



**TDC e sistema di acquisizione per il rivelatore HET di KLOE-2:  
Manuale utente**

L. Iafolla<sup>1)</sup>, A. Balla<sup>1)</sup>, M. Beretta<sup>1)</sup>, P. Ciambrone<sup>1)</sup>, F. Gonnella<sup>2),3)</sup>,  
M. Mascolo<sup>2),3)</sup>, R. Messi<sup>2),3)</sup>, D. Moricciani<sup>3)</sup>

<sup>1)</sup> *INFN-Laboratori Nazionali di Frascati, via E. Fermi 40, 00044 Frascati, Italy*

<sup>2)</sup> *Università Tor Vergata, Italy*

<sup>3)</sup> *INFN Roma-2*

**Abstract**

L'upgrade di KLOE (KLOE-2) prevedeva l'installazione di un sistema di "tagging" per lo studio della fisica  $\gamma\gamma$ , costituito da 2 nuove coppie di rivelatori: HET e LET. Il sistema di acquisizione (DAQ) del rivelatore HET è stato implementato su una scheda VME basata su una FPGA (Virtex-5). Esso consiste principalmente di un TDC, un sistema di elaborazione dati (Zero suppression) e una interfaccia VME. Nella prima parte di questo documento sono descritti i principi di funzionamento e l'architettura del TDC e del DAQ; nella seconda parte sono descritte le modalità di configurazione del sistema e la formattazione dei dati in uscita

Versione 1.0 – 18 maggio 2012

1	Funzionamento del DAQ per il rivelatore HET di KLOE2 .....	3
1.1	Il TDC del sistema di acquisizione .....	3
1.2	Tecnica di misura del TDC .....	4
1.2.1	Interpolazione lineare .....	4
1.2.2	4xOversampling .....	5
1.3	Test di caratterizzazione del TDC.....	5
1.3.1	Errori stocastici: risoluzione e precisione.....	6
1.3.2	Errori sistematici .....	6
1.4	Selezione e registrazione dei dati.....	8
1.4.1	Sistema di selezione dati .....	8
1.4.2	Gestione dei trigger T1 e T2.....	9
1.5	Il readout del sistema di acquisizione .....	10
2	Modalità di test, controllo e stato di funzionamento del DAQ .....	11
2.1	Reset di sistema .....	11
2.2	Modalità di test .....	11
2.3	Controllo del sistema di selezione dati .....	11
2.4	Impostazione dell'offset del TDC.....	12
2.5	Stato di riempimento della fifo ed elaborazione dei trigger.....	12
2.6	Frequenza di conteggio dei singoli canali.....	13
2.7	Stato di errore e di busy .....	13
3	Piedinatura della scheda .....	14
4	Indirizzo e funzione dei registri.....	15
4.1	Funzione dei singoli bit dei registri.....	15
4.2	Formattazione dati nella fifo.....	16
4.2.1	Formattazione delle singole parole nella fifo .....	17

# 1 Funzionamento del DAQ per il rivelatore HET di KLOE2

In questo documento saranno descritte le specifiche e le caratteristiche del sistema di acquisizione del rivelatore HET. Maggiori informazioni in merito al rivelatore e all'esperimento sono reperibili nei documenti citati in bibliografia.

Il DAQ sviluppato per il rivelatore di particelle HET soddisfa i seguenti requisiti:

1. associare gli impulsi rivelati alle intersezione tra bunch da cui hanno avuto origine;
2. selezionare l'insieme di dati corrispondenti agli eventi fisici individuati dalle transizioni dei trigger T1 e T2;
3. trasferire i dati al database di KLOE2.

Nei paragrafi che seguono, saranno argomentati i punti precedenti.

## 1.1 Il TDC del sistema di acquisizione

Per soddisfare il primo requisito del DAQ si può utilizzare un TDC "Time to Digital Converter". Si possono fare, infatti, le due seguenti ipotesi:

1. il tempo tra la rivelazione dell'impulso e l'intersezione fra bunch da cui ha avuto origine è costante;
2. il tempo tra l'intersezione fra bunch e il segnale fiducial di DAΦNE è costante.

Di conseguenza, se si conosce il tempo tra la transizione positiva del fiducial e l'impulso, è possibile risalire all'intersezione tra bunch che ha dato luogo all'impulso stesso.

Il tempo tra la transizione positiva del fiducial e l'impulso può essere misurato con un TDC; la risoluzione di questo TDC deve essere almeno 2,7 ns che è il tempo minimo tra due intersezioni fra bunch. Il range minimo invece è il tempo tra due transizioni positive del fiducial (324 ns).

La distribuzione delle misure aspettata è la somma di 100 gaussiane (una per ogni bunch pieno) i cui valori medi sono posti a intervalli di 2.7 ns (vedi Figura 1). I 20 bunch vuoti non produrranno alcun segnale; assumiamo che il primo bunch pieno, dopo quelli vuoti, è il bunch uno (vedi Figura 2).

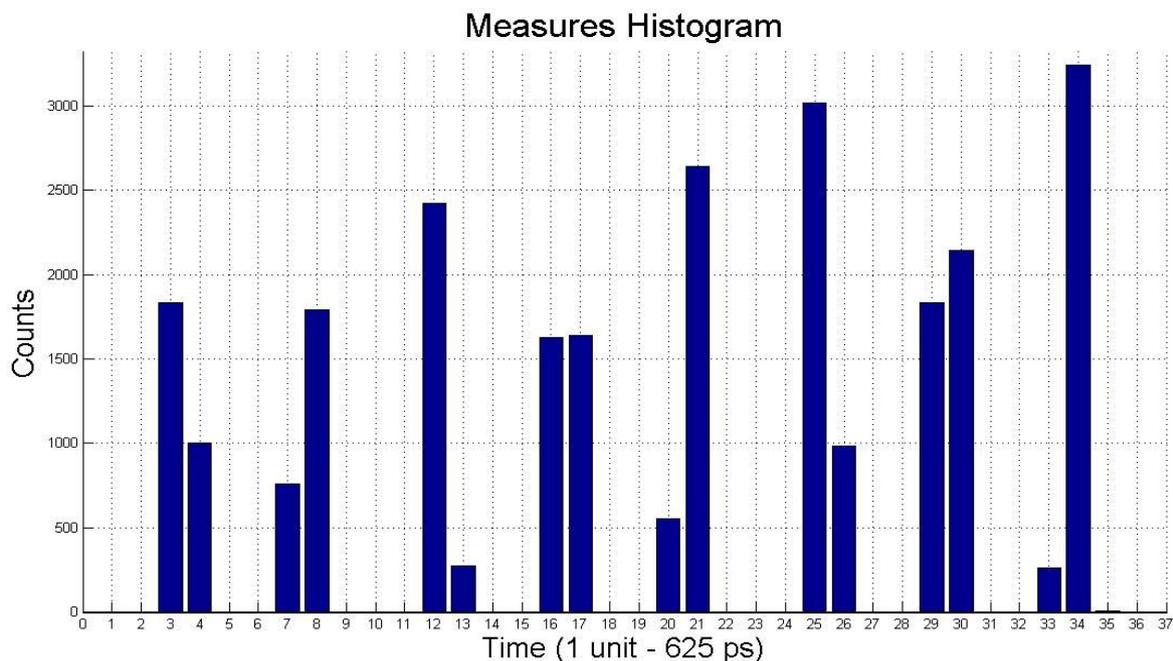


Figura 1. Le misure qui rappresentate sono state ottenute con un test eseguito su una evaluation board. La temporizzazione degli impulsi in ingresso al TDC emulava quella dei segnali che produrrà il rivelatore HET in acquisizione su DAΦNE. Si osservi come si distribuiscono le misure del TDC: i valori medi delle singole distribuzioni sono distanziati di 2.7ns mentre la deviazione standard è sufficientemente piccola da riuscire distinguere quelle relative a bunch diversi. Le distribuzioni sono binomiali, invece che gaussiane, a causa della quantizzazione del TDC.

In generale il tempo misurato per il primo bunch non sarà il più piccolo: per questo motivo l'offset del TDC deve poter essere cambiato in modo che ciò si verifichi (vedi il secondo istogramma di Figura 2 e Figura 12).

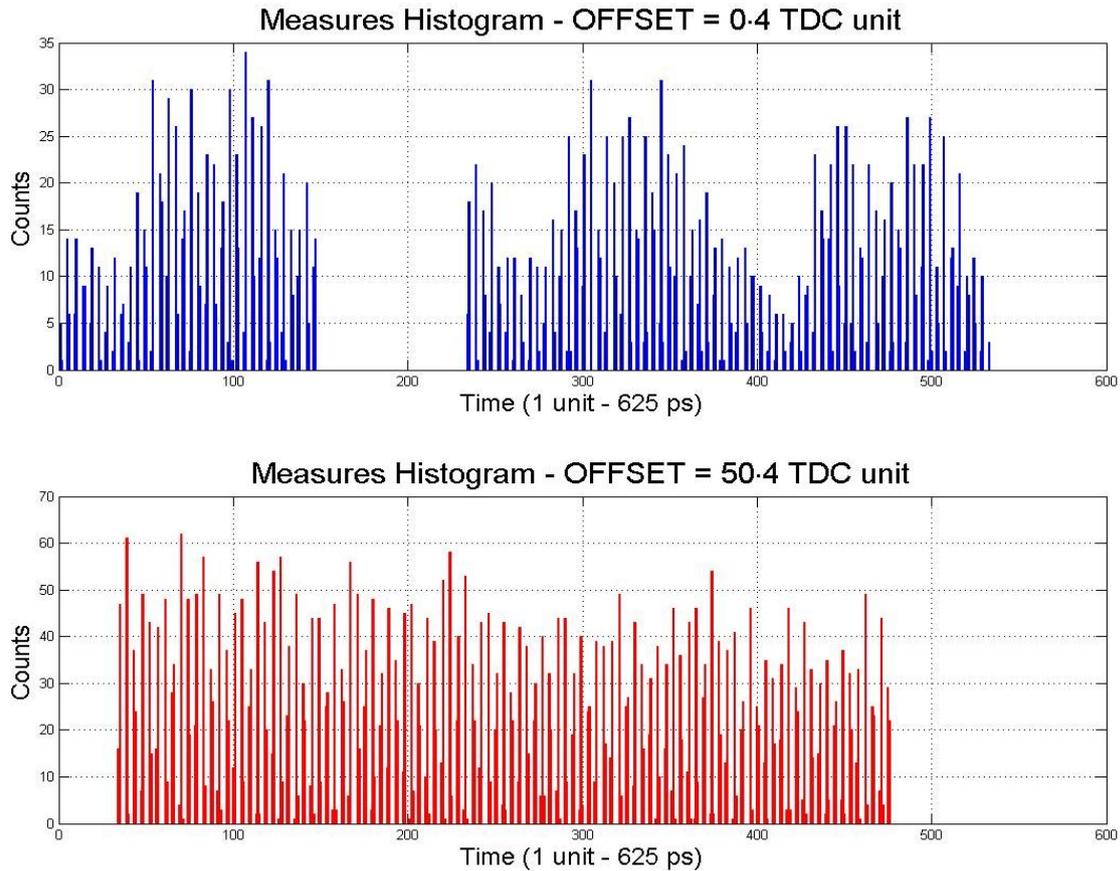


Figura 2. In questo test la temporizzazione degli impulsi in ingresso al TDC emulava la temporizzazione dei segnali che produrrà il rivelatore HET in acquisizione su DAΦNE. Nel primo istogramma è evidente l'intervallo di tempo relativo ai bunch vuoti di DAΦNE: il primo bunch dopo questo intervallo è il bunch uno. Modificando l'offset del TDC si può traslare il primo bunch, e di conseguenza tutti gli altri, in modo che sia il primo dopo l'istante zero.

## 1.2 Tecnica di misura del TDC

In questo paragrafo saranno illustrate brevemente la tecnica dell'*interpolazione lineare* e dell' *4xOversampling*. Le due tecniche sono complementari e servono, rispettivamente, per aumentare la dinamica del TDC e per migliorare la risoluzione.

### 1.2.1 Interpolazione lineare

Definiamo la dinamica  $D$  di un TDC con la seguente relazione:

$$D = \frac{T}{\Delta t} \tag{0.1}$$

in cui  $T$  è il tempo massimo misurabile, mentre  $\Delta t$  è la risoluzione. Le tecniche di misura più precise generalmente hanno una dinamica molto limitata, per cui il tempo massimo  $T$  è insufficiente.

Con il metodo di misura *Nutt* (o anche detto dell'*interpolazione lineare*) si possono realizzare TDC con una dinamica molto grande. Esso consiste nell'eseguire tre misure, di cui, una approssimativa e due molto precise; tutte e tre le misure sono eseguite in riferimento ai fronti positivi del clock del sistema. In Figura 3 è illustrato il principio di funzionamento di questa tecnica. La misura approssimativa  $\Delta t_{12}$  è il conteggio di cicli di clock tra il fronte positivo del clock che viene immediatamente dopo l'impulso di *start* e il fronte positivo del clock che viene immediatamente dopo lo *stop*. Questa misura è affetta dai due errori indicati in Figura 3 con  $\Delta t_1$  e  $\Delta t_2$ .  $\Delta t_1$  deve essere sommato, mentre  $\Delta t_2$  deve essere sottratto per cui:

$$t = \Delta t_1 + \Delta t_{12} - \Delta t_2. \tag{0.2}$$

$t$  rappresenta la misura finale del tempo. Le misure di  $\Delta t_1$  e  $\Delta t_2$  possono essere eseguite con differenti tecniche e la loro risoluzione determina la risoluzione totale del metodo (l'errore totale è la somma degli errori relativi alle misure di  $\Delta t_1$  e  $\Delta t_2$ ).

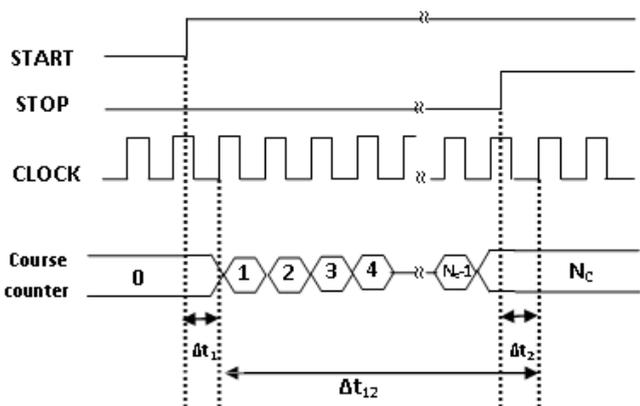


Figura 3. In questo diagramma è rappresentato il principio del metodo di misura di Nutt. La misura definitiva è data dalla combinazione di tre misure di cui una approssimativa e due molto precise.

### 1.2.2 4xOversampling

Le due misure di precisione di cui si è fatto menzione nel paragrafo precedente possono essere eseguite con la tecnica dell'4xOversampling.

La FPGA utilizzata (Virtex5) ha dei dispositivi (DCM, Digital Clock Manager) incorporati in grado di generare fino a quattro segnali di clock con stessa frequenza, ma con relazioni di fase diverse.

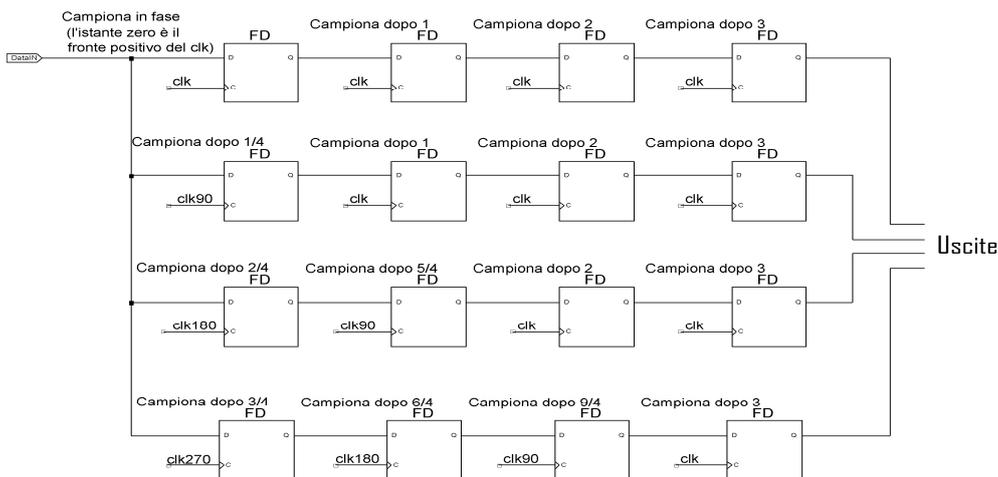


Figura 4. Circuito di sincronizzazione con il clock dei quattro campionamenti eseguiti in corrispondenza delle quattro fasi.

La tecnica dell'4xOversampling consiste nel campionare il segnale con quattro flip-flop sincronizzati ognuno con uno dei clock generati dai DCM: in questo modo la risoluzione temporale migliora di quattro volte rispetto a quella che si otterrebbe campionando il segnale solo in corrispondenza dei fronti positivi di un unico segnale di clock. I quattro campioni così ottenuti sono sincronizzati a quattro clock differenti: per sincronizzarli con un unico clock di sistema si può ricorrere al circuito in Figura 4.

### 1.3 Test di caratterizzazione del TDC

La tecnica di implementazione del TDC introdotta nei paragrafi precedenti è stata già caratterizzata utilizzando una FPGA montata su una *evaluation board* (ML505). Gli stessi test sono stati eseguiti anche utilizzando la scheda definitiva. I quattro parametri fondamentali di caratterizzazione del TDC sono:

1. risoluzione;
2. precisione;
3. linearità integrale;
4. linearità differenziale.

### 1.3.1 Errori stocastici: risoluzione e precisione

La *risoluzione del TDC* è il più piccolo intervallo di tempo che può essere risolto con una singola misura. Con la tecnica di 4xOversampling essa vale  $1/4T_{clk}$ : il  $T_{clk}$  utilizzato vale 2.5 ns perciò la risoluzione è 0.625 ns. L'errore associato alle misure singole può essere valutato osservando le distribuzioni rappresentate in Figura 1: visto che ci aspettiamo distribuzioni larghe 2 unità associeremo alla misura un errore di  $\pm 1$  unità, ovvero  $\pm 0.625$  ns.

La *precisione del TDC* è il valore della deviazione standard per le misure ripetute. Esso può essere stimato con la somma in quadratura di due contributi:

$$\sigma_{TDC} = \sqrt{\sigma_Q^2 + \sigma_R^2},$$

in cui  $\sigma_{TDC}$  è la precisione,  $\sigma_Q$  è il contributo dovuto al solo errore di quantizzazione mentre  $\sigma_R$  è il contributo intrinseco del TDC dovuto alla somma in quadratura di altri contributi (jitter del clock, effetti della non-linearità differenziale, effetti di temperatura, etc.). Il contributo dovuto alla quantizzazione può essere stimato teoricamente e, in unità di TDC, vale:

$$\sigma_Q = \sqrt{c(1-c)};$$

in cui  $c$  è la parte decimale del tempo misurato o, con ottima approssimazione, la parte decimale del valore medio delle misure ripetute.

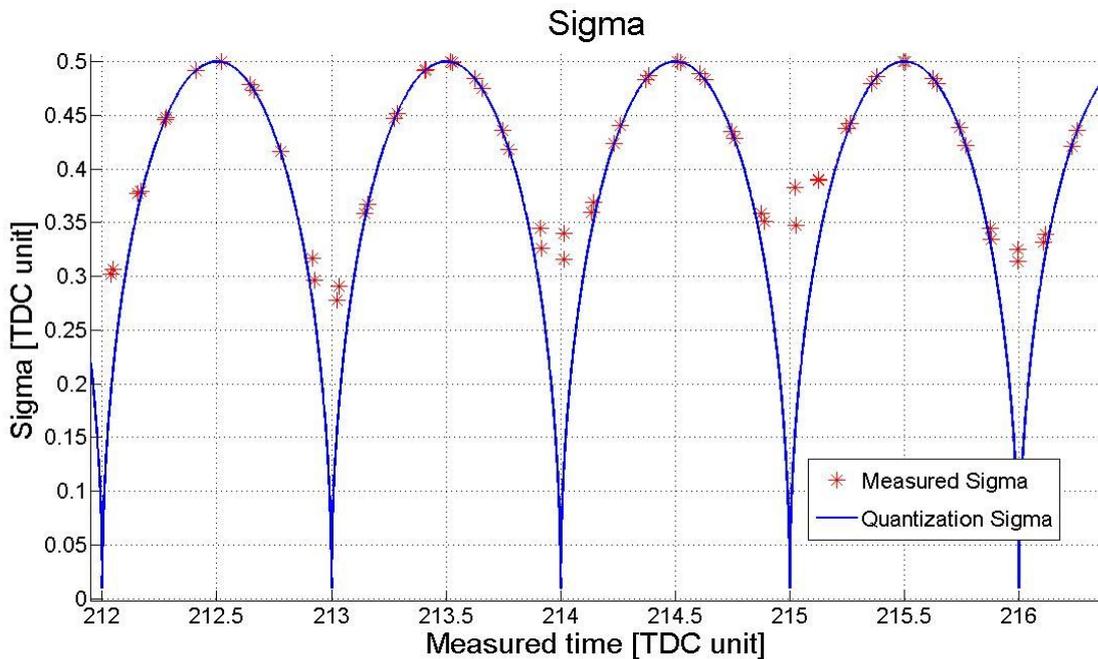


Figura 5. In questo grafico sono rappresentati i valori delle sigma misurate e delle sigma dovute alla quantizzazione in funzione del valore del tempo misurato. L'intervallo delle misure (212-216) è stato scelto in modo del tutto casuale.

In Figura 5 sono rappresentati  $\sigma_{TDC}$  e  $\sigma_Q$ : nei punti in cui  $\sigma_Q$  è trascurabile è più evidente il contributo di  $\sigma_R$ . Nonostante non sia stato ancora elaborato un modello per  $\sigma_R$  esso è sufficientemente piccolo da poter essere trascurato.

### 1.3.2 Errori sistematici

Gli errori sistematici sono di due tipi differenti: *non-linearità integrale* (INL) e *non-linearità differenziale* (DNL).

La *non-linearità integrale* è la deviazione della caratteristica di ingresso-uscita rispetto alla linea retta. Una stima della non-linearità integrale può essere ottenuta nel seguente modo: si eseguono misure ripetute di diversi valori di tempo e se ne calcolano i valori medi; in questo modo gli errori stocastici possono essere trascurati. A questo punto si rappresentano sulle ordinate di un grafico i valori medi ottenuti e sulle ascisse i valori reali dei tempi misurati (vedi il primo grafico di Figura 6). Se il TDC fosse ideale i punti si collocherebbero perfettamente su una retta (caratteristica ingresso-uscita); i parametri della retta ideale possono essere ottenuti interpolando linearmente i punti ottenuti. La stima della INL è ottenuta facendo la differenza tra i punti ottenuti e la retta dell'interpolazione (vedi il secondo grafico di Figura 6). Assumiamo  $\max\{INL\}$  come valore di riferimento della INL: dal secondo grafico di Figura 6 si evince che per il sistema testato  $\max\{INL\}$  vale circa 0.05 ns.

La *non-linearità differenziale* (DNL) è la deviazione di un singolo passo di quantizzazione dal valore ideale del bit meno significativo. In altre parole, a causa della DNL, l'intervallo di tempo associato ad una variazione del bit meno significativo non è sempre lo stesso. Con la tecnica dell'4xOversampling, la DNL può essere causata da uno skew del segnale di ingresso del circuito di Figura 4.

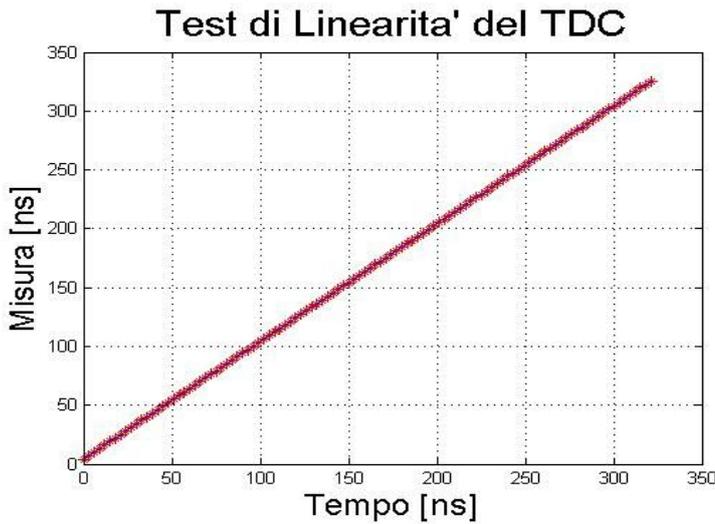
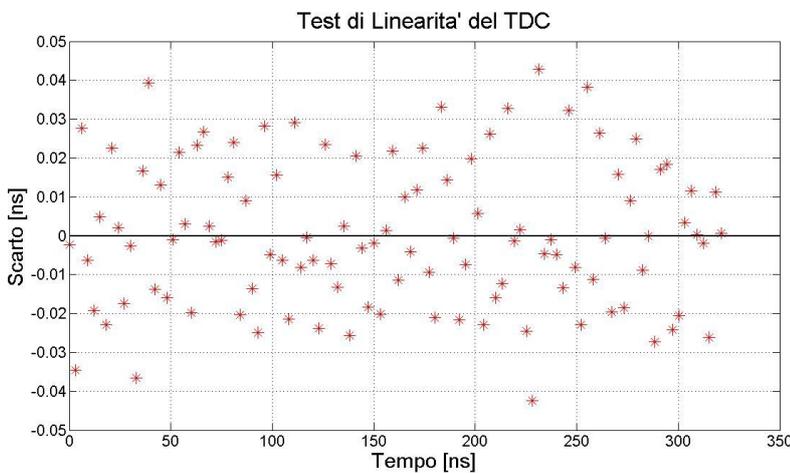


Figura 6. Nel grafico in alto sono rappresentate le medie delle misure ottenute rispetto al valore reale misurato. Nel grafico in basso sono rappresentati gli scarti delle medie rispetto alla retta ideale. Il valore di riferimento della non-linearità integrale è  $INL_{max}$ , ovvero la deviazione massima ( $<0.05$  ns).



La misura di questo errore sistematico può essere svolta nel seguente modo: si eseguono una serie di misure di tempi diversi che incrementano di una quantità molto più piccola del valore ideale del bit meno significativo (*sweep*); per un TDC ideale ci aspettiamo una distribuzione uniforme delle misure ottenute rispetto al tempo (vedi Figura 7). La deviazione della distribuzione ottenuta dalla distribuzione uniforme ci fornisce una misura della non-linearità differenziale.

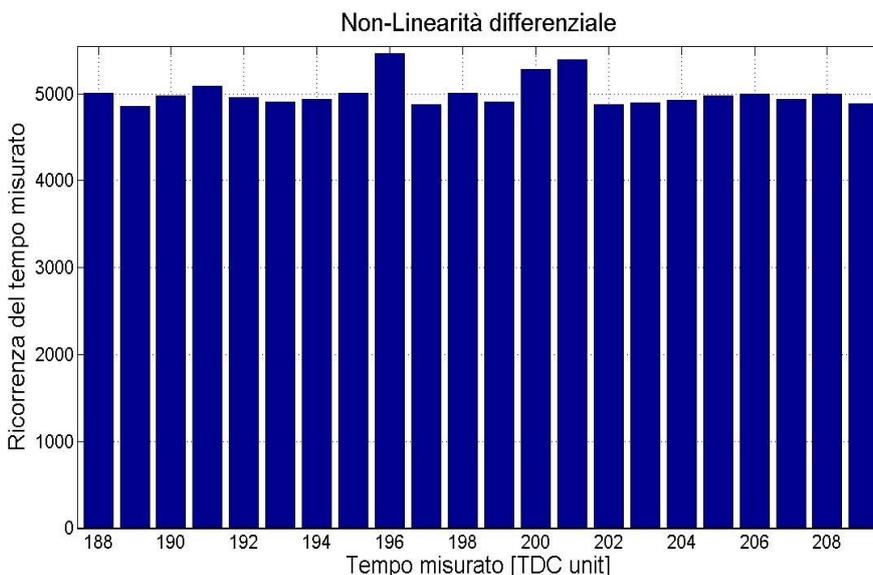


Figura 7. Istogramma delle misure di uno *sweep* con incremento di 80 ps.

La misura della DNL deve essere eseguita con opportuni accorgimenti: se ad ogni incremento del tempo in ingresso non corrispondono sempre lo stesso numero di misure troveremo un effetto del tutto analogo alla non-linearità differenziale; per que-

sto motivo il TDC ed il generatore di impulsi sono stati sincronizzati. Prima di eseguire ogni misura il TDC asserisce un bit, “*inc\_req*”, che determina un incremento in tempo dell’impulsore. Anche la generazione di uno sweep uniforme è di fondamentale importanza per eseguire correttamente la misura: di questo se ne discute nel manuale dell’impulsore.

## 1.4 Selezione e registrazione dei dati

La quantità di dati acquisiti dal rivelatore è molto grande; anche la sola trasmissione di tutti questi dati sarebbe complicata oltre che inutile. A questo scopo sono stati escogitati dei sistemi di selezione e registrazione dei dati significativi di cui si discute nei seguenti paragrafi.

### 1.4.1 Sistema di selezione dati

Il sistema di acquisizione del TDC soddisfa il secondo requisito del DAQ (selezionare l’insieme di dati corrispondenti agli eventi fisici individuati dalle transizioni dei trigger T1 e T2).

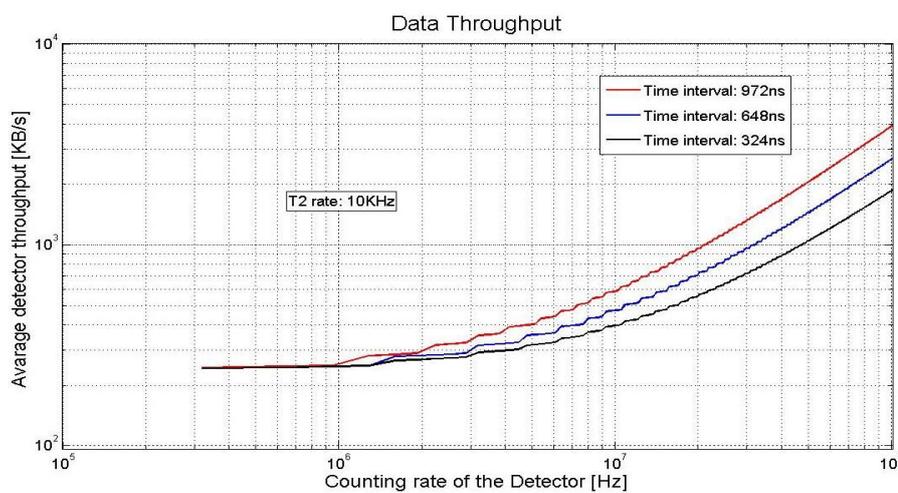


Figura 8. Flusso dati in uscita dal DAQ (throughput) al variare della frequenza di conteggio del rivelatore (numero di impulsi per unità di tempo). Il flusso dipende anche dalla frequenza del T2 e dalla lunghezza dell’intervallo di tempo che si sceglie di registrare. La frequenza di conteggio prevista è inferiore a 10 MHz mentre quella del T2 è inferiore a 10 KHz: se sono valide queste ipotesi il throughput non cambia molto al variare della lunghezza dell’intervallo di tempo registrato.

Il metodo di selezione si basa sulle due seguenti considerazioni:

1. il tempo tra il trigger T1 e il relativo impulso rivelato dall’HET è costante entro un intervallo di circa 20 ns;
2. la quantità di dati in uscita non cambia molto al variare della lunghezza dell’intervallo di tempo registrato\*.

Dalle precedenti ipotesi si evince che la risoluzione del sistema di selezione dei dati può anche essere maggiore dei 20 ns senza che il throughput dei dati cresca molto (vedi Figura 8 per frequenze di conteggio minori di 10MHz). Il sistema di selezione implementato ha una risoluzione uguale al periodo del fiducial (324ns).

Il principio di funzionamento del sistema di selezione è illustrato in Figura 9. Esso può essere riassunto in tre passaggi fondamentali:

- i dati vengono registrati in una memoria a *stack* (pila);
- le transizioni positive del fiducial sono registrate come tutti gli altri eventi utilizzando una parola identificativa (*fiducial tag*) che non corrisponde a nessuna possibile misura;
- la lettura dello stack viene interrotta dopo aver contato un determinato numero di fiducial.

\* Tale considerazione, in realtà, è vera solo per frequenze di conteggio del rivelatore inferiori di ~10MHz.

L'uso di una memoria a stack (LIFO, *Last In First Out*) fa sì che i primi dati che vengono letti sono quelli relativi alle ultime misure eseguite; contando il numero di tag del fiducial è possibile selezionare solo i dati più recenti. L'uso dei tag ha anche l'importante funzione di distinguere misure relative a cicli di fiducial differenti.

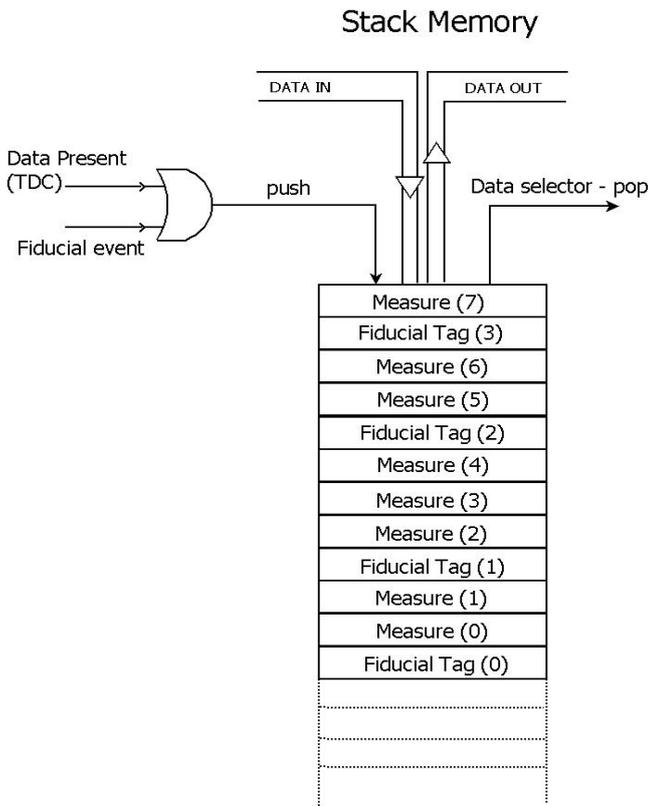


Figura 9. In questo schema è rappresentata la memoria a stack ed un esempio del suo contenuto. In fase di scrittura vengono memorizzate sia le misure che gli eventi di transizione positiva del fiducial. L'operazione di lettura è eseguita fintanto che non sono stati contati un numero prestabilito di fiducial tag.

Riassumendo, il corso delle operazioni è il seguente: durante l'acquisizione le misure ed i tag sono continuamente scritti sulla memoria a stack; la transizione del trigger T1 interrompe le operazioni di scrittura ed innesca la lettura dello stack; la scrittura è riavviata una volta finite le operazioni di lettura.

### 1.4.2 Gestione dei trigger T1 e T2

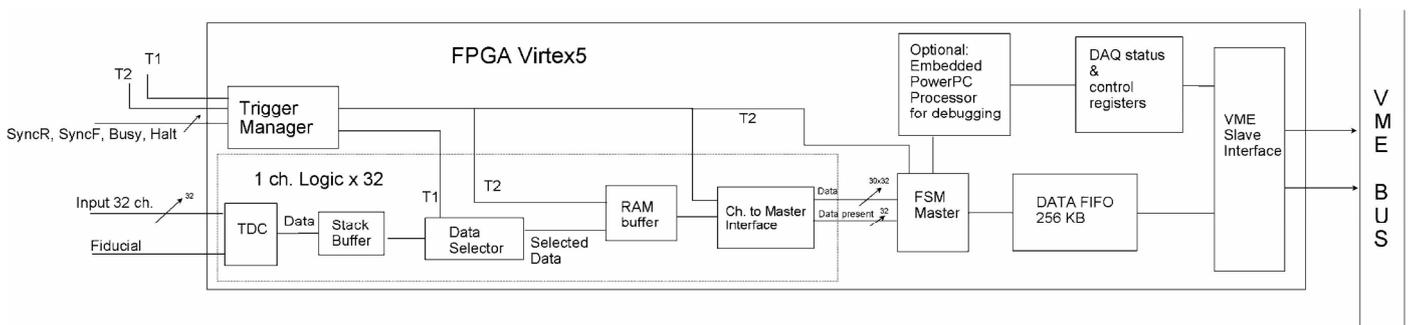


Figura 10. Schema generale del sistema di acquisizione.

In Figura 10 è rappresentato uno schema semplificato dell'intero sistema di acquisizione: come si può vedere per ogni canale è replicata la medesima logica. Il primo stadio di memorizzazione è implementato sulla memoria a stack ed è gestito da una macchina a stati (*Data selector*) che esegue le operazioni descritte nel paragrafo 1.4.1. I dati letti dallo stack vengono momentaneamente memorizzati su una RAM (RAM buffer) ad ogni trigger T1; questi dati saranno convalidati dall'asserzione del trigger di secondo livello T2 per poi essere memorizzati in una memoria FIFO (DATA FIFO).

La gestione dei dati nella RAM buffer è eseguita nel seguente modo: il Data selector, innescato dal T1, scrive sulla RAM buffer a partire dalla prima locazione libera, che chiameremo A, fino alla locazione B; la locazione di partenza rimarrà A finché non è asserito il T2, in quel caso diventerà B+1. Se non ci sarà alcuna transizione del T2 tra un T1 ed il successivo, i dati del primo T1 saranno sovrascritti ed il valore di B sarà aggiornato ad ogni nuovo T1.

In una memoria fifo (non rappresentata in figura) sono scritti gli indirizzi A di tutti gli T1 convalidati dagli T2. Questo registro, equivalente al file system della RAM buffer, consente ad una macchina a stati, “Ch. to Master Interface”, di distinguere ed inviare al “FSM Master” solo i dati relativi ad un evento di T2.

Il FSM Master ad ogni evento di T2 interroga tutti i canali e formatta i dati come illustrato nel paragrafo 4.2. Alcuni canali potrebbero non contenere alcuna misura: i loro dati (che consistono solo di fiducial tag) non vengono scritti sulla DATA FIFO e ciò è segnalato nei footer dell’evento memorizzato (vedi paragrafo 4.2.1).

## **1.5 Il readout del sistema di acquisizione**

Come illustrato in Figura 10, il contenuto della DATA FIFO può essere acceduto tramite una interfaccia VME. In via opzionale può essere utilizzato anche un processore (PowerPC) incorporato sulla FPGA stessa. Questa seconda modalità non è stata implementata nella prima versione del DAQ.

L’interfaccia VME consente accessi a 32 bit in lettura singola (“single shot”) e 64 bit “block transfer”. I registri di configurazione e stato possono essere acceduti in modalità single shot mentre la DATA FIFO anche in modalità burst 64 bit. La DATA FIFO ha parole di 64 bit; visto che i dati relativi agli eventi sono formati da parole da 32 bit (vedi par. 4.2.1), alcuni eventi possono contenere con la parola di riempimento 0x00000000. Un numero dispari di accessi a 32 bit dalla fifo non è consentito; questo provocherebbe un disallineamento e la perdita dei dati.

Alcuni tipi di accesso non sono stati implementati; gli accessi a 32 bit non possono essere a singolo, doppio o triplo byte: l’unico accesso implementato è quello a quattro byte.

Gli indirizzi sono a 32 bit: gli 8 bit più significativi indirizzano la scheda e sono impostati tramite selettori hardware; gli indirizzi dei singoli registri e delle altre aree di memoria sono elencati nel paragrafo 4.

## 2 Modalità di test, controllo e stato di funzionamento del DAQ

L'architettura del sistema di acquisizione implementato è stata illustrata per grandi linee nel capitolo 1. Il funzionamento di tutte le sue parti necessita, ovviamente, di test e debug. A questo scopo sono state implementate diverse modalità di funzionamento. Nel paragrafo 4.1 sono elencati i registri di controllo e stato: il significato della maggior parte è ovvio; altri comandi necessitano di ulteriori spiegazioni.

### 2.1 Reset di sistema

Sono disponibili diversi comandi di reset per le differenti parti del DAQ (memorie, macchine a stati, etc.). La loro esecuzione è abilitata dai relativi segnali di consenso (enable). L'operazione di reset deve quindi avvenire secondo questa procedura:

1. abilitare il reset che si vuole utilizzare scrivendo uno nell'apposito bit del registro `slv_reg1`;
2. eseguire una scrittura sull'apposito bit del registro `slv_reg4` (il segnale di reset è sensibile solo al comando di scrittura non al contenuto di tale registro);
3. disabilitare il reset scrivendo zero nell'apposito bit del registro `slv_reg1`.

Ovviamente l'utente può lasciare i reset abilitati per velocizzare le operazioni; in questo caso gli accessi al registro `slv_reg4` devono essere eseguiti con maggior cautela.

### 2.2 Modalità di test

Diverse modalità di test sono state implementate al fine di provare il corretto funzionamento del sistema. I bit 23 e 24 del registro `slv_reg1` consentono di scrivere tramite VME il suo contenuto o di scrivere pattern predefiniti sulla DATA FIFO. Il pattern, in questo caso, consiste di una sequenza di numeri crescenti a 16 bit replicata 4 volte sulle singole parole a 64 bit (vedi Tabella 1).

Numero parola	Contenuto (hex)
0	00000000
1	01010101
2	02020202
3	03030303
4	04040404
5	05050505
6	06060606
7	07070707
8	08080808
9	09090909
10	0A0A0A0A
11	0B0B0B0B
...	...

*Tabella 1. Contenuto della data fifo in modalità di test 01.*

Si può anche forzare la scrittura della DATA FIFO con un contenuto formattato secondo lo standard KLOE illustrato nel paragrafo 4.2. In questo caso si può eseguire una delle seguenti procedure:

- **Procedura A**

1. assicurarsi che il segnale di Fiducial sia connesso alla scheda;
2. disabilitare il sistema di selezione dati scrivendo 9 nel parametro `Fid_soppress` (bit 20-16 dello `slv_reg3`);
3. abilitare la modalità di generazione automatica dell'ingresso della DATA FIFO scrivendo 1 nel bit 21 di `slv_reg1`;
4. eseguire un reset delle memorie;
5. abilitare la generazione automatica dei segnali di trigger scrivendo 1 nel bit 22 dello `slv_reg1`.

- **Procedura B**

1. assicurarsi che il segnale di Fiducial sia connesso alla scheda;
2. disabilitare i canali del TDC scrivendo tutti zeri nel registro `slv_reg0`;
3. disabilitare il sistema di selezione dati scrivendo 9 nel parametro `Fid_soppress` (bit 20-16 dello `slv_reg3`);
4. eseguire un reset delle memorie;
5. abilitare la generazione automatica dei segnali di trigger scrivendo 1 nel bit 22 dello `slv_reg1`.

L'effetto delle due procedure differisce sostanzialmente nel contenuto della sezione dati del pacchetto illustrato nel paragrafo 4.2: Nel caso A esso consisterà nell'alternarsi dei due pattern [01010101...] e [10101010...]; nel caso B consisterà di una sequenza di fiducial tag. Gli header, i footer e gli altri bit di tag saranno coerenti con quanto riportato nel paragrafo 4.2.

### 2.3 Controllo del sistema di selezione dati

Nel paragrafo 1.4.1 è stato spiegato il principio di funzionamento del sistema di selezione dati. La sua risoluzione è di un periodo di fiducial e consiste semplicemente nel contare il numero di fiducial tag in fase di lettura dalla memoria stack. Questa

fase sarà interrotta dal Data Selector quando esso avrà contato un numero di fiducial tag uguale al valore del parametro di controllo `fid_soppress`. Se ad esempio viene impostato `fid_soppress=3`, il Data Selector conterà tre fiducial tag e per questo motivo i dati registrati sulla RAM buffer saranno quelli relativi all'intervallo di tempo tra la transizione del trigger e tre transizioni positive del fiducial precedenti (vedi Figura 11).

Un funzionamento particolare è riservato alle impostazioni `fid_soppress=0` e `fid_soppress=9`: nel primo caso il Data Selector leggerà tutto il contenuto dello stack senza contare i fiducial tag; nel secondo caso i dati saranno scritti sulla DATA FIFO anche se essi non contengano alcuna misura.

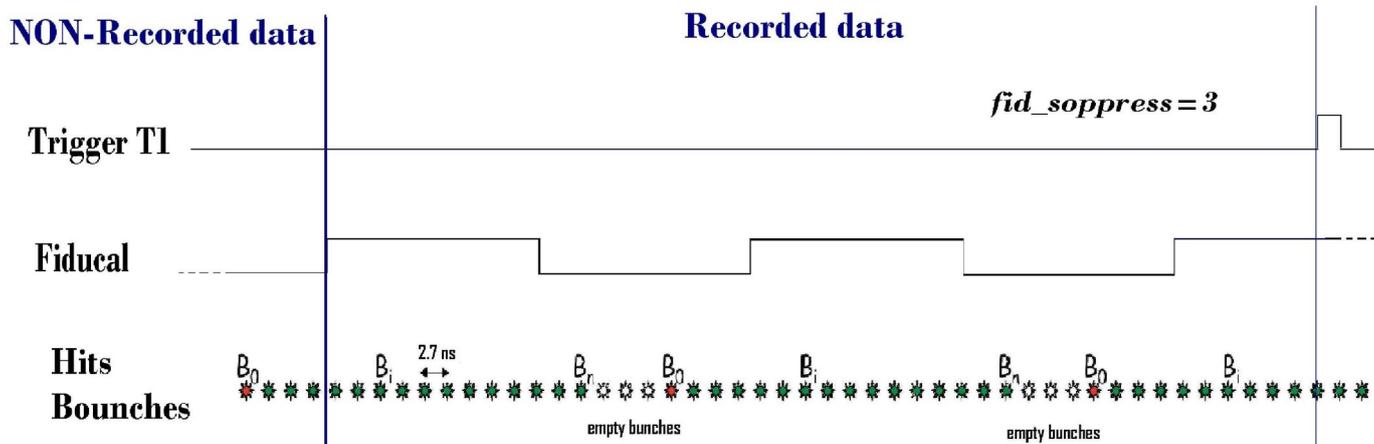


Figura 11. A partire dall'istante di asserzione del trigger, il Data Selector leggerà i dati dallo Stack a ritroso finché non saranno contati tre tag di fiducial che corrispondono alle transizioni positive dello stesso.

## 2.4 Impostazione dell'offset del TDC

La tecnica di memorizzazione utilizzata presenta un limite: durante la scrittura dello stack ci potrebbero essere dei conflitti tra la registrazione dei fiducial tag e le misure. Poiché la precedenza è riservata ai fiducial tag alcune misure non saranno registrate. Come anticipato nel paragrafo 1.1, il TDC è provvisto di un offset regolabile con risoluzione pari ad un periodo di clock. Questo consente di traslare l'istante di registrazione del fiducial tag in corrispondenza di uno dei bunch vuoti (vedi il secondo istogramma di Figura 2 e Figura 12).

Per impostare l'offset del TDC in modo corretto si può utilizzare la seguente procedura:

1. mettere il sistema in misura;
2. ottenere un istogramma delle misure come il primo di Figura 2 per ogni canale;
3. impostare un valore di offset in modo tale che i precedenti istogrammi diventino simili al secondo di Figura 2.

## 2.5 Stato di riempimento della fifo ed elaborazione dei trigger

Durante le normali operazioni di acquisizione la motorola del ??Rock?? interroga il DAQ leggendo lo stato di riempimento della DATA FIFO dai bit 7-2 dello `slv_reg6`. I bit 1 e 0 indicano lo stato della coda dei trigger che ancora devono essere processati. In Tabella 2 riporto i valori possibili su `slv_reg6`.

	Stato della Fifo	Vuota	<1/4	>1/4, <1/2	>1/2, <3/4	>3/4, <1	Piena
	Stato della Fifo (KByte)	0	<65	<131	<197	<262	262
	Codice SR6 (HEX)	D9/DA	DD/DE	D5/D6	E5/E6	A5/A6	25/26
<b>Bit di stato della Fifo</b>							
	7, Not Full <1	1	1	1	1	1	0
	6, Not Prog_Full <3/4	1	1	1	1	0	0
	5, Half Full >1/2	0	0	0	1	1	1
	4, Not Half Full <1/2	1	1	1	0	0	0
	3, Not Prog_Empty <1/4	1	1	0	0	0	0
	2, Not Empty >0	0	1	1	1	1	1

Tabella 2. Nelle prime tre righe di questa tabella sono riportati i 6 stati di riempimento della Data Fifo ed i relativi codici che possono essere letti sugli 8 bit meno significativi di `slv_reg6`. Nelle righe successive sono riportati i bit di stato della Fifo ed i relativi valori.

## 2.6 Frequenza di conteggio dei singoli canali

Ogni canale del TDC è stato provvisto di un contatore di impulsi che è azzerato ad ogni transizione positiva del fiducial. Il numero di conteggio ottenuto alla fine di ogni ciclo di fiducial è scritto sulla Rate Ram. La Rate Ram è di tipo dualport e può essere letta tramite bus VME; durante gli accessi la scrittura è disabilitata.

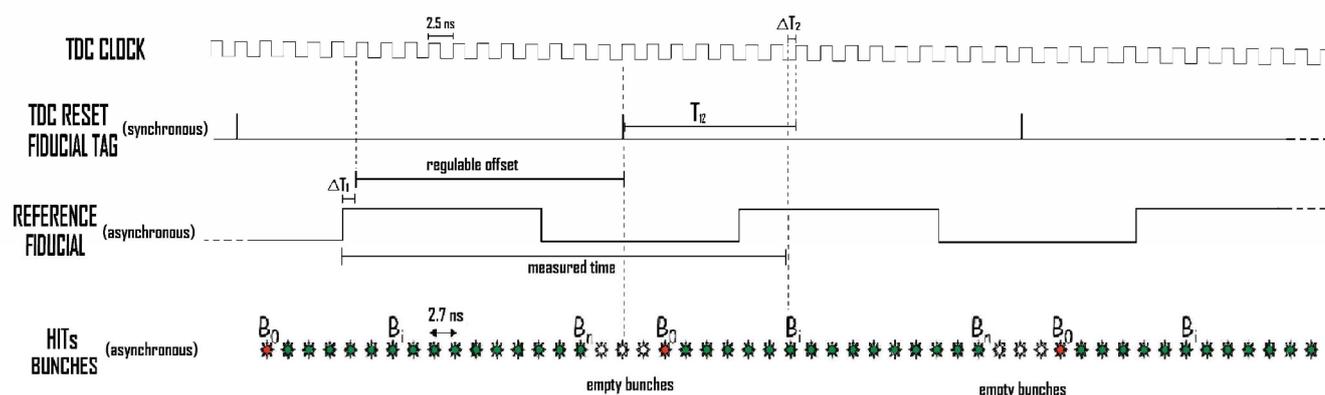


Figura 12. Il TDC implementato utilizza la tecnica dell'interpolazione che consiste nell'eseguire la misura, tra il fronte positivo del fiducial e il generico bunch (hit)  $B_i$ , in tre fasi: prima  $\Delta T_1$ , poi  $T_{12}$  ed infine  $\Delta T_2$ . In questa figura è rappresentata la tecnica generalizzata in cui è stato introdotto un offset configurabile: in questo modo il fiducial tag è scritto nello stack in corrispondenza dei bunch vuoti.

## 2.7 Stato di errore e di busy

Le RAM buffer sono sostanzialmente delle code circolari in cui i dati vengono memorizzati temporaneamente in attesa di essere letti e copiati sulla DATA FIFO. Un flusso di dati o una frequenza di trigger molto grandi possono causare la sovrascrittura di dati non ancora trasferiti alla DATA FIFO. Una seconda situazione anomala prevista è quella in cui la DATA FIFO non sia scaricata sufficientemente spesso e venga riempita in modo da non poter più accettare dati.

Per evitare tali situazioni di errore sono stati implementati due sistemi di allerta: il primo è l'asserzione del segnale BUSY, il secondo è l'asserzione di un bit di errore nel registro `slv_reg2` e del segnale di HALT.

Il segnale di BUSY è asserito quando uno dei buffer del sistema di acquisizione è quasi pieno (le soglie sono state scelte in modo tale che, dopo l'asserzione del BUSY, il sistema sia ancora in grado di accettare almeno un altro evento di T2). Secondo il protocollo di KLOE, questo dovrebbe inibire la generazione di successivi trigger di T2 sinché la situazione di "busy" non sia rientrata.

La situazione di "errore" si verifica quando dati validi sono stati sovrascritti o quando la DATA FIFO è piena.

### 3 Piedinatura della scheda

I segnali di ingresso e uscita della scheda possono essere distinti nelle quattro seguenti categorie:

1. segnali dal rivelatore;
2. segnali da DAΦNE;
3. segnali da e verso KLOE;
4. segnali VME.

I pin della prima e quarta categoria sono disposti su degli appositi connettori. Il fiducial è l'unico segnale proveniente da DAΦNE ed è collegato all'apposito connettore SMA: il segnale dal connettore SMA è discriminato e reso differenziale da un discriminatore montato sulla scheda; i relativi pin sull'FPGA sono K18 e J19. I segnali da e verso KLOE sono disposti su un apposito connettore a cui sarà connesso un disaccoppiatore ottico che garantisce la separazione delle masse\*. In Tabella 3 sono riportati i pin della scheda, i relativi pin della FPGA e i segnali ad essi associati.

PIN sulla scheda	PIN sulla FPGA	Segnale
Y9	Y9	T1
V10	V10	T2
T8	T8	SyncR
P9	P9	SyncF
T11	T11	Busy
R9	R9	Halt
Y8	Y8	Trigger_req

*Tabella 3. Piedinatura dei segnali da e verso KLOE. I pin sulla scheda hanno le stesse sigle dei pin sulla FPGA. L'ultimo segnale è riservato alle operazioni di test.*

Il segnale Trigger\_req è riservato alle operazioni di test e caratterizzazione del TDC: su di esso è semplicemente riportato il contenuto del bit 31 dello slv\_reg3.

---

\* La massa di KLOE è isolata per minimizzare il rumore.

## 4 Indirizzo e funzione dei registri

Registro	Funzione	Addr [23 downto 0]
slv_reg0	abilitazione canali	F80000
slv_reg1	abilitazione trigger, reset, testmode	F80004
slv_reg2	errore canali	F80008
slv_reg3	impostazione ritardi, offset	F8000C
slv_reg4	reset	F80010
slv_reg5	golden number e latch numero T2	F80014
slv_reg6	stato fifo e contatori T2	F80018
slv_reg7	impostazione soglie fifo	F8001C
Rate Ram	ram su cui scrive il rate	18000X
Data Fifo	data fifo	00XXXX

### 4.1 Funzione dei singoli bit dei registri

#### slv\_reg0 (F80000)

Abilita o disabilita i 32 canali del TDC (default: (others=> '0'))

#### slv\_reg1 (F80004)

bit	default	Funzione
31	1	abilita il Trigger di primo livello
30	0	abilita il Trigger di secondo livello
29	0	abilita un reset globale della periferica
28	0	abilita il reset delle memorie della periferica
27	0	abilita il reset delle macchine a stati
26	0	abilita il reset del DCM e del PLL della periferica
25	0	abilita il reset dei registri di controllo
24 downto 23	00	abilita le modalità di test per la FIFO 00 => la fifo è scritta dal TDC 01 => scrive un pattern nella fifo in modo automatico 10 => scrive nella fifo tramite VME 11 => la fifo non scrive nulla.
22	0	abilita modalità di generazione automatica dei trigger
21	0	abilita modalità di generazione automatica dell'ingresso della data fifo

#### slv\_reg2\* (F80008)

Stato di errore dei 32 canali verificatosi durante il T2 precedente.

\* I registri 2 e 6 sono solo in lettura.

## slv\_reg3 (F8000C)

bit	default	Funzione
31	0	Trigger_req utilizzato dal PowerPC in modalità di test
20 downto 16	00011	Fid_soppress indica quanti cicli fiducial devono essere selezionati per ogni T1
14 downto 8	0000000	RitardoTriggers, imposta il ritardo dei trigger in unità di 10 ns
7 downto 0	00000000	OffsetFiducial ovvero offset del TDC. Deve essere calibrato in fase di inizializzazione

## slv\_reg4 (F80010)

bit	default	Funzione
4	0	reset dei registri di controllo
3	0	reset del DCM e del PLL della periferica
2	0	reset macchine a stati
1	0	reset memorie
0	0	reset globale

## slv\_reg5 (F80014)

bit	default	Funzione
31 downto 16	0	Valore del contatore di T2 in caso di errore nel sistema di controllo syncR & syncF
13	1	Inverte internamente lo stato del SyncR
12	0	Inverte lo stato del SyncF
11 downto 0	123 (decimale)	Valore del Golden number

## slv\_reg6\* (F80018)

bit	Funzione
31 downto 20	Numero T2 che e' gia' stato processato.
19 downto 8	Numero T2 che sara' processato.
7 downto 0	registro su cui fare il polling: "(not BigFifoFull) & (not prog_full) & data_countBigFifo(15) & (not data_countBigFifo(15)) & (not prog_empty) & (not BigFifoEmpty) & (not codaT2empty) & (not CodaT2full)"

## slv\_reg7 (F8001C)

bit	default	Funzione
30 downto 16	hex "2000"	prog_full_thresh della fifona (determina il prog_full)
14 downto 0	hex "6000"	prog_empty_thresh della fifona (determina il prog_empty)

## 4.2 Formattazione dati nella fifo

Parola in testa	Header RockM
	Header Rock
	Dati
	...
	...
	Dummy word (eventuale)
	My footer 1
	My footer 2
	Footer Rock
Parola in coda	Footer RockM

\* I registri 2 e 6 sono solo in lettura.

## 4.2.1 Formattazione delle singole parole nella fifo

### Header RockM (FFFFFFF)

bit	31	30	29 - 27	26 - 0
	1, primo tag	1, secondo tag	101, My Header RockM	Header RockM

### Header Rock (FD000XXX)

bit	31	30	29 - 27	26 - 12	11 - 0
	1, primo tag	1, secondo tag	111, My Header Rock	Header Rock	Numero T2

### Dati

bit	31	30	29 - 20	19 - 10	9 - 0
	0, primo tag	X, secondo tag	Dati 1	Dati 2	Dati 3

### Dummy word (eventuale)

bit	31-0
	0x00000000

### My Footer 1

bit	31	30	29 - 16	15 - 0
	1, primo tag	0, secondo tag	Misure Fiducial	Ch. Attivi (31 - 16)

### My Footer 2 (8000XXXX)

bit	31	30	29 - 16	15 - 0
	1, primo tag	0, secondo tag	dummy	Ch. Attivi (15 - 0)

### Footer Rock (FE000000)

bit	31	30	29 - 27	26 - 19
	1, primo tag	1, secondo tag	My footer Rock	Footer Rock

### Footer RockM (F7FXXXXX)

bit	31	30	29 - 27	26 - 19	18 - 0
	1, primo tag	1, secondo tag	My footer RockM	Footer RockM	Nword

### Legenda

- **Primo tag**, bit 31: è zero se la parola contiene dati, uno se contiene footer o header;
- **Secondo tag**, bit 30: per i dati è uno se è la prima parola relativa ad un canale altrimenti è zero, per gli header/footer è zero solo se si tratta di un My footer altrimenti è uno;
- **My Header RockM**, bit (29 down to 27): 101;
- **Header RockM**, bit (26 down to 0): (others=>1);
- **My Header Rock**, bit (29 down to 27): 111;
- **Header Rock**, bit (26 down to 12): 1010000000000000;
- **T2 number**, bit (11 down to 0): numero di T2;

- **Dati n**, l'ordine temporale (decescente) delle misure è dai bit più significativi verso quelli meno significativi e dalla testa verso la coda;
- **Ch. Attivi**, bit (29 downto 16): se un canale non ha misure non viene memorizzato nessun dato e il bit "ch. Attivo" è posto a zero;
- **Misure fiducial**, bit (13 downto 0): ogni misura è da 2 bit e l'ordine temporale (crescente) delle misure è dai bit meno significativi verso quelli più significativi;
- **Dummy**, bit (13 downto 0): (others=>0) non hanno alcun significato;
- **My Footer Rock**, bit (29 downto 27): 111;
- **Footer Rock**, bit (26 downto 0): 1100000000...
- **My Footer RockM**, bit (29 downto 27): 110;
- **Footer RockM**, bit (26 downto 19): (others=>1);
- **Nword**, bit (18 downto 0): numero di parole del pacchetto;
- **Dummy word**, questa parola viene aggiunta se il numero di parole relative ad un evento non è pari: questo accorgimento è necessario poiché la Data Fifo è a 64 bit.