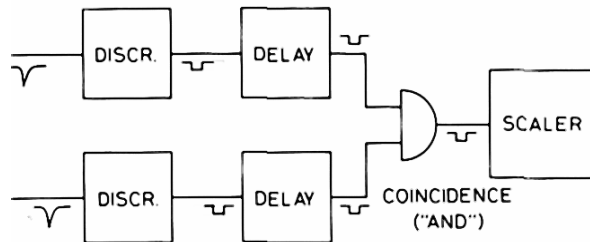


Curva di coincidenza

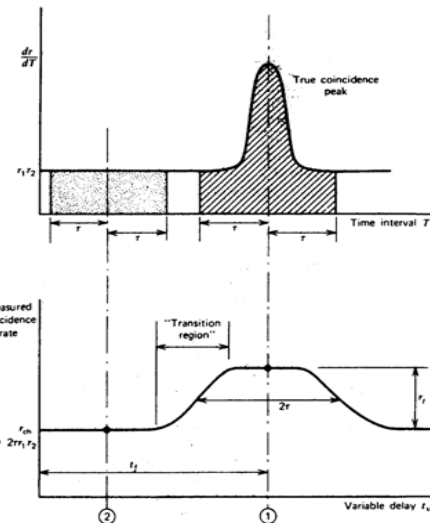
I segnali che formano una coincidenza devono essere sincroni

Errori dovuti a:

- ritardi relativi sbagliati
- coincidenze accidentali



PGI 2005 lect_5_1



Variando il ritardo relativo dei segnali si ottiene una curva di ritardo. Questa curva dipende dalla natura dei segnali stessi e dalla loro frequenza.

PGI 2005 lect_5_2

Le coincidenze vere sono concentrate sotto il picco

Le coincidenze accidentali determinano un innalzamento della linea di base

La larghezza è aumentata dalla durata τ di ciascuno dei segnali di uscita dei discriminatori, supposti uguali

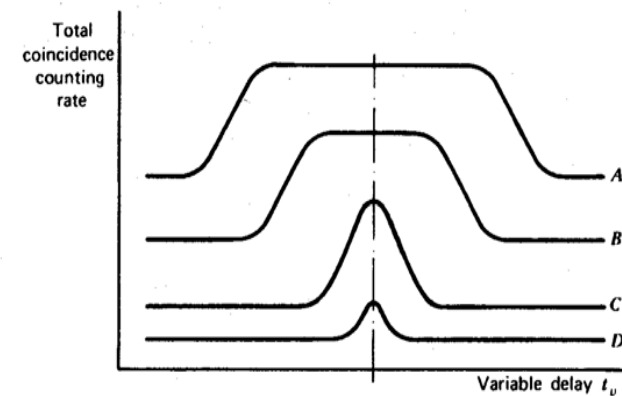
2τ è definita come la risoluzione della coincidenza

Se C_1 e C_2 sono le frequenze (aleatorie) di ciascuno dei canali le coincidenze accidentali C_a sono date da

$$C_a = 2\tau C_1 C_2$$

PGI 2005 lect_5_3

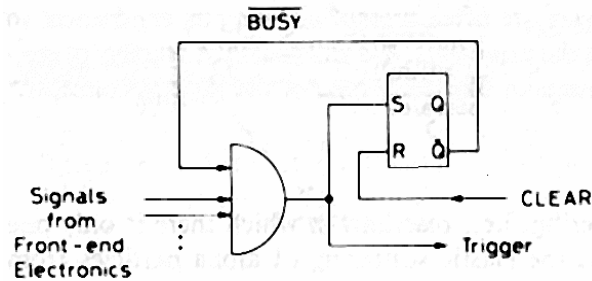
Riducendo τ le coincidenze si perdono



PGI 2005 lect_5_4

Busy

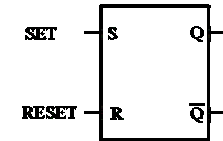
Dopo un *trigger* il sistema richiede un certo tempo per effettuare i calcoli relativi a un *trigger* di livello più elevato o per registrare i dati. Il circuito con un *flip-flop* permette di realizzare questa pausa in attesa di un segnale (*clear*) che rimetta il tutto in funzione



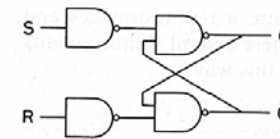
PGI 2005 lect_5_5

Flip-flop

È una memoria a 1 bit

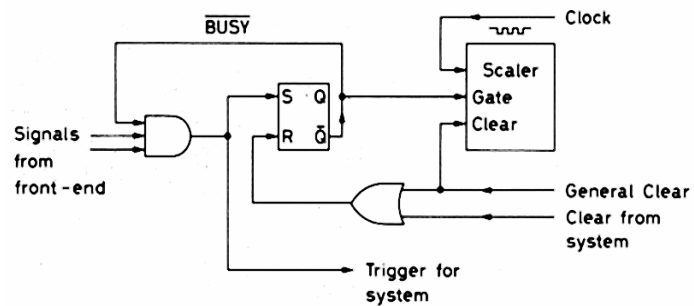


Si può realizzare con quattro **NAND** connessi in controreazione



PGI 2005 lect_5_6

E necessario sapere qual'è la frazione di tempo durante il quale il sistema è inattivo (tempo morto). Si può usare il *busy* associato ad un orologio

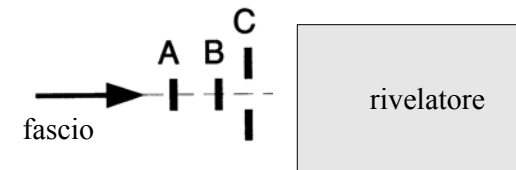


PGI 2005 lect_5_7

VETO

Esempio:
contatore **C** con un buco, in anticoincidenza intorno al fascio

$$A \cdot B \cdot \bar{C}$$



GATE

Esempio:
accettare i segnali in sincronismo con l'acceleratore, solo durante il tempo in cui le particelle arrivano

PGI 2005 lect_5_8

Classificazione dei metodi di *trigger*

• Secondo il processo fisico (1)

➤ Topologia

- Molteplicità delle tracce
- Direzione delle tracce
- Curvatura per misurare il momento
- Evento coplanare o no
- *Minimum bias*

➤ Secondo il processo fisico (2)

➤ Tipo di particella

- Fotoni o pi zero
- Elettroni
- Muoni
- Pioni carichi
- Mesoni K carichi

➤ Energia depositata

➤ *Missing energy*

➤ Vertice o vertice secondario

• Secondo la tecnica elettronica e di calcolo usata (1)

➤ *Fixed flow*

Produce una decisione in un **tempo predeterminato**, indipendente dalla complessità dell'evento

• **Algoritmo logico**

Se usa operazioni di tipo AND e OR (booleane)

• **Algoritmo aritmetico**

Se usa informazioni come numero di contatori, ampiezza (digitizzata) di segnali etc.

➤ Secondo la tecnica elettronica e di calcolo usata (2)

• *Variable flow*

Contiene *loops* logici e programmi: si possono solo definire un tempo di processo minimo e un tempo massimo

• *Data driven*

Un'operazione può essere eseguita soltanto quando tutti i dati necessari sono disponibili: questa condizione è segnalata dai "produttori" di dati con un messaggio semplice come *look-at-me* (LAM), *token* o *data ready*

• *Program driven o demand driven*

Il processore richiede i dati che gli sono necessari

› Secondo il tipo di materiale impiegato

› *Hardware*

Sono costruiti con logica fissa (circuiti elettronici più o meno complessi, cavi e connessioni)
Si applicano al Livello 0, quando c'è, al Livello 1 e talvolta al Livello 2

› *Software*

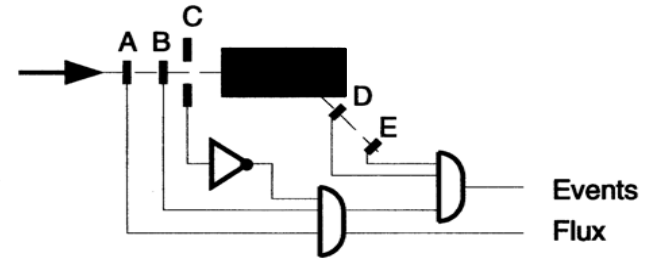
Usano processori programmabili e sono usati dal Livello 2 in su, anche *offline*
Da notare che la flessibilità dei *software triggers* può essere seriamente limitata dalle connessioni non programmabili (*hardware*) alle sorgenti di dati

Esempio 1

Topologia, direzione delle tracce

Fixed flow, logico

Scattering ad angolo fisso



$$\text{Evento} = A \cdot B \cdot \bar{C} \cdot D \cdot E$$

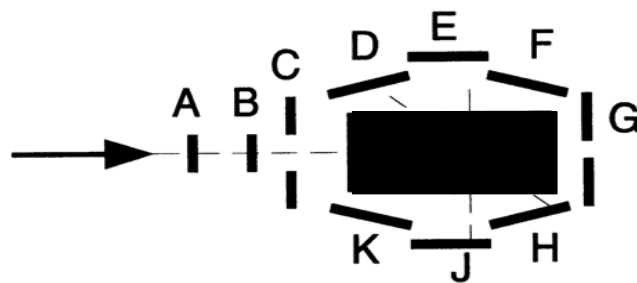
$$\text{Flusso} = A \cdot B \cdot \bar{C}$$

Esempio 2

Topologia, molteplicità delle tracce

Fixed flow, aritmetico

Evento molto inelastico, più di quattro tracce



$$\text{Evento} = A \cdot B \cdot \bar{C} \cdot P$$

ove $P = \text{TRUE}$ se **quattro** dei sette contatori da D a K hanno registrato almeno una particella

P è costruito come

memory look up,

ossia inviando i segnali dei contatori come indirizzi di una memoria nella quale sono stati preregistrati tutti **zeri** salvo un **uno** nelle posizioni

$$15 = D \cdot E \cdot F \cdot G$$

$$23 = D \cdot E \cdot F \cdot H$$

$$39 = D \cdot E \cdot F \cdot J,$$

e così via

Bisogna ritardare A, B e C per aspettare che l'uscita della memoria sia stabile



Esempio 3

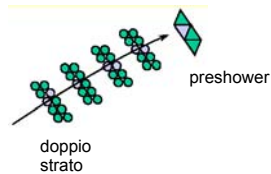
Topologia, curvatura per misurare il momento trasversale

Variable flow, data driven

Selezione di tracce con $p_T > 1.5 \text{ GeV}/c$

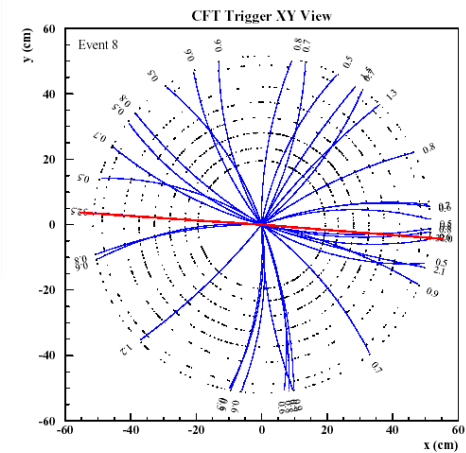
In DØ, il *Central Tracker* è costituito da strati (doppi) cilindrici concentrici di fibre scintillanti. All'esterno del *tracker* si trova un *preshower* cilindrico.

- Il segnale di ciascuna fibra è inviato a un sistema di *gate arrays*
- Le combinazioni di punti, rappresentati da una o due fibre per strato, sono confrontate (*look-up*) in $< 2.6 \mu\text{s}$ con modelli (*patterns*) di tracce precalcolati. Centinaia di *patterns* per $1.5 \text{ GeV}/c$!



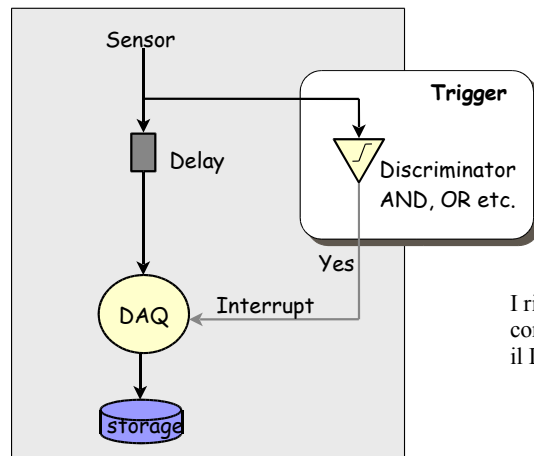
PGI 2005 lect_5 17

Trigger response for $Z \rightarrow ee$ with 4 min.bias



PGI 2005 lect_5 18

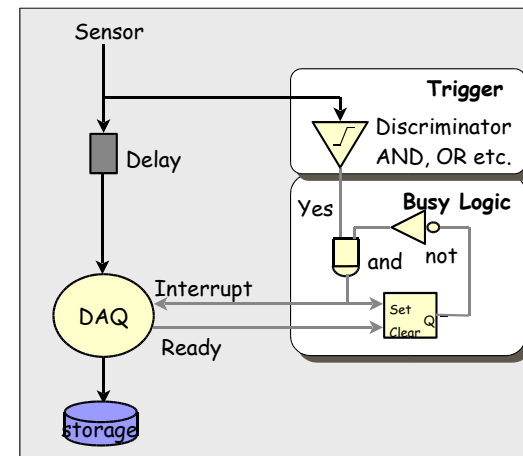
Verso la complessità (0)



I rivelatori devono conservare i dati finché il DAQ non li richiede

PGI 2005 lect_5 19

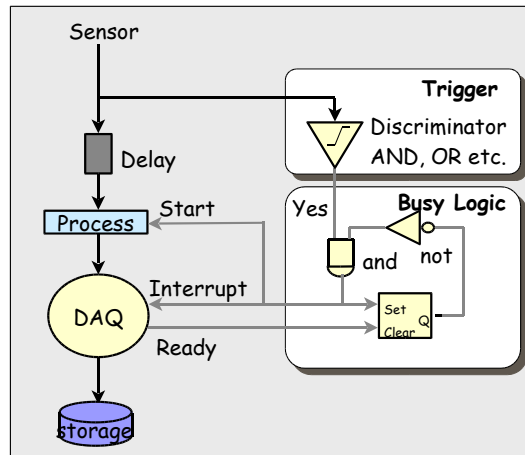
Verso la complessità (1)



Si crea un tempo morto se il DAQ non è pronto

PGI 2005 lect_5 20

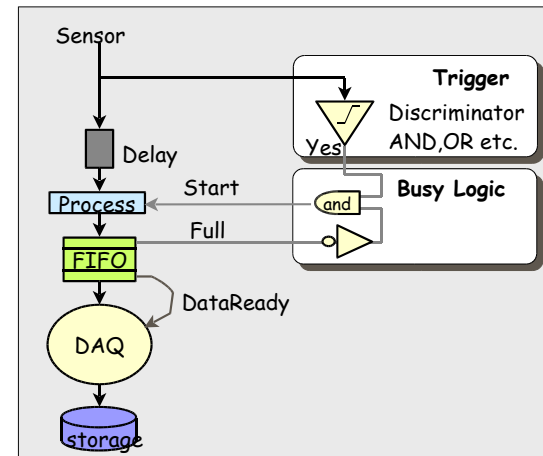
Verso la complessità (2)



Secondo livello di trigger (*process*)

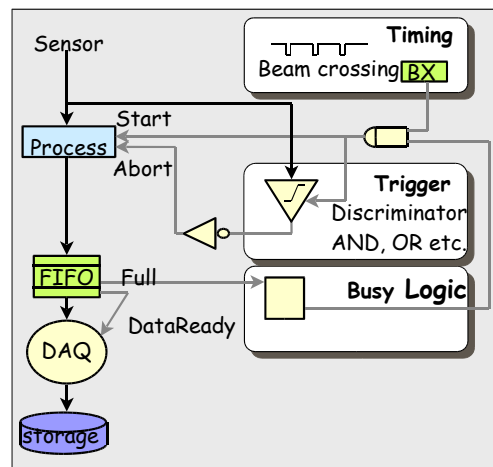
Il sistema è *busy* a causa del livello 2 o a causa del DAQ

Verso la complessità (3)



Livello 2 e DAQ sono disaccoppiati da una FIFO (*derandomizer*)

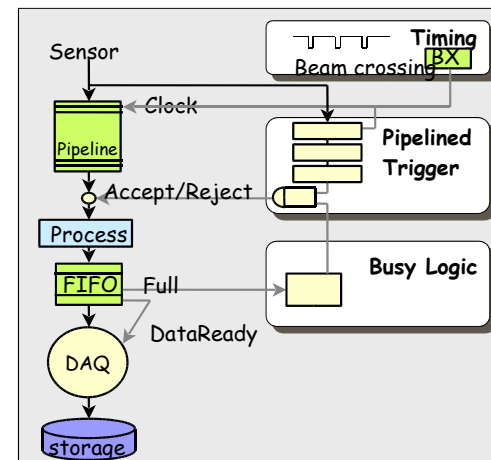
Verso la complessità (4)



Negli anelli di collisione si sa quando gli eventi possono avvenire

Il *trigger* è usato come *veto (abort)*

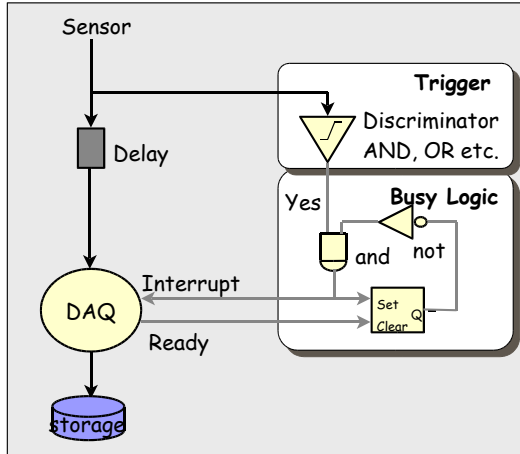
Verso la complessità (5)



Se il tempo di decisione al primo livello e i tempi di trasmissione sono più lunghi della distanza tra i *bunches* sono necessarie *pipelines* nel *front-end*

Analisi del *Trigger* di Secondo Livello

Supponiamo che il *trigger* di primo livello selezioni in media n_e eventi nell'unità di tempo, separati in media da un intervallo $t_e = 1/n_e$.



PGI 2005 lect_5 25

Tutti questi eventi saranno registrati impiegando un tempo medio t_R per ciascuno di essi.

Il numero di *triggers* registrati nell'unità di tempo n_i è per definizione

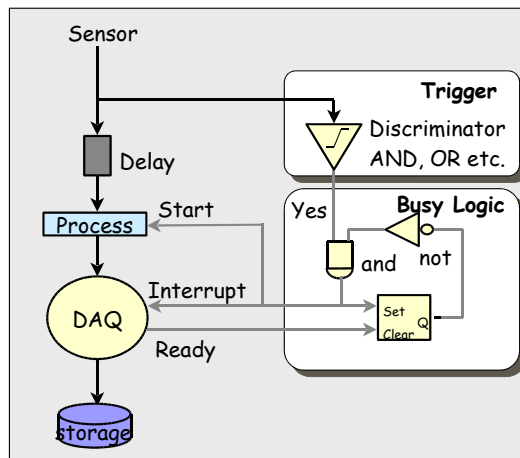
$$1 = n_i(t_R + t_e)$$

La frazione di eventi registrati (efficienza) è

$$E = n_i/n_e = (1 + t_R/t_e)^{-1}$$

PGI 2005 lect_5 26

Il tempo morto t_R si può ridurre introducendo un secondo livello di trigger:



PGI 2005 lect_5 27

Se, con un tempo di calcolo medio al secondo livello $t_p < t_R$, $K n_i$ eventi sono scartati e n_i eventi sono registrati nell'unità di tempo

$$1 = \underbrace{n_i(t_R + t_p + t_e)}_{\text{registrati}} + \underbrace{K n_i(t_p + t_e)}_{\text{scartati}} = \underbrace{n'_i(t'_R + t_e)}_{\text{processati}}$$

dove

$n'_i = n_i(1 + K)$ è il numero di eventi processati nell'unità di tempo

$t'_R = \frac{t_R}{1 + K} + t_p$ è il nuovo tempo morto (ridotto)

PGI 2005 lect_5 28

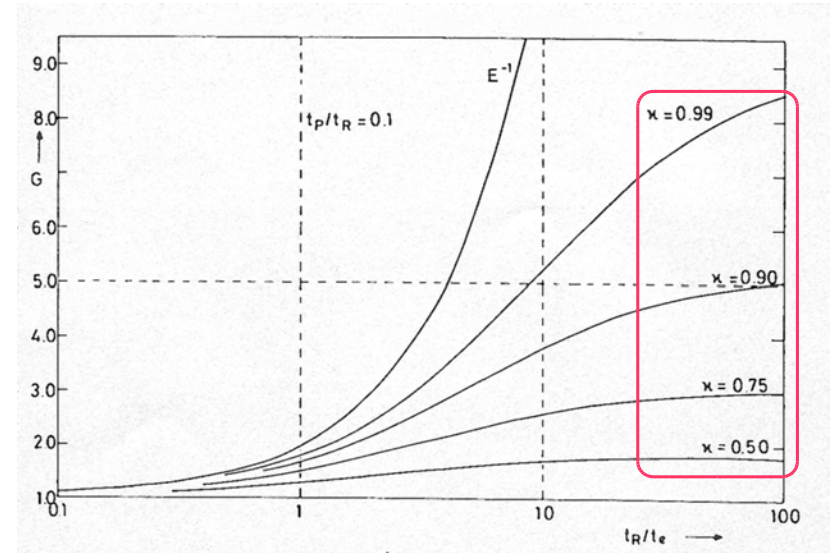
La frazione di eventi processati (nuova efficienza) diventa

$$E' = n \frac{t}{n_e} = \frac{t_e}{\frac{t_R}{1+K} + t_p + t_e} = \frac{t_e}{t_R + t_e} \frac{t_R + t_e}{\frac{t_R}{1+K} + t_p + t_e} = E \left(\frac{t_R + t_e}{\frac{t_R}{1+K} + t_p + t_e} \right)$$

Si ottiene quindi un guadagno G nel numero di eventi processati. Siccome la $E' < 1$, $G < E^{-1}$.

Il numero di eventi registrati è ridotto di un fattore $G/(1+K)$.

La figura presenta il guadagno G in funzione di t_R/t_e per vari valori della frazione di eventi scartati $\kappa = K/(1+K)$ e per un rapporto fisso $t_p/t_R = 0.1$



La tabella presenta il guadagno e la riduzione del volume di dati per diversi valori della frazione di eventi scartati, in due situazioni di rapporti di tempi.

K	$\kappa = \frac{K}{(1+K)}$	$t_e = 1 \text{ ms}$ $t_p = 10 \text{ ms}$ $t_R = 100 \text{ ms}$		$t_e = 0.01 \text{ ms}$ $t_p = 1 \text{ ms}$ $t_R = 100 \text{ ms}$	
		G	$\frac{G}{(1+K)}$	G	$\frac{G}{(1+K)}$
0	0	0.9099	0.9099	0.9901	0.9901
1	0.5	1.656	0.8279	1.961	0.9803
3	0.75	2.805	0.7013	3.845	0.9613
10	0.9091	5.027	0.4570	9.901	0.9001
100	0.9901	8.424	0.0834	50.00	0.4951
1000	0.9990	9.099	0.0091	90.11	0.0900
10000	0.9999	9.173	0.0009	98.05	0.0098

Referenze per l'analisi del secondo livello

G. Lütjens, How Can Fast Programmable Devices Enhance the Quality of Particle Experiments, Topical Conference on the Application of Microprocessors to High Energy Physics Experiments, CERN, Geneva, 1981, CERN 81-07, 236-245

R. Früwirth, M. Regler, R. K.Bock, H. Grote and D. Notz, Data Analysis Techniques for High-Energy Physics, 2nd ed., Cambridge University Press 2000

Regions of Interest

Poichè i dati dei rivelatori non sono disponibili immediatamente e le funzioni di *trigger* sono complesse, si procede per **approssimazioni successive**.

Ogni livello di trigger rappresenta una di queste approssimazioni.

Il trigger di livello $n-1$, avendo trovato un evento potenzialmente interessante, segnala al trigger di livello n in quale area del rivelatore quest'ultimo deve procedere a un'analisi più accurata.

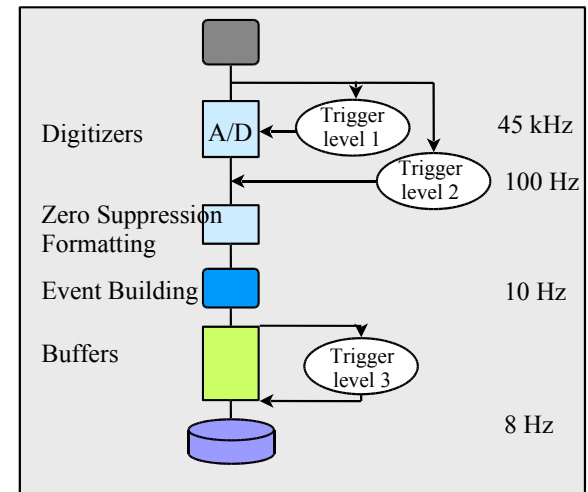
L'area è chiamata **Region of Interest (RoI)**

La procedura implica una trasmissione di informazione abbastanza dettagliata da un livello di *trigger* inferiore a un livello superiore e la possibilità da parte del livello superiore di richiedere, basandosi sull'informazione ricevuta, solo i dati che gli serviranno.

È un esempio di *trigger variable flow, demand driven (program driven)*

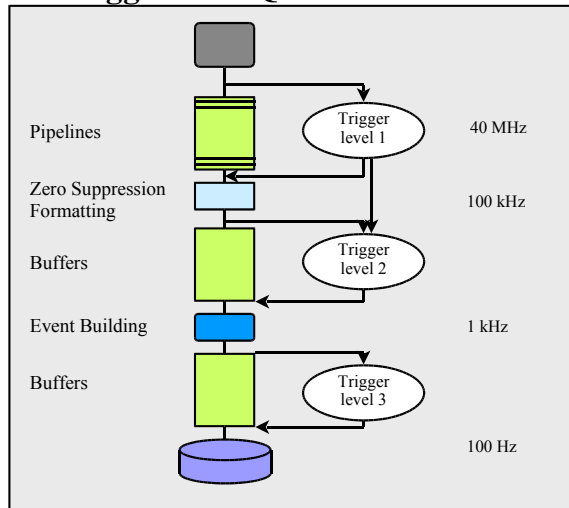
PGI 2005 lect_5 33

Trigger e DAQ al LEP

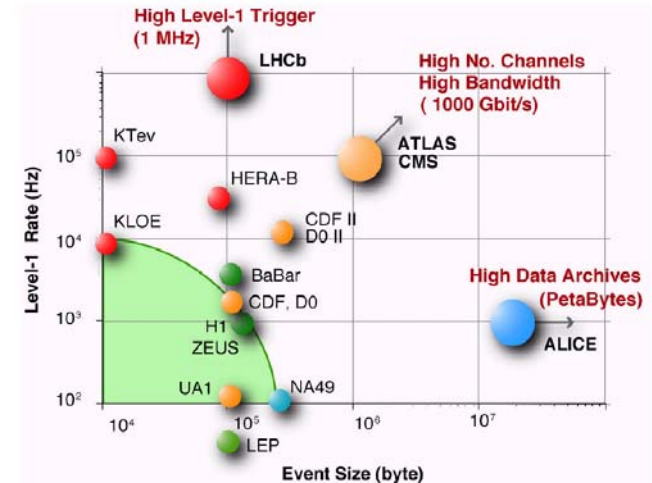


PGI 2005 lect_5 34

Trigger e DAQ all' LHC



PGI 2005 lect_5 35



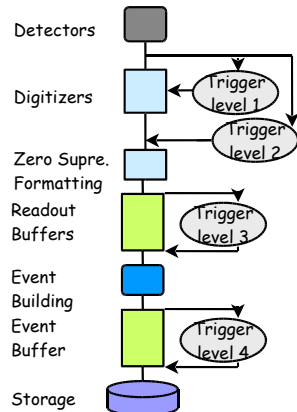
P. Sphicas
Trigger and Data Acquisition

CERN Summer Student Lectures
August 2004

23

PGI 2005 lect_5 36

Livelli di *trigger* in DELPHI (LEP)

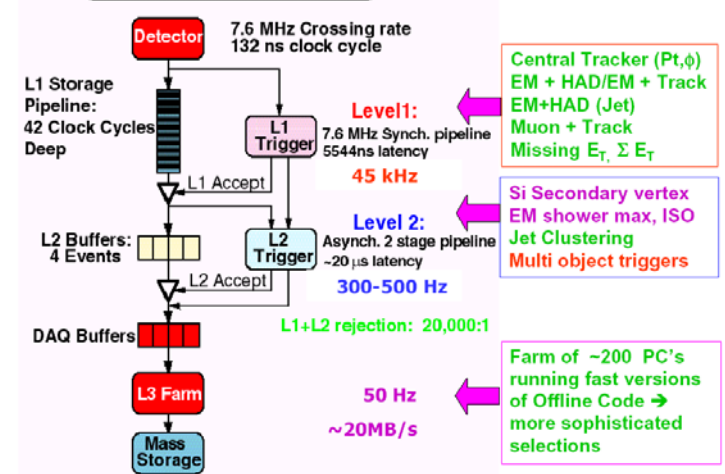


- Livello 1 (3 μ s) *hardware processor*
Informazione dai rivelatori, individualmente:
Energia: calorimetri EM e adronico
Tracce: numero, coincidenze
Muoni: calorimetro adronico e camera a mu
Luminosità: somme analogiche
- Livello 2 (36 μ s) processore hardware
Coincidenze:
Accetta solo tracce che provengono dal punto di interazione.
Rigetta le interazioni beam-gas; usa la TPC
- Livello 3 (~ms) in parallelo per ogni rivelatore, processori OS9
Verifica il trigger Livello 2 con dati digitizzati
- Livello 4 (~ms) un piccolo cluster (farm) di 3 processori alpha
Recostruisce gli eventi usando tutti i dati
Scarta gli eventi vuoti
Etichetta (tagging) le topologie interessanti

PGI 2005 lect_5 37

CDF run 2b

Dataflow of CDF "Deadtimeless" Trigger and DAQ

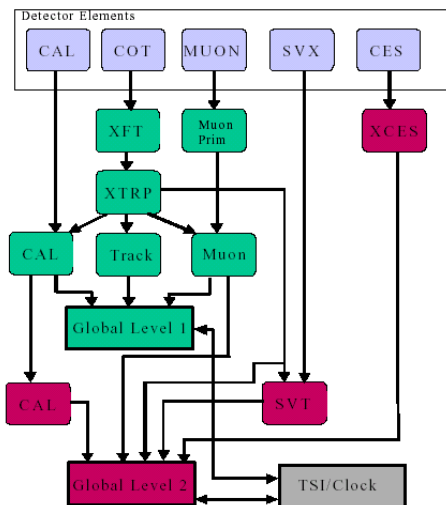


Richard E. Hughes,

FPCP 2002, 05/16-18/2002 p. 9
PGI 2005 lect_5 38

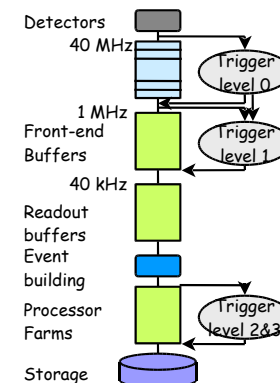
CDF run 2b

- Data samples every 132 ns
- Level 1 trigger decision every 132 ns
- Tracking trigger needs to find tracks with $P_t > 1.5$ GeV/c at every crossing, hence the name: **eXtremely Fast Tracker**



PGI 2005 lect_5 39

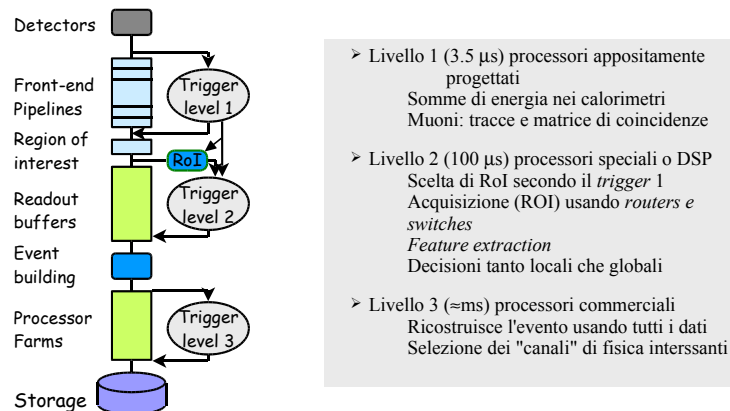
Livelli di *trigger* in LHCb (LHC)



- Livello 0 (4 μ s) processori appositamente progettati
High p_T per elettroni, muoni e adroni
Veto al pileup.
- Livello 1 (1000 μ s) processori speciali
Topologia di vertici primari e secondari
Connette la posizione nel calorimetro con le tracce
- Livello 2 (=ms) processori commerciali
Raffina il livello 1
Rigetta il rumore di fondo
- Livello 3 (=ms) processori commerciali
Ricostruzione degli eventi
Selezione dei "canali" di fisica interessanti

PGI 2005 lect_5 40

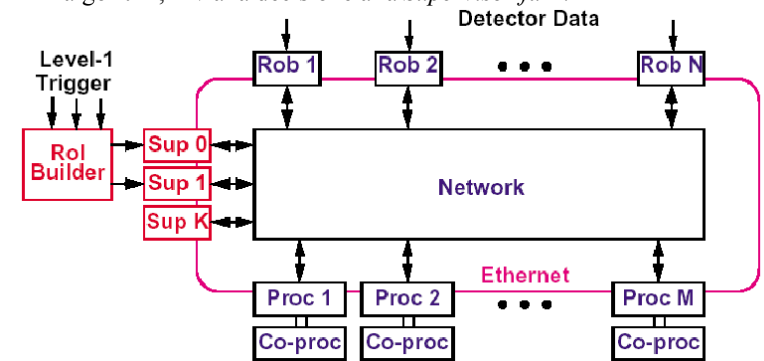
Livelli di *trigger* in ATLAS (LHC)



PGI 2005 lect_5 41

RoI e Livello 2 in ATLAS

- **RoI Builder**: correla i puntatori forniti dal Livello 1
- **Supervisor farm**: riceve l'informazione, assegna l'evento ai processori della *Processor farm*, distribuisce i risultati ai **Rob**
- **Processor farm**: richiede e riceve i dati dai **Rob**, esegue gli algoritmi; Invia la decisione alla *Supervisor farm*



PGI 2005 lect_5 42

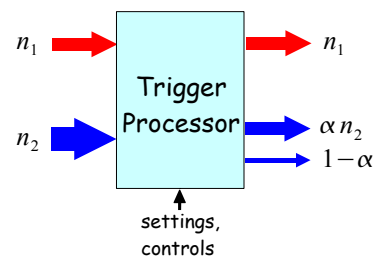
Processore* (o controllore) di *Trigger*

Se si vogliono acquisire simultaneamente due tipi di eventi (canali) prodotti con tasso diverso, può essere necessario limitare il tasso di uno dei due.

Se i tassi entranti sono n_1 e n_2 si riduce il secondo flusso di eventi n_2 di un fattore α che può essere:

- . costante
- . funzione di n_1 e n_2
- . funzione di n_1 e n_2 e di altri parametri come luminosità, tempo etc.

È necessario trasmettere informazioni sugli eventi scartati $1 - \alpha$



* attenzione alla terminologia!
 processore che organizza il *trigger*, come qui,
 o processore che esegue i calcoli di *trigger*

PGI 2005 lect_5 43

La situazione si complica quando:

- . i canali in ingresso sono molti e di ciascuno si conserva la frazione α_i , che può cambiare in modo dinamico
- . esistono canali **prioritari**
- . ciascun canale è trattato in seguito (HLT) con un tempo di processo t_{pi} scartandone una frazione $1 - \beta_i$
- . ci sono perdite per tempo morto nelle code di ingresso ai processori successivi (HLT) e al sistema di acquisizione dati. (DAQ)

Il processore di *trigger* deve funzionare alla velocità di ingresso degli eventi, ma non è necessariamente solo una scatola di elettronica rapida, per quanto con questa tenda ad essere identificato: può essere costruito con vari elementi *hardware* e *software* opportunamente distribuiti.

Il processore di *trigger* produce un'immagine istantanea essenziale per il calcolo del tempo in cui l'esperimento è sensibile (**live time**) e quindi per il calcolo delle **sezioni d'urto** e per le **calibrazioni**. Deve quindi ricevere e trattare anche i segnali di orologio e di *gate* e i *busy* locali e generali.

PGI 2005 lect_5 44

