab. 4) specifica alcune funzioni che impiegano un solo bit. In particolare il bit 0 viene utilizzato per resettare la logica del processore FFT, e deve essere pulsato ad 1 all'accensione dello strumento. Il bit 1 resetta la logica di integrazione. L'integratore è bloccato quando questo bit è posto ad 1, e comincia a funzionare quando viene posto a zero. Le integrazioni successive alla prima partono automaticamente, senza soluzione di continuità, consentendo di realizzare tempi di integrazione arbitrarimente lunghi. Il bit 2 viene usato in congiunzione con il corrispondente bit di stato, per segnalare il termine di un'integrazione. Al termine di una integrazione, il bit 2 di stato viene posto a 1, e viene azzerato quando il bit 2 di controllo viene pulsato. Questo consente al computer esterno di verificare quando nuovi dati sono disponibili. Il bit 3 di stato segnala la presenza di una condizione di overflow, e viene pure resettato dal corrispondente bit di controllo.

Bit	Funzione scrittura	Funzione lettura
0	Reset motore FFT	-
1	Reset integrazione	-
2	Reset fine integr.	Fine integrazione
3	Reset overflow	Overflow
6-4	Risoluzione	-
7	Reset PLL generatore di clock	-

Table 4: Assegnazione dei bit nel registro di controllo e stato

I bit 4-6 servono per selezionare la risoluzione spettrale. In fase di lettura degli spettri, l'interfaccia media tra di loro 2^n canali adiacenti, dove n è il numero rappresentato da questi 3 bit. Il numero di canali letti, e l'indirizzo di lettura, dipende pure da n. Ad esempio ponendo questi bit al valore $n = 5$, e leggendo il canale spettrale all'indirizzo 480, si otterrà una media dei valori contenuti nei canali spettrali da 480 a 511.

Il bit 7, infine, va pulsato all'accensione del sistema, per inizializzare il generatore di clock interno.

Il registro di stato serve per esaminare alcune condizioni del circuito, specificabili con un singolo bit. Sono utilizzati solo i bit 2 e 3, i rimanenti bit copiano i corrispondenti bit del registro di controllo. bit del registro di controllo. Il bit 2 viene posto ad 1 al termine dell'integrazione, per segnalare al microprocessore di controllo che sono disponibili nuovi dati, ed azzerato pulsando il bit 2 del registro di stato. Il bit 3 segnala la presenza di una condizione di overflow nel motore FFT, e viene pure azzerato pulsando il corrispondente bit del registro di controllo.

Il registro di indirizzo specifica la locazione della memoria di transito che verrà letta dalla successiva operazione di lettura dal registro di dati. L'indirizzo viene specificato come un numero di 14 bit, di cui i 12 meno significativi specificano il canale spettrale (che viene ordinato dalla massima frequenza negativa alla massima positiva), e i 2 piú significativi specificano il canale e il tipo di spettro (00 = total power A, 01 = total power B, 10 = real(AB), 11 = imag(AB)) . Quando si utilizza la funzione di media dei canali, gli n bit meno significativi vengono ignorati, con n il valore specificato nei bit 4-6 del registro di controllo.

Ulteriori 2 bit (non specificati) vengono usati dal contatore per selezionare uno dei 4 bytes di cui è composto ciascun risultato. Ogni volta che il registro di indirizzo viene scritto, questi bit vengono posti a zero, in modo da leggere sempre per intero la locazione di memoria a 32 bit. Il registro viene avanzato di una posizione (4×2^n bytes) scrivendo nel registro 3 dello spettrometro. Viene inoltre avanzato automaticamente di un singolo byte ad ogni lettura del dato di memoria. In questo modo, per leggere un blocco di canali è sufficiente specificare l'indirizzo del primo canale spetrale da leggere, ed eseguire un numero di letture pari a 4 volte il numero di canali da esaminare.

La lunghezza di integrazione viene specificata come numero di cicli di FFT da integrare, in una quantità di 18 bit. La durata di ciascun ciclo è pari a 4096 campioni in ingresso, per cui dipende dalla frequenza di questi ultimi. Per una banda analizzata di 62.5 MHz, il massimo tempo di integrazione è pari a $4096 \times (2^{18} - 1)/60\text{MHz} = 17.2$ secondi. Il massimo tempo di integrazione, per una banda di 0.5 MHz, è pari a circa 19 minuti.

Il circuito di integrazione considera il numero di cicli impostati per scalare ii valori in ingresso, eliminando un numero di bit meno significativi adeguato ad evitare overflow.

Il registro di test point consente di inviare due segnali interni del circuito, scelti tra 16 possibili, a 2 pin nel connettore frontale della scheda, dove possono essere esaminati con un analizzatore di stati logici o un oscilloscopio. I segnali esaminabili saranno decisi in fase di debug della scheda. I 4 bit più significativi selezionano la linea TP1, e quelli meno significativi la linea TP0.

Il registro di test point è rileggibile, in modo da poter eseguire semplici test di lettura/scrittura sull'interfaccia.

4.3 Generatore di test

Il circuito funziona utilizzando un sampler Analog Devices AD9433, presente sulla scheda. Può essere però utile disporre di un segnale di test, che può essere un tono a frequenza fissa, un tono a frequenza variabile (sweep), o un rumore pseudo casuale.

Pertanto si è aggiunto al circuito di controllo dell'ADC un generatore sinusoidale (un DDS), con possibilità di generare una sweep in frequenza, e un generatore di rumore pseudo casuale.

Il DDS è composto da un registro di fase a 24 bit, per cui la frequenza generata ha una risoluzione pari a $f_c/2^{24}$ (7.45 Hz per un clock di 125 MHz), e ad ogni millisecondo è possibile incrmentare la frequenza di una quantità prefissata. Siccome i relativi registri sono scritti un byte alla volta (iniziando dal più significativo), il DDS viene impostato in modo sincrono indirizzando il registro 8, con un'operazione di scrittura. Il valore scritto nel registro 8 è irrilevante.

Il generatore di rumore è composto da un linear feedback shiftregister a 35 bit, che genera 32 bit indipendenti ad ogni ciclo di clock. Due bytes sono interpretati come parte reale ed immaginaria di un numero complesso, con distribuzione statistica uniforme. Un sommatore produce ad ogni ciclo di clock la somma di 2 parole da 5 bit e due da 6 bit, ricavati dai 32 bit generati, in modo da ottenere un numero a 7 bit con distribuzione più simile a quella gaussiana.

I segnali prodotti dall'ADC, dal DDS, dal LFSR e dal generatore di distribuzione gaussiana sono moltiplicati per quattro numeri ad 8 bit, specificati in 4 registri, per ottenere il segnale di test inviato al resto del circuito. I segnali del DDS e del LFSR sono complessi, mentre i rimanenti due sono reali.

Il segnale può essere ricampionato, con un fattore di decimazione di 1, 2, 4, 8 o 16, e vene interpretato come numero complesso in 4 possibili modi, come mostrato in tab. 5.

Il circuito riceve dai rimanenti chip della scheda i segnali processati, che possono essere convertiti in forma analogica dal DAC presente sulla scheda. Tramite i bit 7–6 del registro di controllo si può scegliere quale segnale monitorare: il segnale prodotto (componente reale), o i segnali provenienti rispettivamente dai chip 7, 5 o 3.

Infine è possibile scrivere un numero esadecimale nel display a 7 segmenti.

Indir.	Bytes	Funzione scrittura	Funzione lettura
0	1	Test point	Test point
1	1	Controllo	Stato
2	3	Frequenza riga di test	-
3	2	Velocità di sweep	-
4	1	Livello segnale ADC	-
5	1	Livello riga di test	-
6	1	Livello rumore uniforme	-
7	1	Livello rumore gaussiano	-
8	0	Comando Start sweep	-
15	1	Valore visualizzato sul display	-

Bits	Funzione
2–0	Decimazione
	0 = nessuna, 4 = 1/16
4–3	Modo complesso
	0: solo reale
	1: solo immaginario
	2: Reale=immaginario
	3: true complex
6–5	DAC monitor select
	0=generator, 1-3=chip 7,5,3
7	Reset PLL

Table 5: Registri del generatore di test (sinistra) e registro di controllo (destra).

5 Descrizione ad oggetti del sistema

Per valutare in modo rapido il funzionamento del circuito, è stato sviluppato un sistema di classi ed un programma che consentono di programmare il circuito, leggere i risultati, e mostrarli in forma grafica.

Un primo insieme di classi (cap. 5.1) gestisce la comunicazione attraverso l'interfaccia CPLD2. Il programma utilizza un oggetto di questa classe per ogni strumento completo (di solito uno solo) controllato. Ogni elemento dello strumento viene controllato utilizzando un oggetto derivato dalla classe HardwareBlock (cap. 5.2), che utilizza l'oggetto di interfaccia per comunicare con il corrispettivo elemento hardware.

5.1 Gerarchia di oggetti per l'interfaccia CPLD2

L'interfaccia CPLD2 è stata modellizzata con una classe astratta, che definisce metodi generici per leggere e scrivere i registri posti ai vari livelli di indirizzamento (vedi cap. 2.1). La classe è stata quindi implementata in diverse versioni, che comunicano o direttamente con l'interfaccia fisica, o con il programma di server attraverso una socket. In questo modo l'utilizzo dell'interfaccia risulta identico per i programmi posti sul computer di controllo dello strumento, o su un computer client esterno.

La struttura della gerarchia è mostrata in fig. 9. L'interfaccia è descritta dalla classe Cpld2Intf. La classe astratta è specializzata nelle due classi Cpld2IntfHw e Cpld2IntfClient, che rispettivamente si avvalgono di un'interfaccia parallela standard, e di una socket TCP/IP. Quest'ultima è implementata utilizzando l'oggetto SocketClient, che apre una connessione verso un generico server TCP/IP, identificato da un indirizzo di rete e un numero di socket, e permette di trasmettere e ricevere generiche stringhe di testo.

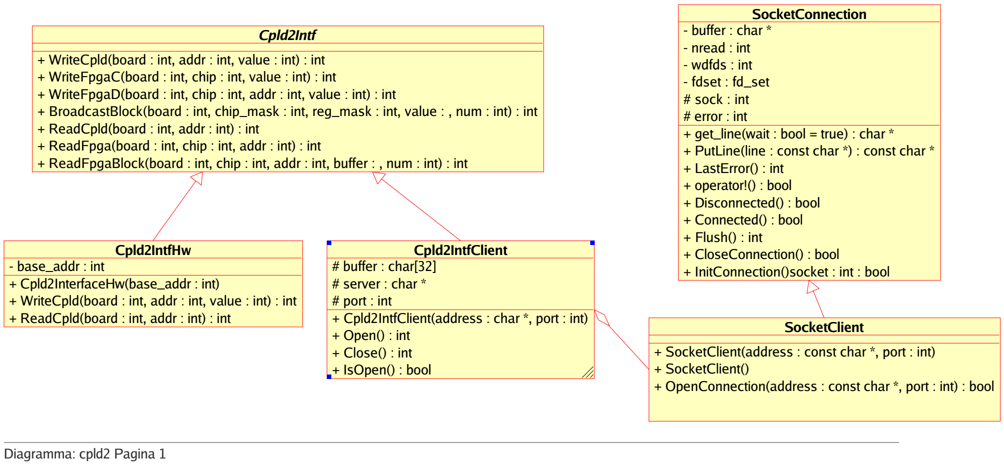

Figure 9: Diagramma statico della classe per l'interfaccia CPLD2

5.2 Elemento hardware generico

Ogni elemento dello spettrometro (filtro digitale, generatore, ADC, blocco FFT) viene descritto come un oggetto, con i metodi relativi. Questi oggetti sono derivati da una classe generica HardwareBlock, che implementa la comunicazione con l'oggetto fisico (attraverso un oggetto Cpld2Intf) ed alcune operazioni generiche.

In fase di istanziazione dell'oggetto, ne viene specificato l'indirizzo hardware, composto dal numero della scheda, del chip e dall' indirizzo di partenza all'interno dello spazio di indirizzamento del chip. L'indirizzo hardware è specificato come un intero, in cui il byte meno significativo viene interpretato come indirizzo di partenza, e i successivi come numero di chip e di scheda. Ad es. l'indirizzo `0x12340` specifica il chip 23 della scheda 1, con un offset di `0x40`. Il costruttore inoltre specifica in oggetto `Cpld2Intf`, creato in precedenza, in modo da creare un'associazione.

Un altro parametro fornito in fase di istanziazione è la frequenza del clock utilizzato, in modo da poter specificare tutti i parametri (banda, frequenza di cielo) in unità fisiche (Hertz, secondi). Tutti gli oggetti

Tutti gli oggetti hanno in comune alcuni metodi:

- Il metodo Reset() inizializza l'hardware ad una situazione nota. Viene eseguito automaticamente alla creazione dell'oggetto, per cui l'hardware viene effettivamente resettato all'avvio del programma.

- Set(): imposta i parametri specificati che, di default, vengono solamente registrati in registri interni dai metodi che li specificano, e non applicati. In questo modo lo strumento entra in uno stato definito, specificato da un insieme autoconsistente di parametri, utilizzando una singola operazione.

- TestPoint(int): imposta il registro di Test Point. L'argomento viene interpretato come un valore esadecimale a 2 cifre, ciascuna delle quali specifica quale di 16 possibili segnali interni viene inviato a due pin di test point hardware. Per alcuni componenti esiste un solo pin di test point, nel qual caso solo i 4 bit meno significativi sono utilizzati.

- bool CheckTP(int): Scrive un valore nel registro di test point, e lo confronta con il valore riletto. Ritorna `true` se il confronto è corretto. In genere, in questo modo, si può controllare se il chip è stato programmato correttamente.

- Control(int): Scrive il registro di controllo dell'elemento. Il registro è composto da 8 bit, con funzioni variabili. Di solito il bit più significativo è utilizzato per resettare il PLL di generazione del clock.

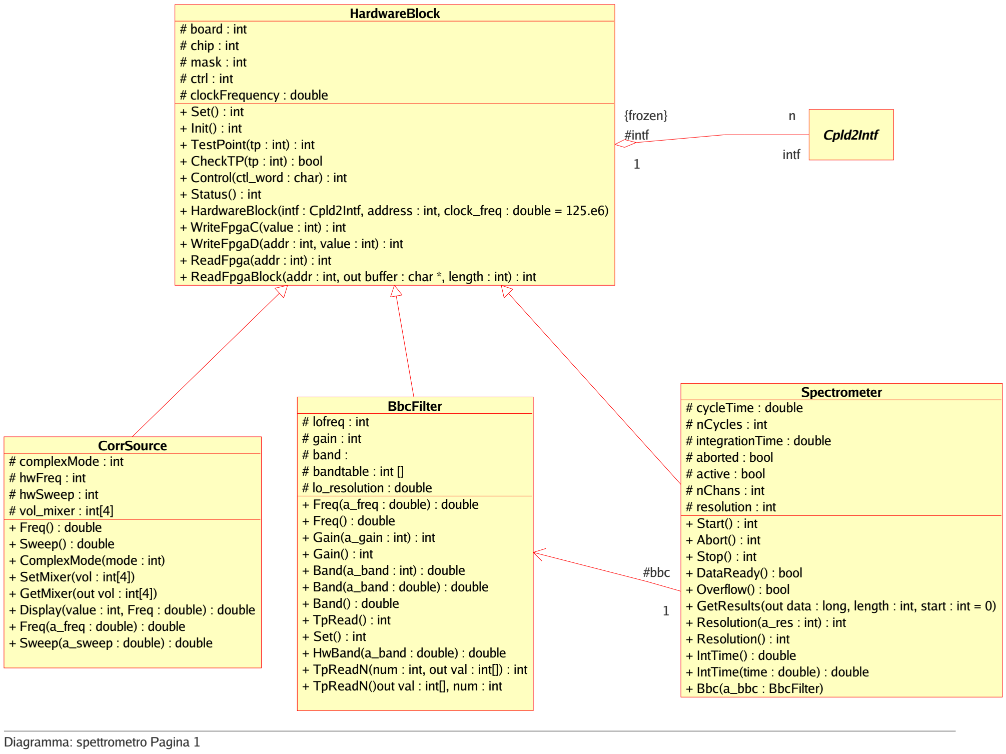

Figure 10: Diagramma statico delle classi che implementano gli elementi hardware dello spettrometro.

- int Status(): Legge il registro di stato. Anche questo registro è composto da bit con funzioni variabili.

Le ultime due operazioni sono generalmente usate all'interno dei metodi propri di ciascun oggetto derivato e non dal programmatore, in quanto ogni oggetto specifico implementa metodi che utilizzano in modo appropriato i bit di controllo/stato, senza imporre al programmatore di conoscerne la funzione specifica.

5.3 Generatore di segnale

Questo elemento (CorrSource) genera un segnale complesso, utilizzando un convertitore ADC, un generatore sinusoidale, e due tipi di generatore pseudocasuale. I metodi Freq(), e Sweep() permettono di specificare la frequenza (in Hz) e il drift di frequenza (in Hz/s) del segnale sinusoidale. Gli stessi metodi senza parametro consentono di rileggere i valori impostati. Lo step di risoluzione di frequenza è pari alla frequenza di clock f_c diviso per 2^{24} che, per $f_c = 125$ MHz, è pari a circa 7.45 Hz. La velocità di sweep è specificata internamente in step al millisecondo, e pertanto viene quantizzata in unità di circa 7.45 kHz/s.

I metodi SetMixer() e GetMixer() permettono di impostare il volume dei 4 segnali miscelati, come un intero nel range $[0 : 255]$, e di rileggere i valori correntemente impostati. Entrambi i metodi accettano come argomento un array di 4 `char`.

Il metodo Display() permette di visualizzare un byte come 2 caratteri esadecimali sul display collegato all'integrato Altera 1S10.

5.4 Convertitore BBC

Questa classe (BbcFilter) implementa il convertitore BBC descritto nel capitolo 4.1.

I metodi Freq, Band e Gain consentono di specificare la frequenza di centro banda e la banda totale (in Hz), e il guadagno dello stadio di uscita (in unità hardware, come intero da 0 a 255). I metodi senza parametro ritornano il valore effettivamente impostato, che risente delle operazioni di arrotondamento. In particolare la banda passante deve essere pari alla frequenza di clock divisa per una potenza di 2, per cui la banda programmata può essere significativamente differente. Il sistema arrotonda la banda al valore consentito più vicino, se questo non differisce di più dell'1%, o altrimenti utilizza la banda immediatamente maggiore a quella specificata. La funzione di servizio HwBand() ritorna il valore di banda adottato in funzione di quello specificato come argomento. È inoltre possibile specificare la banda con un codice numerico da 0 ad 8, dove 0 corrisponde alla frequenza di clock divisa per 256, 7 corrisponde alla minima decimazione (banda uguale a $f_c/2$) e l codice 8 a disabiltare completamente il filtro (modo bypass, ingressi copiati direttamente nelle uscite).

La lettura del misuratore di total power viene effettuata utilizzando i metodi TpRead(), che esegue una singola lettura del misuratore di total power, e TpReadN(), che legge in un vettore un numero arbitrario di misure consecutive.

5.5 Spettrometro

Questa classe (Spectrometer] implementa uno spettrometro FFT. I parametri della FFT specificabili dall'utente sono il tempo di integrazione (in secondi), e la risoluzione (in numero di punti spettrali), tramite i metodi IntTime() e Resolution(). Siccome il tempo di integrazione va specificato all'hardware in numero di cicli, la cui durata dipende dalla frequenza di campionamento del segnale in ingresso, e questa viene selezionata nel convertitore BBC posto davanti allo spettrometro, è necessario stabilire una relazione tra i due oggetti.

L'integrazione viene iniziata dal metodo Start(), e continua indefinitamente fino a che non viene inviato il metodo Stop() o Abort(). Quando i dati sono disponibili nella memoria di transito, viene posto a vero il flag di Data Ready, e i dati possono essere letti con il metodo GetResult(). Fino a che l'integrazione è attiva, sono disponibili nuovi dati ogni volta che è trascorso il tempo di integrazione specificato. Se i dati non sono letti in tempo, vengono sovrascritti dall'integrazione seguente. Il metodo Stop() blocca lo spettrometro al termine dell'integrazione in corso, e il successivo trasferimento di dati, Abort() la interrompe in modo drastico, senza trasferimento dei dati.

I metodi DataReady() e Overflow() controllano se i rispettivi flag sono attivi, ed eventualmente li azzerano.

6 Esempi di programma

Di seguito sono descritti due programmi. Il primo, puramente testuale, è un semplice programma senza parametri che esegue una misura della banda passante del filtro. Ne viene dato il codice completo come esempio di utilizzo degli oggetti descritti.

Il secondo è un programma che utilizza il sistema di interfaccia utente Qt. Permette di specificare tutti i parametri di un semplice sistema, composto da un generatore, due filtri e uno spettrometro, e di leggere e graficare i risultati della misura. Entrambi i programmi utilizzano componenti implementati sulla scheda pre-prototipo, secondo lo schema concettuale mostrato in figura 11.

6.1 Programma di misura della banda passante

Il programma in esempio consente di misurare la risposta spettrale di un filtro. Tutti i parametri sono specificati con costanti nel programma, per semplificarne la struttura.

Il programma inizia dichiarando gli oggetti che verranno utilizzati, e i parametri del test. Il filtro è impostato con una banda pari a 31.25 MHz, e la frequenza di centro banda a 31.25 MHz. La banda verrà analizzata campionandola in 1000 punti spettrali, separati tra di loro di 1/2000 della frequenza di clock (125 MHz).

```
//
#include "CorrSource.h"
```

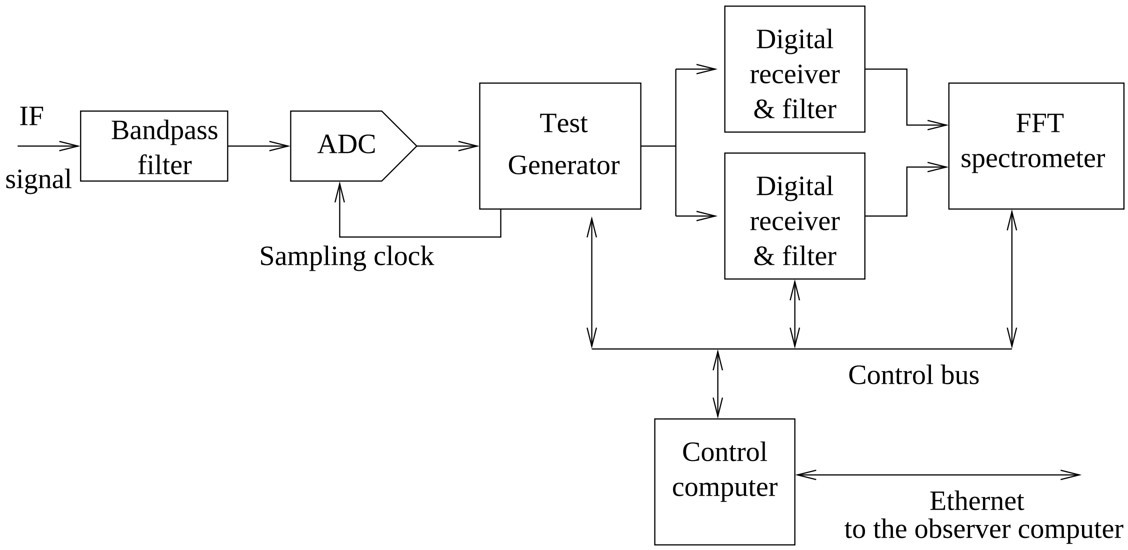

Figure 11: Schema concettuale dello strumento utilizzato dai programmi di test.

```
#include "BbcFilter.h"
#include "Cple2IntfClient.h"
#include <iostream>
using namespace std;

int main (int argc, char **argv)
{
   const int nchans=1000;               // Number of points to measure
   const double clockfreq=125.0e6
   double band = 31.25e6;
   double flo = 31.25e6;               // Generator center frequency
   const double step=clockfreq/nchans/2.0;  // Step for frequency
```

Quindi gli oggetti necessari vengono istanziati ed inizializzati. Innanzitutto viene inizializzata l'interfaccia, specificandone il nome di rete (`xarcos`) e la porta della socket (1234). Quindi sono istanziati il generatore e il filtro (quest'ultimo posto nel chip all'indirizzo 7). Infine vengono impostati la frequenza e la banda del filtro, e i livelli del generatore (solo il segnale del DDS ha un livello differente da zero). I metodi `Set()` rendono effettivi i valori impostati.

```
//
   Cpld2IntfClient intf(nodename, port);       // Interface to spectrometer
   CorrSource source(&intf, 0x000b00, 125e6);  // Test source (on chip 0xb)
   BbcFilter filt(&intf, 0x00720);             // Filter1 (on chip 0x7-2)
//
// Initialize hardware with meaningful settings
//
   filt.Band(band);
   filt.Gain(1);                       // For CW tone, very small gain needed
   filt.Freq(flo);                     // LO set to zero frequency. DC output
   filt.Set();
   int volume[4]={0,10,0,0};            // Sinewave level: adjust as needed
   source.SetMixer(volume);
   source.Set();
//
```

Infine il loop di misura imposta la frequenza del generatore, che viene incrementata ogni volta di 62.5 KHz, e legge il misuratore di total power del filtro. Il display a 7 segmenti indica il procedere della misura.

```
    double freq;
    for (freq=0.0; freq < 0.5*clockfreq; freq += step) {
         source.Freq(freq);
         int tpvalue= filt1.TPRead();
         cout << i << ' ' << freq << ' ' << tpvalue << '\n';
         source.Display((i*256)/nchans);
    }
    intf.Close();
    return 0;
}
```

6.2 Programma GUI

Il sistema controllato dal programma consiste di un generatore, due filtri, e uno spettrometro, connessi come in fig. 11.

Il programma consente di impostare un'integrazione, utilizzando come sorgente di dati il convertitore ADC o un generatore di test, analizzando due regioni indipendenti, ma con la stessa banda passante, poste in una posizione arbitraria della banda di ingresso. Lo spettro osservato viene visualizzato in un'area di plot, realizzata utilizzando l'insieme di classi IRA_plot, sviluppate presso la sezione IRA di Cagliari.

La schermata del programma è mostrata in fig. 12.

Il programma è stato sviluppato utilizzando il tool QtDevelop. Il metodo di inizializzazione crea gli oggetti utilizzati, ed inizializza lo strumento in una configurazione standard. Sono state scritte alcune *slot*, routines che vengono collegate dal tool di sviluppo all'interfaccia, e vengono eseguite in seguito ad eventi generati da quest'ultima.

Eventi vengono generati principalmente dai pulsanti, mentre in genere i controlli dei parametri vengono letti esplicitamente solo quando necessario. Pertanto i parametri devono essere esplicitamente inviati all'hardware, premendo i pulsanti "SET" e "INIT" (o "START"). Un'eccezione è costituita dai controlli del mixer nel generatore, che vengono applicati immediatamente (anche durante un'integrazione).

Quando un valore viene impostato, il relativo controllo viene aggiornato con il valore effettivamente utilizzato (ad es. per la frequenza), incluso eventuali errori di arrotondamento.

Il programma esegue un numero arbitrario di integrazioni elementari (parametro "Repeat"), mostrando al termine di ciascuna gli spettri ottenuti nei due canali. Lo spettro per il canale A è mostrato in azzurro, quello per il canale B in violetto.

Gli slider "Window start" e "Window width" controllano la posizione di partenza e l'ampiezza della porzione di spettro visualizzato, espresse come frazione della banda passante impostata.

Il pulsante "START" abortisce un'eventuale integrazione in corso, imposta i valori specificati ai filtri, e inizia l'integrazione. Viene inoltre impostato un timer, che ogni secondo controlla se sono presenti dei dati, ed aggiorna gli indicatori nella finestra "Progress". Al termine del numero prefissato di integrazioni, lo spettrometro viene bloccato e il timer fermato. È possibile fermare in ogni momento la serie di integrazioni con il pulsante "Abort".

References

[1] C. Broadwell (2004) "Alma Correlator Control Bus Manual" ALMA document CORL-60.00.00.00-020-B-MAN

[2] G. Comoretto, R. Nesti, F. Palagi, A. Russo: "A modular multichannel spectrometer - design study", Arcetri Technical Report n. 4/2006.

[3] G. Comoretto: "Programming Manual for the Tunable Filter Bank", ALMA document CORL-60.01.07.00-002-C-MAN

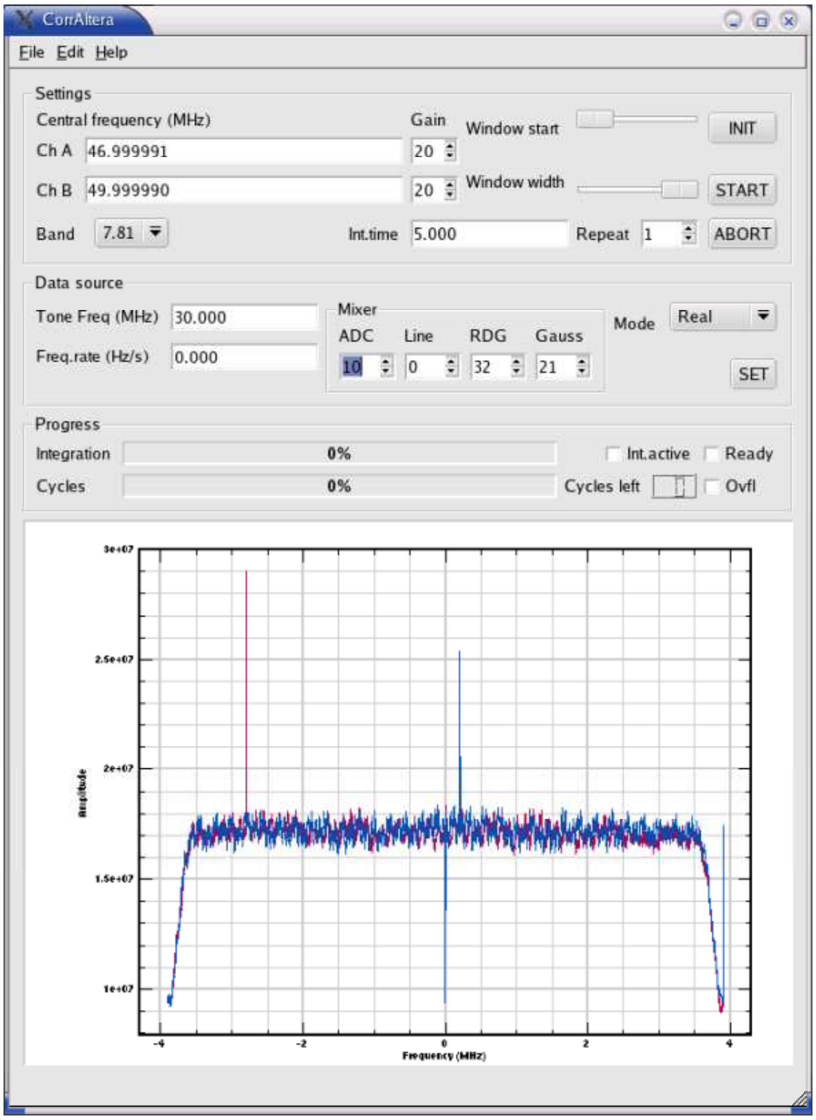

Figure 12: Interfaccia utente del programma dimostrativo di controllo dello spettrometro.

[4] J. Mock: "JFFT: A tool for generating synthesizeable verilog for streaming FFTs and polyphase filter banks" `http://www.mock.com/setistuff/jfft`

Contents