



Rapporti Tecnici INAF INAF Technical Reports

Number	347
Publication Year	2025-11-13
Acceptance in OA@INAF	2025-11-24T14:06:50Z
Title	ACQUISITION & CONTROL SYSTEM PROJECT FOR COSMIC RAY DETECTOR
Authors	RUSSO, Francesco
Publisher's version (DOI)	https://doi.org/10.20371/INAF/TechRep/347
Handle	http://hdl.handle.net/20.500.12386/45208

PROGETTO DI UN SISTEMA DI ACQUISIZIONE E CONTROLLO PER RILEVATORE DI RAGGI COSMICI

Francesco Russo (INAF/IASF-Palermo) - Technical staff (francesco.russo@inaf.it)

Rev. Documento 1.9 (Descrizione del FIRMWARE) — 13/11/2025

Sommario

INTRODUZIONE	3
DESCRIZIONE DEL SISTEMA.....	4
MODULO COUNTERS.....	5
MODULO SAMPLER.....	7
MODULO PEAK READER	9
MODULO TRIGGER MANAGER	11
MODULO I/O COMMUNICATION	12
MODULO I/O INTERFACE	14
MODULI DI SERVIZIO	15
CONCLUSIONI	16

Introduzione

Questo documento descrive il progetto di un sistema di acquisizione e controllo per strumenti di rivelazione di raggi cosmici. In particolare questo documento descrive, dal punto di vista del firmware, i vari moduli-IP (Intellectual Property) necessari per realizzare l'acquisizione di eventi rapidi caratterizzati da una durata temporale dell'ordine dei nanosecondi, generati ad esempio da radiazione Cherenkov prodotta da sciami elettromagnetici e, contemporaneamente, l'acquisizione di eventi con una durata più lunga, dell'ordine delle decine di nanosecondi o maggiori, generati ad esempio da fluorescenza atmosferica prodotta da sciami adronici. Inoltre questo documento descrive i moduli-IP dedicati alla lettura dei picchi analogici, alla gestione dei trigger, alla lettura degli housekeeping, alla generazione di impulsi di test, alla gestione della modulistica di servizio, ed infine alla gestione della comunicazione con un computer Host.

I moduli-IP firmware descritti in questo documento sono stati interamente sviluppati, testati e utilizzati dallo staff tecnico del Laboratorio di Elettronica dell'Istituto IASF-PA dell'INAF. La descrizione funzionale dei moduli è realizzata in codice VHDL, mentre il collegamento di tutti i moduli è stato realizzato in schematico, in modo da ottenere uno schema grafico dettagliato e facilmente fruibile dell'intero progetto. I dispositivi FPGA utilizzati sono della famiglia CYCLONE-V di ALTERA (Intel), come pure l'ambiente di sviluppo Quartus.

Per lo sviluppo di questo progetto si è fatto largo uso di strutture RTL e di molteplici macchine a stati dedicate che lavorano contemporaneamente in parallelo. In tal modo è stato possibile implementare e testare ogni modulo-IP in maniera isolata l'uno dall'altro, a beneficio di uno sviluppo ottimizzato alle prestazioni e anche all'eventuale riutilizzo dei singoli moduli-IP per progetti diversi da quello in oggetto. Inoltre si è preferito evitare l'uso di processori virtuali tipo NIOS o processori fisici SoC embedded, in quanto i primi richiederebbero una notevole quantità di risorse per un uso in parallelo come richiesto dal progetto, mentre i secondi sono presenti come singola unità all'interno di un chip FPGA, e quindi inutilizzabile in questo ambito. In ogni caso, l'utilizzo di questi processori nel progetto ne comprometterebbe un'eventuale portabilità.

Un ruolo determinante per il raggiungimento delle caratteristiche e performance descritte è svolto dalla complessa e completa definizione, a livello di progetto, di tutta una serie di constraint e assignment, per istruire il compilatore ad ottenere il miglior routing e fitting dell'intero progetto. Senza queste definizioni risulta impossibile portare a termine correttamente la compilazione di un così denso progetto, ed ottenere un sistema funzionante alle velocità di clock richieste.

Con suddetti dispositivi FPGA è stato possibile, per un totale di 64 canali di acquisizione, raggiungere le seguenti caratteristiche e prestazioni:

- Conteggio a 32 bit degli eventi generati dai raggi cosmici (+ il dark del rivelatore) fino a 250MHz per ogni canale di acquisizione
-
- Eventi rapidi di durata minore di 10ns:
 - Campionamento diretto e contemporaneo di tutti i trigger dei canali di acquisizione a 1GSMP/s, su memoria Free-Run ad anello
 - Algoritmo di trigger di pixel majority, di tipo pipeline a 1GSMP/s a bassa latenza, sul totale dei canali di acquisizione
 - Algoritmo per la produzione degli Hits di read-out per Peak-Detectors multipli
 - Lettura dei valori analogici dei Peak-Detectors, tramite controllore programmabile dedicato
 - Salvataggio continuo degli eventi (una porzione di campioni + i valori analogici dei Peak-Detectors) su Stack di memoria a doppio buffer
-
- Eventi di durata maggiore di 10ns:
 - Campionamento dei conteggi di ogni canale di acquisizione a 6-8 bit fino a 20MSMP/s, su memoria Free-Run ad anello
 - Algoritmo di trigger a soglia max con media mobile, sulla sommatoria dei canali di acquisizione
 - Salvataggio continuo degli eventi (una porzione di campioni di conteggio) su Stack di memoria a doppio buffer
-
- Gestione delle acquisizioni a partire dai trigger provenienti dai moduli di conteggio, di campionamento e da input esterno, tramite controllore dedicato
- Generazione di un Timestamp univoco a 64 bit per la marcatura temporale degli eventi, 32 bit per i secondi e 32 bit per i decimi di microsecondo
- Generazione di un ID-counter univoco a 32 bit per la marcatura sequenziale degli eventi
- Gestione della sincronizzazione dei segnali PPS e 10MHz per la generazione del Timestamp
- Trigger rate medio tipico sostenibile fino a 100KHz circa o superiore (in base al numero di canali triggerati ed al numero di campioni da leggere)
- Finestra di coincidenza programmabile e lettura dei Rate-meters dei triggers
-
- Gestione di dispositivi di servizio esterni come sensori di temperatura, voltage e current sensing, configurazione di ASIC, moduli di alta tensione, etc
- Possibilità di interfacciamento con detti dispositivi esterni tramite bus SPI, I2C, UART
- Generazione di impulsi di test programmabili in larghezza, periodo, ampiezza e numero finito o continuo di impulsi
-
- Gestione del protocollo di comunicazione con un computer Host tramite controllore dedicato
- Interprete dei comandi, risposte e payload, controllo di flusso, timeouts
- Flusso di I/O medio sostenibile fino a 100MB/s
-
- Possibilità di interfacciamento, con un computer Host, di tipo parallelo come FT245-SYNC e ASYNC, o seriale SPI HS, UART a 10Mbaud o superiore

Descrizione del Sistema

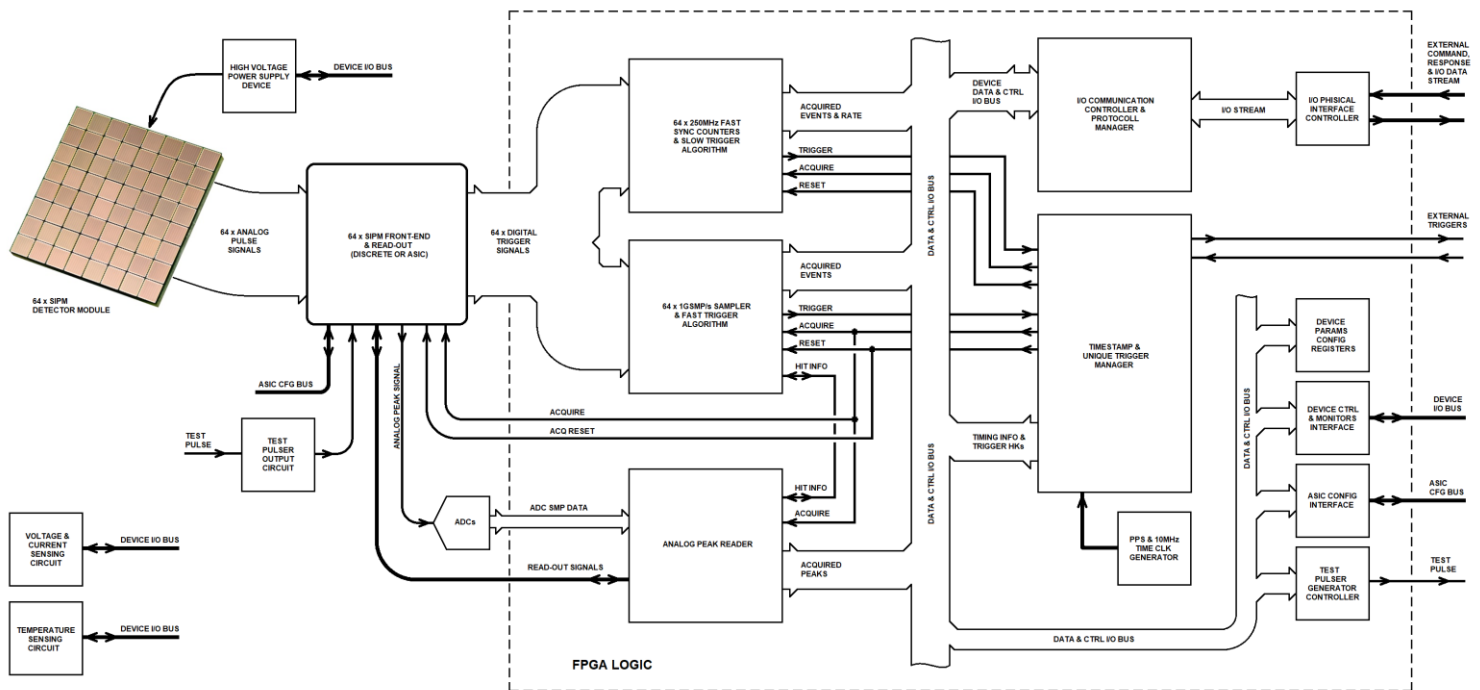


Fig 1: tipico sistema di acquisizione e controllo del rivelatore di raggi cosmici

Uno strumento di rivelazione della radiazione prodotta dai raggi cosmici è composto prima di tutto da un detector, interfacciato con una circuiteria elettronica di Front-End ad esso dedicata, tipicamente un ASIC, a sua volta connesso ad una elettronica di back-end adibita all'acquisizione, tipicamente un FPGA. L'FPGA è impiegato anche per l'interfacciamento con un computer Host, così da poter controllare lo strumento e leggere i dati scientifici acquisiti. Infine è presente una circuiteria secondaria di servizio che permette di interfacciarsi con sensoristica di temperatura, voltage e current sensing, moduli di HV, test pulse, etc. Anche questa circuiteria di servizio è asservita dall'FPGA.

È evidente che l'FPGA ricopre un ruolo fondamentale per la progettazione di uno strumento scientifico come appunto un cosmic ray detector. Di seguito sono descritti i moduli-IP che sono stati sviluppati per questo progetto, e come questi moduli sono interconnessi tra di loro dentro l'FPGA, al fine di ottenere un firmware di controllo e acquisizione completo.

Come visibile in Fig. 1, tutti i segnali di trigger in uscita dalla circuiteria di Front-End, di ciascun canale di acquisizione, sono connessi ai corrispondenti ingressi dell'FPGA. Internamente questi segnali digitali afferiscono contemporaneamente a due moduli-IP distinti: il modulo "Counters" ed il modulo "Sampler", che sono il cuore del sistema di acquisizione.

Il modulo-IP "Peak-Reader" realizza la lettura del picco dei segnali analogici, se questa funzionalità è presente nella circuiteria di Front-End. Questo modulo gestisce un ADC e i segnali di controllo per la selezione del canale da leggere.

I moduli Counters, Sampler e Reader sono interconnessi al modulo-IP "Trigger", che ne gestisce le acquisizioni e l'accumulo degli eventi in base alla formazione dei relativi trigger di acquisizione. Questo modulo si occupa anche della creazione di un Timestamp a 64 bit, con una risoluzione di 100ns, ed un ID sequenziale univoco a 32 bit per la marcatura temporale degli eventi.

Nel progetto sono presenti anche una serie di moduli-IP di servizio, necessari per la gestione della circuiteria secondaria di servizio prima descritta.

Tutti i moduli descritti sono interconnessi al modulo-IP "Communication-Manager", che realizza sia la gestione della configurazione di tutti i vari moduli, sia la lettura di tutti i dati scientifici acquisiti. Questo modulo a sua volta si occupa di colloquiare con un computer Host, tramite un protocollo proprietario sviluppato all'occorrenza.

Di seguito sono descritti nel dettaglio i vari moduli-IP presenti nel progetto.

Modulo Counters

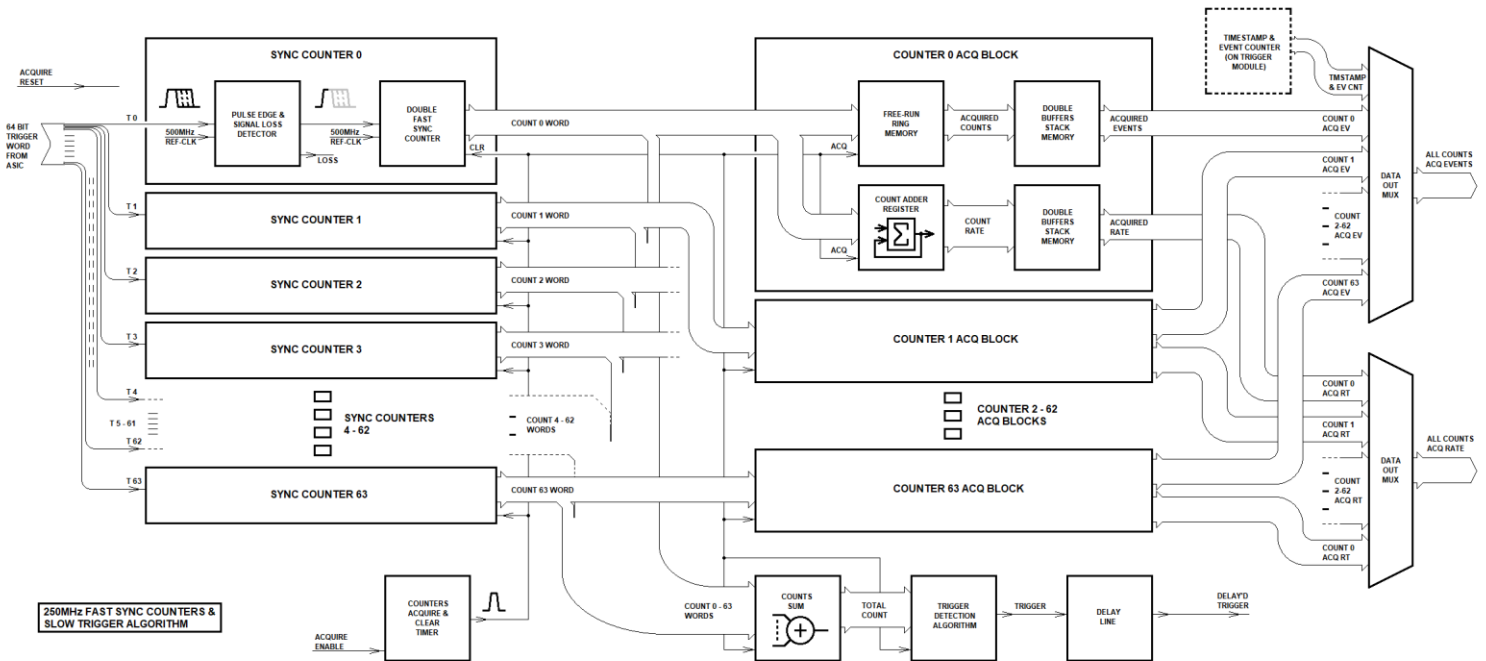


Fig 2: Modulo-IP con algoritmo di trigger per eventi >10ns e contatori veloci sincroni

Il modulo-IP "Counters" è adibito alla misura della radiazione prodotta dai raggi cosmici e contemporaneamente all'acquisizione degli eventi caratterizzati da una durata temporale dell'ordine delle decine di nanosecondi o maggiori. Queste due funzioni sono implementate per tutti i 64 segnali digitali in ingresso e provenienti dalle uscite trigger del Front-End.

Come visibile in Fig. 2, ogni impulso di trigger prodotto dal Front-End entra in un formatore di impulsi in grado di rilevarne il fronte di attacco. L'impulso così formato è contato tramite un contatore sincrono il cui valore è inviato in uno stadio formato da due catene indipendenti. Una si occupa di accumulare il conteggio su un apposito registro che servirà per la misura del fondo cosmico, mentre l'altra si occupa di registrare ogni conteggio su una memoria Free-Run ad anello.

La cadenza con cui i valori di conteggio vengono accumulati, registrati ed infine azzerati, è stabilita da un timer programmabile, controllato dal segnale di abilitazione dell'acquisizione proveniente dal modulo-IP di Trigger.

Oltre ad essere inviato allo stadio prima descritto il valore del contatore concorre, insieme ai valori dei contatori degli altri 63 canali di acquisizione, alla formazione del trigger di acquisizione degli eventi adronici. L'algoritmo di trigger è prodotto della somma dei 64 valori di conteggio, processati di continuo tramite una media mobile programmabile, il cui risultato infine è comparato con un valore soglia. L'uscita del comparatore servirà come ingresso al modulo di Trigger. Di ritorno da quest'ultimo arriverà lo stop dell'acquisizione che avvierà la procedura di arresto della memoria ad anello e il trasferimento dell'evento, cioè una porzione preimpostata dei conteggi in essa registrati, insieme a una etichettatura temporale, in uno Stack di memoria a doppio buffer, per poi riprendere l'attività di acquisizione fino al prossimo trigger, e così via.

Lo Stack di memoria a doppio buffer è dimensionato opportunamente per consentire l'accumulo di diversi eventi. La presenza del doppio buffer ha il duplice vantaggio di consentire l'acquisizione degli eventi senza introduzione di alcun tempo morto, ed inoltre di permettere lo scaricamento di questi buffer da parte del computer Host in maniera concentrata, occupando così il bus di comunicazione il meno possibile, a beneficio di altri processi di I/O.

Note:

Implementare ben 64 contatori a 32 bit ognuno dentro un FPGA, non è una operazione semplice. Intanto bisogna escludere a priori l'utilizzo di contatori asincroni. Per asincrono si intende un dispositivo sequenziale che non ha alcuna relazione con un clock di riferimento. Introdurre l'impulso che si vuole contare direttamente nell'ingresso del clock di un contatore è un'operazione sconsigliata, se non completamente errata. Un contatore a 32 bit è composto da 32 Flip-Flop più una logica combinatoria per realizzarne l'incremento voluto. Non c'è nessuna garanzia che l'impulso da contare venga distribuito dentro l'FPGA in maniera omogenea a zero spread su tutti e 32 i clock dei FF di un contatore. A maggior ragione per 64 contatori e soprattutto quando un FPGA è già denso di altra circuiteria logica da asservire. La corretta

distribuzione di un segnale di clock dentro un FPGA deve avvenire esclusivamente attraverso le linee dedicate che ogni dispositivo FPGA mette a disposizione per questo scopo. Ma queste linee sono un numero esiguo e ben limitato.

C'è poi il problema della lettura e del successivo azzeramento del valore raggiunto dal contatore. Questo processo sottostà sempre ad una operazione sincrona effettuata da un controllore dedicato che trasferisce il valore di conteggio in una memoria. Non c'è nessuna garanzia che questa operazione non legga un valore errato dal contatore, dovuto al suo incremento asincrono.

Riepilogando questo procedimento di sviluppo è impraticabile. Bisogna pensare per forza ad un contatore sincrono, in cui l'ingresso di clock del contatore è sempre sottoposto ad un clock di riferimento, e l'impulso da contare, opportunamente trattato, agisca solamente da abilitazione per l'incremento.

Detto questo bisogna considerare che l'impulso di trigger minimo in larghezza prodotto da alcuni ASIC è di circa 2ns, e che il fondo (+ il dark del rivelatore) può raggiungere la frequenza di diverse decine di MHz per ogni canale. Per soddisfare tutti questi requisiti, per tutti i 64 canali del Front-End, è stato sviluppato un doppio Fast-Counter sincrono operante a 500MHz di clock. Per poter funzionare a queste frequenze ognuno dei due contatori è stato pensato a 6-8 bit massimo, ed il più semplice possibile dal punto di vista circuitale, quindi con reset asincrono e senza controllo del ricircolo. La necessità del doppio contatore è dovuta proprio alla presenza del reset asincrono che impedisce l'azzeramento del contatore entro un ciclo di clock. I due contatori si alternano consecutivamente nel conteggio, e mentre uno è pronto per contare, l'altro è in stato di stop con il proprio valore disponibile per il trasferimento nello stadio successivo della catena di acquisizione e di accumulo del conteggio, per poi essere immediatamente azzerato con il reset asincrono. Tutto questo entro il periodo del timer di acquisizione, calcolato opportunamente in modo da evitare il ricircolo del contatore.

Il Fast-Counter è stato concepito per poter essere compilato in due modalità: come contatore standard o come shift-register più encoder. Questa possibilità è stata implementata per consentire una maggiore flessibilità in fase di compilazione, e quindi una maggiore probabilità di portare a termine con successo la compilazione del progetto. Molteplici fattori concorrono a questa riuscita, come ad esempio la famiglia di FPGA che si utilizza, la densità del progetto, la velocità di clock, etc.

Come già anticipato è stato necessario implementare diversi accorgimenti tecnici, nonché definire diverse constraint e assignment, per poter far funzionare correttamente ben 64 catene di conteggio così composte. La descrizione completa di queste note tecniche è demandata ad altro documento.

Modulo Sampler

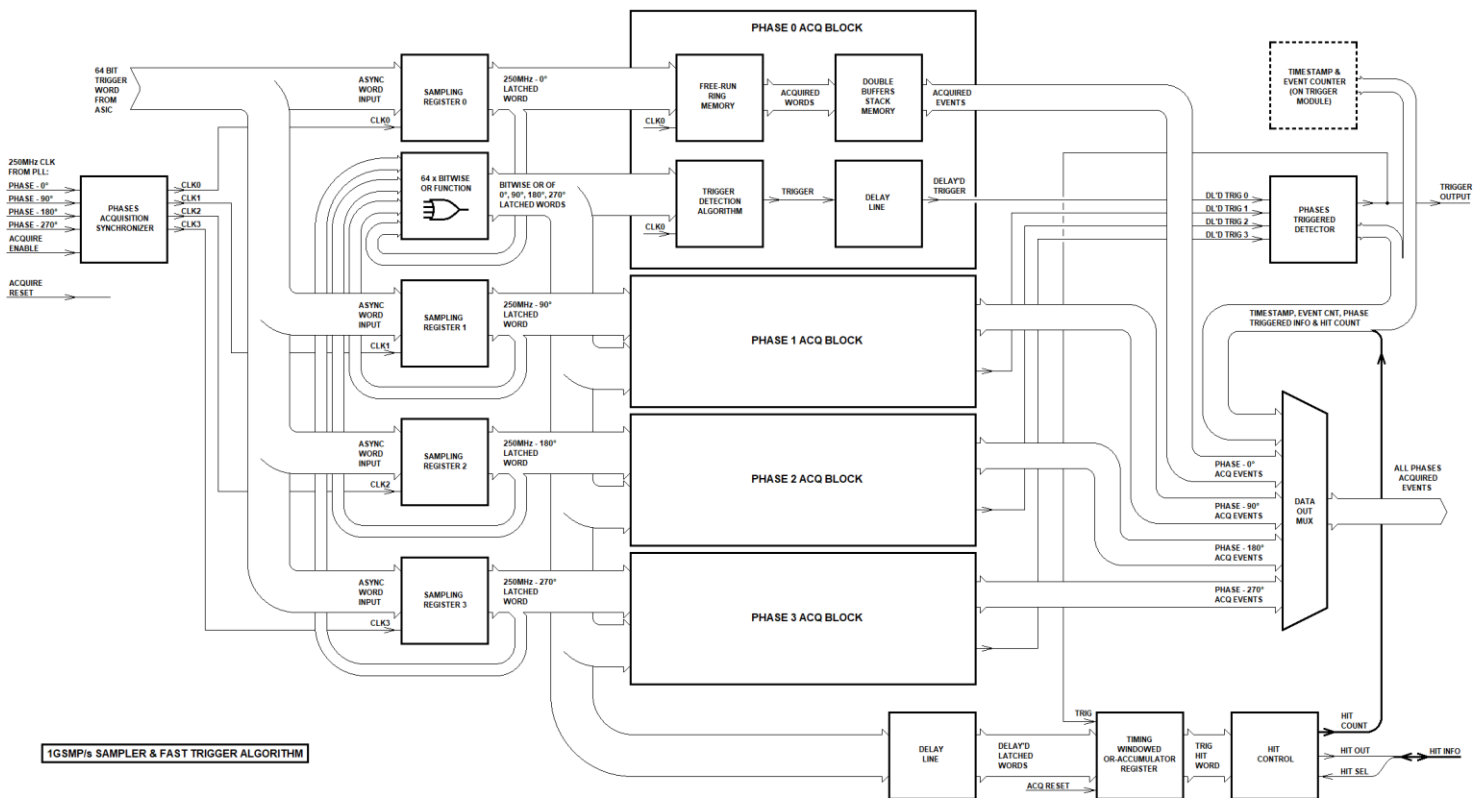


Fig 3: Modulo-IP con acquisizione a campionamento da 1 GSMP/s e algoritmo di trigger rapido

Il modulo-IP "Sampler" si occupa dell'acquisizione degli eventi rapidi caratterizzati da una durata temporale dell'ordine dei nanosecondi.

Il nome stesso di questo modulo ne descrive il comportamento. In pratica in questo modulo avviene il campionamento diretto, contemporaneo e continuo ad alta velocità di tutti i 64 segnali digitali in ingresso, provenienti dalle uscite trigger del Front-End. In particolare in questo progetto il campionamento avviene alla velocità di 1GSMP/s.

I campioni ottenuti dal processo di campionamento vengono registrati su una memoria Free-Run ad anello, e contemporaneamente indirizzati verso la logica di formazione del trigger di acquisizione. L'algoritmo del trigger è la somma aritmetica degli "1" logico presenti in ogni singolo campione, questo valore è poi comparato con una soglia di majority. L'uscita del comparatore servirà come trigger in uscita verso il modulo di Trigger. Di ritorno da quest'ultimo arriverà lo stop dell'acquisizione che avvierà la procedura di arresto della memoria ad anello e il trasferimento dell'evento, cioè una porzione preimpostata dei campioni in essa registrati insieme a una etichettatura temporale, in uno Stack di memoria a doppio buffer, per poi riprendere l'attività di acquisizione fino al prossimo trigger, e così via.

Come per il modulo "Counter", lo Stack di memoria a doppio buffer è dimensionato opportunamente per consentire l'accumulo di diversi eventi. La presenza del doppio buffer ha il duplice vantaggio di consentire l'acquisizione degli eventi senza introduzione di alcun tempo morto, ed inoltre di permettere lo scaricamento di questi buffer da parte del computer Host in maniera concentrata, occupando così il bus di comunicazione il meno possibile, a beneficio di altri processi di I/O.

Sebbene il compito di questo modulo-IP sia semplice nella sua funzione, la sua implementazione non lo è altrettanto.

Per raggiungere la frequenza di campionamento di 1GHz su un FPGA di fascia economica (ma non solo per questo...), è necessario suddividere il campionamento in più parti operanti a frequenze più basse, e poi fondere insieme i campioni delle singole parti così da ottenere il risultato iniziale voluto.

Come visibile in Fig.3, la word a 64 bit formata dai trigger è campionata da 4 registri di ingresso identici. Ogni registro campiona lo stesso segnale ad una frequenza di 250MHz, ma con uno sfasamento di clock di 90° l'uno dall'altro. In pratica i quattro campioni ottenuti sono distanti temporalmente 1ns, quindi è come se fosse un solo registro che opera alla frequenza di 1GHz. Successivamente i campioni di ogni registro sono indirizzati verso altrettanti stadi di registrazione operanti ognuno sempre a 250MHz e con sfasamento incrementale di 90°.

Per ottenere la funzione di trigger prima descritta, i campioni ottenuti dai quattro registri di ingresso sono computati bit per bit (bitwise) con una funzione di OR logica. Questa operazione è l'equivalente funzionale di una finestra di integrazione temporale di 7ns, dovuta alla sovrapposizione di 4 finestre da 4ns sfasate di 1ns, che scorre con una risoluzione di 1ns. In poche parole tutti gli "1" logico delle word di 4 campioni consecutivi concorrono alla formazione del trigger di acquisizione. In questo modo si evita la potenziale perdita di trigger quando un evento si distribuisce su più campioni consecutivi.

Per operare a queste frequenze, anche la logica di trigger è quadruplicata e fatta lavorare ad un quarto della frequenza nominale opportunamente sfasata. Inoltre l'ottenimento della somma degli "1" logico avviene attraverso più stadi sommatori sequenziali in configurazione pipeline. In questo modo è possibile ottenere una computazione del trigger di acquisizione per ogni campione di 1ns, ma con una latenza iniziale di qualche decina di ns dovuta alla profondità della pipe. Latenza poi compensata calcolando opportunamente l'offset di trasferimento dei campioni dalla memoria ad anello allo Stack.

Riepilogando, si è riusciti ad ottenere il campionamento a 1GSMP/s tramite 4 stadi di campionamento identici operanti ad un quarto della frequenza nominale opportunamente sfasati. Per raggiungere questo obiettivo è fondamentale che questi 4 stadi siano sincronizzati nell'acquisire i campioni rispetto al segnale ACQUIRE che stabilisce l'inizio e lo stop dell'acquisizione. Questa funzione indispensabile e basilare si è ottenuta tramite un circuito sincronizzatore di fase opportunamente progettato e sviluppato per l'occorrenza. In questo sincronizzatore il numero di stadi da sincronizzare è parametrizzabile in fase di progetto.

Un altro circuito indispensabile è il rivelatore di fase dei trigger. Data la natura dell'algoritmo del trigger di acquisizione che integra i campioni in bitwise-OR tra le quattro fasi di campionamento all'ingresso dei quattro stadi di trigger, questi ultimi inevitabilmente produrranno tutti un trigger indipendentemente da quale campione delle quattro fasi contiene l'informazione utile a superare la soglia di majority. È necessario quindi un circuito che consente di identificare quale dei quattro stadi di trigger ha superato la soglia per primo, e fornisca così l'informazione che permette a posteriori di associare esattamente il trigger al campione corrispondente, compreso nei campioni dell'evento acquisito, con la precisione del ns.

Infine in questo modulo-IP Sampler è presente la circuitazione adibita alla creazione delle Hit-Info necessarie ad identificare, da parte del modulo-IP "Peak Reader", i canali del Front-End da cui leggere i valori analogici dei picchi, tramite un opportuno ADC. Tale circuitazione può operare in diverse modalità:

1. Solo i campioni (in bitwise-OR) dell'evento che producono il primo trigger sono considerati
2. I campioni del primo evento più tutti quelli degli eventi successivi sono considerati
3. I campioni del primo evento più tutti i campioni successivi (indipendentemente se producono un trigger) sono considerati

In più è possibile abilitare una finestra temporale programmabile, durante il quale effettuare l'integrazione in OR delle word.

Questo circuito provvede a supplire la mancanza in alcuni Front-End di tale funzionalità, o addirittura di rimpiazzare lo stesso circuito dove già presente, in quanto questa versione digitale risulta molto più precisa e avanzata nella sua funzione.

Note:

Un aspetto importante della catena di trigger è che questa è affetta da una latenza dovuta alla pipeline. Affinché sia possibile costruire un algoritmo di computazione veloce, capace di valutare il superamento della majority per ognuno dei campioni di 1ns, è inevitabile l'utilizzo di una pipeline e la latenza che questa comporta. Ma è fondamentale che questa latenza sia sempre precisa e costante. Il fatto di avere una latenza per quanto breve, viene comunque compensato tramite l'aggiunta di un offset di lettura in fase di trasferimento, ma solo se quest'offset non cambia mai indipendentemente dalla conformazione del campione da valutare. La logica di trigger di questo modulo-IP è stata progettata proprio in modo da avere sempre una latenza predeterminata.

Uno dei motivi che limita la velocità di campionamento è quello della velocità massima di scrittura della memoria Free-Run ad anello. In questo progetto si è deciso di limitare a 1GSMP/s la velocità di campionamento in quanto adeguata per l'ottenimento dei risultati scientifici richiesti. Con tale frequenza ognuno dei quattro stadi lavora ad una frequenza di 250MHz. Questa frequenza è perfettamente sostenibile dalla memoria embedded presente in un FPGA della famiglia CYCLONE V (C-6). Volendo estrapolare al massimo questa filosofia di campionamento, sfruttando il massimo delle risorse disponibili in un CYCLONE V, e cioè la frequenza massima di scrittura di 315MHz, e 8 clock a 315MHz sfasati di 45°, si potrebbe raggiungere una velocità di campionamento di oltre 2.5GSMP/s. Se la stessa estrapolazione la calcoliamo per un FPGA di fascia alta ben più performante e costoso come uno STRATIX10 sempre di Altera, con una fmax di scrittura di 600MHz, e ben 16 clock sfasati di 22.5°, otterremmo una ragguardevole velocità di campionamento di quasi 10GSMP/s.

Anche per questo modulo-IP è stato necessario implementare diversi accorgimenti tecnici, nonché definire diverse constraint e assignment, per poter far funzionare correttamente il campionamento a 1GSMP/s così come esposto. La descrizione completa di queste note tecniche è demandata ad altro documento.

Modulo Peak Reader

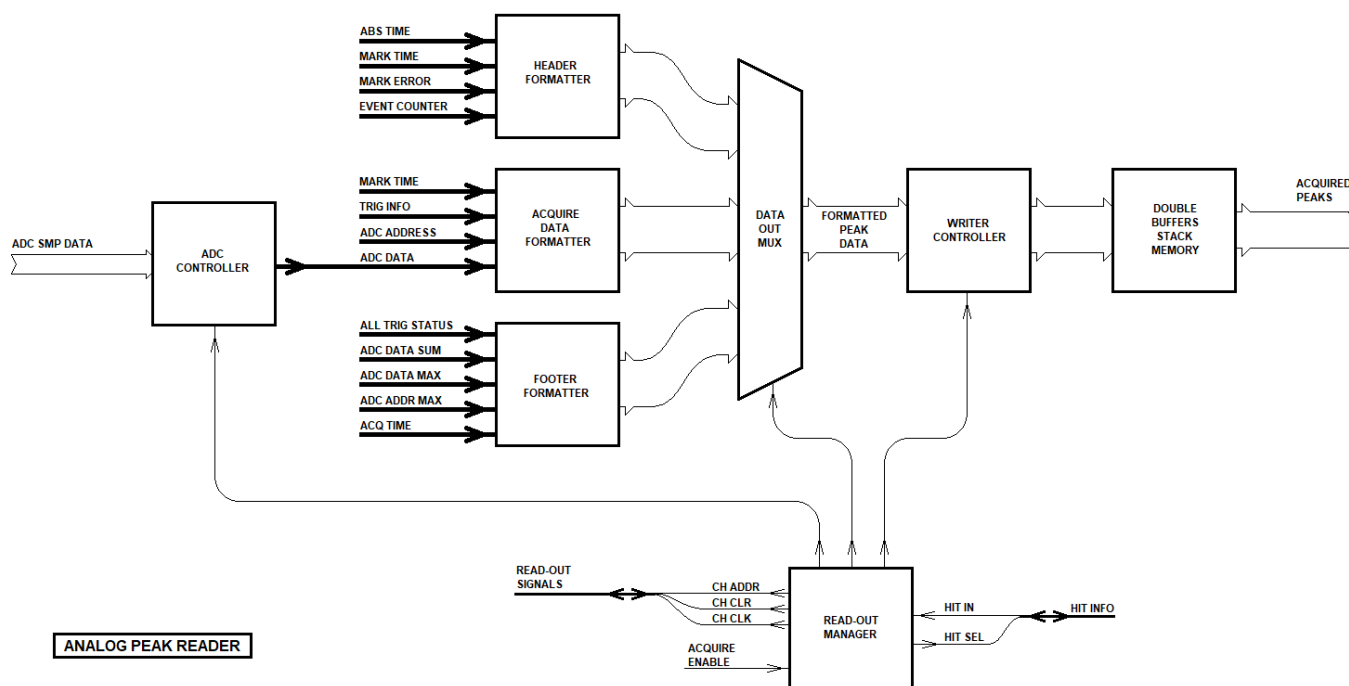


Fig 4: Modulo-IP lettore dei picchi analogici

Il modulo-IP "Peak-Reader" si occupa di leggere i valori analogici dei Peak-Detectors presenti nella circuitazione elettronica di Front-End. Questa operazione è composta da queste principali fasi sequenziali:

1. selezione del canale del Peak-Detector da leggere attraverso un mux
2. attesa del tempo di assestamento del segnale in uscita del mux
3. avvio conversione del segnale tramite un ADC
4. attesa della fine conversione dell'ADC
5. trasferimento del valore convertito in memoria
6. ritorno al punto 1 per il passaggio al canale successivo

Da questa lista si evince che le fasi 2, 4 e 5 sono costose in termini di tempo. Tempo che dipende dalle caratteristiche elettriche del mux, dalle performance dell'ADC utilizzato, e dalla velocità di scrittura della memoria di trasferimento più la quantità di dati da trasferire. Essendo tutte le fasi necessariamente sequenziali si desume che il tempo totale per il processamento di tutti i canali dei Peak-Detectors può essere notevole e determinare un punto critico nell'intento di ottenere un tempo morto di acquisizione più ridotto possibile.

Per ovviare a questo problema è stato progettato un controllore di lettura a pipeline capace di interlacciare le fasi "lente" dei vari canali da leggere tra di loro, e rendere così il tempo di processamento unico per tutte e tre le fasi. In pratica mentre si attende il tempo di assestamento del mux del canale n, si attende contemporaneamente il tempo di conversione dell'ADC del canale n-1, e anche contemporaneamente avviene il trasferimento del valore convertito in memoria del canale n-2. E così via.

Inoltre sempre al fine di minimizzare il tempo morto, questo controllore ha la possibilità di decidere se procedere alla lettura del canale selezionato, o passare al successivo, in base allo stato del segnale di Hit che proviene dalla circuitazione dedicata del modulo-IP Sampler, oppure se disponibile dall'elettronica di Front-End stessa. Lo stato di Hit non è altro che la segnalazione, nel canale selezionato, dell'avvenuta fuoriuscita del trigger, dovuto al superamento del segnale di ingresso ad una determinata condizione di soglia. Questa funzionalità è comunque bypassabile se si vogliono leggere tutti i canali indipendentemente dallo stato prima descritto.

Con questo controllore è anche possibile effettuare una pre-scansione rapida dei canali da leggere, monitorando solamente gli Hit, e successivamente eseguire un indirizzamento diretto dei canali da leggere memorizzati preventivamente tramite il pre-scan.

Altra funzionalità è la possibilità di leggere un eventuale canale complementare insieme al canale in stato di Hit. Utile quando più canali sono affacciati in coppia ad un array di rivelatori, ma solamente uno dei due supera la soglia di trigger che ne segnala lo stato di Hit.

Altre funzionalità sono disponibili in questo controllore, come di seguito riassunte:

- da 1 a 255 canali di lettura possibili
- lettura sequenziale interlacciata tramite Hit-info
- scansione preventiva dei canali da leggere tramite Hit-info, e successivo indirizzamento diretto
- scrittura in memoria con header iniziale
- scrittura in memoria con footer finale
- lettura del canale complementare (se presente)
- lettura di tutti i canali indipendentemente dallo stato dello Hit
- calcolo del valore analogico massimo dei canali (con indirizzo del canale corrispondente)
- calcolo della somma analogica di tutti i canali
- report dello stato degli Hit di tutti i canali
- report della durata temporale dell'intero processo di lettura
- accesso alla memoria di trasferimento in modalità programmabile

Come visibile in Fig.4, è possibile formattare i dati da scrivere nella memoria di trasferimento in maniera versatile, a seconda delle informazioni richieste in fase di progetto. È possibile formattare la scrittura con un HEADER di intestazione dei dati, il corpo dei dati analogici stessi, ed infine un FOOTER in chiusura. Per ogni sezione sono disponibili diversi parametri, utilizzabili in una qualsiasi combinazione richiesta.

Per quanto riguarda l'HEADER sono disponibili questi parametri:

- Timestamp in secondi a 32 bit
- Timestamp in decimi di microsecondi a 32 bit
- Errore di shift del Timestamp
- Valore del contatore sequenziale di evento

Mentre per il corpo dei dati analogici sono disponibili questi parametri:

- Timestamp in decimi di microsecondi a 32 bit
- Indirizzo del canale
- Hit info relativo
- Valore analogico letto dall'ADC

Infine per il FOOTER sono disponibili questi parametri:

- Status degli Hit di tutti i canali
- Somma di tutti i valori analogici letti
- Valore massimo di tutti i valori analogici letti
- Indirizzo del canale col valore massimo
- Tempo totale di acquisizione trascorso

Anche in questo caso la memoria di trasferimento è uno Stack di memoria a doppio buffer, e come per i moduli precedenti, questo Stack è dimensionato opportunamente per consentire l'accumulo di molteplici dati formattati dei picchi analogici acquisiti.

Note:

La procedura di formattazione dei dati da scrivere è descritta in codice VHDL su un package separato dal resto del corpo che definisce la struttura RTL e la macchina a stati del controllore. Questo in modo da semplificare tale procedura da parte di un qualsiasi utilizzatore interessato semplicemente a definire la formattazione dei dati dello strumento, senza dover necessariamente mettere le mani su tutto il codice VHDL dell'intero controllore.

Questo è utile per il riutilizzo di questo controllore su progetti diversi da quello in oggetto.

Modulo Trigger Manager

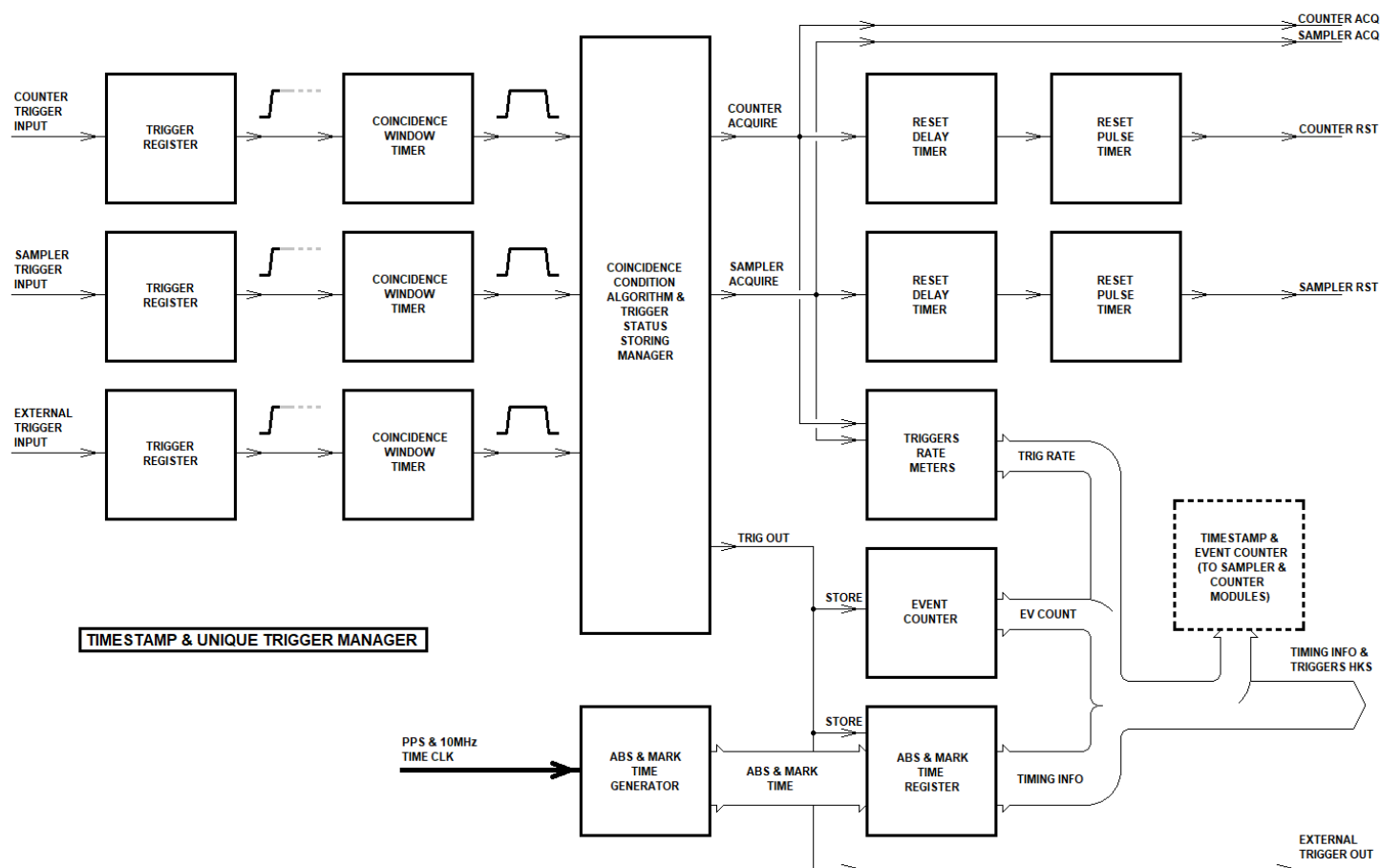


Fig 5: Modulo-IP di gestione dei trigger

Il modulo-IP "Trigger" si occupa di gestire le acquisizioni nei moduli Counters e Sampler.

Come visibile da Fig.5, da questi due moduli proviene il segnale di trigger di acquisizione prodotto dai relativi algoritmi circuitali presenti in ognuno. Questi segnali di trigger sono inizialmente processati per individuarne il fronte di attacco e successivamente fatti entrare in un temporizzatore al fine di creare una finestra di coincidenza programmabile.

È disponibile un segnale di trigger esterno, proveniente da altre eventuali strumentazioni scientifiche, che è trattato allo stesso modo dei due precedenti, al fine di contribuire alla gestione delle acquisizioni.

Le finestre di coincidenza sono valutate da un algoritmo programmabile, superato il quale il modulo produce il segnale di uscita Trig-Out. Questo segnale memorizza il valore del Timestamp a 64 bit e il valore del contatore sequenziale su appositi registri per la marcatura temporale degli eventi. Inoltre Trig-Out fuoriesce come trigger esterno ad uso di altre eventuali strumentazioni scientifiche.

In contemporanea alla produzione del Trig-Out, il modulo gestisce i segnali ACQUIRE di inizio e fine acquisizione verso i moduli Counters e Sampler. Infine produce il relativo segnale di reset programmabile.

Note:

Questo modulo-IP si occupa anche di una funzione indispensabile, e cioè la sincronizzazione, con un clock di riferimento, dei segnali PPS e 10MHz esterni, provenienti da un eventuale modulo CSAC o RTC esterno.

Modulo I/O Communication

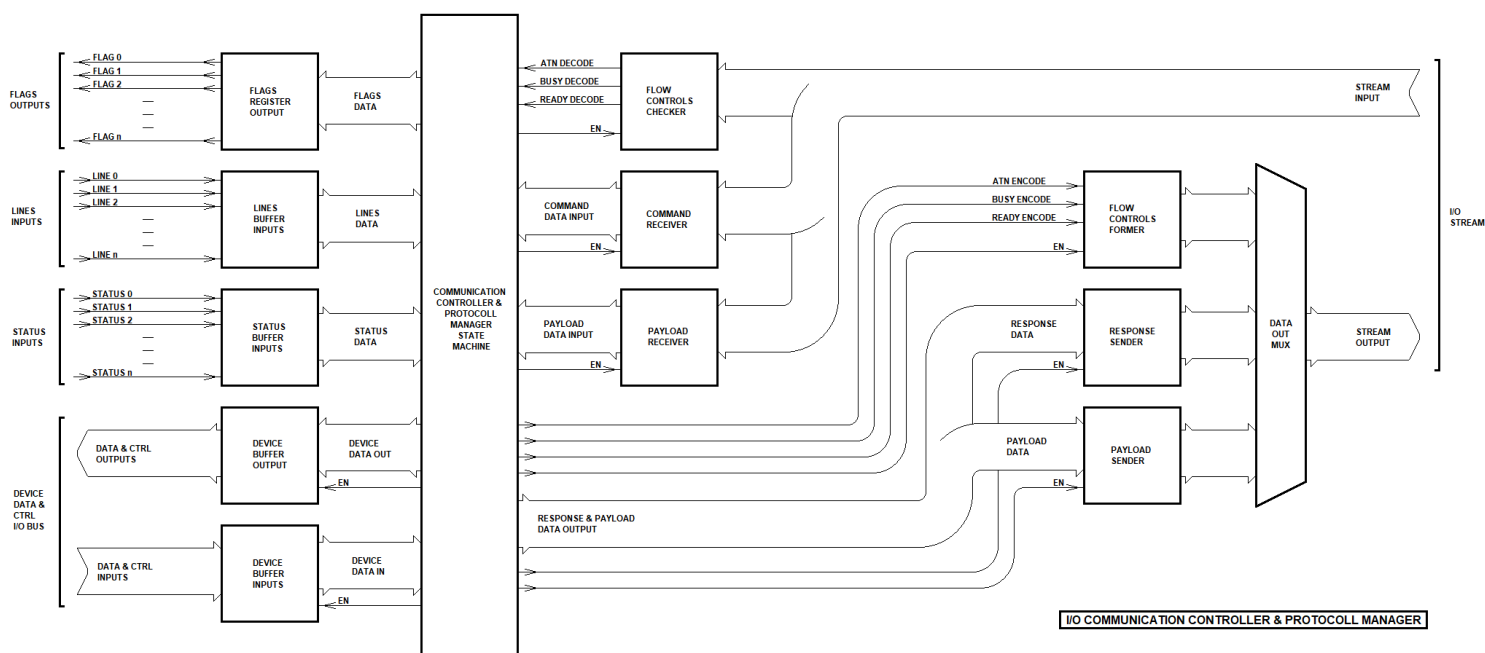


Fig 6: Modulo-IP di gestione della comunicazione I/O

Il modulo-IP "I/O Communication" si occupa di gestire la comunicazione interna tra i moduli-IP del progetto, nonché di gestire la comunicazione con un computer Host esterno, attraverso il modulo-IP "I/O Interface".

Qualunque operazione di I/O, da e verso i vari moduli-IP ad esso connesso, è avviata da una relativa transazione di I/O creata dal computer Host. In poche parole è il computer che gestisce interamente lo strumento attraverso la scrittura e lettura di Stream di dati opportunamente formattati.

Come visibile in Fig.6, lo Stream in input dei dati provenienti dal computer Host afferra contemporaneamente a tre stadi di controllo. Il primo in alto serve al riconoscimento e decodifica di determinate sequenze di byte nello Stream, atte a realizzare il controllo del flusso di I/O. Il secondo serve alla ricezione della stringa del comando da eseguire, e il terzo serve alla ricezione della stringa di payload dei dati. Questi tre stadi di controllo afferiscono ad un controllore di comunicazione, che a sua volta si connette ad altri tre stadi di uscita per la composizione dello Stream di uscita verso il computer Host. Il primo codifica la sequenza di byte per il controllo di flusso, il secondo l'invio della stringa in risposta al comando, il terzo l'invio della stringa di payload richiesto.

Il controllore di comunicazione è essenzialmente una macchina a stati, che tramite i segnali di controllo di flusso, regola l'avvicendamento delle transazioni tra il computer e lo strumento scientifico in uso. Ogni transazione sottostà ad un protocollo proprietario sviluppato all'occorrenza, che prevede tre tipi di transazioni: Una di tipo TASK senza trasferimento di dati di PAYLOAD, una di tipo WRITE con trasferimento del PAYLOAD inviato dall'Host e ricevuto dal modulo-IP, ed una di tipo READ con trasferimento del PAYLOAD ricevuto dall'Host e inviato dal modulo.

Ogni transazione è avviata dall'Host e prevede queste fasi:

- Invio di ATN
- Ricezione ATN_RET
- Invio del COMMAND
- Attesa del processamento del COMMAND
- Ricezione del RESPONSE
- Attesa di preparazione del PAYLOAD
- Invio o ricezione del PAYLOAD, se previsto dal tipo di transazione
- Attesa di completamento transazione

L'ATN è un segnale di attenzione, inviato dall'Host al controllore, allo scopo di risvegliare lo strumento e sincronizzarlo con lo Stream della transazione in arrivo. Il controllore risponde a sua volta con un ATN_RET ad indicare che è pronto a ricevere lo Stream. Il COMMAND è composto da una stringa predefinita di byte e rappresenta

il comando della transazione. Il RESPONSE è anch'esso composto da una stringa di byte che rappresentano la risposta al comando ricevuto. Ed infine il PAYLOAD è il blocco dati vero e proprio da trasferire.

Tra le varie fasi è necessario inserire delle pause allo scopo di dare il tempo sia al controllore che all'Host di processare il comando e la risposta, nonché di prepararsi all'invio o ricezione dei dati di PAYLOAD. Durante questo intervallo Host e controllore possono inviarsi vicendevolmente il byte di controllo BUSY. Al termine di questa attesa bisogna inviare il byte READY, ad indicare la ripresa del flusso della transazione.

La decodifica della stringa del COMMAND, come pure la codifica del RESPONSE, sono completamente definibili in fase di progetto in base alle esigenze specifiche del software di controllo, sviluppato dall'utente utilizzatore dello strumento. I parametri a disposizione sono molteplici e permettono di realizzare in maniera flessibile e efficiente la lista di comandi e risposte necessarie per la gestione completa dello strumento.

Il controllore consente una velocità di transazione con l'Host di oltre 100MB/s.

Per quanto riguarda la comunicazione interna tra i moduli-IP del progetto, il controllore dispone di un bus proprietario full-duplex ad alta velocità che permette di interagire con ogni modulo del progetto nel minor tempo possibile. Necessario quando i moduli da gestire sono diverse unità, al fine di ridurre al minimo il tempo morto di acquisizione totale.

In più il controllore mette a disposizione molteplici Flags logici utilizzabili per configurare in maniera semplice e rapida le varie modalità di funzionamento dei moduli-IP del progetto. È pure possibile eseguire la lettura dello stato di varie linee statiche che possono essere utilizzate per monitorare le condizioni di vari segnali in ingresso all'FPGA. Come pure eseguire la lettura dello stato di vari status logici interni che sono utilizzati specificamente dal controllore per monitorare, e quindi agire di conseguenza, ad una condizione di busy o error dei vari moduli-IP da gestire. È possibile realizzare tutte queste operazioni attraverso semplici transazioni di tipo TASK senza trasferimento di payload, e quindi in maniera molto rapida dal punto di vista del tempo di comunicazione col computer Host.

Riepilogando è stato realizzato un controllore di comunicazione, tra i moduli-IP interconnessi del progetto e il computer Host di gestione, molto versatile ed efficiente, che è possibile sfruttare per molteplici progetti anche diversi da quello in oggetto.

Note:

La procedura di decodifica e codifica delle stringhe di byte del COMMAND e del RESPONSE è descritta in codice VHDL su un package separato dal resto del corpo che definisce la struttura RTL e la macchina a stati del controllore. Questo in modo da semplificare tale procedura da parte di un qualsiasi utilizzatore interessato semplicemente a definire la lista di comandi e risposte dello strumento, senza dover necessariamente mettere le mani su tutto il codice VHDL dell'intero controllore.

Questo è utile per il riutilizzo di questo controllore su progetti diversi da quello in oggetto.

La descrizione completa del protocollo di comunicazione, del bus proprietario per i moduli-IP interconnessi, e dei parametri di codifica e decodifica dei comandi e risposte, è demandata ad altro documento.

Modulo I/O Interface

Il modulo-IP "I/O Interface" si occupa dell'interfacciamento tra il modulo-IP "I/O Communication", che ricordiamo essere adibito specificamente alla gestione del protocollo di comunicazione, ed un computer Host.

Sono stati progettati diversi controllori di interfacciamento per soddisfare plurime esigenze di collegamento. Questi controllori, sviluppati nel corso degli anni dal Laboratorio di Elettronica dell'IASF-PA per lo sviluppo di molteplici esperimenti scientifici, sono in grado di interfacciarsi direttamente o attraverso dei comuni chip commerciali, a bus di comunicazione standard con diverse velocità di trasmissione, tra cui:

- UART, fino a 10MBaud
- SPI, fino a 100Mbit/s
- USB2, attraverso il chip FT232H di FTDI in modalità 245-Async, fino a 10MByte/s, e 245-Sync, fino a 40MByte/s
- eSATA + USB2, attraverso i chip JM20330 e JM20338, fino a 100MByte/s (sistema visto dal S.O. come una unità HDD standard, senza necessità di driver)
- USB3, attraverso i chip FT60x di FTDI, fino a 400MByte/s

In più è stato sviluppato un modulo-IP Master & Slave proprietario in grado di realizzare una connessione Point-To-Point seriale ad alta velocità ed efficienza, pensato per effettuare collegamenti plurimi di diverse unità di strumenti scientifici ad una unità madre di controllo. Il controllore ha le seguenti caratteristiche:

- Doppio buffer di I/O a bassa latenza
- Trasmissione a pacchetti con parametri riconfigurabili
- Controllo del CRC
- Ritrasmissione automatica in presenza di errori di trasmissione
- Handshake con temporizzazione adattativa

Tramite questo controllore è possibile gestire diversi tipi di chip SERDES gigabit ad alta velocità, come i chip Texas della famiglia TLK, in grado di offrire una velocità di trasmissione di 2.5Gbit/s e oltre. Ma è anche possibile gestire i SERDES embedded integrati di alcune famiglie di FPGA, come anche il CYCLONE V-GX, in grado di spingersi oltre i 6Gbit/s. Normalmente questi collegamenti sono realizzati solamente con due linee di trasmissione TX e RX differenziali a bassa emissione EMI, tramite semplici doppini in rame intrecciati e accoppiati in AC, permettendo così anche un utile isolamento galvanico tra l'unità madre e le varie unità di strumenti collegati. Con questo tipo di trasmissione GIGABIT a 2 linee è altresì possibile sfruttare diversi moduli SFP ottici commerciali, tipo AVAGO AFBR-57R5APZ o simili, e realizzare così un collegamento in fibra ottica, necessario quando la lunghezza dei collegamenti è notevole, volendo mantenere contemporaneamente una elevata velocità di trasmissione.

Infine è stato realizzato un ulteriore modulo-IP di interfacciamento in grado di emulare il comportamento dei chip SERDES della famiglia TLK della Texas, in modo da essere abbinato al controllore di SERDES descritto prima. Con questo controllore è possibile realizzare sempre un collegamento seriale ad alta velocità tra unità madre e varie unità di strumenti, limitando però la velocità massima a 500Mbit/s, e attraverso 4 linee di trasmissione invece che 2, ma sempre differenziali e in AC.

Questa ulteriore possibilità di collegamento può essere utile come compromesso tra velocità e semplicità, quando le esigenze di velocità di trasmissione non sono molto spinte, non si vogliono aggiungere complicazioni tecniche e di spazio nell'utilizzo di chip SERDES esterni, e si vuole rimanere su famiglie di FPGA a basso costo senza SERDES embedded integrati.

Moduli di Servizio

In questo progetto sono stati altresì implementati diversi moduli-IP dedicati alla gestione di circuiterie elettroniche esterne all'FPGA, come:

- La comunicazione con sensori di varia natura presenti nello strumento
- La gestione di ADC e DAC per il monitor di tensioni e correnti di funzionamento dello strumento, e per il controllo di alte tensioni e bias esterni
- La comunicazione con un eventuale ASIC per gestire la configurazione dei suoi registri interni
- La generazione di impulsi di test

Allo scopo sono stati realizzati diversi controllori proprietari per l'interfacciamento con protocolli di trasmissione standard come SPI, I2C e UART. Questi controllori, realizzati senza l'ausilio di processori virtuali o embedded, sono molto flessibili nel loro utilizzo e si prestano bene per un loro riutilizzo in svariati progetti.

In più è stato realizzato un controllore per la generazione di impulsi, programmabile in larghezza ampiezza e periodo, utile per la verifica funzionale del Front-End.

Conclusioni

Come già anticipato, questo progetto è stato completamente sviluppato, testato ed utilizzato con successo dallo staff tecnico del Laboratorio di Elettronica dell'Istituto IASF-PA dell'INAF, su piattaforma Altera di Intel, e con ambiente di sviluppo Quartus. Per i test sono stati utilizzati rivelatori sia di tipo SIPM che MaPMT, connessi ai relativi Front-End sia in forma di ASIC, come il RADIOROC della Weeroc, sia con elettronica discreta custom ad alta velocità.

Sebbene tutti i moduli-IP siano stati descritti in linguaggio VHDL standard, quindi facilmente compilabili in altri tipi di piattaforme, l'eventuale porting di codesti moduli deve essere studiato con attenzione, in quanto il loro sviluppo è stato modellato intorno alle risorse disponibili specificamente negli FPGA Altera. Inoltre, l'aver sfruttato un ambiente grafico per la realizzazione del Top-Level Entity e per la connessione dei vari moduli-IP tra di loro, operazione che semplifica e velocizza lo sviluppo, non permette un porting immediato del progetto. In più, la definizione delle constraint temporali, obbligatoriamente necessarie per un corretto funzionamento dei moduli, è intrinsecamente legata alla piattaforma di sviluppo, e quindi non riutilizzabile direttamente in altri contesti.

Rimanendo invece su piattaforma Altera, tutti i moduli-IP del progetto sono facilmente e pienamente riutilizzati, singolarmente o interamente, su svariati tipi di progetti anche diversi da quello in oggetto.

La documentazione ausiliaria di cui si è fatto cenno nella descrizione è in corso di sviluppo e completamento. Altra documentazione è ampiamente disponibile direttamente sul sito Altera.com.