

Università degli Studi di Ferrara  
Facoltà di Scienze Matematiche, Fisiche e Naturali  
Corso di Laurea in Informatica



Realizzazione di un  
Sistema di Acquisizione Dati  
per un Rivelatore di Muoni

*Primo Relatore:*

Dott. MIRCO ANDREOTTI

*Secondo Relatore:*

Dott. GIANLUIGI CIBINETTO

*Laureando*

MIRCO TAGLIANI

Anno Accademico 2005-2006



*Ai genitori, che mi hanno così sostenuto  
e alla musica, che mi ha dato tali sensazioni..*



# Indice

<b>Introduzione</b>	<b>vii</b>
<b>1 Apparato sperimentale</b>	<b>1</b>
1.1 I rivelatori di muoni (LST)	1
1.2 Ricostruzione tridimensionale	3
1.3 Lettura dei segnali	4
<b>2 Scheda di acquisizione dati</b>	<b>7</b>
2.1 Caratteristiche fisiche	7
2.1.1 Caratteristiche principali	7
2.1.2 Disposizione e utilizzo delle linee	9
2.2 Software di controllo della scheda	10
2.3 Funzionamento in lettura	12
2.3.1 Lettura a singola linea	12
2.3.2 Lettura a linee parallele	13
2.4 Funzionamento in Scrittura	18
<b>3 Applicazione di acquisizione dati</b>	<b>21</b>
3.1 Requisiti dell'applicazione di acquisizione	21
3.2 Applicazione di lettura	22
3.3 Simulazione	26
3.3.1 La scheda hardware di simulazione della FEC	27
3.3.2 Applicazione di scrittura sulla scheda di simulazione	29
3.4 Realizzazione dell'applicazione per la sola lettura	31
3.4.1 Modifiche alla scheda di simulazione della FEC	32
3.5 Monitoring di acquisizione	33
3.6 Interfaccia grafica dell'applicazione	34
<b>4 Prestazioni del sistema di acquisizione</b>	<b>39</b>
4.1 Misura della frequenza di acquisizione	39

---

4.1.1	Test con simulazione . . . . .	39
4.1.2	Test senza simulazione . . . . .	40
4.2	Errori di acquisizione . . . . .	42
4.2.1	Test con simulazione . . . . .	42
4.2.2	Test senza simulazione . . . . .	44
<b>5</b>	<b>Conclusioni</b>	<b>47</b>
<b>A</b>	<b>Dettagli della scheda di simulazione</b>	<b>49</b>
	<b>Bibliografia</b>	<b>51</b>

# Introduzione

In questa tesi presentiamo la realizzazione del sistema di acquisizione dati per un rivelatore di muoni. Tutto il lavoro di tesi è stato svolto nel laboratorio di criogenia ed LST del Dipartimento di Fisica dell'Università degli Studi di Ferrara. In questo laboratorio si svolgono attività di ricerca e sviluppo nel campo della rivelazione delle particelle, in particolare nella rivelazione dei muoni.

Le attività di ricerca sono rivolte allo sviluppo di sempre più efficienti sistemi di rivelazioni di tali particelle in vista di un futuro utilizzo negli esperimenti internazionali di fisica delle particelle di nuova generazione.

In generale quando si utilizzano rivelatori di particelle si devono trattare segnali elettrici molto veloci, quindi inevitabilmente si devono utilizzare sistemi di elettronica in grado di manipolare tali segnali con frequenze molto alte. In conseguenza di questo fatto l'acquisizione, l'elaborazione e la memorizzazione dei dati corrispondenti deve essere affidata ad un calcolatore in grado di gestire con sufficiente rapidità tutte queste operazioni. Il calcolatore dovrà essere interfacciato con il sistema di rivelazione attraverso opportuni sistemi di interfaccia e dovrà quindi gestire l'acquisizione dei dati utilizzando un'applicazione software adeguatamente progettata.

Nel progetto trattato in questa tesi il sistema di rivelazione dei muoni è affidato ad una configurazione a telescopio di rivelatori chiamati tubi di Iarocci o LST (Limited Streamer Tube). L'apparato di rivelazione fornisce segnali elettrici corrispondenti ai punti di impatto delle particelle. Il progetto prevede l'utilizzo di una scheda di elettronica di prima elaborazione (FEC) per la manipolazione dei segnali elettrici provenienti dal sistema di rivelazione.

Per quanto riguarda l'interfaccia con il calcolatore si è scelto di utilizzare una scheda della National Instruments (DIO-6533) con ingressi/uscite digitali programmabili ad alta velocità di trasferimento. Il software utilizzato per la realizzazione dell'applicazione di acquisizione dati è LabView della National Instruments, il quale è provvisto dei drivers e di tutte le librerie per il controllo della scheda di interfaccia.

La prima fase del lavoro di tesi si è concentrata sullo studio delle principali funzioni della scheda di acquisizione al fine di individuarne le modalità di funzionamento che meglio si adattano al nostro scopo. Quindi si è proceduto con la realizzazione del sistema di acquisizione dati.

Durante la fase di realizzazione di questa tesi non è stato possibile utilizzare la scheda FEC in questione, in quando la sua preparazione deve ancora essere terminata. Al fine di collaudare il sistema di acquisizione si è quindi provveduto alla realizzazione della simulazione del sistema di rivelazione e alla costruzione di un circuito che riproducesse con buona fedeltà le principali caratteristiche della FEC. Il circuito di simulazione della FEC è stato realizzato con semplici circuiti integrati, quali AND, OR, FLIP-FLOP e registri a scorrimento. L'applicazione del sistema di acquisizione è stata sviluppata sul sistema operativo Linux Fedora Core 3.

Nel Capitolo 1 riportiamo una descrizione del sistema di rivelazione dei muoni e delle caratteristiche della scheda di prima elaborazione (FEC).

Nel Capitolo 2 descriviamo in dettaglio le principali funzioni della scheda di interfaccia, in particolare trattiamo più dettagliatamente quelle utilizzate per la realizzazione del sistema di acquisizione.

Nel Capitolo 3 descriviamo la realizzazione dell'applicazione per l'acquisizione dei dati e distingueremo le due versioni studiate: una con la simulazione e una, più fedele a quella che sarà la configurazione finale, senza la simulazione.

Nel Capitolo 4 riportiamo i risultati ottenuti da test di collaudo svolti sulla scheda al fine di valutarne le prestazioni, quali frequenza massima di acquisizione e frazione di errori nella lettura dei dati.

Nel Capitolo 5 concludiamo commentando i risultati ottenuti e riportando i progetti per un futuro miglioramento del sistema.

Riportiamo inoltre un'appendice (A) nella quale descriviamo altri dettagli del circuito di simulazione della FEC.

Notazioni usate nella tesi: I/O (input/output) ingresso/uscita.



# Capitolo 1

## Apparato sperimentale

In questo capitolo si descriverà l'apparato sperimentale per il quale si è realizzato il sistema di acquisizione dati. Attraverso questo apparato, utilizzato nel laboratorio di criogenia e LST del dipartimento di Fisica, si svolgono attività di ricerca e didattica sulla rivelazione dei muoni <sup>1</sup> [1].

In particolare si distinguono: il sistema di rivelatori di muoni, la configurazione per la ricostruzione tridimensionale del passaggio di particelle e l'elettronica di prima elaborazione (front end) dei segnali.

### 1.1 I rivelatori di muoni (LST)

Per rivelare muoni o più in generale particelle cariche, in questo laboratorio si utilizzano i rivelatori a gas LST (Limited Streamer Tube), conosciuti anche come tubi di Iarocci [2].

In generale questi rivelatori sono composti da un certo numero di celle (Tipicamente 8) a sezione quadrata o rettangolare di dimensioni che possono variare da  $10 \times 10 \text{ mm}^2$  a  $10 \times 10 \text{ cm}^2$ . Le pareti interne delle celle (3 o 4 a seconda del tipo) sono rese conduttrici con l'apporto di uno strato di grafite. Al centro di ogni cella è posizionato un filo conduttore, tipicamente d'argento, che percorre tutta la lunghezza della cella. Il tutto è contenuto in un involucro di PVC che permette la circolazione di una miscela di gas all'interno delle celle. All'estremità i tubi sono sigillati con tappi provvisti di connettori per il gas e per l'alta tensione di al-

---

<sup>1</sup>I muoni sono particelle cariche della famiglia dei leptoni. I muoni sono molto simili agli elettroni, anch'essi leptoni, in particolare si distinguono da questi per la massa che è circa 200 volte più grande. I muoni studiati in questo laboratorio sono prodotti nell'alta atmosfera terrestre a seguito di interazioni nucleari dei raggi cosmici con i nuclei dei gas dell'atmosfera stessa.

imentazione. I tubi utilizzati in questo laboratorio hanno 7 o 8 celle di dimensioni  $15 \times 17 \text{ mm}^2$  e sono lunghi 350 cm. Il filo ha funzione di anodo e viene mantenuto ad una tensione dell'ordine di 5600 V, mentre la grafite funge da catodo ed è collegata a massa. Per il funzionamento voluto si fa circolare nelle celle una miscela ternaria di  $\text{Ar}/\text{C}_4\text{H}_{10}/\text{CO}_2$  (3/9/88)% [2].

Al passaggio di una particella carica all'interno di una cella, si verifica la ionizzazione del gas. Per effetto del campo elettrico tra anodo e catodo si ha una migrazione di ioni negativi ( $e^-$ ) verso l'anodo e ioni positivi verso il catodo. Durante questa migrazione grazie, alla particolare scelta della miscela di gas, si verificano ulteriori ionizzazioni secondarie, con conseguente moltiplicazione della carica raccolta su anodo e catodo. La miscela di gas è inoltre scelta affinché la moltiplicazione degli ioni avvenga in una zona ristretta in corrispondenza del passaggio della particella. Gli impulsi elettrici dovuti agli ioni permettono di ricavare il punto di impatto della particella carica. In Fig.1.1 è mostrato uno schema delle celle di un tubo LST (immagine in alto) e una vista esterna di un tubo singolo (immagine in basso).

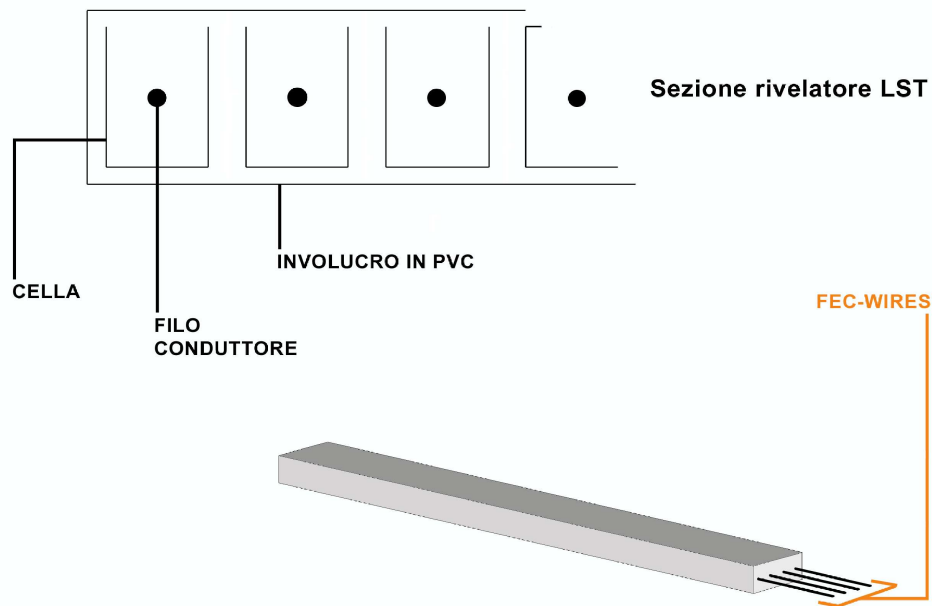


Figura 1.1: Sezione di un LST, in alto, prospettiva di un LST, in basso.

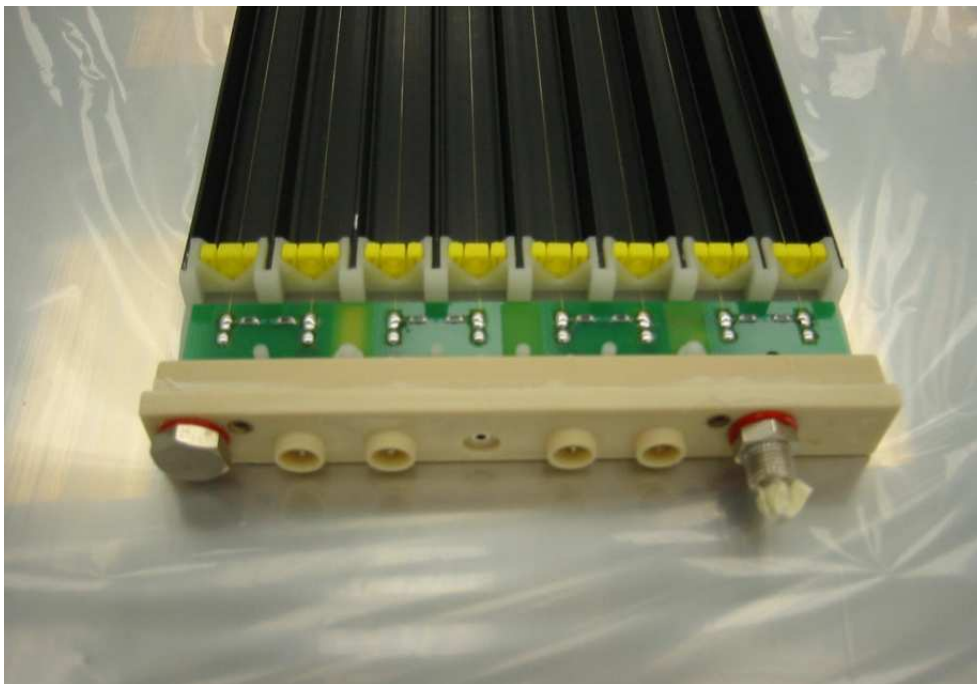


Figura 1.2: Particolare di un rivelatore LST.

## 1.2 Ricostruzione tridimensionale

In ogni cella di un LST il movimento e la raccolta di carica producono sui fili degli impulsi elettrici che possono essere usati come indicazione del punto di impatto della particella in moto. Un secondo metodo di rivelazione del passaggio di una particella è quello di osservare l'impulso elettrico indotto su strisce di rame poste all'esterno del tubo e in contatto con esso. Queste strisce, solitamente chiamate *strip*, vedi Fig.1.3 possono essere parallele o ortogonali alle celle.

Secondo il sistema di riferimento cartesiano indicato in Fig.1.3, i fili forniscono l'informazione sulla posizione  $x$  del punto di impatto della particella, mentre le strip poste ortogonalmente forniscono la posizione  $y$ . Configurando un sistema di rivelamento a più strati tubi/strips possiamo rilevare il punto di impatto sul primo strato  $(x_1, y_1)$ , sul secondo  $(x_2, y_2)$  e così via fino allo strato  $n$ -esimo  $(x_n, y_n)$ , potendo così ricostruire una traccia tridimensionale della particella in esame. Si può vedere un esempio di sistema a più strati in Fig.1.3.

In sistemi con più strati di tubi alcuni piani di strip possono essere posti, come accennato precedentemente, parallelamente alle celle degli LST per aumentare l'efficienza di rivelazione. In generale ogni singolo strato di strip è isolato dagli altri attraverso un foglio di rame collegato a massa. Tale isolamento è necessario

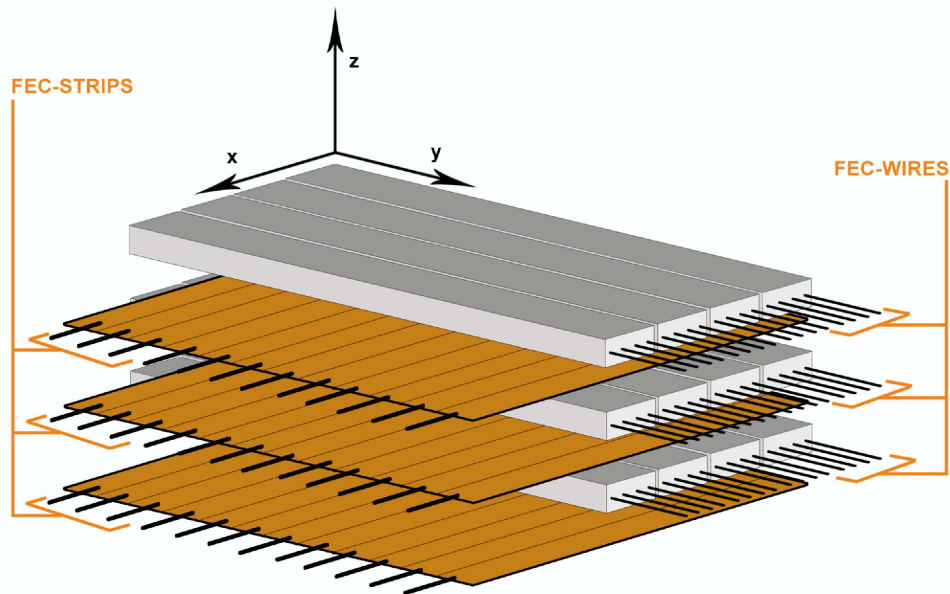


Figura 1.3: Sistema di LST in configurazione per la ricostruzione tridimensionale.

affinchè non ci sia influenza tra gli strati.

### 1.3 Lettura dei segnali

I segnali provenienti dai tubi e dalle strips sono impulsi elettrici, cioè segnali analogici, che devono essere digitalizzati, in quanto lo scopo della ricostruzione è quello di individuare solo se una particella ha attraversato un certo punto. A tale scopo si utilizza una scheda di prima elaborazione dei segnali (FEC, *Front End Card*) [3] la quale esegue le seguenti funzioni:

- amplificazione dei segnali
- discriminazione dei segnali
- memorizzazione dei dati

La FEC è composta da una scheda madre principale nella quale sono alloggiati 4 schede secondarie ognuna delle quali dispone di 16 ingressi analogici. Gli ingressi vengono collegati direttamente ai tubi e alle strip tramite cavi conduttori. In ogni scheda secondaria gli impulsi in ingresso attraversano un primo modulo

che li amplifica, segue quindi un secondo modulo che discrimina i segnali sopra una certa soglia. Una delle principali caratteristiche della FEC è la possibilità di modificare la soglia di discriminazione.

Ogni segnale discriminato rappresenta lo stato della cella da cui deriva. Lo stato viene memorizzato come un bit 1 se si è rivelato un impulso (situazione che corrisponde a una cella attraversata da una particella) o un bit 0 se non si è rivelato nulla. Vengono così memorizzate 4 parole di 16 bit (una per ogni scheda secondaria) in 1 registro a scorrimento veloce diviso in 4 linee. Il registro si tro-

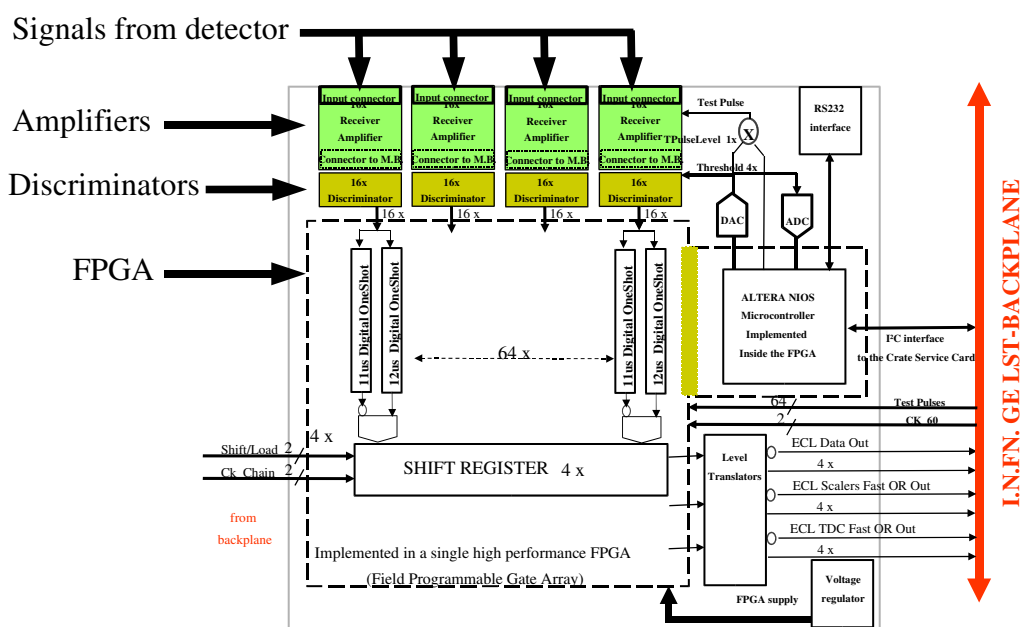


Figura 1.4: Scheda madre della FEC.

va all'interno di un componente della scheda madre chiamato FPGA (Filed Programmable Gate Array) il quale gestisce tutte le funzioni richieste per la memorizzazione dei dati, gestione dei tempi di latenza e ritardo nella trasmissione delle informazioni binarie.

La FEC presenta 4 uscite le quali sono le uscite seriali delle linee del registro a scorrimento interno al FPGA. Lo svuotamento seriale del registro è regolato da un clock inviato dall'esterno per mezzo di un ingresso dedicato della FEC. La frequenza del clock si può impostare a seconda delle esigenze. La Fec presenta

inoltre altre 4 uscite ognuna delle quali è l'OR logico dei 16 bit memorizzati nella corrispondente linea del registro. È disponibile un'ulteriore ingresso chiamato *bit enable* il quale ogni volta che rileva un fronte di salita o di discesa di un segnale in ingresso, rispettivamente congela e scongela i dati binari nel registro.

# Capitolo 2

## Scheda di acquisizione dati

In questo capitolo si descriverà la scheda di acquisizione dati utilizzata come interfaccia tra la FEC e il calcolatore (PC) sul quale è installato il software di controllo del sistema di acquisizione. La scheda in questione è la DIO-6533 [4], una scheda PCI della National-Instrument. Verranno in particolare descritte la scheda utilizzata e le funzioni di lettura e scrittura utilizzando il software **LabView**.

### 2.1 Caratteristiche fisiche

L'interfaccia di acquisizione utilizzata tra la FEC e il PC è la scheda PCI DIO-6533 della National-Instruments. Questo device mette a disposizione funzioni di I/O digitale ad alta velocità, generazioni di waveform digitali e funzioni di handshaking relativamente flessibili.

#### 2.1.1 Caratteristiche principali

La scheda DIO-6533 dispone di 40 linee di I/O digitale, le quali possono essere suddivise in 2 gruppi principali:

1. linee di I/O utilizzate per la lettura/scrittura di dati digitali (32 linee)
2. linee di handshaking<sup>1</sup> (8 linee).

---

<sup>1</sup>Nel caso citato per handshaking si intende il processo che garantisce la sincronizzazione ed il controllo temporale delle operazioni di I/O.

Le linee appartenenti al primo gruppo vengono utilizzate per la lettura e scrittura di dati digitali da e sulla scheda, esse possono essere usate sia singolarmente che in parallelo. Quando usate singolarmente ogni lettura/scrittura su una linea avviene indipendentemente dalle altre, lettura/scrittura su linee differenti non hanno quindi nessun legame temporale. Al contrario, quando le linee vengono utilizzate in parallelo si ha una dipendenza temporale tra lettura/scrittura su linee differenti, cioè ogni lettura/scrittura viene eseguita simultaneamente su tutte le linee utilizzate. Ogni linea viene interpretata come un bit di una parola binaria, che può essere scritta o letta dalla scheda, la posizione e il peso del bit sono determinati dalla posizione della linea. Attraverso un utilizzo di questo tipo si può raggiungere un ingresso (o un'uscita) dati parallela di 32 bit.

Le linee appartenenti al secondo gruppo sono linee per l'handshaking, utilizzabili per l'importazione o l'esportazione di clock e per l'importazione di segnali per il controllo (trigger) del software che gestisce le operazioni di lettura/scrittura. Queste linee possono essere usate come controlli solamente quando le linee di I/O del primo gruppo vengono utilizzate in parallelo. Al contrario, quando le linee I/O del primo gruppo vengono utilizzate in singola linea, non è ammesso l'utilizzo delle linee di handshaking, le quali devengono linee di I/O normali (come quelle del primo gruppo).

La scheda DIO-6533 è corredata da una morsettiera esterna, la quale ha funzione di interfaccia tra le linee della scheda e l'esterno. Scheda e morsettiera sono collegate tramite un cavo.

La scheda dispone di un clock hardware interno, il quale si può raggiungere una frequenza massima di 20MHz e può essere esportato su una delle linee dedicate all'handshaking. Quando le linee vengono utilizzate in parallelo, è possibile inviare/ricevere successioni di parole binarie sincronizzando la velocità di trasferimento attraverso questo clock. La frequenza del clock si imposta con il software di controllo della scheda.

Sulla DIO-6533 sono inoltre disponibili 20 morsetti collegati a massa e un vcc di alimentazione a 5v.



## 2.1.2 Disposizione e utilizzo delle linee

Le 32 linee dedicate all'input/output, appartenenti al primo gruppo, sono divise in 4 blocchi di 8, chiamati *porte*. Ad ogni porta è assegnata una lettera, dalla A alla D e le linee sono identificate dalla lettera della porta a cui appartengono e da un numero che va da 0 a 7. Ad esempio A0, A5, B4. Tutte le 16 linee che vanno da A0 a B7 fanno parte del *gruppo 1* mentre le 16 da C0 a D7 fanno parte del *gruppo 2*.



Figura 2.1: Disposizione dei canali di I/O e di controllo.

Le 8 linee di handshaking sono 4 per ogni gruppo:

- ACK<sub>x</sub> (STARTTRIG<sub>x</sub>)
- STOPTRIG<sub>x</sub>
- PCLK<sub>x</sub>
- REQ<sub>x</sub>

dove x può essere 1 o 2 a seconda del gruppo.

Gli ACK1 e ACK2 sono due uscite digitali che assumono lo stato 1 per il periodo in cui il gruppo di porte a cui sono associati sta eseguendo un'operazione di I/O.

I CLK1 e CLK2 sono linee di I/O dedicate all'esportazione del clock generato dall>DIO-6533 o all'importazione di un clock esterno per la sincronizzazione delle funzioni di lettura/scrittura di successioni di parole binarie nella modalità della scheda a linee parallele .

I REQ1 e REQ2 possono essere utilizzati come ingressi di un segnale di trigger esterno, per la sincronizzazione delle funzioni di lettura/scrittura utilizzate del gruppo di linee corrispondenti.

Come sopra accennato, se la scheda viene utilizzata con le linee di I/O in singola linea anche queste otto linee possono divenire linee di I/O comuni.

Tutte le funzioni della scheda sopra descritte sono accessibili attraverso l'ambiente di sviluppo software fornito da LabView anch'esso della National-Instruments, e in particolare attraverso le librerie di acquisizione dati NI-DAQmx.

## 2.2 Software di controllo della scheda

Per il controllo della scheda si utilizza il software LabView. Per la gestione di sistemi di acquisizione LabView mette a disposizione le librerie NI-DAQmx. Tali librerie sono utilizzabili per il controllo di varie schede della National Instruments e possono essere utilizzate anche per la scheda in esame. In particolare dallo studio delle applicazioni della scheda risulta che alcune particolari librerie sono comuni ad ogni determinata operazione richiesta. Le librerie comuni che si utilizzano in ogni applicazione sono proprio quelle che determinano quale operazione deve svolgere un determinato canale e in che modalità deve operare.

Per chiarire l'utilizzo di queste librerie comuni, forniremo qui una schematizzazione di come deve essere impostata un'applicazione LabView per un qualsiasi modalità di funzionamento della scheda. Ogni libreria DAQmx è in grado di creare, manipolare e accettare un task, cioè un pacchetto di informazioni relative al compito da svolgere.

In generale possiamo pensare ogni applicazione suddivisa in 3 stadi principali:

1. creazione del task da eseguire
2. opzioni del task da eseguire
3. esecuzione, reset ed eliminazione del task creato

Il primo punto è di fondamentale importanza, in quanto in questa fase definiamo se uno o più canali della scheda devono essere utilizzati come ingressi o uscite. Una volta definito questo task, i canali interessati possono essere utilizzati solo in questo modo finchè il task non viene annullato. Tale libreria è rappresentata dall'icona più a sinistra in Fig. 2.2, nella quale si mette in evidenza il parametro di ingresso.

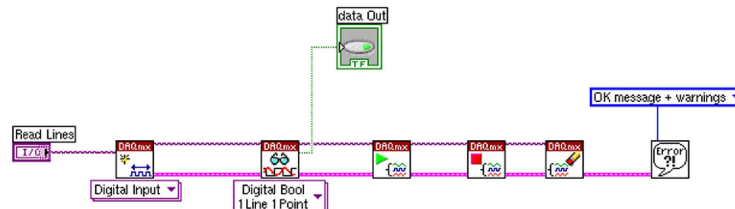


Figura 2.2: Diagramma a blocchi di un esempio di applicazione LabView per la lettura di una singola linea.

Il parametro d'ingresso rappresenta appunto i canali della scheda interessati in questo task. Il parametro può essere definito per esempio in questo modo: /Dev1/port0/line0, dove /Dev1 rappresenta il device DIO-6533, port0 rappresenta la porta A (port1 per la B, port2 per la C e port3 per la D), mentre line0 rappresenta la linea 0 della porta A (A0), line1...line7 sono le rimanenti linee della porta in esame.

Come detto in precedenza queste librerie possono essere utilizzate anche per altre schede, infatti la creazione del task fornisce diverse opzioni, quali per esempio analog I/O. È chiaro che essendo la scheda qui utilizzata una digital I/O non può essere utilizzata come analogica, quindi le altre opzioni relative ad altre schede non dovranno essere tenute in considerazione.

La struttura del secondo punto dipenderà da quale modalità della scheda si vuole utilizzare, per esempio la struttura di una modalità hadshaking sarà diversa da una modalità in linea singola. Per quanto riguarda questo secondo punto analizzeremo nel dettaglio alcuni esempi nei paragrafi successivi, concentrandoci in particolar modo sulle modalità utilizzate per la realizzazione del sistema di acquisizione. È ovvio che in questo stadio le funzioni da eseguire saranno in relazione al canale e al suo utilizzo di I/O definite nel task creato dal punto 1.

Nel terzo punto si utilizzano quelle librerie che mettono in esecuzione tutte le impostazioni che provengono dai punti 1 e 2, quindi si procede all'esecuzione

delle funzioni di stop e clear del task per rendere le linee interessate libere per altre modalità di funzionamento.

Nei paragrafi seguenti mostreremo esempi di base di utilizzo della scheda in diverse modalità, con particolare attenzione a quelle che saranno poi utilizzate per la realizzazione del sistema di acquisizione.

## 2.3 Funzionamento in lettura

La lettura dei dati digitali attraverso la scheda DIO-6533 può avvenire, come accennato precedentemente, in due modi:

- in singola linea
- a linee parallele

### 2.3.1 Lettura a singola linea

Nella modalità di utilizzo in singola linea, ogni linea può essere utilizzata in modo indipendente dalle altre. Le funzioni software di lettura fornite da LabView permettono l'acquisizione dello stato 0 o 1 (a seconda del segnale posto in ingresso) della linea indicata.

In Fig.2.2 riportiamo un esempio di applicazione LabView per la lettura di una singola linea. Nel diagramma a blocchi di tale applicazione si utilizzano alcune delle principali librerie NI-DAQmx. La funzione di creazione del canale di ingresso restituisce un task contenente le informazioni sul canale selezionato. Il canale può essere ad esempio creato sulla linea di indirizzo: /Dev1/port0/line0.

Il task viene poi inviato alla funzione di lettura (seconda icona grande da sinistra in Fig.2.2) che lo incrementa aggiungendo le informazioni necessarie per eseguire una lettura. La funzione di lettura restituisce inoltre il valore letto in una variabile numerica di LabView. Questa funzione polimorfica di NI-DAQmx, in questo caso viene impostata come *single channel-single sample*, che significa un solo valore su un singolo canale.

È da notare che le funzioni descritte finora non eseguono nessuna operazione effettiva sulla scheda, ma impostano semplicemente il task per le operazioni da svolgere. Il task è un'oggetto programmabile contenente le informazioni per svolgere determinate operazioni che vengono trasmesse tra una funzione e l'altra. Ogni funzione a sua volta può incrementare il task di informazioni relative ai

compiti da svolgere. Il task viene quindi eseguito dalla funzione *Play* (terza icona grande).

Procedendo verso destra nell'esempio di Fig. 2.2 seguono la funzione di stop e di eliminazione del task e la gestione dei messaggi di errore, rispettivamente rappresentate dalla quarta, quinta e sesta icona grande da sinistra.

Riepilogando, per la lettura in singola linea ci si attiene ai seguenti passi:

- Creazione del canale di input per una singola linea
- Impostazione della funzione di lettura come *single channel single sample*
- Avvio del task
- Stop del task
- Eliminazione del task per liberare le risorse occupate

Come esposto precedentemente in questa modalità di utilizzo non è possibile associare alle linee nessun tipo di controllo proveniente dalla scheda. Non è ammesso infatti l'utilizzo delle funzioni di handshaking tipiche delle librerie DAQmx, poichè le linee CLK, ACK e REQ non possono essere utilizzate con funzione di handshaking. In questa modalità la scrittura o la lettura possono essere gestite e sincronizzate attraverso controlli software generati dall'utente attraverso LabView, per esempio attraverso cicli *while* di attesa su determinate linee di I/O comuni, che rilevano un bit 1 o 0 e rispettivamente abilitano e disabilitano determinate procedure. Seguendo questa strada le prestazioni della scheda si riducono notevolmente poichè la frequenza di acquisizione risente fortemente dei tempi di esecuzione della macchina su cui è attiva l'applicazione di lettura. Nei test effettuati in laboratorio, utilizzando un calcolatore con un processore da 850 MHz e 256 MB di RAM, non si possono superare i 20 Hz di frequenza di acquisizione per ogni singolo bit su una linea. Siccome il sistema di acquisizione deve leggere parole da 16 bit, la frequenza di acquisizione per singola parola si ridurrebbe all'ordine di 1-2 Hz. La modalità a linee singole non risulta quindi consona al tipo di progetto illustrato in questa tesi, vista la necessità di frequenze di acquisizione superiori di almeno un ordine di grandezza.

### 2.3.2 Lettura a linee parallele

La lettura in modalità a linee parallele permette l'acquisizione simultanea di un determinato numero di linee. In questo caso le funzioni di lettura vedono il gruppo di linee come un'insieme di bit che costituiscono una parola binaria. Quindi

ogni linea rappresenta il bit della parola nella posizione indicata dalla linea stessa. Per esempio, utilizzando le linee 0-7 la parola binaria sarà a 8 bit e la linea 0 corrisponderà al bit meno significativo mentre la linea 7 al più significativo. La funzione di lettura restituisce la parola binaria convertita nel corrispondente numero decimale. Per esempio, in corrispondenza di una parola 00000011 la funzione di lettura restituirà il valore 3.

Le linee utilizzabili in parallelo devono essere necessariamente tutte le linee di una porta, un numero inferiore non viene accettato. Se si usa la porta A le linee utilizzate devono essere prese nell'intervallo A0.....A7. Esiste un ulteriore limite all'utilizzo delle porte: se si utilizzano le linee della porta B è necessario anche l'utilizzo delle linee della porta A, poichè fanno parte dello stesso gruppo, relativo ai controlli di CLK1, ACK1, REQ1, STOP1. Lo stesso discorso vale per le porte C e D, le quali fanno riferimento ai controlli CLK2, ACK2, REQ2, STOP2. Quindi nella modalità in parallelo si possono utilizzare al massimo 2 gruppi di parole da 16 bit. Sempre in questa modalità è possibile associare un clock di lettura che sincronizza l'acquisizione di una successione temporale di diversi stati assunti dalle linee. Il clock può essere importato da una fonte esterna attraverso la linea di controllo CLK1 per il gruppo di porte 1 e CLK2 per il gruppo 2, o esportato dalla scheda per la sincronizzazione.

Nell'esempio di applicazione LabView per la lettura simultanea di più linee, riportato in Fig.2.3, partendo da sinistra incontriamo come prima funzione, la creazione del canale di ingresso. Questa è presente come nella modalità di scrittura a singola linea, l'unica differenza sta nella variabile contenente il nome della linea, in questo caso si deve specificare il gruppo di linee in questo modo: /Dev1/port0/line0:7. La differenza è sostanzialmente solo la parte dell'indirizzo riguardante la linea, che ora /line0:7 rappresenta un intervallo di linee che devono essere utilizzate nella lettura/scrittura.

La seconda icona grande da sinistra in Fig.2.3 imposta le ulteriori funzioni di handshaking. Le impostazioni di questa funzione LabView sono:

- Sample clock output terminal (/Dev1/PFI4 o /Dev1/PFI5)
- Clock rate (Hz)
- Sample mode (Finite Sample/Continuos Sample)
- Sample per channel (Numero intero)
- Sample clock pulse polarity (Active High/Active Low)

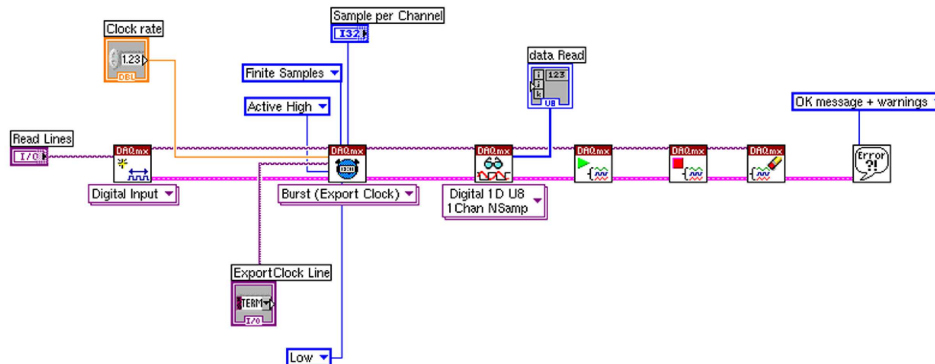


Figura 2.3: Diagramma a blocchi di un'applicazione LabView per la lettura parallela di più linee.

- Pause when (High/Low)

*Sample clock output terminal* rappresenta la linea di controllo sulla quale esportare <sup>2</sup> il clock, se si usa il gruppo di linee 1 sarà /Dev1/PFI4 dove /PFI4 è la linea CLK1, mentre se si usa il gruppo 2 sarà /Dev1/PFI5, corrispondente a CLK2.

*Clock rate* è la frequenza in Hz del clock esportato (max 20Hz) <sup>3</sup>.

*Sample mode* rappresenta la modalità di lettura che può essere di un numero infinito di valori (Continuous Samples) o di un numero stabilito (Finite Samples). Se *Sample Mode* è impostato con *Finite Samples* il numero di valori deve essere inserito nell'input *Samples per channel*, mentre se *Sample Mode* è impostato con *Continuous Samples* il valore *Samples per channel* non deve essere inserito, poichè viene impostato di default a -1 (numero infinito di valori).

*Sample clock pulse polarity* distingue se la lettura dello stato delle linee deve avvenire sul fronte di salita (Active High) o su quello di discesa (Active Low) del clock, vedi Fig.2.4, dove riportiamo una rappresentazione grafica della sincronizzazione tra ACK e CLK.

<sup>2</sup>Nel esempio citato si è esportato un clock, ma è possibile eseguire le stesse operazioni importando un clock esterno sulla stessa linea.

<sup>3</sup>Se il clock viene importato da una sorgente esterna questo input non esiste.

*Pause when* rappresenta la polarità del trigger. Il task si ferma e rimane in attesa anche dopo la funzione play se la linea di REQ relativa al gruppo di linee utilizzate (a cui è implicitamente legata) si trova nello stato indicato da questo input. Quando risulta vero il contrario il task riprende l'esecuzione.

Quando il task è in attesa, risulta comunque impegnato in un'operazione di I/O quindi l'ACK relativo al gruppo di linee in uso rimane nello stato 1 fino al termine del task.

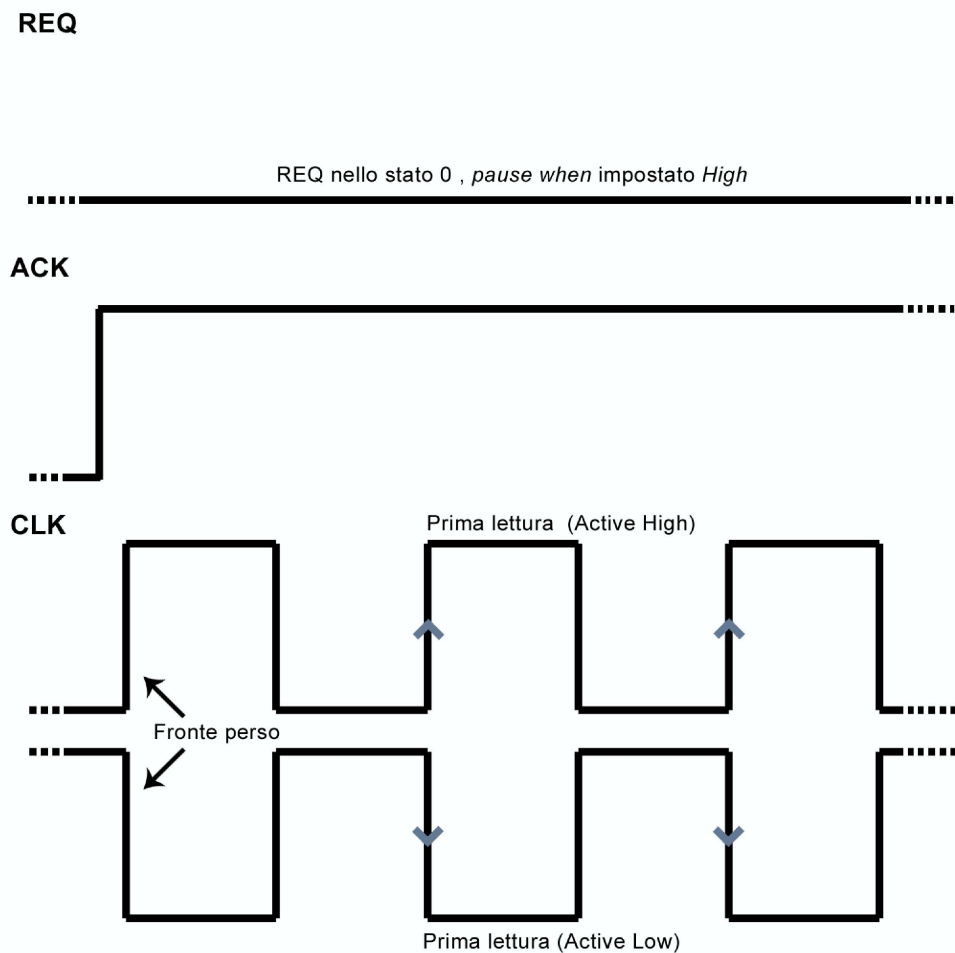


Figura 2.4: Rappresentazione dei segnali REQ, ACK e CLK nella modalità *Active High / Active Low*.

La terza icona da sinistra in Fig.2.3 rappresenta la funzione di impostazione del task per la lettura, qui la funzione viene configurata come *single channel multiple sample*, atta a ricevere una successione di più valori su ogni sigola linea. Il



dato in uscita da questa funzione risulta essere un vettore contenente elementi che sono la rappresentazione decimale delle parole di bit lette in successione dal gruppo di linee ad ogni fronte di salita o discesa del clock. Le icone successive eseguono le stesse funzioni indicate nella modalità in singola linea, vedi Sez. 2.3.1.

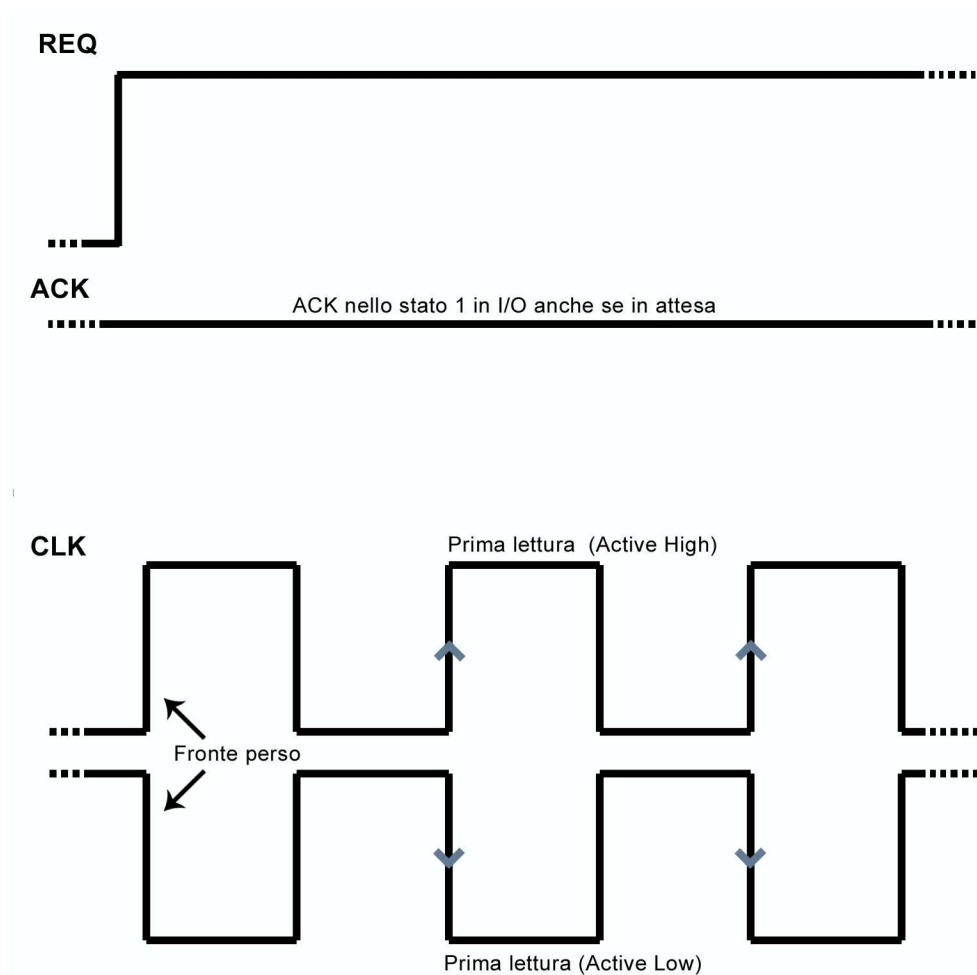


Figura 2.5: Rappresentazione dei segnali REQ, ACK e CLK nella modalità *Active High / Active Low*.

Dalle prove effettuate in laboratorio si sono evidenziate alcune caratteristiche fondamentali della modalità di lettura con handshaking: eseguendo una lettura di una successione di dati con l'esportazione di un clock sulla linea CLK, la linea ACK rimane nello stato 1 esattamente per il tempo necessario per l'acquisizione del vettore.

L'esportazione e avvio del clock avvengono simultaneamente al fronte di salita dell'ACK, tuttavia l'acquisizione del primo dato avviene al secondo fronte di salita o di discesa del clock, a seconda che sia impostata la modalità *Active High* o *Active Low*. In termini di acquisizione quindi, il primo fronte di clock viene irrimediabilmente perso vedi Fig.2.5, dove riportiamo una rappresentazione grafica della sincronizzazione tra REQ, ACK e CLK.

Nell'esempio descritto è sottinteso che la linea REQ fosse a 0 e che l'ingresso *pause when* della funzione di handshaking DAQmx fosse impostato come *High*.

Si è inoltre evidenziato che se la stessa operazione viene eseguita con la linea REQ a 0 e l'ingresso *pause when* della funzione di handshaking DAQmx impostato come *Low*, il task si blocca immediatamente in attesa di uno stato 1 sulla linea REQ. La linea ACK invece passa allo stato 1 appena viene avviato il task, rimanendo questo in attesa del REQ. Quando si trasmette lo stato 1 alla REQ il task si sblocca e viene eseguita l'operazione di lettura, ma anche in questo caso c'è la perdita del primo fronte di clock, vedi Fig.??, dove riportiamo una rappresentazione grafica della sincronizzazione tra REQ, ACK e CLK. La lettura del primo bit avviene quindi sempre un periodo di clock in ritardo rispetto all'arrivo del REQ (o ACK nel caso non si usi il trigger). Dopo il termine della lettura, in entrambi i casi descritti, la linea ACK torna allo stato 0 ma la linea CLK continua ad emettere il segnale di clock. Nell'istante in cui avviene una seconda lettura e la linea ACK passa di nuovo allo stato 1, viene rigenerato un nuovo clock con le stesse caratteristiche.

## 2.4 Funzionamento in Scrittura

Le impostazioni della funzione di scrittura della scheda sono in generale le stesse della funzione di lettura. Anche in questo caso c'è la possibilità di utilizzare le linee singolarmente o in parallelo, con le stesse modalità della lettura. Nelle Fig.2.6 e 2.7 sono mostrati rispettivamente i diagrammi a blocchi di scrittura singola e parallela. Da queste figure si può notare che nei diagrammi per la scrittura non ci sono sostanziali differenze dai diagrammi per la lettura. La creazione di un canale di input è sostituita dalla creazione di un canale di output, seguendo le stesse regole citate per il funzionamento in lettura.

La seconda funzione (dedicata all'Handshaking) è esattamente identica a quella utilizzata per il funzionamento in lettura. La funzione di impostazione del task per le opzioni di lettura (Terza icona grande nelle Fig.?? e 2.3) viene sostituita nelle Fig.2.6 e 2.7 con una funzione NI-DAQmx per l'impostazione del task

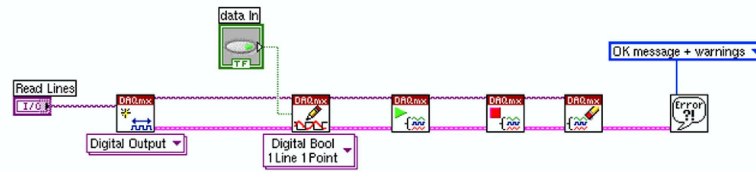


Figura 2.6: Diagramma a blocchi di un esempio di applicazione LabView per la scrittura di una singola linea.

di scrittura, quindi si deve fornire a questa funzione un input di dati da scrivere sulle linee.

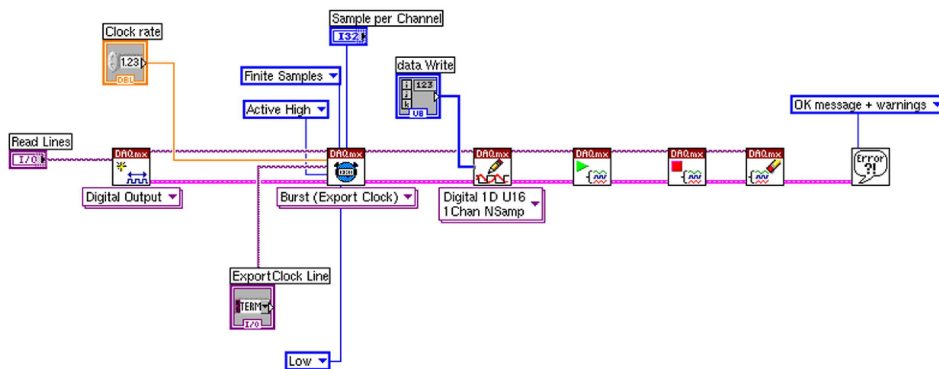


Figura 2.7: Diagramma a blocchi di un esempio di applicazione LabView per la scrittura parallela di più linee.

Il resto del diagramma e il concetto di base rimangono pressochè invariati. Nella modalità di scrittura con funzione di handshaking, il task rimane in attesa di uno stato del REQ che gli permetta di proseguire abilitando il clock in uscita dalla linea CLK e iniziando la scrittura dei dati ad ogni fronte. Anche in questo caso la scelta del fronte di scrittura avviene preventivamente con l'impostazione del task attraverso la funzione di handshaking.

Una differenza da tenere in considerazione tra la modalità di scrittura e quella di lettura è nella sincronizzazione della scrittura in corrispondenza dell'arrivo di un trigger all'ingresso REQ. In questo caso la scrittura parte al primo fronte di clock, quindi non viene perso come nel caso della lettura. In scrittura quindi il

primo fronte utile si ha un semiperiodo di clock in ritardo rispetto all'arrivo del REQ (o ACK nel caso in cui non si usi il trigger). Il funzionamento in scrittura è descritto dal diagramma in Fig.2.8, dove riportiamo una rappresentazione grafica della sincronizzazione tra fronte di salita in ingresso alla linea REQ e inizio della scrittura.

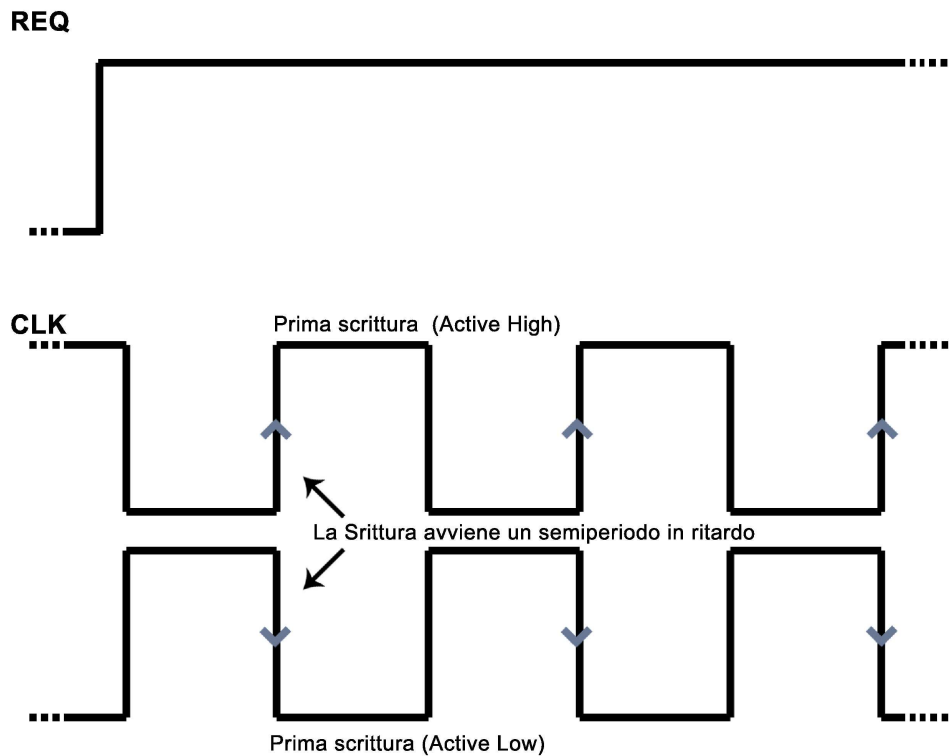


Figura 2.8: Rappresentazione dei segnali REQ, ACK e CLK nella modalità *Active High / Active Low*.

# Capitolo 3

## Applicazione di acquisizione dati

In questo capitolo si descriverà l'applicazione software realizzata per il sistema di acquisizione. La necessità di collaudare tale applicazione ha portato alla costruzione di un sistema di simulazione dell'apparato di rivelamento e dell'elettronica di Front End. È stata costruita una scheda hardware contenente dei registri a scorrimento come simulazione delle linee del registro della FEC, inoltre si è realizzata un'applicazione software che simula il funzionamento del sistema di LST e riempie i registri della scheda. Anche la scheda hardware di simulazione e l'applicazione di scrittura sulla stessa saranno trattate in questo capitolo.

### 3.1 Requisiti dell'applicazione di acquisizione

L'applicazione di acquisizione ha il compito di scaricare e memorizzare i dati binari dalla FEC attraverso la scheda DIO-6533. L'applicazione deve procedere allo svuotamento delle linee del registro della FEC all'occorrenza di un *evento*, la cui presenza è individuata da un opportuno *trigger*. Un esempio di trigger potrebbe essere la richiesta di avere almeno un certo numero di strati (2 o 3) del rivelatore colpiti. Uno specifico numero di strati colpiti potrebbe significare un evento reale in cui un muone attraversa il rivelatore. Diversamente un solo strato colpito potrebbe essere provocato da rumore in generale (del rivelatore o dell'elettronica) oppure essere l'inizio di un evento, quindi è necessario attendere altre informazioni dal rivelatore prima di acquisire l'evento. Nella pratica il trigger verrà realizzato con una logica esterna sia alla FEC sia alla scheda di acquisizione, e il segnale di trigger verrà utilizzato, attraverso la scheda, dal programma di acquisizione per procedere all'acquisizione dell'evento.

A seguito dell'arrivo del segnale di trigger, l'applicazione deve procedere al-

la lettura dei registri della FEC. Durante il periodo di tempo che intercorre tra l'arrivo del trigger e la fine della lettura dei registri è necessario che la FEC non aggiorni questi. In questo modo si può eseguire lo scaricamento completo dei dati senza che la FEC scriva nuovi bit, relativi ad un diverso evento, sui registri.

Questa necessità è soddisfatta dalla presenza nella FEC del *bit enable*, il quale viene controllato attraverso lo stesso segnale di trigger. Al fronte di salita del segnale di trigger il registro viene congelato, mentre al fronte di discesa ritorna abilitato a ricevere nuovi dati.

Il programma di acquisizione, all'arrivo del trigger, deve comandare la scheda di acquisizione affinché si scarichino le linee del registro della FEC e simultaneamente si proceda alla memorizzazione delle parole per poi elaborarle e/o salvarle direttamente in un file. Per eseguire questa operazione l'applicazione deve inviare un clock alla FEC attraverso la scheda di acquisizione. Il fronte di memorizzazione del clock deve essere successivo a quello di emissione del dato da parte del registro, per esempio se il registro espelle un bit ad ogni fronte di salita del clock, il bit è reso disponibile al fronte di salita, ma deve essere letto al fronte di discesa, istante in cui vi è la sicurezza che non stà avvenendo un cambio di stato. Se il bit fosse letto sullo stesso fronte di emissione si potrebbe correre il rischio di leggere prima dell'emissione di questo, rischiando quindi di acquisire il valore precedente, piuttosto che quello attuale,

Riassumendo, per ogni evento si devono verificare le seguenti operazioni:

1. arrivo del trigger e simultaneo congelamento del registro della FEC
2. invio del clock e sincronizzata memorizzazione della parola

Come visto nel capitolo precedente la scheda di acquisizione impostata in configurazione di lettura presenta la possibilità di svolgere tutte le funzioni richieste.

## 3.2 Applicazione di lettura

L'applicazione di acquisizione deve scaricare i buffer della FEC in sincronia con un clock inviato alla stessa, per questo motivo si è deciso di impostare un tipo di lettura parallela delle linee. Si è scelto il solo utilizzo delle 8 porte A0.....A7,poichè le linee del registro della FEC sono 4.

Nella funzione di creazione del canale, nell'applicazione di lettura,si è quindi passato il valore: /Dev1/port0/line0:7. La FEC ad ogni ciclo di clock espelle

simultaneamente un bit per ogni linea. Ad ogni fronte di clock si ha quindi a disposizione una parola di tanti bit quante sono le linee del registro. L'acquisizione procede alla memorizzazione di queste parole in successione, le quali vengono salvate in base decimale all'interno di un vettore di LabView. Successivamente l'applicazione si dedicherà alla conversione in binario dei suddetti valori decimali: ogni valore decimale viene convertito in una parola binaria (4 bit di lunghezza) e con i bit corrispondenti si costruisce un vettore a 4 componenti. Infine questo vettore viene memorizzato come colonna di una matrice di 4 righe, che ha tante colonne quanti sono i bit di ogni parola memorizzata in una singola linea del registro. La scelta della linea della matrice in cui memorizzare il bit avviene in base alla posizione dello stesso nella parola di 4: posizione 0, cioè bit meno significativo, derivante dalla linea A0, sarà memorizzato nella linea 0 della matrice; posizione 3, cioè bit più significativo derivante dalla linea A3 sarà memorizzato nella linea 3 e così via per le posizioni 1 e 2. I bit da A4 ad A7 non sono utilizzati poichè le linee sono scollegate dalla FEC. Una descrizione grafica dell'operazione eseguita è riportata in Fig.3.1.

Infine le linee della matrice sono concatenate l'una con l'altra creando una parola unica che verrà salvata successivamente in un file. In questa nuova parola i bit meno significativi sono quelli derivanti dalla linea A0, vedi schema riportato in Fig.3.1.

Ritornando ai dettagli dell'applicazione di lettura, nella funzione di handshaking sono stati impostati i seguenti parametri:

- Sample clock output terminal (/Dev1/PFI4)
- Clock Rate (Modificabile)
- Sample Mode (Finite Sample)
- Sample per channel (16)
- Sample clock pulse polarity (Active Low)
- Pause when (Low)

Per il *sample clock output terminal* si utilizza la linea CLK1, la frequenza di clock (*clock rate*) viene inserita attraverso l'interfaccia grafica dell'applicazione.

Il *sample mode* è stato impostato con il valore *Finite Sample*, per permettere l'acquisizione di una successione di elementi di numero finito. Il numero di elementi

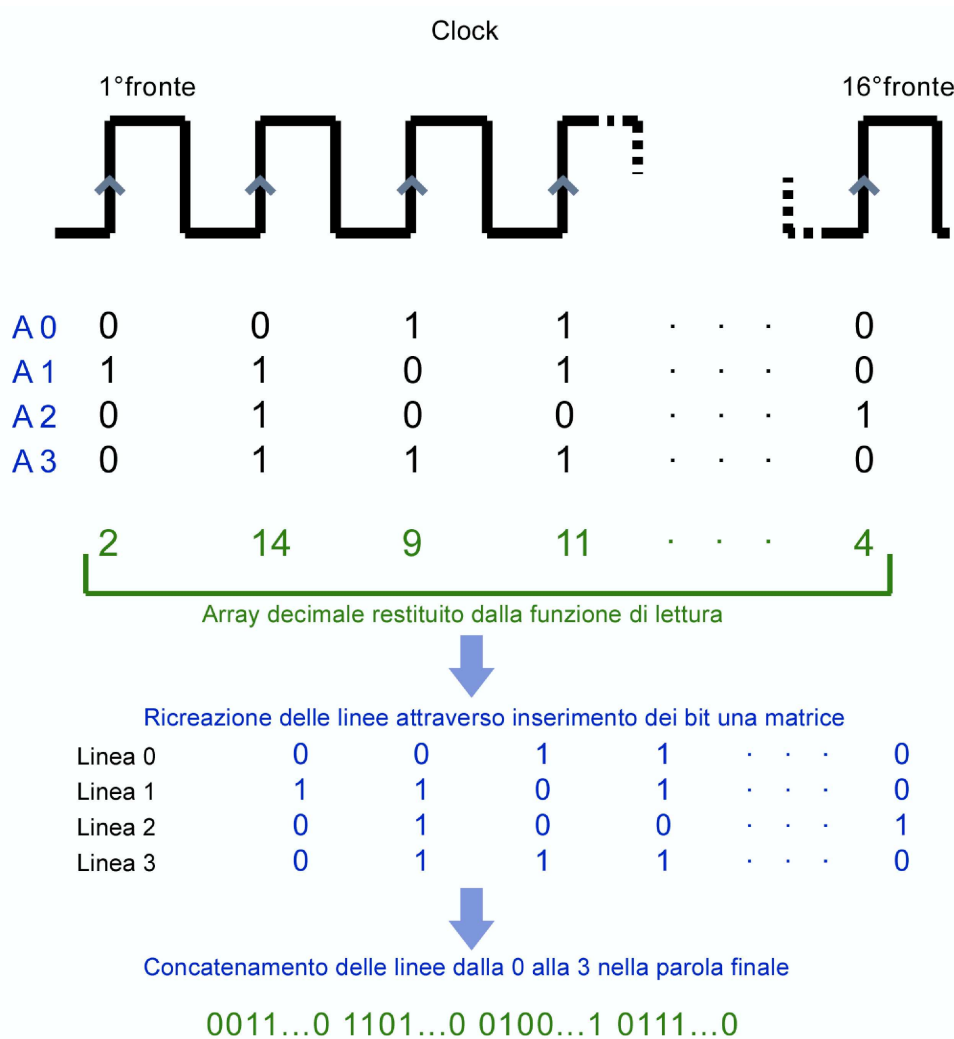


Figura 3.1: Descrizione grafica della gestione dei dati ricevuti.

acquisiti è impostato a 16 attraverso il controllo *sample per channel*, 16 bit è la dimensione del registro della FEC.

Il parametro *sample clock pulse polarity* viene impostato come *Active Low* per forzare l'acquisizione del dato sul fronte di discesa, visto che il registro della FEC espelle un bit ad ogni fronte di salita.

Si è deciso inoltre di impostare il controllo *pause when* con il valore *Low*, per forzare l'attesa dell'acquisizione fino all'istante in cui un segnale di trigger la abilita. Il trigger viene rilevato dall'applicazione attraverso la linea REQ1 della scheda.



di acquisizione.

Il clock per lo scaricamento sincronizzato, viene generato all'inizio dell'acquisizione della successione di 16 parole, ma al termine dell'operazione non si conclude e continua a generare fronti incontrollati. Questi fronti rischierebbero di scaricare il registro della FEC prima dell'avvento di una nuova lettura. Per risolvere il problema si è costruito un *AND* logico tra clock e trigger. Il trigger al termine della lettura dei dati passa allo stato 0, forzando l'abbattimento del clock in uscita della *AND*. Il clock quindi sarà rigenerato all'acquisizione successiva. Un esempio di questo sistema è visibile in Fig.3.2, nella quale riportiamo rispettivamente dall'alto verso il basso: i due segnali di ingresso (clock e trigger) e il segnale di uscita della *AND* (clock *AND* trigger).

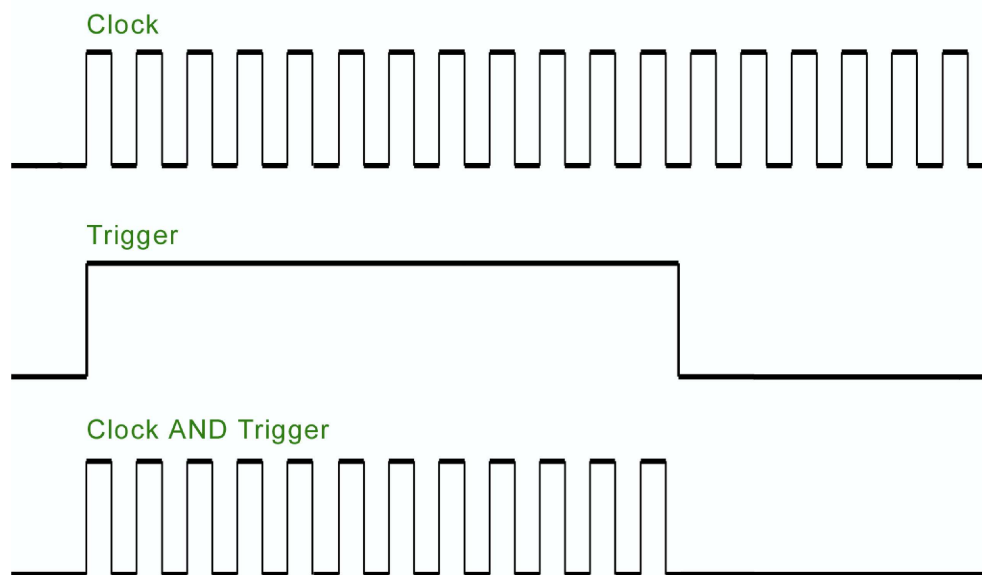


Figura 3.2: Rappresentazione grafica dei segnali clock e trigger e *AND* logico tra i due.

Tutte le operazioni dell'applicazione di acquisizione sono ripetute per un determinato numero di volte attraverso un ciclo *while* che le contiene. Il numero di acquisizioni eseguite può essere un numero finito o infinito. Ad ogni ciclo la funzione di lettura si blocca, in attesa di un segnale di trigger sulla linea REQ1. Ricevuto il segnale, si esegue l'acquisizione e si rimette in attesa di un nuovo segnale.

È da notare che se l'applicazione esegue un'acquisizione in un tempo minore

della durata del segnale di trigger, possono avvenire più letture al posto di una. La soluzione a questo problema sarà trattata nel prossimo paragrafo, dove si parlerà della simulazione hardware della FEC e delle logiche di clock e trigger.

L'applicazione di lettura contiene inoltre un'ulteriore procedura per il controllo degli errori di acquisizione il cui algoritmo di funzionamento sarà descritto nel prossimo capitolo riguardante i test effettuati sulle prestazioni del sistema di acquisizione. Il diagramma a blocchi che descrive l'applicazione di lettura è riportato in Fig.3.3.

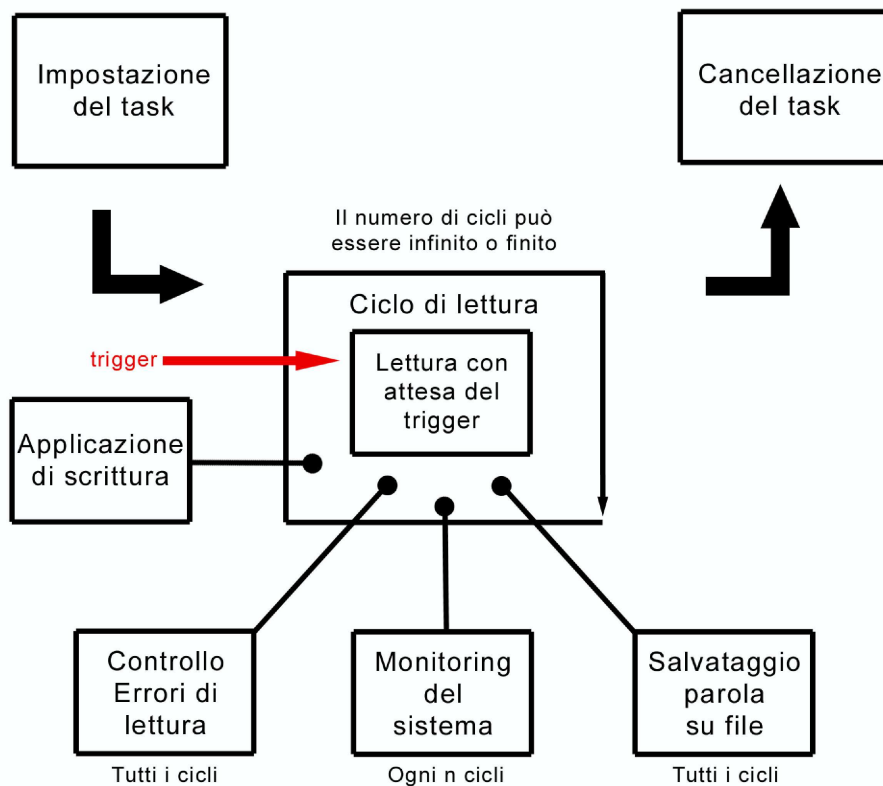


Figura 3.3: Diagramma a blocchi dell'applicazione di lettura

### 3.3 Simulazione

La simulazione della FEC è divisa in due parti:

1. la scheda hardware di simulazione delle linee del registro della FEC
2. l'applicazione di scrittura dei dati binari sulla scheda hardware

Sulla scheda realizzata sono integrate anche la logica del clock, trattata nel paragrafo precedente, e la logica del trigger, utilizzato per la sincronizzazione dell'applicazione di lettura con quella di scrittura. Un'immagine dell'interfaccia grafica dell'applicazione di lettura è riportata in Fig.3.10 nel paragrafo 3.6.

### 3.3.1 La scheda hardware di simulazione della FEC

La scheda hardware di simulazione della FEC contiene due registri a scorrimento da 8 bit ciascuno, collegati in cascata, al fine di simulare una linea del registro a 16 bit della FEC. Si è scelto di simulare una sola linea, più che sufficiente per valutare le prestazioni dell'applicazione di lettura. L'utilizzo di più linee non comporta nessun cambiamento in termini di trattamento dei dati e di gestione della lettura, quindi le linee A1, A2, A3, sono semplicemente rimaste scollegate, e portano sempre il valore 0.

I registri a scorrimento, dispongono di un ingresso per il clock di scaricamento seriale, che permette l'espulsione di un bit ad ogni fronte di salita (come il registro della FEC). L'ultimo bit della linea è disponibile sull'ultima uscita del secondo registro. Come spiegato nel capitolo precedente la funzione di lettura DAQmx Read genera un primo ciclo di clock in corrispondenza del quale non viene acquisito nessun valore, cioè la lettura parte al secondo ciclo, in questo caso al secondo fronte di discesa. Questo fatto porta alla perdita sistematica del bit disponibile sull'ultima uscita del secondo registro. Per ovviare a questo inconveniente si è deciso di inserire un flip-flop di tipo D alla fine della cascata dei due registri, il quale permette la memorizzazione dell'ultimo bit, reso disponibile all'uscita dello stesso flip-flop (vedi Fig.3.4) al secondo fronte di salita, e letto dall'applicazione di acquisizione al secondo fronte di discesa. Entrambi i registri e il flip-flop finale hanno in ingresso lo stesso segnale di clock. L'apparato sopra descritto è riportato in Fig.3.4. Uno schema e una descrizione più dettagliata della scheda di simulazione sono riportati in Appendice A, nella quale si tratta il problema della presenza di rumori che influiscono sul corretto funzionamento della scheda di simulazione.

La linea di memorizzazione realizzata con i registri, la quale simula una linea del registro della FEC, deve essere riempita attraverso un'applicazione di scrittura che sarà descritta in dettaglio nel sottoparagrafo successivo. Lettura e scrittura della linea devono essere sincronizzate, la lettura deve avvenire soltanto dopo che la scrittura è terminata. Inoltre deve essere presente un *multiplexer* per smistare i clock di lettura e scrittura al registro, rispettivamente nelle fasi di let-

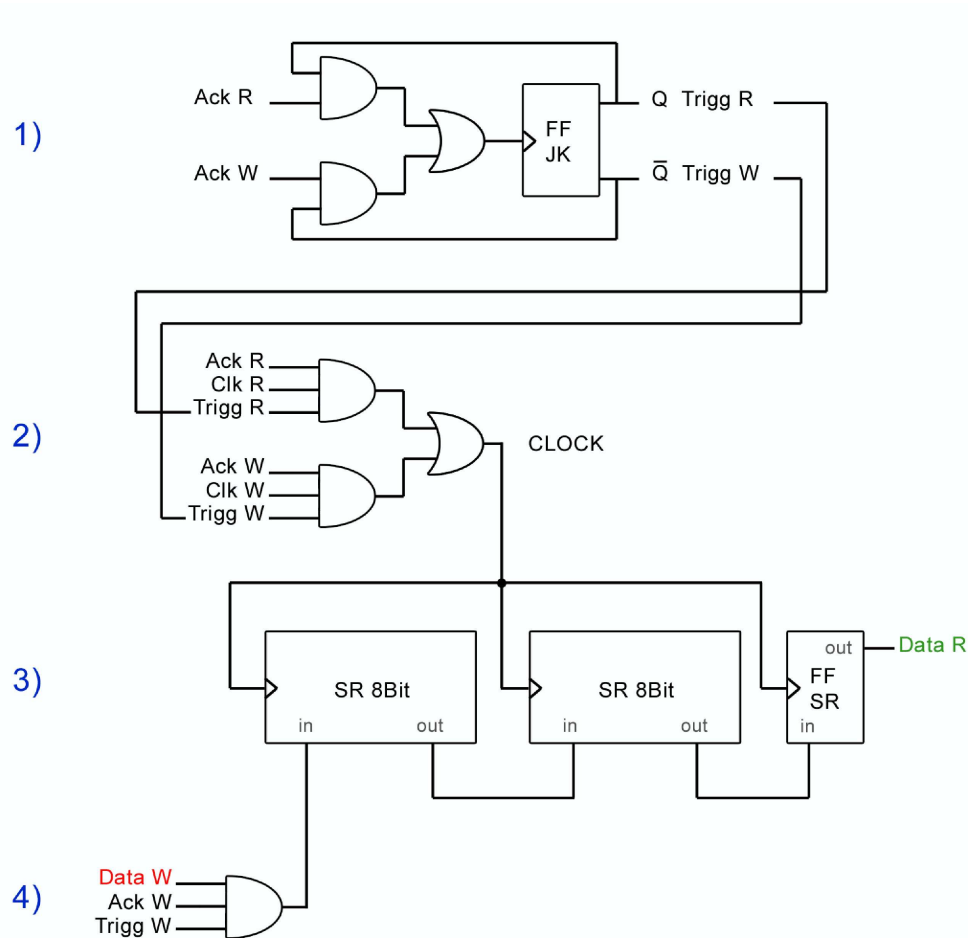


Figura 3.4: Schema della scheda di simulazione della FEC

tura e scrittura. Si è progettata e realizzata la scheda di simulazione hardware della FEC in modo da contenere i seguenti due ulteriori moduli:

1. logica del trigger
2. logica del clock

La logica del trigger rappresentata al punto 1 della Fig.3.4 è stata realizzata utilizzando la retroazione delle uscite  $Q$  e  $\bar{Q}$  di un flip-flop di tipo J-K, il quale inverte il valore delle uscite ogni volta che rivela un fronte di discesa nel segnale di clock in ingresso. Il segnale di clock viene generato attraverso l'OR logico di:

- $ACK1 \text{ AND } Q$

- $ACK2 AND \bar{Q}$

ACK1 rappresenta l'acknowledge di lettura, esso passa allo stato 1 durante l'operazione di acquisizione e torna allo stato 0 terminata la lettura. Un istante dopo si trova di nuovo a 1, visto che la funzione di lettura all'interno di un ciclo *while* è forzata a fermarsi, ad un nuovo ciclo, in attesa di segnale di trigger sulla linea di REQ1. La stessa cosa succede per l'ACK2 della scrittura, ne consegue che le due linee ACK1 e ACK2 restano attive per la maggior parte del tempo.

La retroazione (Q e  $\bar{Q}$ ) serve ad abilitare il passaggio dello stato 1 di uno delle due linee ACK attraverso una delle due porte *AND*. Nell'istante in cui la linea ACK abilitata passa allo stato 0, l'uscita della porta *AND* a cui è connessa passa allo stato 0, la seconda porta *AND* si trovava già a 0, quindi l'*OR* logico delle uscite delle *AND* passa a 0. Si è quindi generato un fronte di discesa del clock in ingresso al flip-flop JK che inverte il valore delle uscite, abilitando il passaggio dell'altra linea CLK. Le uscite Q e  $\bar{Q}$  oltre ad essere utilizzate nella retroazione, risultano essere due trigger complementari, i quali sincronizzano le funzioni di lettura e scrittura.

La logica del clock visibile al punto 2 di Fig.3.4 è progettata attraverso l'ingresso in porte *AND* separate di:

1. ACK1,CLK1 e Trigger di lettura (Q)
2. ACK2,CLK2 e Trigger di scrittura ( $\bar{Q}$ )

questa logica è necessaria per il motivo descritto nel paragrafo precedente, il clock esportato dalla funzione di handshaking non termina alla fine di una operazione di I/O, con la logica *AND* si forza la terminazione dello stesso al termine del segnale di trigger. I clock in lettura e scrittura vengono quindi smistati attraverso i relativi segnali di trigger e una porta *OR* che ha in ingresso le uscite delle due *AND*.

### 3.3.2 Applicazione di scrittura sulla scheda di simulazione

L'applicazione di scrittura delle parole di bit sulla linea del registro funziona allo stesso modo dell'applicazione di lettura, anche la funzione DAQmx di handshaking è impostata con gli stessi parametri le uniche differenze sono:

- la scelta delle linee di uscita della scheda DIO-6533, e di conseguenza l'utilizzo delle linee di handshaking appartenenti al gruppo di linee scelto.

- la funzione *DAQmx Read* è sostituita dalla funzione *DAQmx Write*

Le linee della scheda di acquisizione utilizzate per la scrittura sono quelle da C0 a C7, quindi la funzione di creazione del canale ha in ingresso il valore /Dev1/port2/line0:7, le linee di handshaking relative sono quindi ACK2, REQ2, CLK2. La funzione *Write* riceve in ingresso un vettore contenente le rappresentazioni decimali delle parole che saranno scritte in parallelo sulle linee utilizzate. In realtà, sebbene di default si utilizzino 8 linee per la scrittura, i valori contenuti nell'array non possono essere diversi da 0 o 1, in quanto stiamo usando una sola linea di scrittura. Le rimanenti porte C1..C7 rimangono scollegate. Anche l'applicazione di scrittura attende un trigger per l'avvio dell'esecuzione sulla linea REQ2.

### Modalità di scrittura sulla scheda di simulazione

La scrittura delle parole di 16 bit sulla scheda di simulazione della FEC è stata creata per essere utilizzabile secondo due modalità:

- scrittura di una parola singola, la quale rimane fissa ad ogni altra successive scrittura.
- scrittura di parole random: ad ogni scrittura viene generata una nuova parola.

Nella modalità di utilizzo di scrittura di una parola data, l'applicazione di scrittura attiva sull'interfaccia grafica un controllo per l'inserimento dei bit della parola in un vettore LabView.

Nella modalità di scrittura con parole random, vengono generate parole *online* attraverso un modulo software LabView, integrato nell'applicazione di scrittura. Ad ogni ciclo il modulo genera la parola, segue la scrittura delle stessa sui registri della scheda di simulazione della FEC. Ad ogni ciclo si ha quindi la scrittura di una nuova parola.

Un'analisi dettagliata dei risultati ottenuti attraverso queste due modalità di scrittura è riportata nel capitolo 4.

Il diagramma a blocchi che descrive l'applicazione di scrittura è riportato in Fig. 3.5.

Nelle Fig.3.12 e Fig. 3.13 nel paragrafo 3.6, sono riportate le immagini dell'interfaccia grafica dell'applicazione di scrittura, nelle modalità a parola singola e a parole random.

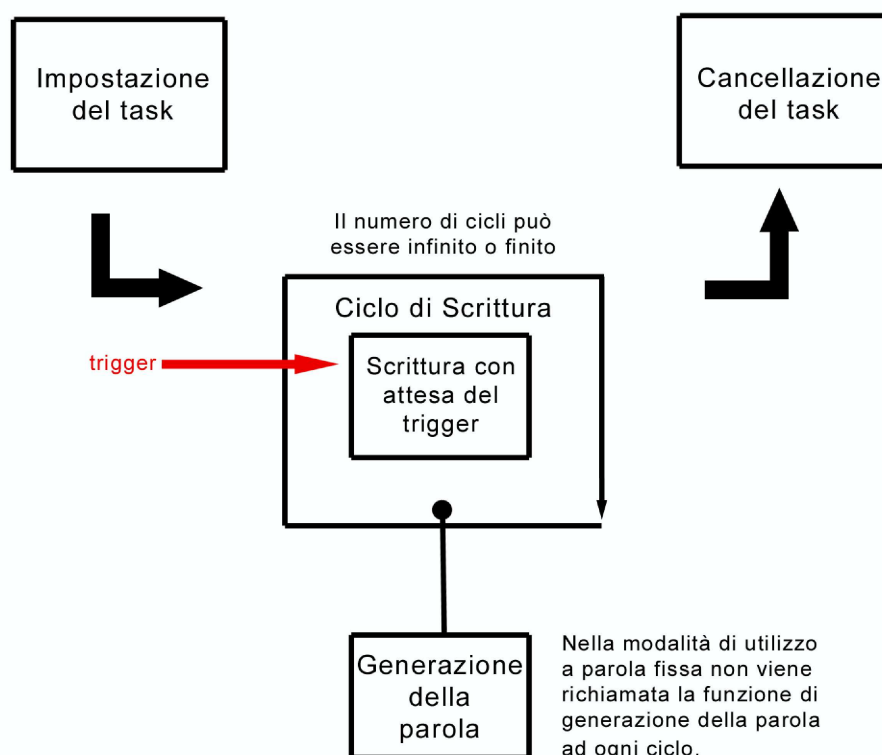


Figura 3.5: Diagramma a blocchi dell'applicazione di scrittura.

### 3.4 Realizzazione dell'applicazione per la sola lettura

La versione dell'applicazione di lettura della scheda di simulazione della FEC, descritta precedentemente, deve svolgere le operazioni di lettura dei registri, solamente dopo che l'applicazione di scrittura ha riempito i registri, operazione che richiede un determinato intervallo di tempo. Il tempo di esecuzione dell'applicazione di scrittura influisce quindi sulla frequenza di acquisizione. L'applicazione di lettura è inoltre appesantita dalla procedura di controllo degli errori eseguita ad ogni acquisizione, la quale introduce un secondo decremento della frequenza di acquisizione. Nella realtà l'applicazione di lettura deve semplicemente scaricare i registri della FEC, per effettuare i test sulle prestazioni reali del sistema di acquisizione si è voluto quindi creare una seconda versione dell'applicazione di lettura.

La seconda versione dell'applicazione di lettura è stata liberata dalla dipendenza temporale dell'applicazione di scrittura, che viene eseguita solo una volta.

In seguito viene posto manualmente il registro della scheda di simulazione in una configurazione circolare, dove l'ultima uscita del secondo registro viene collegata all'ingresso del primo. Viene quindi avviata l'applicazione di lettura che invia un clock al registro e acquisisce i bit. Ad ogni ciclo di clock il bit sull'uscita del registro, grazie alla configurazione circolare, passa all'ingresso del registro. Al termine della lettura, la parola si trova di nuovo nel registro, e al prossimo ciclo di acquisizione sarà di nuovo letta, e così via fino al termine dell'applicazione. Il diagramma a blocchi della seconda versione dell'applicazione di lettura è riportato in Fig.3.6.

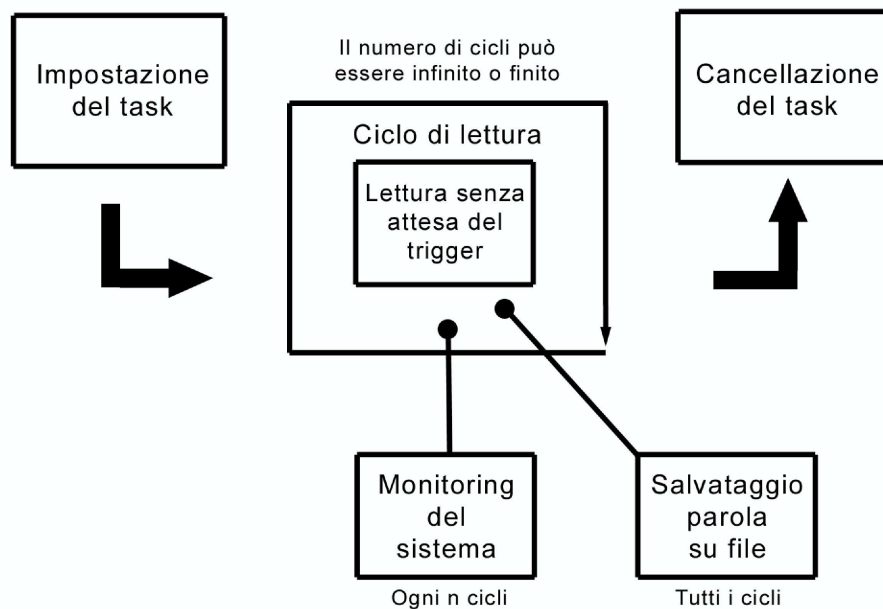


Figura 3.6: Diagramma a blocchi dell'Applicazione di lettura senza simulazione

Un'immagine dell'interfaccia grafica dell'applicazione di lettura senza simulazione è riportata in Fig.3.10 nel paragrafo 3.6.

### 3.4.1 Modifiche alla scheda di simulazione della FEC

La creazione di questa nuova versione dell'applicazione di lettura senza modulo di scrittura ha portato alla necessità di introdurre modifiche alla scheda progettata come simulazione dei registri della FEC. Dalla scheda è stata tolta la logica del trigger, visto che l'applicazione di lettura funziona indipendentemente da quella di scrittura, cioè non c'è più necessità di sincronizzazione tra le due. La logica che



smista il clock al registro è rimasta pressochè invariata. L'unica differenza è che lo smistamento non è più controllato dal trigger lettura/scrittura ma da un bit di controllo manuale. Quindi come rappresentato nel circuito in Fig.3.7, agli ingressi delle *AND* della logica di controllo abbiamo rispettivamente *ACK1,CLK1* bit di controllo negato e *ACK2, CLK2* bit di controllo non negato. Le uscite delle due porte *AND*, come prima, sono ingressi di una porta *OR*, la cui uscita è il clock in ingresso alla linea del registro. È stato quindi costruito un altro multiplexer controllato dallo stesso bit per smistare i dati da scrivere nel registro. Un ingresso della prima porta *AND* è collegato alla linea di scrittura dalla scheda di acquisizione, mentre uno dei due ingressi della seconda porta *AND* è collegato all'uscita dell'ultimo registro (retroazione). Le uscite di queste due porte *AND* sono ingressi di una porta *OR*, la cui uscita è collegata all'ingresso del primo registro. Manualmente si abilita quindi la scrittura della parola o la scrittura della retroazione. Una rappresentazione grafica della seconda versione del circuito è riportata in Fig.3.7.

### 3.5 Monitoring di acquisizione

Il monitoring di acquisizione rappresenta la parte dell'applicazione di acquisizione dedicata al controllo del funzionamento di tutte le linee di uscita degli LST. Dopo aver convertito i valori decimali contenuti nel vettore di output in una parola binaria (conversione identica a quella descritta nel paragrafo 3.2 dell'applicazione di lettura), il modulo dell'applicazione di lettura dedicato al monitoring si occupa della creazione di un istogramma con tante classi quanti sono i canali di lettura LST+Strip. Ogni classe corrisponde ad un canale ed il corrispondente numero di eventi viene incrementato ogni volta che il canale è nello stato 1. Al fine di non limitare la frequenza di acquisizione si è scelto di eseguire l'analisi di un evento ogni *N* eventi registrati. *N* è un numero di eventi che possiamo impostare a seconda delle esigenze. È chiaro che più *N* è grande meno il processo di monitoring influenzerà le prestazioni dell'acquisizione. Il numero *N* è modificabile dall'interfaccia grafica dell'applicazione di lettura.

Una volta ricevuta la parola da analizzare, si controlla quale dei bit della stessa è un 1 e, calcolata la posizione del bit nella parola, si incrementa di 1 il valore dell'istogramma alla stessa posizione. Questa procedura consente di conoscere *online* lo stato di tutti i tubi e delle strip. Questo monitoring è in grado quindi di fornire lo stato di funzionamento di ogni canale del rivelatore. Per esempio, un canale spento sarà individuato dal corrispondente valore nell'istogramma che

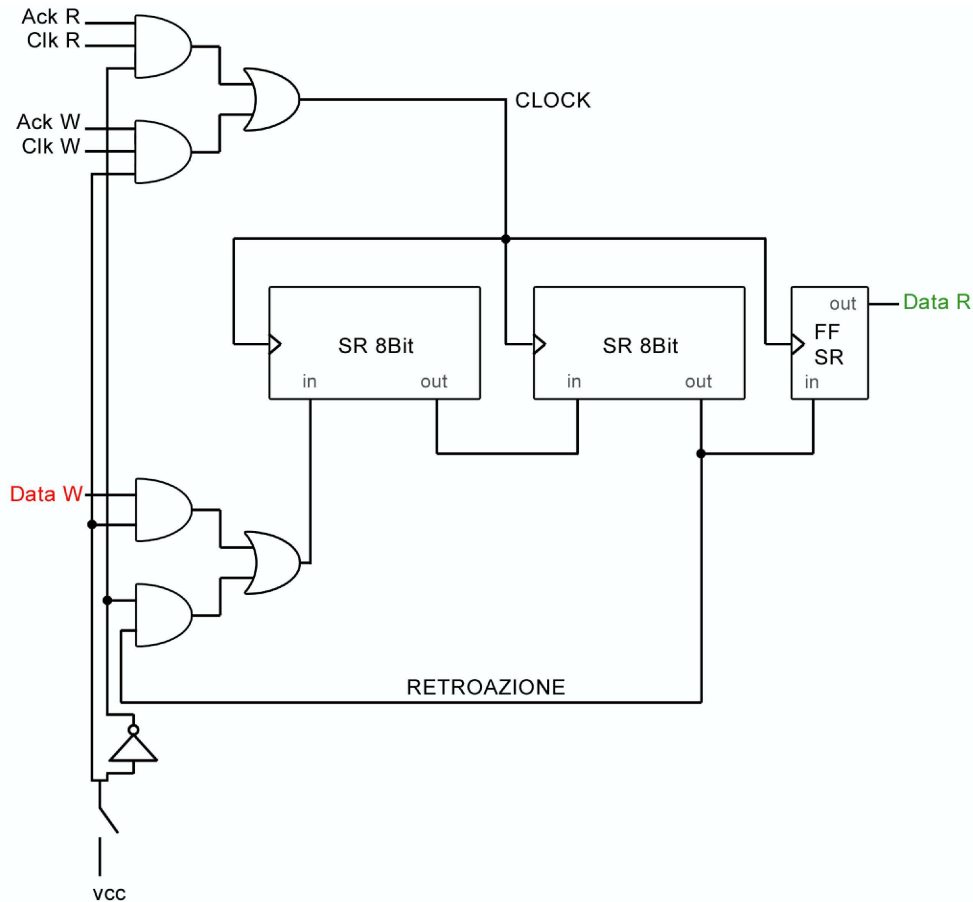


Figura 3.7: Schema grafico della seconda versione del circuito di simulazione della FEC

rimane a zero. Un esempio di istogramma che rivela il corretto funzionamento di tutti i canali dei rivelatori del sistema è riportato in Fig.3.8. Come si può notare non ci sono valori 0.

Al contrario, in Fig.3.9 è riportato un istogramma che rivela il malfunzionamento di due canali. I canali in questione sono il canale 2 e il canale 8, come si può notare dalla figura. Una situazione di questo tipo può essere attribuita a problemi sui canali in questione, per esempio connessioni rotte, alta tensione spenta, ecc.

### 3.6 Interfaccia grafica dell'applicazione

In questo paragrafo si riportano quattro immagini dell'interfaccia grafica delle applicazioni di lettura e scrittura.

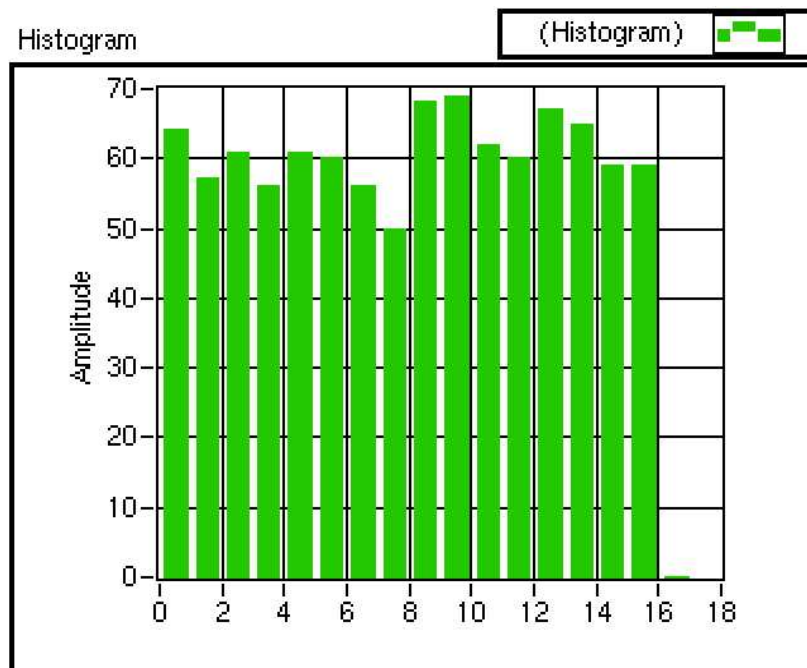


Figura 3.8: Esempio di monitoring in cui si riporta un istogramma nel quale non si rileva la presenza di canali spenti

La Fig.3.10 rappresenta l'interfaccia grafica dell'applicazione di lettura implementata per l'uso con la simulazione. Sulla parte sinistra è presente il pannello di impostazione. Partendo da in alto a sinistra sono presenti sul pannello i seguenti controlli:

- gruppo di linee sulle quali effettuare la lettura
- linea di esportazione del clock
- indirizzo del file di output
- numero di linee su cui leggere
- lunghezza della parola da leggere
- frequenza del clock da esportare
- numero di eventi da acquisire

In alto a destra è presente il pannello di visualizzazione degli errori di acquisizione e delle frequenze di acquisizione. In basso si trova invece il pannello di

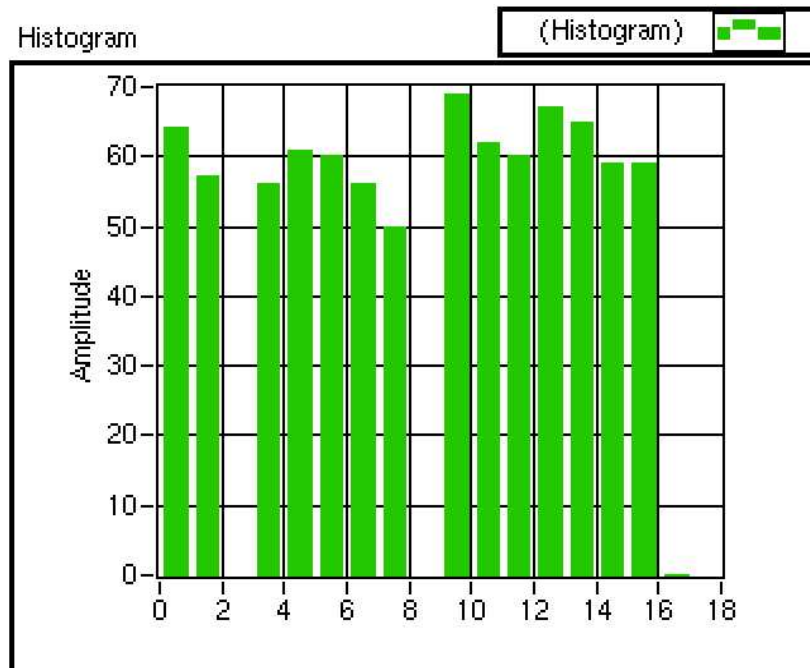


Figura 3.9: Esempio di monitoring in cui si riporta un istogramma nel quale si rileva la presenza di 2 canali spenti, il 2 e l'8

monitoring del sistema di acquisizione.

In Fig. 3.11 si trova l'interfaccia dell'applicazione di lettura implementata per il funzionamento senza simulazione. Il pannello di impostazione e il pannello di monitoring risultano identici a quelli riportati nell'interfaccia dell'applicazione di lettura con simulazione. Manca invece l'indicatore del numero di errori di acquisizione, poichè il calcolo di questi errori si esegue *offline*.

Nelle Fig. 3.12 e 3.13 è riportata l'interfaccia grafica dell'applicazione di scrittura rispettivamente nella configurazione di scrittura di una singola parola e nella configurazione di scrittura di parole random. Partendo dall'alto sono presenti sul pannello i seguenti controlli:

- gruppo di linee sulle quali effettuare la scrittura
- linea di esportazione del clock
- numero di linee su cui scrivere
- lunghezza della parola da scrivere

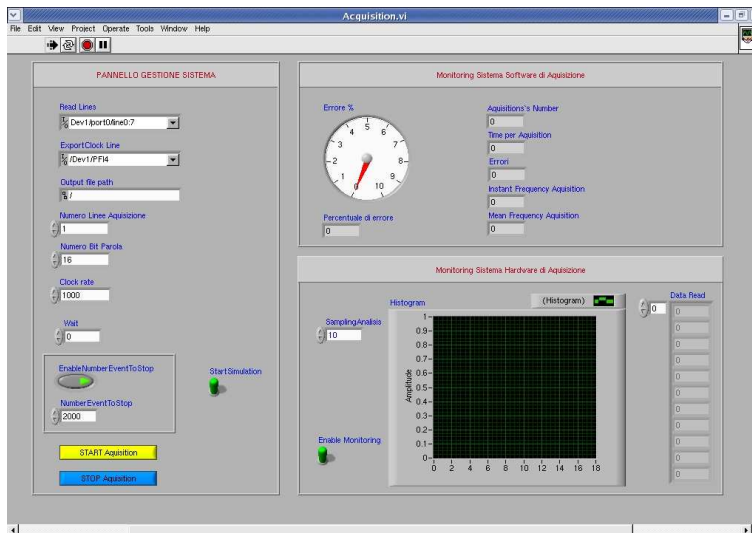


Figura 3.10: Interfaccia grafica dell'applicazione di lettura con la simulazione.

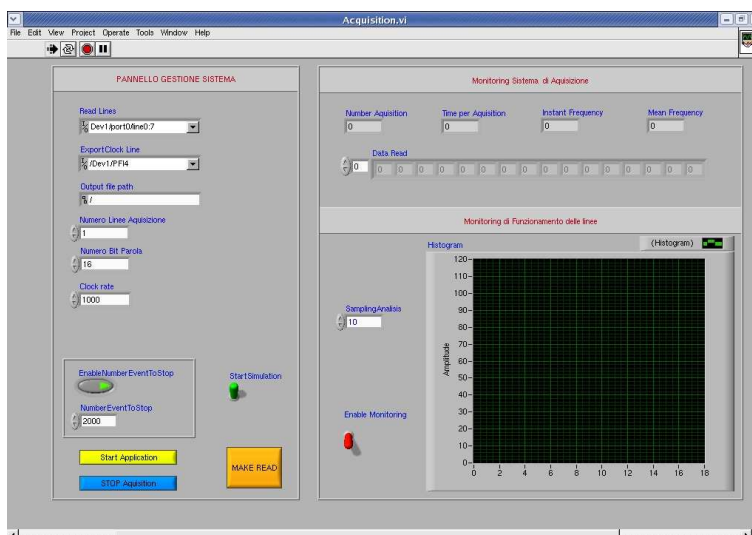


Figura 3.11: Interfaccia grafica dell'applicazione di lettura senza simulazione.

- frequenza del clock da esportare
- numero eventi da generare

L'interfaccia grafica dell'applicazione di scrittura nella configurazione di scrittura di parole random, dispone di un'ulteriore controllo per l'inserimento delle componenti del vettore da scrivere. L'interfaccia grafica dell'applicazione di scrittura compare solo quando dal pannello di lettura con simulazione si avvia la simulazione.

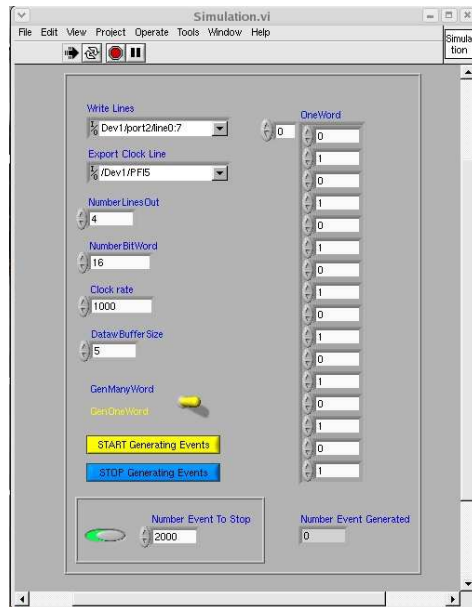


Figura 3.12: Interfaccia grafica dell'applicazione di scrittura in configurazione di scrittura di una parola singola.

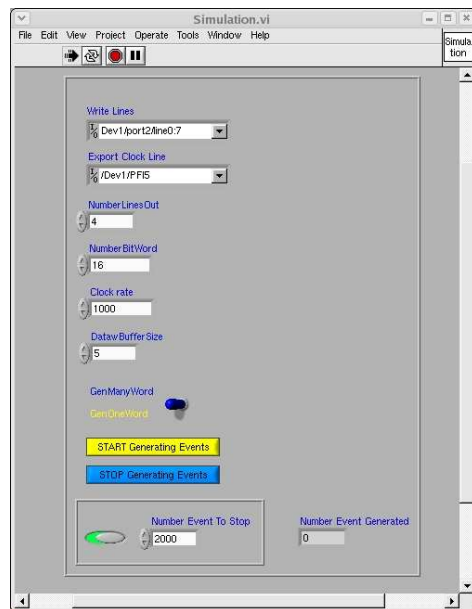


Figura 3.13: Interfaccia grafica dell'applicazione di scrittura in configurazione di scrittura di parole random.

# Capitolo 4

## Prestazioni del sistema di acquisizione

In questo capitolo si descriveranno i test effettuati per la valutazione delle prestazioni del sistema di acquisizione. In particolare valuteremo la frequenza di acquisizione e la frazione di errori di acquisizione in funzione della frequenza di clock. Distingueremo inoltre i test effettuati sul sistema di acquisizione con e senza la simulazione.

### 4.1 Misura della frequenza di acquisizione

#### 4.1.1 Test con simulazione

La frequenza di acquisizione viene misurata come il rapporto fra un determinato numero di letture e il tempo impiegato per effettuare tali letture. I tempi sono misurati con le appropriate applicazioni di LabView. Le misure sono state eseguite su campioni di 2000 acquisizioni a diversi valori della frequenza di clock nell'intervallo fra 300 Hz e  $10^5$  Hz. Questi test sono stati eseguiti in due diverse configurazioni della simulazione, cioè utilizzando la generazione random delle parole scritte e con la scrittura di una singola parola impostata all'inizio del test. I risultati sono mostrati in Fig. 4.1 nella quale riportiamo i grafici della frequenza di acquisizione in funzione della frequenza di clock per la configurazione parola random ( Fig.4.1.(a) ) e parola singola ( Fig.4.1.(b) ).

Dalla Fig. 4.1 si può notare che la frequenza di acquisizione dipende dalla frequenza di clock fino a un valore di quest'ultima di circa  $10^3$  Hz, oltre questo valore la frequenza di acquisizione diviene indipendente dal clock. Questo comportamento era prevedibile, in quanto, fin tanto che la frequenza di clock è bas-

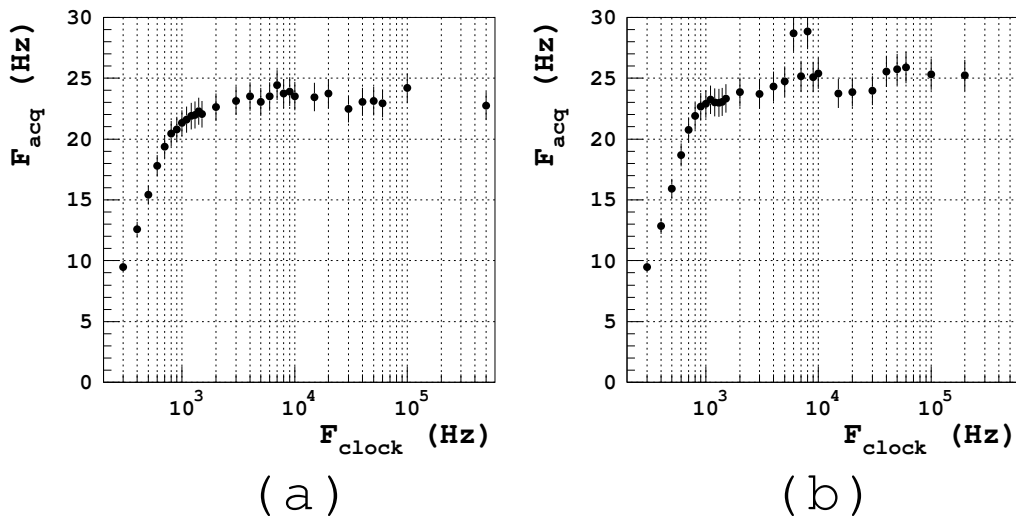


Figura 4.1: (a) Grafico della frequenza di acquisizione in funzione della frequenza di clock per la configurazione parola random. (b) Grafico della frequenza di acquisizione in funzione della frequenza di clock per la configurazione parola singola.

sa, il tempo di scrittura e lettura della parola vanno a contribuire sostanzialmente al tempo complessivo che intercorre fra un ciclo di scrittura/lettura e l'altro. Differentemente ad alte frequenze di clock il tempo impiegato per la scrittura/lettura della parola diventa trascurabile rispetto al tempo complessivo fra cicli diversi, quindi la frequenza di acquisizione non risente più delle variazioni del clock, ma dipende solamente dalla frequenza dei cicli di scrittura e lettura, i quali dipendono dalla velocità del calcolatore utilizzato.

Si può notare una piccola differenza nella zona di plateau della frequenza di acquisizione fra le due modalità. In modalità parola singola la frequenza di acquisizione risulta maggiore rispetto alla modalità random di circa 10%. Questo differenza va attribuita al fatto che in modalità random in fase di scrittura si deve generare la parola da scrivere, operazione che richiede un determinato tempo.

#### 4.1.2 Test senza simulazione

Lo stesso test descritto per la configurazione con la simulazione è stato effettuato anche con il sistema di acquisizione sprovvisto della simulazione. Si è voluto effettuare questo test perchè il sistema di acquisizione verrà nella realtà utilizzato senza simulazione, quindi tutta la parte relativa ai cicli di scrittura non contribuirà più al tempo di acquisizione.



Come descritto nel capitolo precedente il sistema senza simulazione scrive una volta sola una parola nel registro a scorrimento, quindi il registro viene posto in configurazione circolare (uscita collegata all'ingresso). A seguito della prima scrittura inizia l'acquisizione e la misura della frequenza di acquisizione viene eseguita esattamente come nel test precedente. Per questo test usiamo la sola modalità di singola parola, in quanto con una parola random non avrebbe nessun senso, visto che questa viene scritta una volta sola all'inizio. I risultati ottenuti sono mostrati in Fig. 4.2 ,

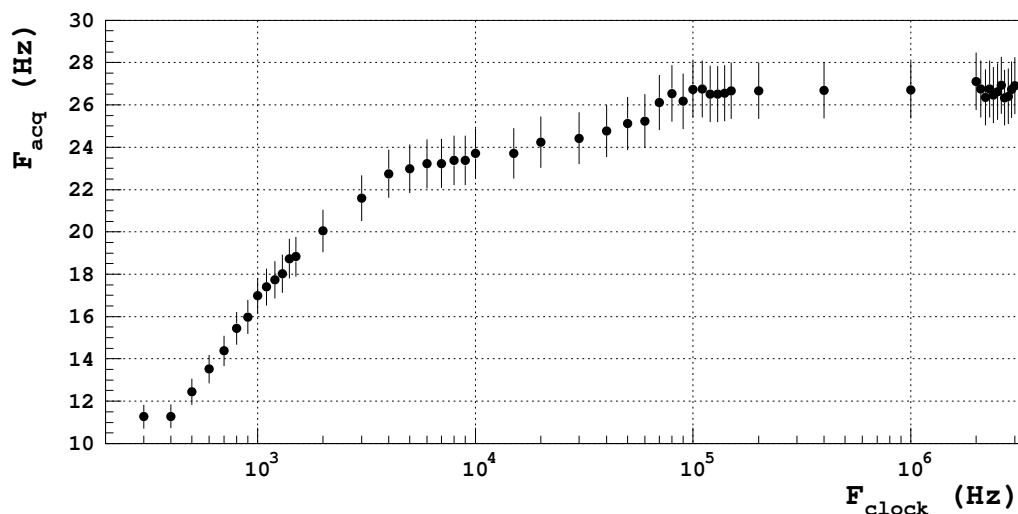


Figura 4.2: Grafico della la frequenza di acquisizione in funzione della frequenza di clock.

nella quale riportiamo la frequenza di acquisizione in funzione della frequenza di clock. Come ci si aspetta l'andamento è del tutto simile a quello ottenuto dai precedenti test e le frequenza di plateau risulta leggermente più alta rispetto ad entrambe le configurazioni con la simulazione. Questo incremento è ovviamente dovuto al fatto che non si deve più eseguire la scrittura della parola. L'incremento non è comunque così notevole in quanto il valore massimo della frequenza di acquisizione dipende fortemente dalla velocità dei cicli di LabView e quindi dalla velocità del calcolatore.

## 4.2 Errori di acquisizione

### 4.2.1 Test con simulazione

Per valutazione degli errori di acquisizione si intende dare una stima della frazione di parole che non sono lette correttamente (sbagliate). Con l'utilizzo della simulazione si può notare, in particolar modo ad alte frequenze di clock, che il sistema di scrittura esegue più cicli del sistema di lettura. Questo difetto è da attribuire alla configurazione hardware adottata per controllare il clock in ingresso al registro e in generale a tutta la configurazione del circuito esterno [5]. In particolar modo è da notare che per le connessioni fra scheda e circuito sono stati utilizzati cavi non schermati e di lunghezze arbitrarie. Questi accorgimenti non presi, inevitabilmente introducono rumori nel circuito, motivo per cui a volte il registro non vede il fronte di discesa del fine scrittura, quindi una seconda scrittura viene immediatamente eseguita.

Per i motivi appena esposti con questo test valutiamo la frazione di parole sbagliate rispetto ad un campione di 2000 parole effettivamente lette, quindi valuteremo anche la differenza fra il numero di cicli di scrittura e lettura.

Questi test sono stati effettuati, come in precedenza, a diversi valori della frequenza di clock e nelle due modalità di simulazione.

Nella Fig. 4.3 riportiamo la frazione di parole sbagliate (a,b) e la differenza percentuale (c,d) fra il numero di cicli scrittura/lettura in funzione della frequenza di clock ottenute dalle due modalità di simulazione.

Per quanto riguarda la frazione di parole sbagliate possiamo notare in entrambe le modalità di simulazione che questo valore aumenta a frequenze elevate. Nella modalità singola gli errori sono praticamente pari a 0% con un massimo inferiore al 0.5% in corrispondenza alla frequenza più alta di 30 kHz. Diversamente nella modalità random l'errore rimane attorno allo 0.25% per crescere fino ad un massimo inferiore al 2% a 500 kHz. Nella modalità random si può notare che la frazione di errori è, anche se di poco, superiore alla frazione della modalità singola. Questo fatto deriva dalla temporanea perdita di sincronizzazione che si verifica fra la memorizzazione della parola random nella variabile globale di LabView e il confronto di questa variabile con la parola letta dalla scheda. In realtà le parole random generate vengono memorizzate in un buffer di 5 parole per ridurre l'effetto di questa temporanea perdita di sincronizzazione. In particolare può avvenire che si esegua il confronto della parola letta con la variabile globale, quando questa o non è ancora stata aggiornata o è già stata aggiornata più di 5

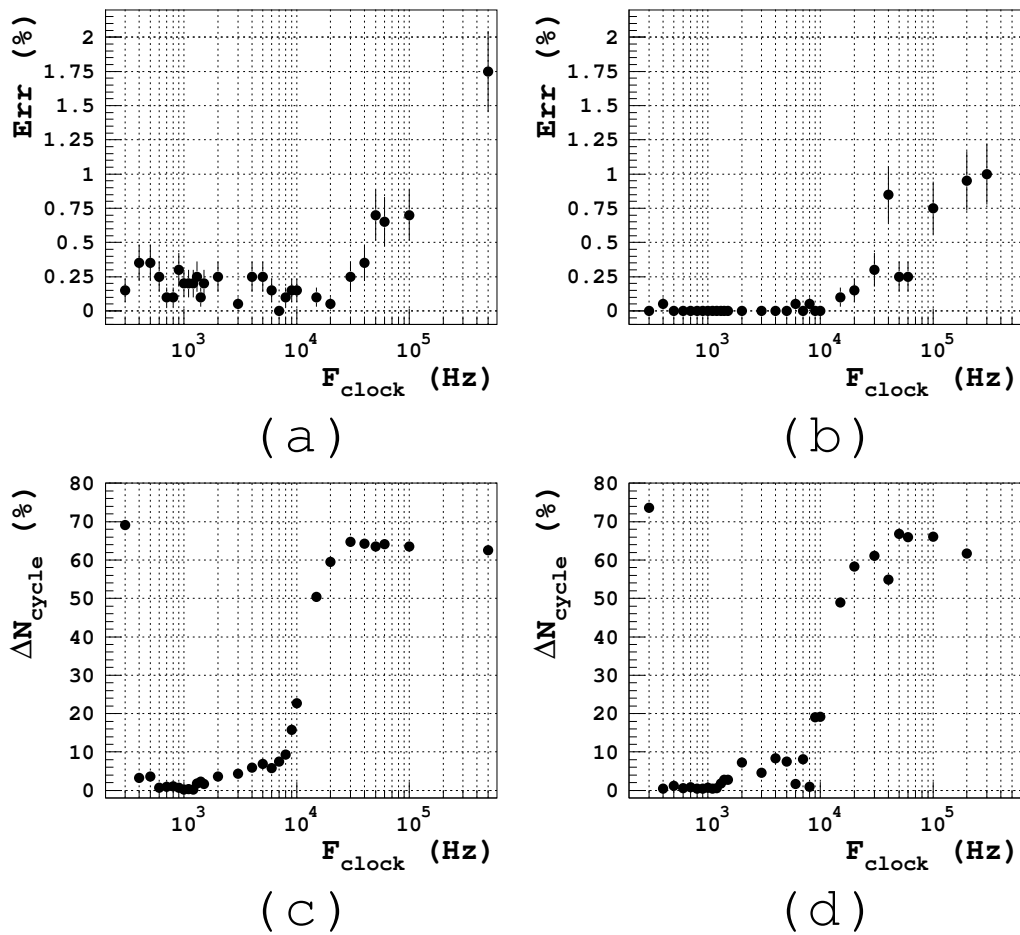


Figura 4.3: (a) Errori % in modalità a parole random (b) Errori % in modalità a parole singola (c) Differenza % di cicli in modalità a parole random (d) Differenza % cicli in modalità a parole singola

volte e quindi perde la parola vera. Tutte queste fonti di errore fanno sì che la valutazione della frazione di errori nella modalità random sia in realtà una valutazione globale di errori introdotti dal sistema di simulazione, sistema di circuito elettronico e sistema di lettura. In generale si può concludere comunque che la frazione di errori è limitata e quindi il sistema risulta affidabile per le operazioni di acquisizione che dovrà eseguire.

Per quanto riguarda la differenza percentuale fra il numero di cicli di scrittura e lettura, si può notare dai grafici in Fig. 4.3 che tale differenza subisce un brusco aumento in corrispondenza di frequenze di circa 20 kHz in entrambe le modalità. Questo comportamento può essere associato ai vari componenti del circuito che

sono configurati in un modo tale da permettere il passaggio di eventuali rumori di alta frequenza.

#### 4.2.2 Test senza simulazione

La valutazione degli errori di lettura nella configurazione del sistema di acquisizione senza simulazione viene eseguita in modo diverso da come si sono eseguiti i test precedenti.

Anzitutto il controllo delle parole scritte viene effettuato *offline* dopo l'acquisizione di campioni di 2000 parole. In questa fase di lettura si possono verificare due tipi di errori. In primo luogo il vero errore di lettura che si verifica quando la parola letta è effettivamente diversa dalla parola memorizzata nel registro. In secondo luogo rumori sulla linea del clock possono causare fronti di clock accidentali in concomitanza dell'inizio della lettura, causando uno shift di un bit della parola. In seguito a questo evento la parola letta risulta essere shiftata di un bit rispetto alla parola di riferimento, perchè al momento della lettura la parola registrata era effettivamente shiftata. Quindi definiamo **errori di lettura** tutte le volte che incontriamo una parola diversa dalla parola di riferimento o dalla parola di riferimento shiftata, mentre definiamo **errori di clock** ogni volta che si verifica uno shift della parola.

Entrambi questi errori si potranno verificare anche nella realtà quando useremo la FEC.

In Fig. 4.4 riportiamo la percentuale di errori di clock calcolata su campioni di 2000 acquisizioni per diversi valori della frequenza di acquisizione da 300 Hz a 3 MHz. Non riportiamo nessun grafico relativo agli errori di lettura in quanto dai test si è riscontrato che le letture sbagliate avvengono solo oltre frequenze di clock di 2.8 MHz e comunque inferiore allo 0.5%. Differentemente oltre i 3 MHz si verifica l'azzeramento del registro, quindi oltre questo valore non si può utilizzare il circuito realizzato.

Dalla Fig. 4.4 si può dedurre che il circuito presenta qualche sorta di filtri passa banda, in quanto notiamo un notevole aumento, fino al 50%, degli errori di clock nell'intervallo di frequenza fra 10 kHz e 60 kHz. Nell'intervallo di frequenze fra 2 kHz e 10 kHz invece la percentuale di errore si riduce al di sotto del 10%, rimanendo comunque oltre il 5%. Al di fuori di questi intervalli critici la percentuale di errore rimane sempre al di sotto dell'1%.

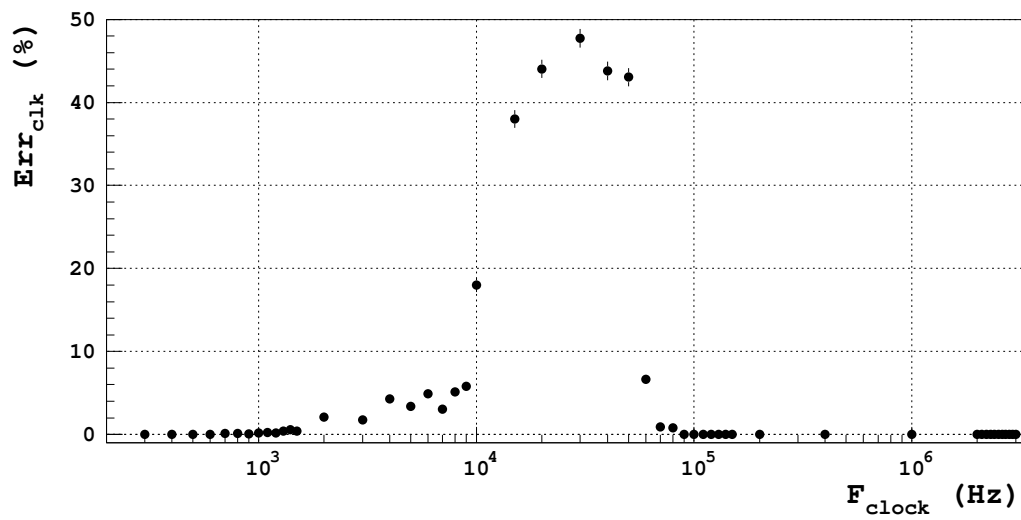


Figura 4.4: Grafico della percentuale di errori di clock calcolata su campioni di 2000 acquisizioni.



# Capitolo 5

## Conclusioni

La fase iniziale di questa tesi si è concentrata sullo studio delle funzionalità della scheda di acquisizione DIO-6533. Ciò ha permesso l'individuazione delle principali funzioni soddisfacenti i requisiti richiesti per lo sviluppo del sistema di acquisizione dati, obiettivo finale del lavoro qui presentato.

Contemporaneamente allo sviluppo dell'applicazione software di acquisizione, al fine di verificarne il funzionamento e le prestazioni, si è reso necessario realizzare due prototipi che simulassero la scheda FEC. Per ognuno di questi prototipi è stata quindi messa a punto un'adeguata versione del sistema di acquisizione.

Dai test condotti sulle diverse versioni del sistema di acquisizione possiamo concludere che, eseguita l'applicazione su un calcolatore da 850 MHz, questa è in grado di raggiungere una frequenza massima di acquisizione di 25 Hz, adeguata agli studi svolti nel laboratorio. Da successivi test volti alla valutazione degli errori in lettura si è dimostrato che, nella versione con simulazione di scrittura, la frazione di errori è inferiore all'1% nell'intervallo di frequenze di clock 300 Hz - 0.1 MHz, mentre oltre questa frequenza massima il sistema assumeva un comportamento non controllabile. Gli stessi test effettuati con la versione senza simulazione hanno dimostrato che la frazione di errori risulta essere inferiore all'1% negli intervalli di frequenza di clock 300 Hz - 2 kHz e 0.1 MHz - 2.8 MHz. I differenti risultati dei test possono essere attribuiti sia alle diversità dei circuiti utilizzati per ciascuna versione, sia alle differenze fra le versioni stesse.

Nonostante i circuiti utilizzati siano ben lontani da una configurazione ottimale, il sistema di acquisizione è in grado di funzionare a frequenze di clock dell'ordine di 1 kHz, dando buoni risultati in termini di frequenza di acquisizione e di errori di lettura.

I futuri sviluppi di questo progetto sono tesi al miglioramento dei risultati

fin qui ottenuti, utilizzando un calcolatore più veloce, la reale scheda FEC e più adeguati accorgimenti all'intero sistema di acquisizione, quali, per esempio, l'uso di cavi coassiali per la riduzione dei rumori nel trasferimento dei dati.



# Appendice A

## Dettagli della scheda di simulazione

La scheda di simulazione della FEC così come viene presentata nel capitolo 3 risulta non funzionare correttamente, in quanto nella realtà si verificano rumori che corrompono il segnale di clock entrante nei registri e nel flip-flop di tipo D.

Anche la logica di trigger risulta essere soggetta a disturbi che ne provocano il malfunzionamento. Sebbene non si sia raggiunta una situazione ideale, totalmente libera da rumori, con l'inserimento di alcuni condensatori sulle linee di clock e agli ingressi di certe porte si è risolto il problema. I condensatori funzionano come filtri, e annullano una parte rilevante dei rumori che disturbano il funzionamento della scheda di simulazione della FEC. In Fig.A.1 è riportato lo schema del circuito di Fig.3.4 descritto nel paragrafo 3.3.1 con l'aggiunta delle modifiche qui descritte. I condensatori posti sulla logica di trigger sono da 100 pF, mentre il condensatore posto sul clock di ingresso degli shift-register e del flip-flop ha capacità di 1 nF.

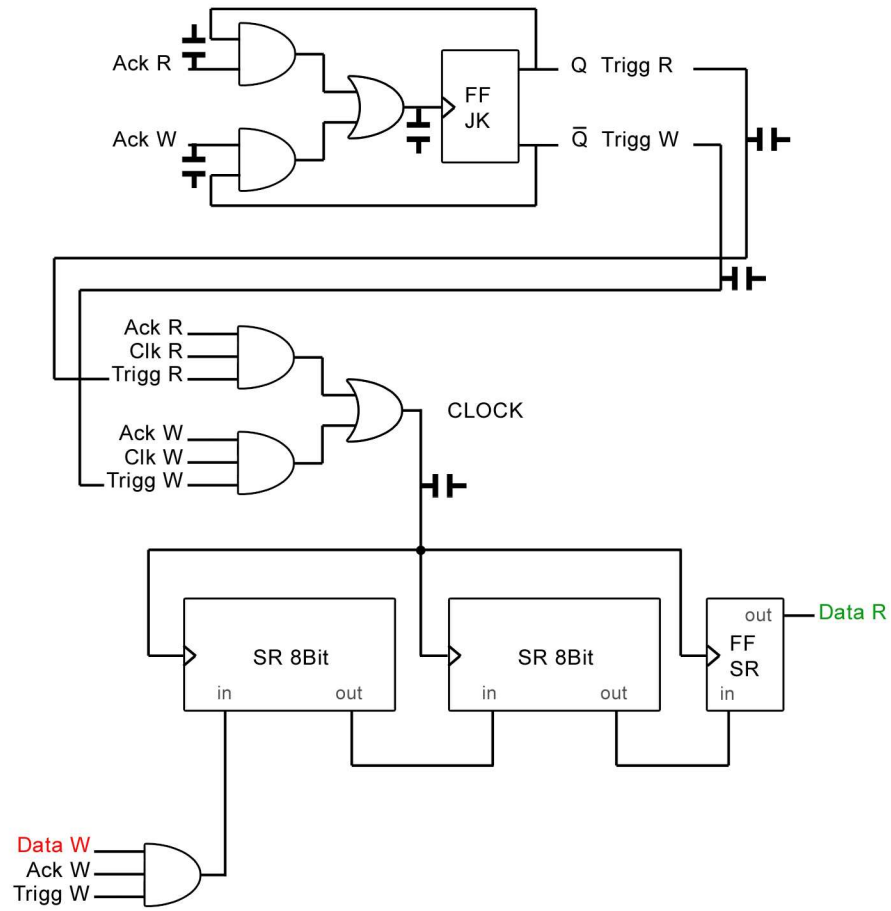


Figura A.1: Schema della scheda di simulazione della FEC

# Bibliografia

- [1] Particle Data Group, S. Eidelman *et al.*, Phys. Lett. B **592**, 1 (2004).
- [2] G. Battistoni *et al.*, *Operation Of Limited Streamer Tubes*, Nucl. Instrum. Meth. **164** (1979) 57.  
G. Battistoni *et al.* *Resistive Chatode Detectors With Bidimensional Strip Readout: Tubes and Drift*, Nucl. Instrum. Meth. **176** (1980) 297.  
E. Iarocci *et al.*, Nucl. Instrum. Meth. **217** (1983) 30.  
G. Battistoni *et al.*, *Influence of Gas Mixture and Cathode Material on Limited Streamer Operation*, Nucl. Instrum. Meth. **217** (1983) 433.  
M. Caria *et al.*, *Large Series test of Limited Streamer Tubes*, CERN-EP/87-76, 10 April 1987.
- [3] The LST Team, *A Barrel IFR Instrumented with Limited Streamer Tubes*, (Proposal from the BaBar Collaboration to the SLAC Experimental Program Advisory Committee), 2003.  
<http://www.slac.stanford.edu/BFROOT/www/Detector/LST/LSTprop-A4-May30-ajss.pdf>  
[http://www.fe.infn.it/electron/babar\\_ifr.htm](http://www.fe.infn.it/electron/babar_ifr.htm)  
G. Cibinetto *et al.*; “*A Barrel IFR instrumented with limited streamer tubes for BaBar experiment*”, Int. J Mod. Phys. A20 3834-3838, 2005.
- [4] Manuale della scheda DIO-6533 - National Instruments  
Manuale delle librerie DAQmx - National Instruments
- [5] William R. Leo, “*Techniques for Nuclear and Particle Physics Experiments*”, Springer-Verlag, 1993.