

# Sistemi di trigger

**Sistemi di trigger :**

**Generalità sui sistemi di trigger e di acquisizione**

**Connessioni con simulazioni degli eventi**

**Sistemi di trigger per esperimenti di fisica nucleare di bassa energia**

**Sistemi di trigger a più livelli**

**Cenni su sistemi di trigger per le alte energie**

**Hardware per sistemi di trigger veloci**

**Sistemi di trigger software**

**Test pratici con modulistica standard NIM/CAMAC**

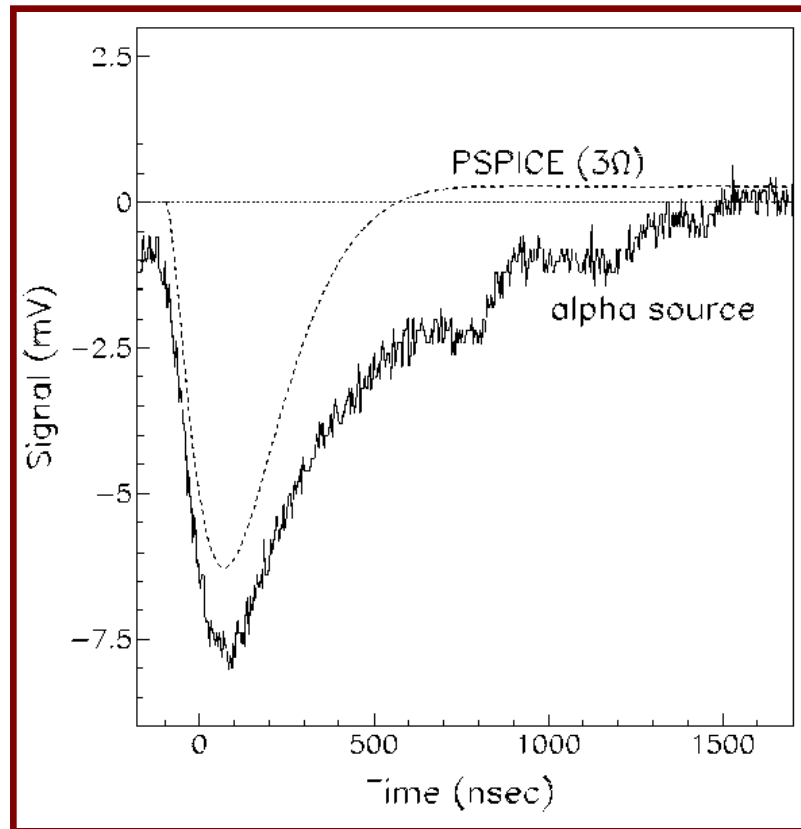
# Generalità

**I sistemi di trigger nascono per migliorare l'efficienza di utilizzo dei sistemi di rivelazione**

**Un rivelatore ideale deve essere sempre acceso e non perdere alcun evento.**

**Se avviene una reazione devo leggere il mio rivelatore e conservare i dati per l'analisi successiva. Idealmente dovrei farlo durante il tempo in cui il mio rivelatore si ripristina ed è pronto per rivelare una nuova particella**

# Generalità



**Se il mio segnale dura  $1\mu\text{s}$  dovrei riuscire a registrarlo durante questo tempo ed essere pronto a misurare un nuovo evento subito dopo per non perdere nulla.**

## Generalità

**Purtroppo non siamo ancora in grado di fare questo. Con i digitalizzatori siamo in grado oggi di convertire segnali 12-14 bit (100MHz-1GHz)**

**1 detector  $1\mu\text{s}$  =  $12*100$  bit =150byte**

**Max readout VME = 80MB/s**

**Tempo di lettura VME 80B =  $1\mu\text{s}$**

**Impieghiamo quindi  $2\mu\text{s}$  per leggere i dati raccolti in  $1\mu\text{s}$  da 1 rivelatore ma di solito ne abbiamo tanti, in un crate VME possiamo inserire 20 moduli ciascuno con 4-8 canali ....**

## Generalità

**Supponiamo poi di riuscire a leggere questi dati molto velocemente, dobbiamo poi scriverli, se abbiamo dati dal nostro VME con 20 moduli e 8 canali 150B/ $\mu$ s al canale**

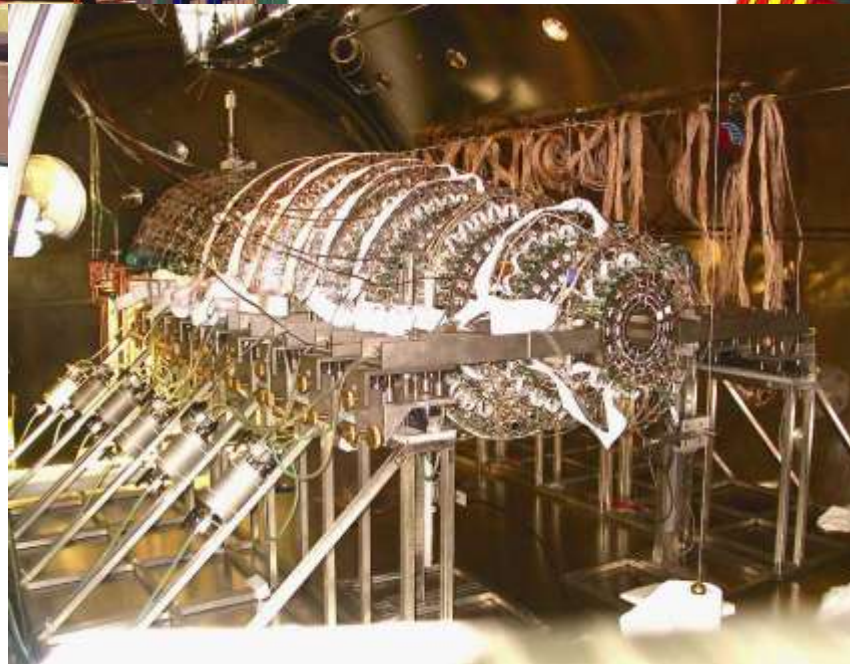
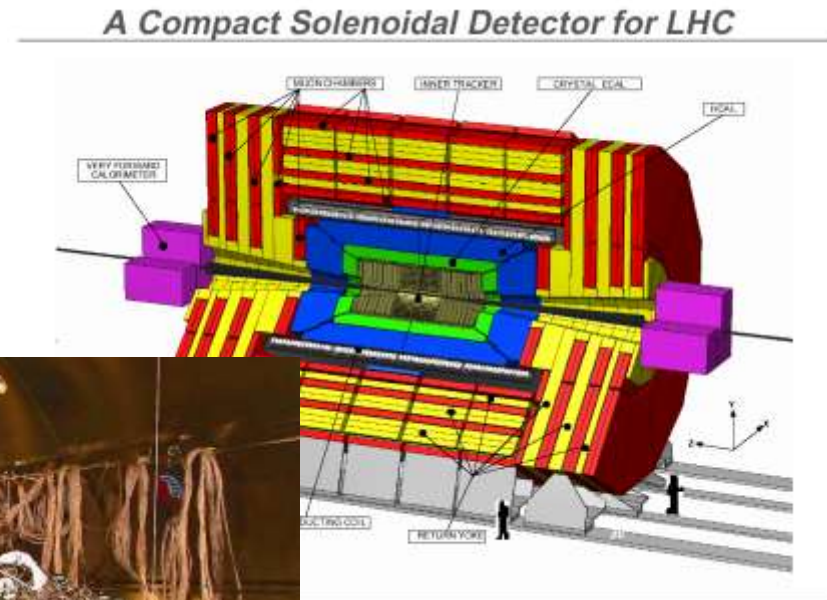
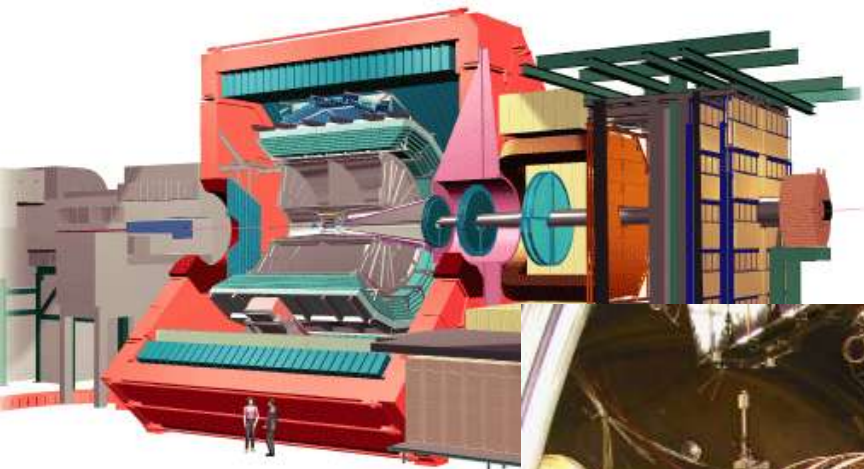
**150MB/s \* 8 \* 20 = 24 GB/s in 10s riempiamo il disco del vostro portatile**

**I moduli elettronici devono essere equipaggiati da sistemi di soppressione di zeri ( non dobbiamo perdere tempo a trasferire zeri)**

**Occorre poi un sistema che ci dica quando ci sono dati da leggere - trigger**

# Generalità

**Il sistema di trigger deve tenere conto della struttura del rivelatore e della fisica che si vuole studiare**



## Generalità

**connessione tra le lezioni di trigger e di simulazioni a N-corpi**

**I primi passi per la costruzione di un buon sistema di trigger sono:**

**Studio del processo fisico da investigare;**

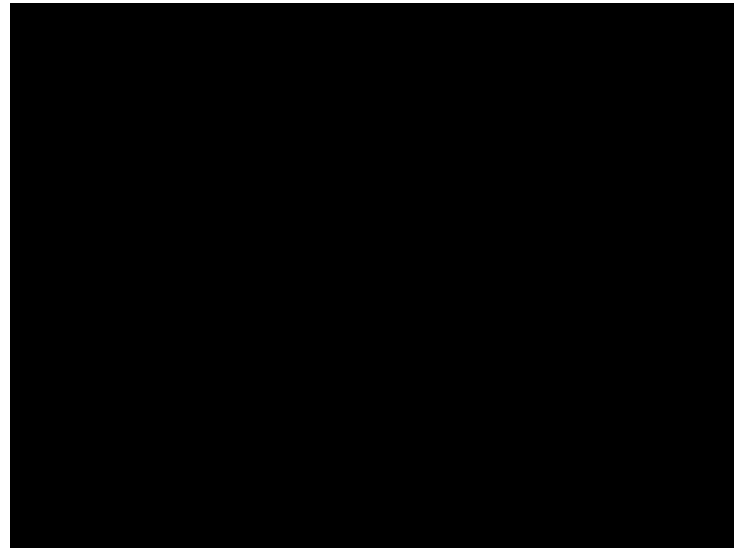
**Studio della risposta dei rivelatori utilizzati;**

**Studio delle velocità di readout e storage disponibili.**

**Come costruiamo un sistema di trigger per lo studio di reazioni nucleari ad energie intermedia con un sistema a  $4\pi$  come CHIMERA**

# Sistema di trigger per energie intermedie

**Studio del processo fisico da investigare:**

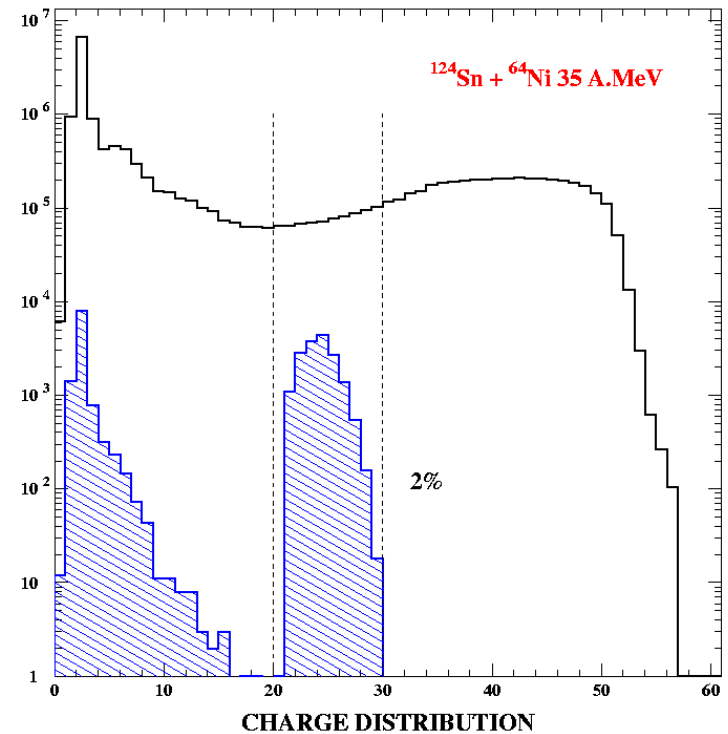
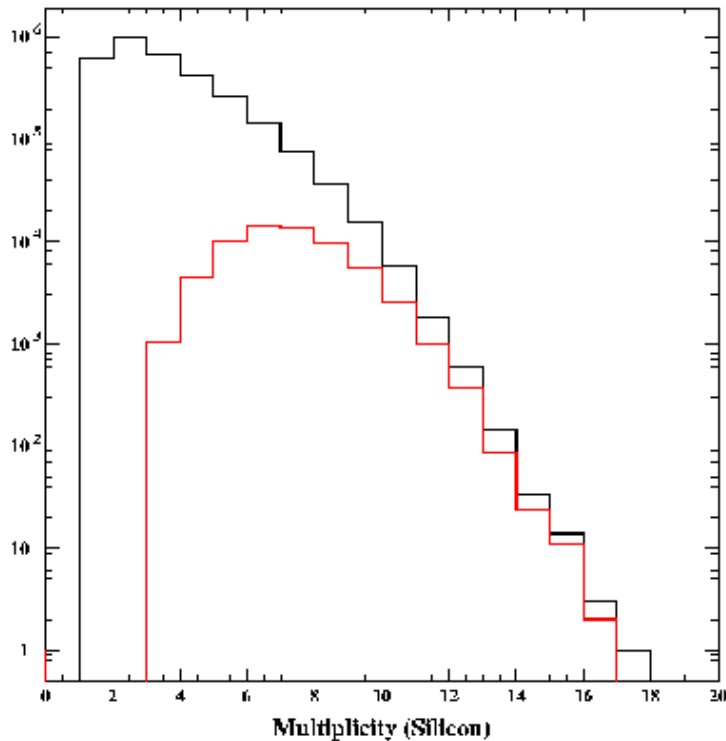


**Uno dei processi più interessanti ad energie intermedie è la multiframmentazione di cui avrete parlato o parlerete con Massimo Papa**



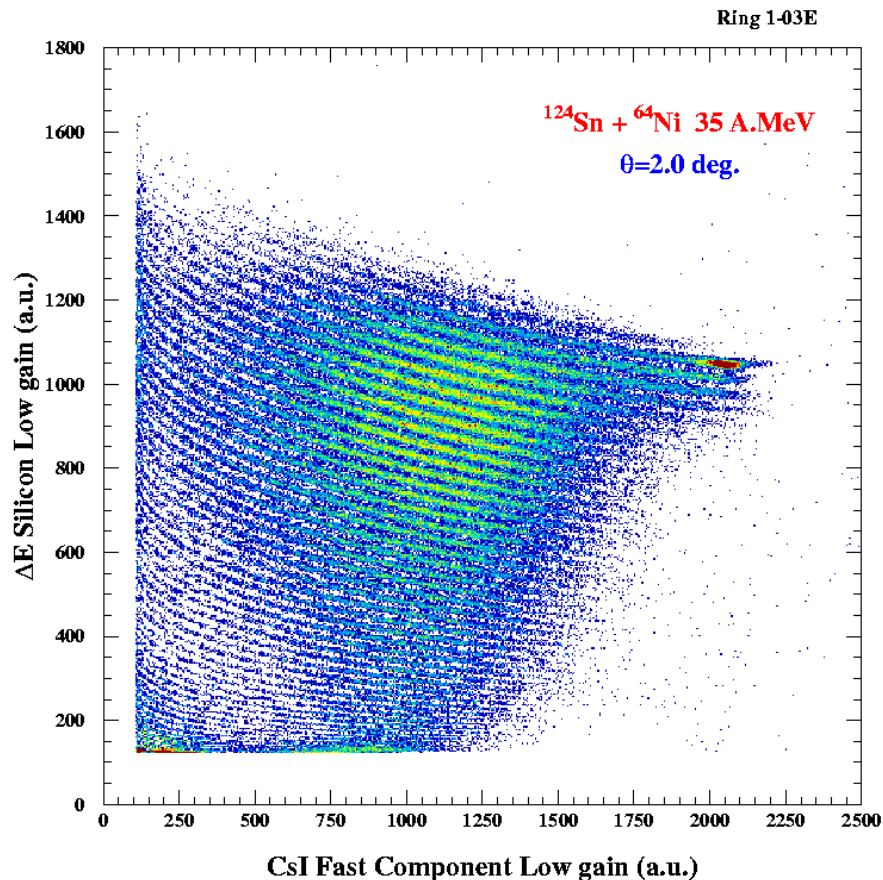
# Sistema di trigger per energie intermedie

Negli eventi di multiframmentazione avremo produzione di un gran numero di particelle ( non quante a LHC!!! ) ed una distribuzione di cariche medio-bassa



# Sistema di trigger per energie intermedie

Dobbiamo però fare attenzione, vogliamo vedere eventi di multiframmentazione ma questi rappresentano solo una piccola percentuale della sezione d'urto totale



Lo scattering elastico è sempre presente

Ci sono tanti eventi di reazioni binarie

## Su cosa si può basare il nostro trigger?

Possiamo usare dei discriminatori che ci dicono se ci sono o no segnali nei nostri rivelatori

Modo semplice → OR dei segnali

Ad esempio se ho un fascio di  $^{120}\text{Sn}$  da 30 MeV/A di 1.6 nA ed un bersaglio di Au da  $200\mu\text{g}/\text{cm}^2$  - a  $1^\circ$  la sezione d'urto per scattering Coulombiano è di quasi  $4 \times 10^5 \text{b}/\text{sr}$  - Chimera ha a  $1^\circ$  rivelatori da  $2 \times 10^{-4} \text{sr}$  ( rivelatori da  $5 \times 5 \text{cm}^2$  a 350cm dal bersaglio ) il tasso di conteggio atteso è quindi:

$$N_i = 1.6 \times 10^{-9} / 25 \times 1.6 \times 10^{-19} = 4 \times 10^8 \text{ p/s}$$

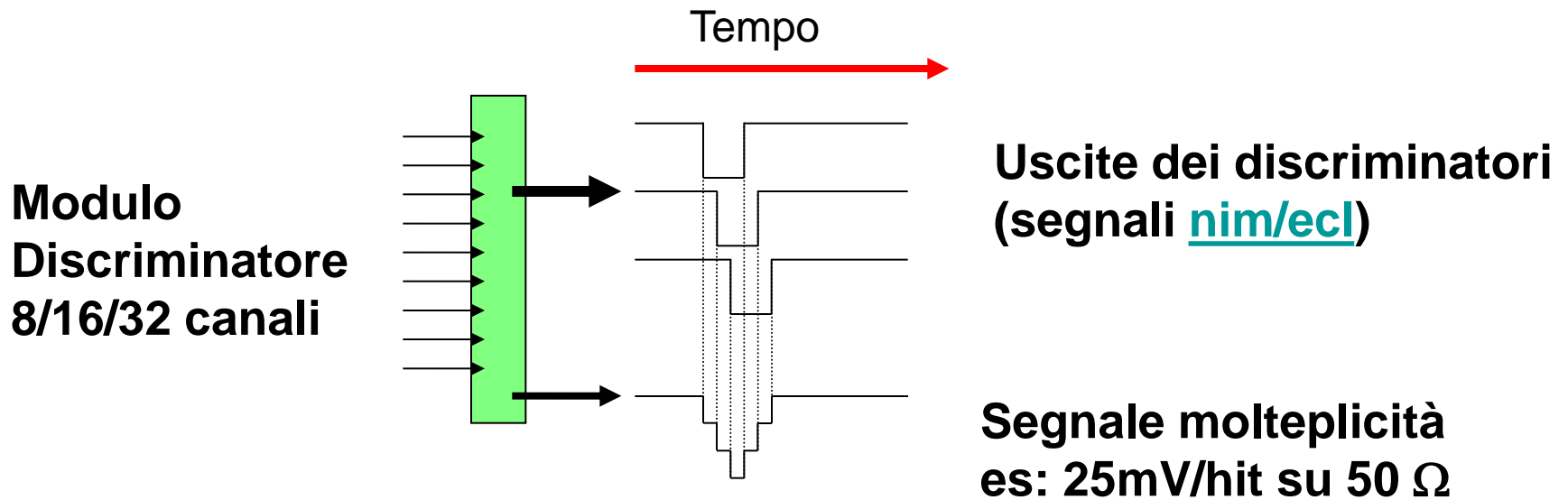
$$N_t = (200 \times 10^{-6} / 197) \times 6 \times 10^{23} = 6 \times 10^{17} \text{ p/cm}^2$$

$$N = N_i \times N_t \times \sigma \times \Delta\Omega = 4 \times 10^8 \times 6 \times 10^{17} \times 4 \times 10^{-19} \times 2 \times 10^{-4} = 20 \text{ khz}$$

## **Su cosa si può basare il nostro trigger?**

**Molto meglio contare I rivelatori che hanno dato segnale**

# Sistema di trigger di primo livello (molteplicità)



**Posso mandare questo segnale ad un discriminatore e impostando la soglia corretta ottenere il mio trigger di molteplicità. Molteplicità 3 soglia maggiore di 50 mV. In un rivelatore di Multiframmentazione come CHIMERA questo segnale è il segnale master di acquisizione**

**Devo avere almeno 3 frammenti emessi per acquisire l'evento mentre devo rigettare eventi con 2 soli frammenti ( scattering elastico in cui rivelo proiettile e bersaglio che non si sono frammentati )**

# Classificazione del trigger

Trigger primo livello

Analizza le uscite dei discriminatori secondo pattern predefiniti  
Permette la conversione dei segnali  
Tempi caratteristici 100 ns-1  $\mu$ s

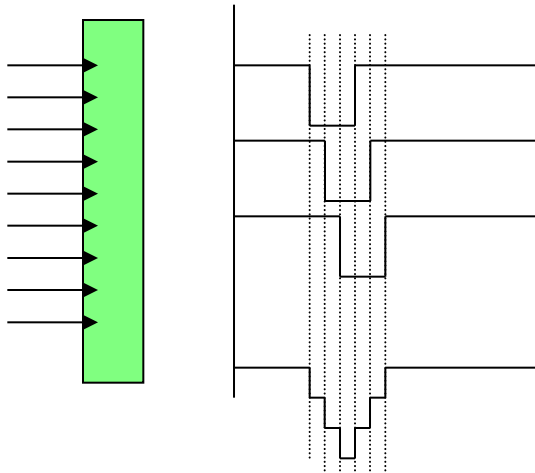
Trigger secondo livello

Analizza i segnali convertiti costruisce variabili globali per selezionare eventi validi da trasferire al livello superiore  
Tempi caratteristici 10-100  $\mu$ s

Trigger n-livello

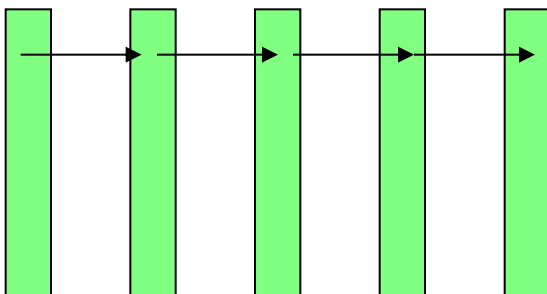
Analizza le variabili globali costruite  
E decide se salvare su nastro-disco i dati

## Sistema di trigger di primo livello (molteplicità avanzata)



Notiamo che il segnale di molteplicità viene costruito correttamente solo se le particelle arrivano in coincidenza entro il tempo dato dalla larghezza delle uscite dei discriminatori

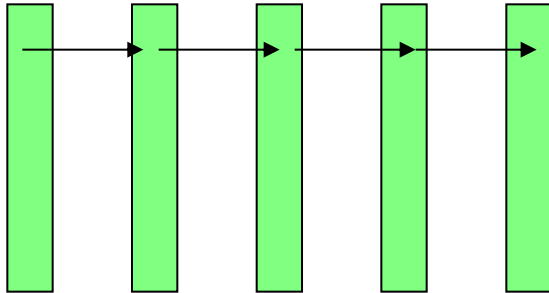
**Occorre ovviamente ricordarsi pure dei tempi di transito dei segnali**



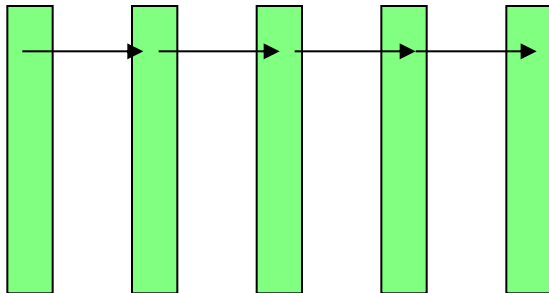
Per mandare il segnale dal primo all'ultimo dei discriminatori ci metto del tempo (ritardo circa  $1\text{ ns}/20\text{ cm}$ )

Se dobbiamo collegare 80 moduli il ritardo può diventare più di 100 ns

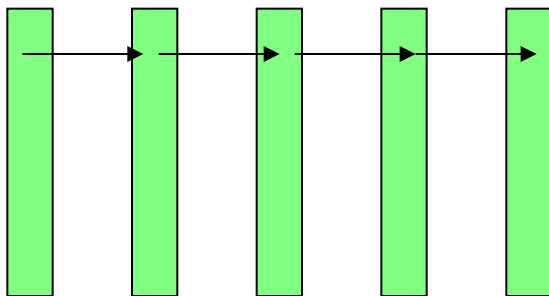
## Sistema di trigger di primo livello (molteplicità avanzata)



Conviene costruire più catene anche per diminuire il rumore e rendere più semplice la ricerca dei guasti



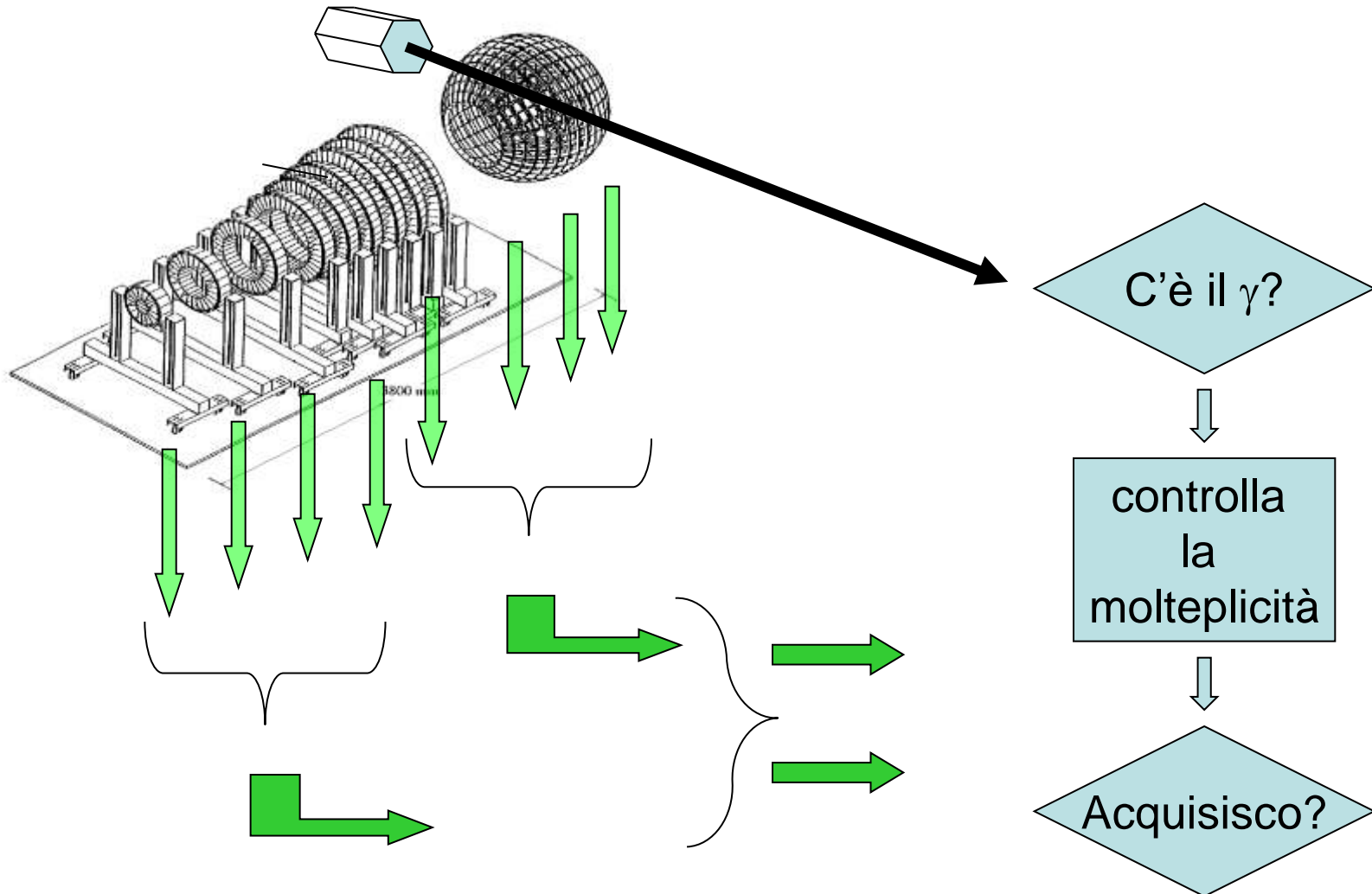
Se le catene corrispondono a pezzi del rivelatore si può sfruttare il tutto per ottenere pure l'informazione sulla molteplicità di particelle in funzione dell'angolo di rivelazione



Ovviamente per esperimenti particolari in cui accoppiamo il rivelatore  $4\pi$  con altri rivelatori occorre mettere nella giusta priorità l'informazione raccolta da tali rivelatori "master"



# Sistema di trigger di primo livello (molteplicità avanzata + rivelatore aggiuntivo)



## Sistema di trigger di livello superiore

**Il sistema di trigger di primo livello autorizza la conversione degli eventi che può durare da 10 a 100  $\mu$ s secondo il tipo di convertitore scelto.**

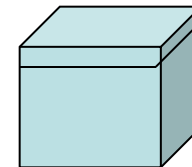
**Dopo la conversione il dato va letto/trasferito e conservato per l'analisi**

**Conservare troppi dati può costare troppo sia in termini di tempo che di "spazio" ( spazio disco, spazio fisico per immagazzinare i nastri, lavoro per rileggerli ) se si può evitare di scrivere dati non interessanti è molto meglio.**

Cerca l'ago nel pagliaio?

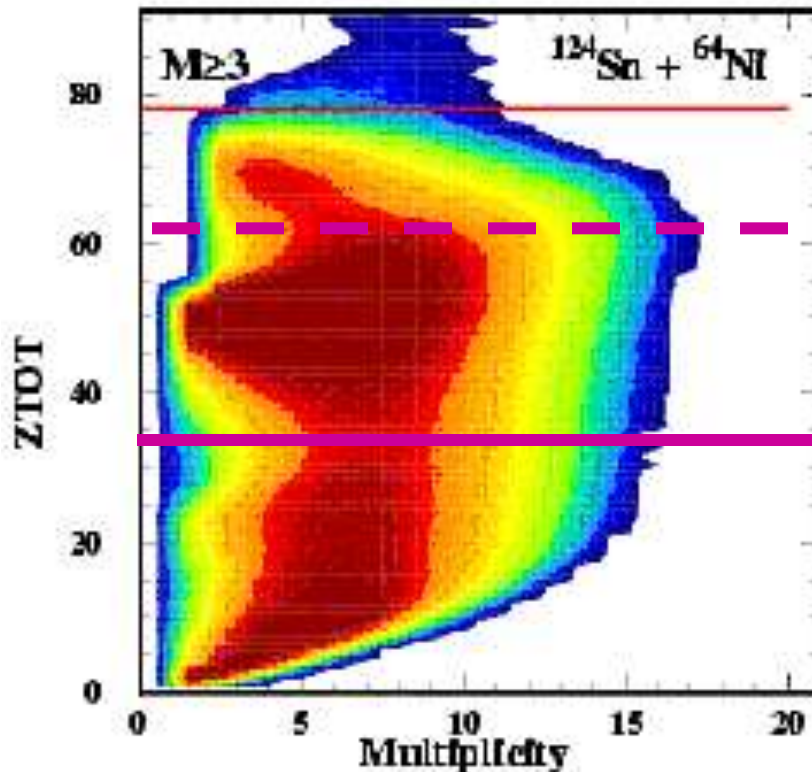


Meglio cercarlo nello scatolino



## Sistema di trigger di livello superiore

Un trigger di livello superiore analizza i dati convertiti e sfrutta le informazioni fisiche contenute nei dati per effettuare una accettazione o rejezione dell'evento. Ad esempio se la carica totale raccolta in un evento è minore della carica di proiettile e bersaglio l'evento non è completo, non potrò analizzarlo, meglio buttarlo e non perdere tempo a scriverlo su banda



Rivelato solo proiettile  
Eliminabile secondo la  
fisica cercata

Eventi eliminabili

## **Sistema di trigger di livello superiore**

**Nei sistemi di rivelazione preparati per LHC ed in generale per gli esperimenti di alta energia esistono vari livelli di trigger superiore che si basano sulle risposte attese dai vari rivelatori**

**Si ricercano prima eventi validi nei singoli pezzi di rivelatore dedicati a varie tipologie di particelle ( rivelatori che identificano muoni, particelle strane, elettroni ....)**

**Un livello superiore poi mette insieme le informazioni provenienti dai vari sottorivelatori per cercare l'evento composto da 4 mu, 5 elettroni 2 gamma .... Che potrebbe identificare ...l'higgone o il Rubbione o ....  
L'LHCone**

**Capite bene come i 10 anni spesi nel costruire un rivelatore di alta energia servano tutti a coloro che fanno simulazioni per preparare gli algoritmi su cui devono lavorare i livelli di trigger avanzati per evitare di buttare eventi da Nobel**

# **Sistema di trigger di livello superiore**

**Come si costruisce un trigger di livello superiore?**

**Occorre utilizzare una grande potenza di calcolo in linea**

**DSP ( digital signal processor ) sono dei chip che effettuano computazione, sono privi di sistema operativo, vengono programmati in base alle esigenze,,**

**FPGA –(Field Programmable Gate Array) sistemi fatti da logiche programmabili**

**Per i livelli più sofisticati si possono usare anche Farm di CPU che oggi raggiungono livelli di velocità notevoli e semplificano il compito del programmatore per quanto riguarda interconnessione, scambio di dati programmazione parallela**

## Ottimizzazione del tempo morto

**Abbiamo già accennato ai tempi tipici con cui abbiamo a che fare nelle misure di fisica nucleare**

**Ogni 100 ns mediamente arriva un burst di particelle che può generare reazioni, accettato l'evento il sistema di trigger genera le finestre di conversione per i QDC/TDC, Per Chimera occorrono circa 10-15  $\mu\text{s}$  per generare questi gate correttamente**

**I nostri QDC hanno un tempo di conversione fisso di circa 100  $\mu\text{s}$**

**L'attuale readout legge gli eventi convertiti a circa 250  $\mu\text{s}/\text{ev}$**

**Sommando tutti i tempi morti arriveremmo a quasi 500  $\mu\text{s}/\text{ev}$  che porterebbe ad un tempo morto medio del 50% a 1kHz**

## Ottimizzazione del tempo morto

Con  $500 \mu\text{s/ev}$  possiamo acquisire al massimo 2 KHz

Gli eventi arrivano in sequenza random ad un rate del 100% del massimo rate teorico il tempo morto è del 50% secondo la formula:

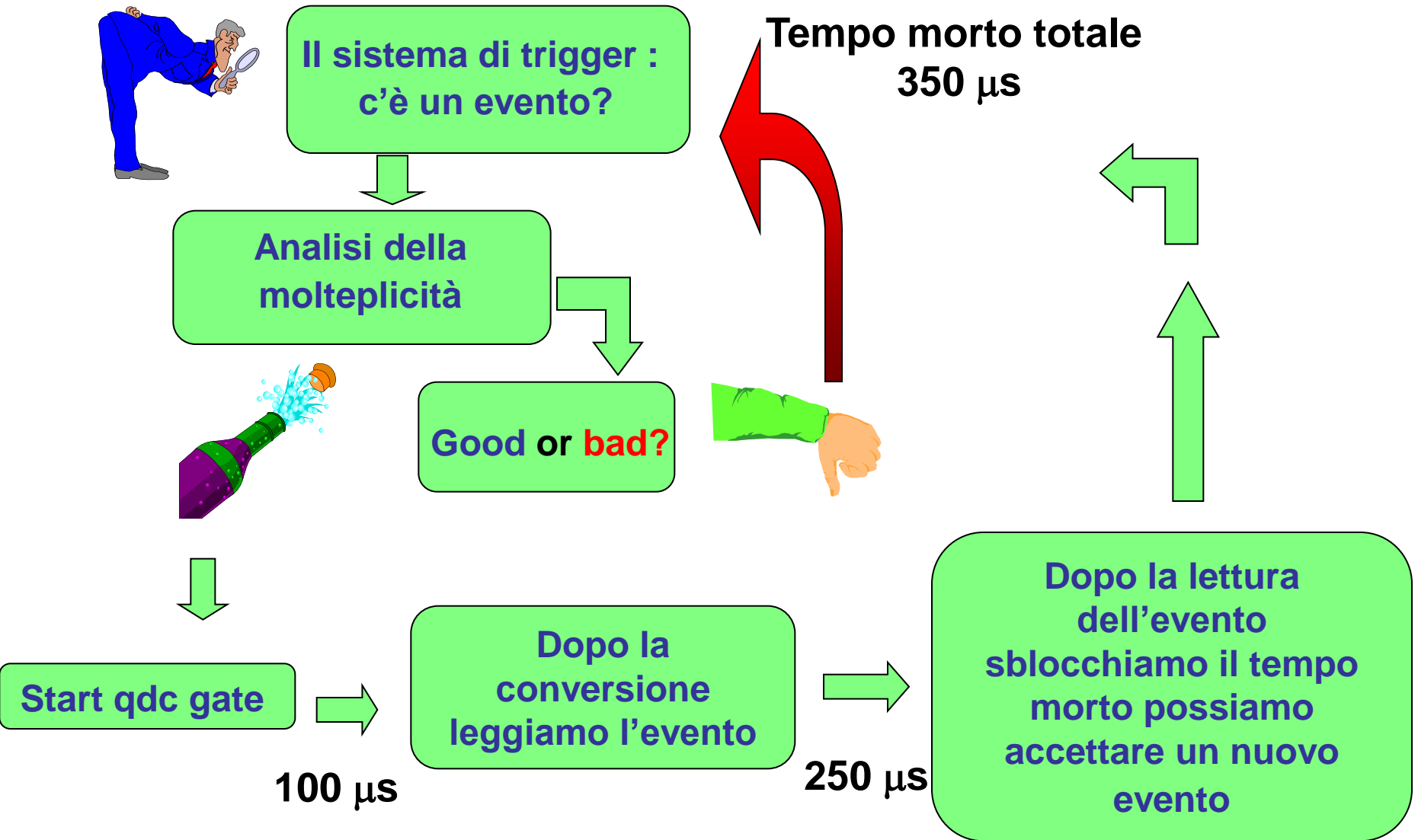
$$N/M = 1 / (1 + M/M_{\max})$$

Dove N sono gli eventi acquisiti, M il tasso di conteggi in input  $M_{\max}$  il tasso di conteggi massimo ( nel nostro caso 2kHz ) se  $M/M_{\max} = 1$   $N = 0.5M$

Per migliorare le performances dell'acquisizione si possono ovviamente migliorare le velocità dei vari processi, ma la soluzione vera è renderli paralleli in modo da non sommarne i tempi di esecuzione

# Ottimizzazione del tempo morto

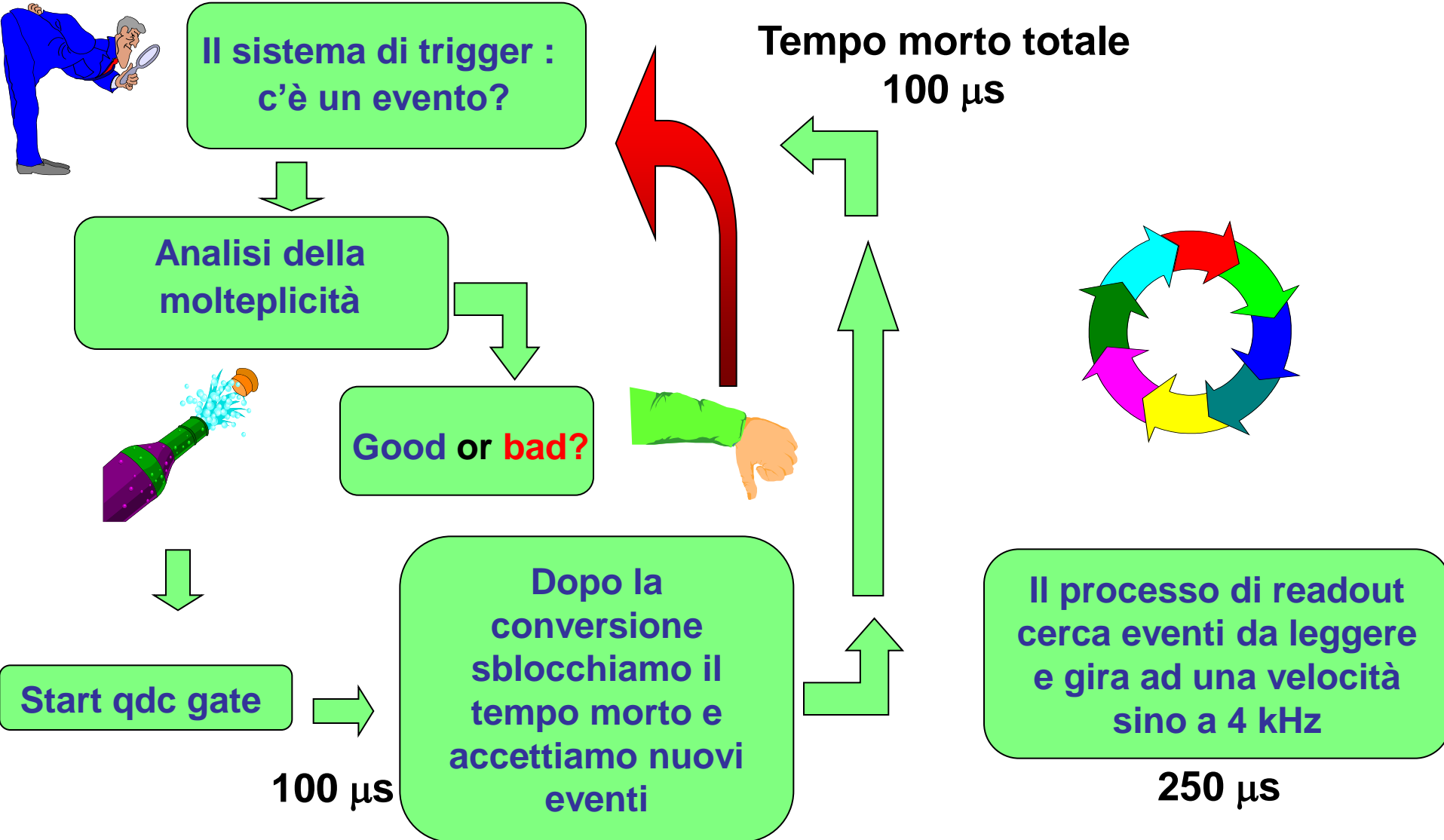
## Conversione e readout sequenziali



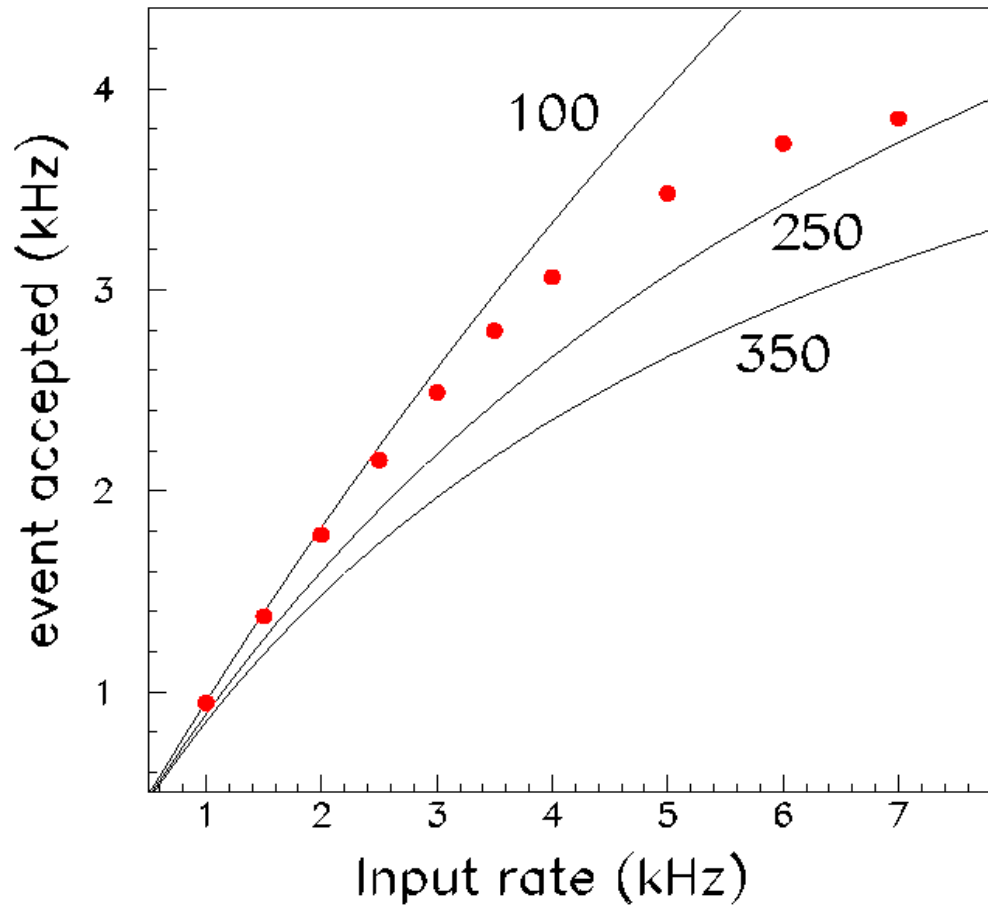


# Ottimizzazione del tempo morto

## Conversione e readout paralleli



# Ottimizzazione del tempo morto



**Il tempo morto segue la curva dei 100  $\mu$ s fino a 1-2 kHz poi si curva per saturare verso i 4 kHz che è il massimo rate sopportabile dal sistema di readout**

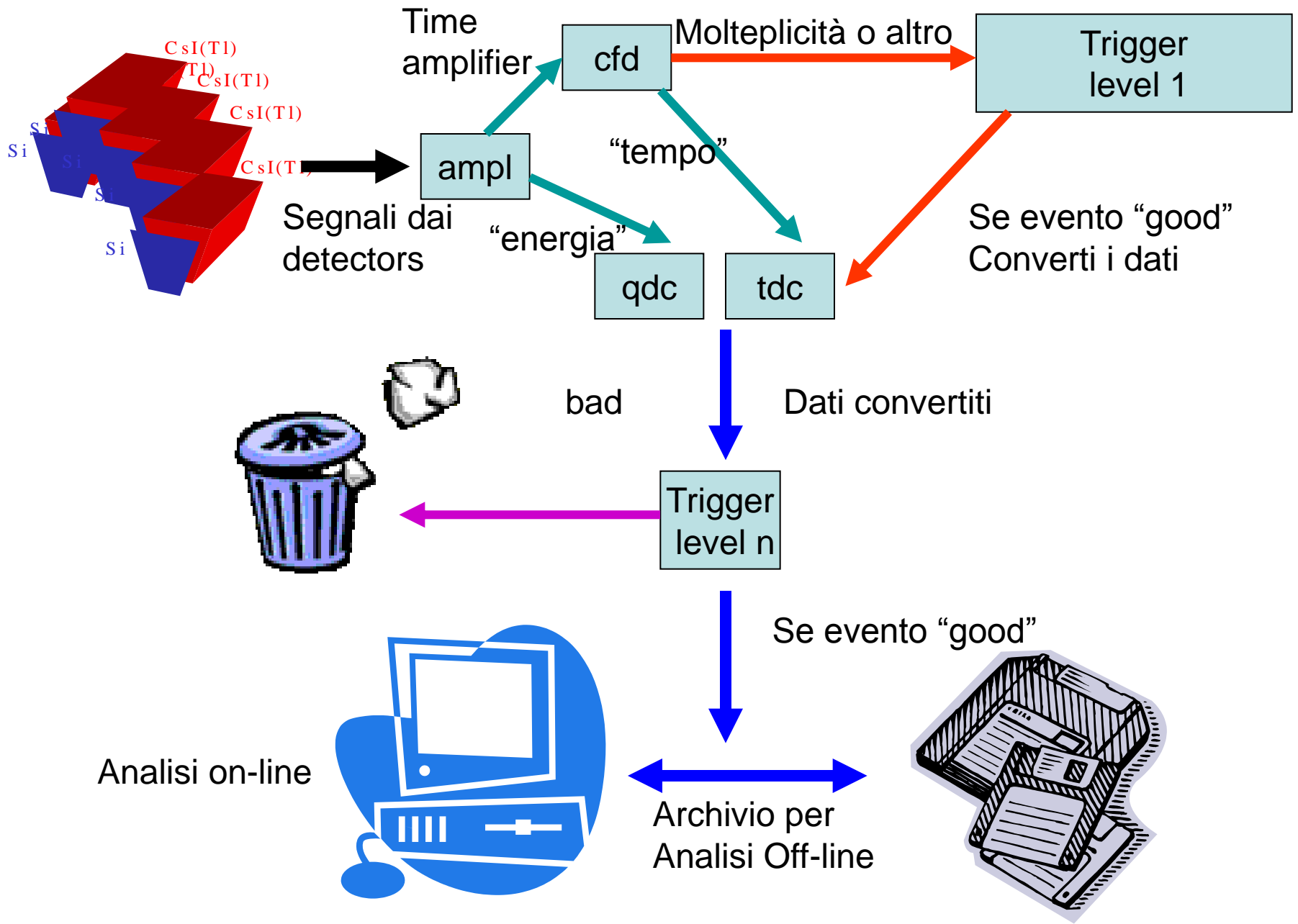
## Ottimizzazione del tempo morto

**La parallelizzazione della conversione e del readout si è realizzata in CHIMERA sfruttando un buffer di 16 eventi dei QDC. Gli eventi convertiti vengono conservati dai QDC nel buffer, il processo di readout quindi può leggere gli eventi dal buffer durante la conversione di altri eventi**

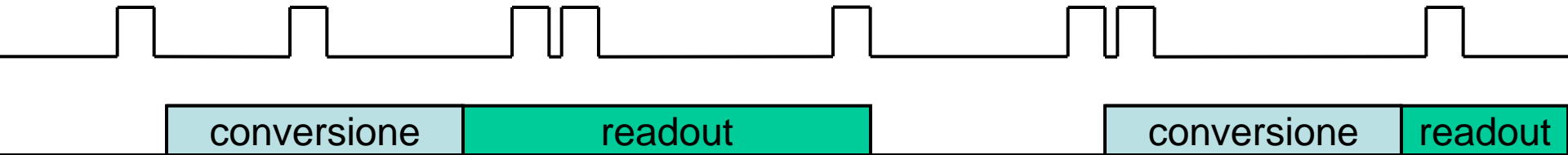
**Il vantaggio ulteriore del readout parallelo alla conversione è che il processo di readout è derandomizzato, può cioè procedere sequenzialmente non è costretto a seguire il rate random di arrivo degli eventi, solo il tempo morto di conversione segue la statistica random di arrivo degli eventi.**

**In questo modo con 1 kHz di eventi in input ne riusciamo ad acquisire 900 abbiamo un tempo morto del 10% solamente guadagnamo almeno 300 eventi/s**

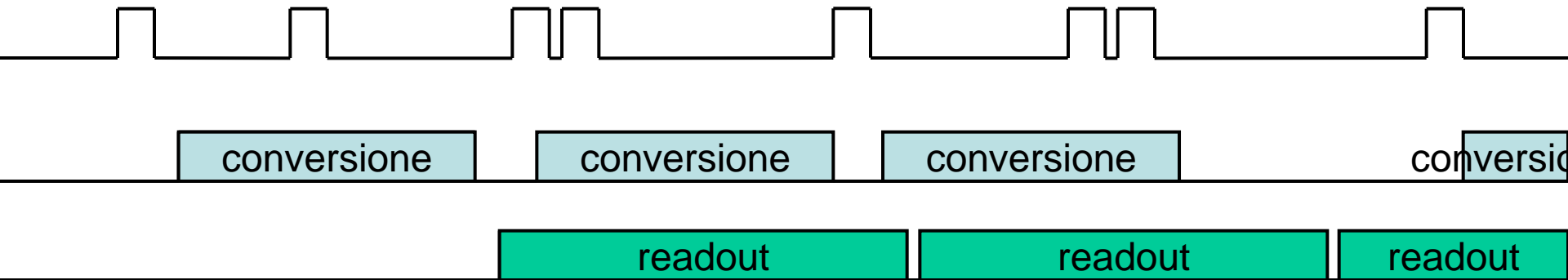
# Schema acq



# Temporizzazione readout seriale/parallelo



**Conversione e readout seguono l'andamento random dei segnali**



**Conversione random - Readout derandomizzato**