



Università degli Studi di Roma
“La Sapienza”

Facoltà di Scienze Matematiche, Fisiche e Naturali

Corso di Laurea in Fisica

Studio del sistema di allineamento
temporale tramite impulsaggio
dell'elettronica di FE delle camere a muoni
nell'esperimento LHCb

Relatori:
Dott. Valerio Bocci
Prof. Franco Meddi

Laureando:
Francesco Messi
matricola 688429

Anno Accademico 2006-2007

Tesine

Tesina 1: Le Pulsar come generatrici di onde gravitazionali.

Relatore: Dott. Federica Antonucci

Tesina 2: Radio Detection of Neutrinos.

Relatore: Prof. Antonio Capone

Indice

Indice	vi
Elenco delle figure	xi
Elenco delle tabelle	xiii
Introduzione	1
1 L'esperienza LHCb	3
1.1 La violazione di CP	3
1.2 La violazione di CP nel Modello Standard	4
1.3 La violazione di CP nei mesoni B	6
1.4 Il Collisore Adronico LHC (<i>Large Hadron Collider</i>)	9
1.4.1 La struttura di LHC	9
2 I Rivelatori di LHCb	13
2.1 I processi di <i>Trigger</i> e <i>Tagging</i>	14
2.1.1 Il processo di <i>Trigger</i>	14
2.1.2 Il processo di <i>Tagging</i>	16
2.2 Il rivelatore di vertice (<i>VELO</i>)	17
2.3 Il sistema di tracciamento	18
2.4 I rivelatori RICH (<i>Ring Imaging Cherenkov</i>)	19
2.5 Il sistema di calorimetri	20
2.6 Il Rivelatore di Muoni	20
3 L'elettronica del rivelatore di Muoni	23
3.1 L'elettronica di <i>Front End</i> : le schede <i>CARDIAC</i>	25
3.1.1 Il <i>chip</i> <i>CARIOCA</i>	26
3.1.2 Il <i>chip</i> <i>DIALOG</i>	27
3.2 Le schede <i>IB</i> (" <i>Intermediate Board</i> ")	28
3.3 Il Sistema di Controllo (<i>ECS</i>)	29
3.3.1 L' <i>ELMB</i> (<i>Embedded Local Monitor Board</i>)	30
3.3.2 L' <i>ODE-System</i> : le schede <i>Off-Detector Electronics</i>	32
3.3.3 L' <i>SB-System</i> : le schede <i>Service Board</i> e il modulo <i>Pulse Distribution Module</i>	33
3.4 Il sistema <i>TFC</i> (<i>Timing and Fast Control</i>)	36
3.4.1 Specifiche richieste	36

4	Il Pulse Distribution Module	39
4.1	Generalità	39
4.1.1	Funzioni	40
4.1.2	Architettura	42
4.2	Componenti	43
4.2.1	Personalizzazione della scheda ELMB	43
4.2.2	Il sistema TTC	44
4.2.3	Il <i>chip</i> FPGA	46
4.2.4	Registri I ² C: i PCF	50
4.3	Funzionalità ed Utilizzo	52
4.3.1	Il <i>firmware</i> implementato nella scheda ELMB	52
4.3.2	Il <i>firmware</i> implementato nel <i>chip</i> FPGA	53
4.4	Primo sistema di test: TTCvi e TTCvx	55
4.5	Secondo sistema di test: ODIN	56
5	Il PVSS (“<i>Prozess Visualisierungs- und Steuerungs-System II</i>”)	59
5.1	Generalità e funzioni del programma PVSS	59
5.2	Il programma PVSS al Cern	63
5.2.1	FRAMEWORK e FWELMB	63
5.3	Configurazioni ed interfaccia per il <i>Pulse Distribution Module</i> : il progetto “PDM”	64
5.3.1	L’OPC CANOpen Server	64
5.3.2	Lo schema del progetto “PDM”	64
5.3.3	L’interfaccia utente del progetto “PDM”	64
5.4	Il progetto finale per l’ECS	69
6	L’allineamento temporale	71
6.1	il “ <i>Coarse Delay</i> ” ed il “ <i>Fine Delay</i> ”	73
6.2	la procedura di test del PDM	75
6.2.1	le misure del <i>jitter</i>	75
6.2.2	Il sistema di misura	79
A	Il Protocollo I²C	85
B	Il CAN-bus ed il protocollo CANOpen	89
B.1	Il CAN-bus	89
B.2	Il protocollo CANOpen	93
C	Gli objects del PDM	97
D	L’OPC (<i>OLE for Process Control</i>)	99
D.1	Il file di configurazione per l’OPC Server	100
	Glossario	103
	Acronimi	107
	Bibliografia	113
	Ringraziamenti	115

Elenco delle figure

1.1	Due triangoli di unitarietà nella parametrizzazione di Wolfenstein con una approssimazione valida fino a $\mathcal{O}(\lambda^5)$	6
1.2	Il collisore adronico LHC (<i>Large Hadron Collider</i>); è visibile la collocazione dei quattro maggiori esperimenti: ALICE, ATLAS, CMS ed LHCb.	10
2.1	Gli angoli polari dei quark b e \bar{b} calcolati dal generatore di eventi PYTHIA. Si vede chiaramente che la distribuzione è fortemente piccata intorno all'angolo nullo ed è quindi conveniente scegliere, per il rivelatore, una geometria a braccio singolo.	13
2.2	Distribuzione dell'impulso per $B_d^0 \rightarrow \pi^+\pi^-$ in 4π e quella misurata rivelando sia π^+ che π^- nello spettrometro a braccio singolo.	14
2.3	Disposizione dei rivelatori nell'esperimento LHCb.	15
2.4	Diagramma di flusso dei quattro livelli di <i>trigger</i> dell'esperimento LHCb. Da una frequenza iniziale di $40MHz$ si arriva ad una frequenza di $200Hz$	15
2.5	Distribuzione del p_t per muoni da $b \rightarrow \mu$, da $b \rightarrow c \rightarrow \mu$ e da altri eventi in cui il B decade in $\pi^+\pi^-$	16
2.6	Distribuzione del parametro d'impatto per kaoni con $p_t > 0.8GeV/c$ dal decadimento del b e da altre fonti, in eventi in cui un B decade in $\pi^+\pi^-$	17
2.7	Il sensore del VELO. La parte dedicata alla misura radiale (a destra) è composta da strisce equidistanti, mentre la parte dedicata alla misura radiale (a sinistra) è composta di strisce radiali ad angoli tra 10° e 20°	18
2.8	Sezione del RICH1 (a sinistra) e del RICH 2 (a destra).	20
2.9	Ricostruzione della traccia di due muoni che colpiscono la stazione M3 nello stesso punto; sono evidenziate in grigio le FOI (<i>Field Of Interest</i>).	22
3.1	Posizione dei sottosistemi dell'elettronica in LHCb	24
3.2	Schema dell'elettronica del rivelatore.	25
3.3	Diagramma dell'elettronica di <i>Front End(FE)</i> . I <i>chip</i> CARIOCA ricevono l'uscita delle camere formando i <i>canali fisici</i> (digitali) che vengono inviati al <i>chip</i> DIALOG. Qui i <i>canali fisici</i> sono allineati temporalmente e messi in OR per andare a formare i <i>canali logici</i>	26
3.4	La scheda <i>CARDIAC</i> : sulla scheda sono montati, su di un lato due <i>chip</i> CARIOCA (foto di sinistra), sull'altro un <i>chip</i> DIALOG (foto di destra).	26
3.5	Schema del <i>chip</i> CARIOCA: i segnali analogici provenienti dal rivelatore di muoni vengono amplificati, confrontati con una soglia programmabile e resi come segnali digitali al resto dell'elettronica dell'esperimento.	27

3.6	Schema del <i>chip</i> DIALOG: i segnali digitalizzati dai CARIOCA vengono allineati temporalmente per compensare i diversi “percorsi temporali” che i <i>canali fisici</i> compiono. Inoltre, viene effettuata una prima riduzione dei <i>canali fisici</i> in <i>canali logici</i>	27
3.7	La Scheda Intermedia (<i>Intermediate Board, IB</i>).	29
3.8	Schema logico delle connessioni del sistema ECS (Experiment Control System) per le camere per i muoni. È indicata la logica di connessione dei vari canali del <i>bus</i> CAN per l’interconnessione dei <i>crate</i> con i calcolatori. Inoltre, saranno presenti altri due calcolatori per il controllo delle tensioni, uno per temperature e pressioni ed uno per la gestione globale del rivelatore.	30
3.9	Schema a blocchi della scheda ELMB	31
3.10	La scheda “ELMB digitale” - lato superiore (a sinistra) e lato inferiore (a destra).	31
3.11	La scheda “ELMB analogica” - lato superiore (a sinistra) e lato inferiore (a destra).	31
3.12	Diagramma di funzionamento delle schede ODE.	33
3.13	L’SB-crate: ogni crate può alloggiare fino a 20 schede SB; al centro è ben visibile la scheda PDM con la connessione per la fibra ottica e i quattro connettori per l’interfacciamento tramite altrettanti <i>bus</i> di tipo CAN. È visibile un <i>crate</i> completo di 20 schede SB e un modulo PDM, utilizzato in laboratorio, e i primi due <i>crate</i> montati a giugno 2007 nel pozzo 8.	34
3.14	Schema dell’SB-crate: la scheda PDM permette la connessione dei calcolatori grazie a quattro <i>bus</i> di tipo CAN. Ogni unità PDM alloggia una scheda ELMB, mentre ogni unità SB ne alloggia quattro. In totale, su ogni SB-crate sono presenti fino a 81 nodi CAN.	34
3.15	La comunicazione fra il Sistema di Controllo dell’Esperimento (ECS) ed il <i>front-end</i> (FE) del rivelatore di muoni avviene tramite la <i>Service Board</i> (SB). Ogni scheda SB dispone di 12 uscite verso altrettanti rami del FE. Ogni ramo può gestire fino a 8 schede CARDIAC	35
3.16	Schema del sistema “ <i>LHCb Online system</i> ”.	36
3.17	Diagramma logico delle funzioni del <i>TFC Readout Supervision</i> . Sono indicate le funzioni base della scheda ODIN.	37
4.1	Il modulo <i>Pulse Distribution Module (PDM)</i>	39
4.2	Schema d’iniezione delle particelle nel collisore LHC: sono previsti 2808 <i>bunch</i> per fascio, il cui intervallo temporale è di 25ns (considerando anche i <i>bunch</i> vuoti.	40
4.3	Secondo schema possibile per l’iniezione delle particelle nel collisore LHC; questo schema è estremamente simmetrico, ma prevede solo 2592 <i>bunch</i> per fascio.	41
4.4	Funzioni principali del modulo PDM	41
4.5	Diagramma a blocchi del modulo PDM	42
4.6	Il bus <i>LVDS-bus</i> nella configurazione con doppia terminazione	43
4.7	Il modulo di ricezione del segnale nel sistema TTC: il TTCrq	46
4.8	Struttura delle interconnessioni locali delle FPGA <i>Pro Asic^{plus}</i> della ditta ACTEL; è visibile il sistema denominato “ <i>Ultra Fast local line</i> ”.	47
4.9	Struttura delle interconnessioni locali delle FPGA <i>Pro Asic^{plus}</i> della ditta ACTEL; è visibile il sistema denominato “ <i>Long Line</i> ”.	48

4.10	Struttura delle interconnessioni locali delle FPGA <i>Pro Asic^{plus}</i> della ditta ACTEL; è visibile il sistema denominato “ <i>High Speed very long line</i> ”.	48
4.11	Struttura delle interconnessioni locali delle FPGA <i>Pro Asic^{plus}</i> della ditta ACTEL; è visibile il sistema denominato “ <i>High Performance Global Network</i> ”.	49
4.12	Schema della <i>tegola logica</i> di una ACTEL <i>Pro Asic^{plus}</i>	50
4.13	schema dello <i>switching gate</i> di una ACTEL <i>Pro Asic^{plus}</i>	50
4.14	Schema di reset delle schede ELMB installate sulle <i>Service Board (SB)</i> : grazie alla logica implementata nelle FPGA dei moduli SB, è possibile eseguire un <i>reset</i> od un <i>power cycle</i> delle schede ELMB presenti sulle SB inviando il giusto segnale dal modulo PDM. La figura di sinistra rappresenta le connessioni nella scheda SB, mentre quella di destra le connessioni nella scheda del PDM.	51
4.15	Diagramma a blocchi dell’FPGA implementata sul modulo PDM: in particolare sono visibili l’ <i>interfaccia I²C slave</i> , per l’accesso ai registri interni, e il <i>multiplexer</i> , per selezionare la sorgente del segnale <i>BC_Pulse</i> .	53
4.16	Misura all’oscilloscopio del segnale <i>BC_Pulse</i> : si vede la generazione del segnale <i>broadcast</i> da parte del sistema TTC (<i>CALIBRATION_A</i>), la successiva ricezione del segnale da parte del <i>chip</i> TTCrq (<i>BRCST_STR1</i>) e generazione dell’impulso <i>BC_Pulse</i> da parte del <i>chip</i> FPGA del modulo PDM; è visibile anche il segnale rilevato all’uscita dei cavi delle schede SB e destinate alle schede CARDIAC (<i>FE_TSTPLS</i>)	54
4.17	Schema dei collegamenti del banco di test: tramite l’applicativo “ <i>CAENV-MEdemo</i> ” è possibile interfacciare il calcolatore al modulo <i>V1718</i> ; tramite il <i>bus</i> VME questi è in grado di pilotare il modulo TTCvi; i segnali generati dal TTCvi sono codificati sulla fibra ottica dal TTCvx e ricevuti dal TTCrq, che provvede a decodificarli e renderli disponibili al modulo PDM.	56
4.18	Schema dei collegamenti tra la scheda ODIN e l’SB-crate, nei test svolti in laboratorio. Il sistema PVSS sviluppato per ODIN, vi si interfaccia tramite rete LAN; i segnali generati dalla scheda ODIN sono codificati sulla fibra ottica dal TTCvx e ricevuti dal TTCrq, che provvede a decodificarli e renderli disponibili al modulo PDM.	57
5.1	Struttura gerarchica del <i>software</i> impiegato; si noti la centralità del programma PVSS II.	60
5.2	Schema logico della struttura del programma PVSS II. Sono indicati i vari moduli che ogni progetto PVSS può utilizzare.	60
5.3	Il programma PVSS è in grado di gestire sistemi complessi, dotati di stazione di controllo e sviluppo anche connessi in remoto secondo qualsivoglia tecnologia.	61
5.4	Ruolo del PVSS in un sistema automatizzato.	62
5.5	I DataPoint type del progetto PDM	65
5.6	Il pannello di accesso per la gestione del modulo PDM: è possibile raggiungere due categorie di pannelli, una per l’amministrazione ordinaria e l’altra per una gestione “esperta” della scheda.	65
5.7	Il pannello “ <i>PULSE</i> ” tramite cui l’utente ordinario può settare la sorgente del segnale <i>BC_Pulse</i> .	66

5.8	Il pannello “ <i>CRATE</i> ” tramite cui l’utente ordinario può eseguire un “ <i>reset</i> ” o un “ <i>power cycle</i> ” delle schede ELMB del <i>crate</i> ; è possibile selezionare una sola scheda, le schede di un’intero modulo SB, di un intero canale CAN o di tutto il <i>crate</i>	66
5.9	Il pannello “ <i>TTCrx</i> ” tramite cui l’utente esperto ha accesso completo ai registri del modulo TTCrq.	67
5.10	Il pannello “ <i>PCF</i> ” tramite cui l’utente esperto può settare il valore dei due registri PCF del modulo PDM; tramite il primo si abilita o meno l’alimentazione del TTCrq, il secondo è usato per la manutenzione delle schede ELMB presenti nel <i>crate</i>	68
5.11	Il pannello “ <i>ACTEL</i> ” tramite cui l’utente esperto può accedere ai registri implementati nella logica del <i>chip</i> FPGA del modulo PDM.	68
5.12	Esempio di utilizzo del parametro \$ nel “pannello padre”(figura superiore) e nel “pannello figlio” (figura inferiore). Grazie a questa sintassi, è possibile passare un parametro da un pannello padre ad uno figlio e generare il nome dei <i>DataPoint</i> in maniera standardizzata per tutti gli apparati dell’esperimento.	70
6.1	Schema logico della procedura di allineamento temporale in assenza di fisica.	73
6.2	Schema della cella TDC implementata nel <i>chip</i> SYNC.	74
6.3	Schema dell’istogrammatore del <i>chip</i> SYNC per l’istogramma del <i>Fine Time Histogram</i> : il range temporale dell’istogramma è di $25ns$ e sarà diviso in 16 parti, una per contatore. La risoluzione dell’istogramma corrisponde così alla risoluzione del TDC: $1.56ns$	74
6.4	Schema logico della procedura di test per l’allineamento temporale dell’elettronica in assenza della fisica dei fasci.	75
6.5	Tempi di propagazione del segnale <i>BC_Pulse</i> dalla sua generazione nel modulo PDM ai connettori del BP: il <i>jitter</i> temporale dei segnali è inferiore ai $100ps$	76
6.6	Tempi di propagazione del segnale <i>BC_Pulse-TSTPLS</i> dalla generazione nel modulo PDM fino ad un metro di cavo dopo le schede SB. Distribuzione “in Y” per SB. Il <i>jitter</i> temporale dei segnali è dell’ordine dei $100ps$	77
6.7	Tempi di propagazione del segnale <i>BC_Pulse-TSTPLS</i> dalla generazione nel modulo PDM fino ad un metro di cavo dopo le schede SB. Distribuzione “in X” per canale.	78
6.8	Tempi di propagazione del segnale <i>BC_Pulse-TSTPLS</i> dalla generazione nel modulo PDM fino ad un metro di cavo dopo le schede SB. Distribuzione di più schede a parità di posizione all’interno dell’ <i>SB-crate</i>	78
6.9	Apparato di misura utilizzato in laboratorio. È stato utilizzato un <i>LeCroy WR104Xi</i> equipaggiato con una sonda <i>AP034</i> ed una <i>PP007</i> , una scheda ODIN completa di interfaccia PVSS, una scheda TTCvx ed un SB-crate completo (20 schede SB e 1 modulo PDM).	79
6.10	Sistema PVSS per la gestione della scheda ODIN: è indicata la procedure per la generazione manuale dei segnali <i>broadcast</i>	81
6.11	Sistema PVSS per la gestione della scheda ODIN: è indicata la procedure per richiamare le configurazione pre-impostate nel sistema.	82
A.1	Schema del funzionamento del protocollo I^2C	85
A.2	il <i>Schema di invio di un pacchetto di dati secondo il protocollo I^2C</i>	85
A.3	Generazione del segnale di <i>Acknowledge</i> nel protocollo I^2C	86

A.4	Trasferimento completo di un pacchetto nel protocollo I^2C	87
B.1	Schema generale del sistema di comunicazione CAN.	90
B.2	Il formato standard per i messaggi su bus di tipo CAN.	91
B.3	Schema del protocollo CANOpen	94
B.4	Diagramma degli stati di un nodo CANOpen.	94
B.5	Relazione tra CANbus, <i>Object Dictionary</i> e applicazioni <i>software</i>	96

Elenco delle tabelle

1.1	Le famiglie di fermioni. Come si nota, le famiglie sono ordinate nel senso crescente della massa.	4
1.2	Il sistema dei quattro mesoni B . Sono riportate la massa e la vita media di ogni particella.	7
1.3	Principali caratteristiche dell'acceleratore adronico LHC , nel caso di fasci di protoni circolanti, messe a confronto con quelle dell'acceleratore $Tevatron$. . .	10
3.1	Rapporto tra <i>canali logici</i> e <i>canali fisici</i> nei quadranti di ogni camera. Si noti che la riduzione maggiore dei canali avviene nelle zone interessate dalle IB.	28
3.2	Specifiche richieste dall'esperimento LHCb per il sistema TFC messe a confronto con quelle offerte dal progetto RD12-TTC.	37
4.1	Lista dei moduli TTC supportati	45
4.2	Elenco dei comandi <i>broadcast</i> usati nel sistema TTC.	55
6.1	Principali caratteristiche della macchina $LeCroy\ WR104Xi$	80
6.2	Parametri relativi al segnale $CALIBRATION_A$, utilizzati per la simulazione del funzionamento del collisore LHC durante le misure effettuate in laboratorio sull'SB-crate.	81

*Bisogna aver rinunciato al buon senso
per non convenire che non conosciamo nulla
se non attraverso l'esperienza.
(François Voltaire, il filosofo ignorante)*

Introduzione

Dove gli strumenti forniti direttamente dalla natura non sono stati sufficienti ad ampliare le proprie conoscenze, l'uomo ha costruito macchine che potessero dargli risposte, con le quali investigare l'infinitamente grande o l'infinitamente piccolo.

Proprio verso quest'ultima strada, si pone la grande macchina costruita al CERN, il collisore adronico LHC (*Large Hadron Collider*). Ormai superata l'indivisibilità dell'atomo, grazie a questo tipo di macchine, si è in grado di far scontrare tra loro diversi tipi di particelle con un'energia tale da scinderle, per piccolissimi istanti, nei loro costituenti fondamentali. Lo studio di quello che accade dopo la collisione di due adroni è lo scopo degli esperimenti situati lungo il percorso dei fasci di LHC.

I valori sperimentali finora ottenuti negli esperimenti di fisica delle alte energie, confermano, entro la risoluzione possibile, i dati teorici previsti dal *Modello Standard*.

Tuttavia, alcune questioni non collimano esattamente con quanto previsto. In particolare, detto modello non è in grado di generare la giusta quantità di *violazione CP*, necessaria a giustificare l'asimmetria materia-antimateria presente nell'universo.

LHCb è un esperimento di fisica delle alte energie, progettato allo scopo di approfondire le attuali conoscenze sulla violazione di CP nell'ambito del Modello Standard, mediante lo studio dei decadimenti rari nei sistemi dei mesoni B neutri.

Grazie alla grande luminosità che verrà raggiunta ($L_{LHCb} = 2 \cdot 10^{32} \text{cm}^{-2} \text{s}^{-1}$), LHCb avrà a disposizione più di 10^{12} eventi all'anno. Una statistica così elevata permetterà di ridurre gli errori sperimentali sulle misure già effettuate, ottenendo conferme o meno della validità della teoria finora affermata.

Il mio lavoro di tesi, svolto nel gruppo LHCb della sezione di Roma dell'INFN, è stato incentrato sul rivelatore di muoni e sul sistema di test per l'allineamento temporale dell'elettronica di *Front-End* del rivelatore; in particolare sul modulo PDM (*Pulse Distribution Module*), l'ottimizzazione del suo *software* e la sua interfaccia utente.

Ho strutturato questo elaborato di tesi in sei capitoli:

Nel primo capitolo ho inserito uno sguardo generale sull'esperimento LHCb, dalla teoria della Fisica su cui si basa l'esperimento (la violazione di CP, la sua interpretazione nel Modello Standard, la matrice CKM ed i triangoli di unitarietà), alla struttura del collisore adronico LHC; il secondo capitolo l'ho dedicato ai rivelatori presenti nell'esperimento LHCb (i processi di *trigger* e *tagging*, il VELO, il sistema di tracciamento, i RICH, il sistema dei calorimetri), dedicando l'intero terzo capitolo al rivelatore di muoni (l'elettronica di FE, le schede IB, l'ECS ed il TFC).

Gli ultimi tre capitoli riportano il lavoro fattivamente svolto nel periodo di tesi: il quarto capitolo è interamente dedicato al modulo PDM, di cui ho contribuito a conclude-

re la parte *software* (*firmware* dell'ELMB e logica dell'FPGA), fino alla versione attuale; il quinto capitolo è dedicato al programma PVSS, tramite cui ho sviluppato l'interfaccia utente per la gestione del modulo PDM; infine l'ultimo capitolo riguarda la procedura di allineamento temporale, scopo principe di tutto il lavoro svolto.

Ho ritenuto utile aggiungere anche quattro appendici:

Le prime due riguardano i protocolli di comunicazione utilizzati dalla logica del modulo PDM: l'appendice A riporta una breve trattazione sul protocollo I²C, mentre l'appendice B una sul *bus* di comunicazione di tipo CAN e sul protocollo CANOpen;

Le ultime due sono inerenti al lavoro "*software*" svolto sul modulo PDM e riportano, l'*Object Dictionary* del PDM (appendice C) e una breve trattazione, seguita dal listato del file di configurazione, dell'OPC Server (appendice D).

Capitolo 1

L'esperimento LHCb

Al momento del Big Bang, la nascita del nostro universo, materia e antimateria furono creati in ugual misura; ma per qualche ragione, l'antimateria scomparve e il nostro universo è ora composto dalla materia rimanente. Questo fatto comporta un notevole interesse scientifico; infatti, le simmetrie e le relative leggi di conservazione costituiscono uno degli aspetti fondamentali della descrizione fisica della natura, insieme alle particelle elementari e alle loro interazioni [1, 2]. I continui progressi nell'ambito della ricerca dei componenti fondamentali della materia e nello studio delle forze che la governano, attraverso l'analisi delle particelle elementari, sono culminati nella costruzione della teoria nota con il nome di *Modello Standard* [3, 4, 5].

L'asimmetria tra materia ed antimateria, è spiegabile grazie alla *violazione di CP*, scoperta nel 1964 Christenson, Cronin, Fitch e Turlay [6] nel sistema dei K neutri osservando il decadimento del K_L in $\pi^+\pi^-$. Il Modello Standard non è però in grado di generare la quantità di violazione di CP necessaria per giustificare l'entità di tale asimmetria [8]; questo limite rappresenta un forte impulso alla ricerca di una "nuova fisica".

La realizzazione di una nuova generazione di acceleratori di particelle in grado di produrre grandi quantità di quark pesanti, come il quark b (*bottom* o *beauty*), apre nuove possibilità di investigazione su questo fenomeno. Il sistema di mesoni B neutri, costituiti da un anti-quark di tipo b e da un quark di tipo d o s , è del tutto analogo a quello dei mesoni K , eccetto per il fatto che la massa del quark b è circa 30 volte maggiore di quella del quark s [9]; questa differenza permette di ridurre le incertezze nelle previsioni teoriche e, in particolar modo per alcuni canali di decadimento, le previsioni del Modello Standard sono estremamente precise, perchè poco influenzate dalle interazioni forti [10].

Risulta dunque chiaro come il sistema di mesoni B neutri possa rappresentare il luogo ideale per approfondire la conoscenza del Modello Standard e verificarne la consistenza con il fenomeno della violazione di CP [11].

1.1 La violazione di CP

Nell'ambito del Modello Standard, la materia è interamente composta da due diversi tipi di particelle elementari, le une note come costituenti fondamentali della materia (i *fermioni*) e le altre come particelle di scambio (i *bosoni*).

I fermioni hanno *spin* $1/2$ e si dividono in due categorie: i *leptoni*, soggetti alle interazioni elettromagnetiche ed a quelle deboli, ed i *quark*, soggetti anche alle interazioni forti. All'interno di ognuna di queste categorie, esistono tre famiglie, ciascuna composta da due membri, secondo lo schema illustrato in tabella 1.1. Per i quark, i membri di ogni famiglia

famiglia	1	2	3	Q/e
quark	$u(\approx 3MeV/c^2)$	$c(\approx 1.2GeV/c^2)$	$t(\approx 175GeV/c^2)$	2/3
	$d(\approx 6MeV/c^2)$	$s(\approx 130MeV/c^2)$	$b(\approx 4.2GeV/c^2)$	-1/3
leptoni	$e(\approx 0.5MeV/c^2)$	$\mu(\approx 105MeV/c^2)$	$\tau(\approx 1.8GeV/c^2)$	-1
	$\nu_e(\approx 0)$	$\nu_\mu(\approx 0)$	$\nu_\tau(\approx 0)$	0

Tabella 1.1: Le famiglie di fermioni. Come si nota, le famiglie sono ordinate nel senso crescente della massa.

sono distinti in base ad una caratteristica chiamata *sapore (flavor)*. Per i leptoni, invece, si distingue tra leptone carico e suo neutrino.

Le interazioni forti forniscono ai quark un ulteriore numero quantico, noto come *colore*, che può assumere tre differenti valori; poichè in natura non c'è manifestazione di questo numero quantico, è necessario che i quark siano confinati in particelle prive di *colore*: gli *adroni*. Gli adroni si dividono in *barioni*, fermioni composti da tre quark di colore diverso, e *mesoni*, bosoni composti da coppie quark-antiquark.

Le particelle appartenenti al secondo tipo sono note come particelle di scambio e, lasciando da parte le interazioni gravitazionali, sono tutte bosoni di spin 1: il *fotone* (γ), particella scambiata nelle interazioni elettromagnetiche, i *gluoni* (g), relativi alle interazioni forti, e i tre bosoni W^\pm e Z^0 che mediano le interazioni deboli.

Osserviamo che, nelle interazioni forti, descritte dalla *Cromodinamica Quantistica (QCD)*, il colore gioca lo stesso ruolo giocato dalla carica elettrica nelle interazioni elettromagnetiche.

Consideriamo gli operatori C , P e T che cambiano il segno, rispettivamente, alla carica, alle coordinate spaziali ed a quelle temporali. Secondo il teorema di *CPT*, applicando tutte e tre queste simmetrie il risultato non cambia, ossia si ottengono due sistemi equivalenti, indistinguibili fra loro.

Gli esperimenti dimostrano che nelle interazioni deboli, nè l'inversione di parità P nè quella di carica C conservano le simmetrie; nel 1964 James Cronin e Val Fitch, studiando il decadimento dei Kaoni neutri al *US Brookhaven laboratory*, scoprirono che neanche la combinazione di simmetria carica-parità era conservata [6]: questo fenomeno fu chiamato *violazione di CP*. Secondo Sacharov la violazione di CP è condizione necessaria per uno squilibrio nella simmetria materia-antimateria.

1.2 La violazione di CP nel Modello Standard

Nell'ambito del Modello Standard, la violazione di CP si interpreta secondo la *matrice di Cabibbo-Kobayashi-Maskawa (CKM)* [12], una matrice unitaria 3×3 i cui elementi descrivono l'accoppiamento nelle interazione deboli fra i vari quark. La matrice CKM collega gli autostati delle interazioni elettrodeboli dei quark *down*, *strange* e *bottom* (d' , s' , b'), con i loro autostati di massa (d , s , b) mediante la seguente trasformazione unitaria:

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \begin{pmatrix} d \\ s \\ b \end{pmatrix} = V_{CKM} \begin{pmatrix} d \\ s \\ b \end{pmatrix} \quad (1.1)$$

Gli elementi della matrice CKM descrivono gli accoppiamenti tra correnti cariche, come si vede scrivendo la lagrangiana delle interazioni deboli in termini degli autostati elettrodeboli:

$$\mathcal{L}_W^{cc} = -\frac{g}{\sqrt{2}}(\bar{u}, \bar{c}, \bar{t})\gamma^\mu V_{CKM} \begin{pmatrix} d \\ s \\ b \end{pmatrix} W_\mu^\dagger + h.c. \quad (1.2)$$

dove W_μ^\dagger descrive il bosone carico W^\pm e $h.c.$ rappresenta l'hermitiano coniugato del primo addendo.

La matrice CKM è caratterizzata da nove parametri complessi, ossia diciotto variabili reali, ma alcune condizioni, fra cui l'unitarietà, riducono il numero totale dei parametri indipendenti a quattro: tre angoli e una fase. In sostanza, la matrice CKM è unicamente determinata da quattro parametri reali, di cui uno è certamente una fase. È proprio quest'ultima che consente di generare la violazione di CP nell'ambito del Modello Standard, come fu evidenziato da Kobayashi e Maskawa nel 1973: se la fase fosse nulla non si avrebbe Violazione di CP.

La matrice CKM descrive gli accoppiamenti tra correnti cariche; non tutte le transizioni possibili, però, sono egualmente favorite: sperimentalmente si osserva che le transizioni tra famiglie diverse sono soppresse (*regola di Cabibbo*), mentre risultano favorite quelle tra elementi della stessa famiglia.

I coefficienti della matrice CKM vanno misurati sperimentalmente; tra le diverse caratterizzazioni possibili della matrice CKM, la *parametrizzazione di Wolfenstein* [13] risulta essere una delle più utili; essa corrisponde ad uno sviluppo in potenze della quantità $\lambda \equiv |V_{us}| \equiv \sin \theta_C \approx 0.22$, dove θ_C è l'*angolo di Cabibbo* [14]:

$$V_{CKM} \approx V_{CKM}^{(3)} + \delta V_{CKM} \quad (1.3)$$

dove lo sviluppo al terz'ordine in λ è dato da:

$$V_{CKM}^{(3)} = \begin{pmatrix} 1 - \frac{1}{2}\lambda^2 & \lambda & A\lambda^3(\rho - i\eta) \\ -\lambda & 1 - \frac{1}{2}\lambda^2 & A\lambda^2 \\ A\lambda^3(1 - \rho - i\eta) & -A\lambda^2 & 1 \end{pmatrix} \quad (1.4)$$

ed il resto da:

$$\delta V_{CKM} = \begin{pmatrix} 0 & 0 & 0 \\ -iA^2\lambda^5\eta & 0 & 0 \\ \frac{A}{2}\lambda^5(\rho + i\eta) & (\frac{1}{2} - \rho)A\lambda^4 - iA\lambda^4\eta & 0 \end{pmatrix} \quad (1.5)$$

Dai decadimenti dei mesoni B sono stati misurati gli elementi della matrice $|V_{cb}| = 0.0402 \pm 0.0019$ e $|V_{ub}/V_{cb}| = 0.090 \pm 0.025$, dai quali si ricava:

$$A = 0.823 \pm 0.042 \\ \sqrt{\rho^2 + \eta^2} = 0.41 \pm 0.11$$

Esplicitando la condizione di unitarietà $V_{CKM}V_{CKM}^\dagger = 1$, si ottengono nove equazioni, delle quali due sono di particolare interesse per la fisica del b :

$$V_{ud}V_{ub}^* + V_{cd}V_{cb}^* + V_{td}V_{tb}^* = 0 \quad (1.6)$$

$$V_{tb}V_{ub}^* + V_{ts}V_{us}^* + V_{td}V_{ud}^* = 0 \quad (1.7)$$

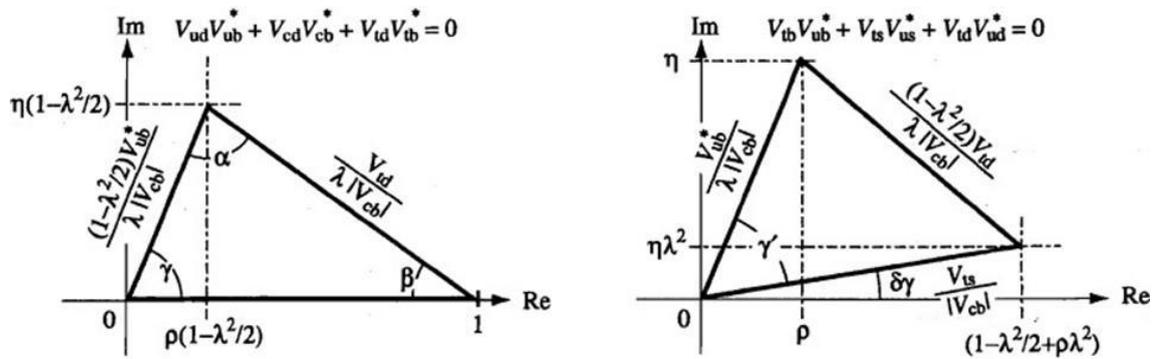


Figura 1.1: Due triangoli di unitarietà nella parametrizzazione di Wolfenstein con una approssimazione valida fino a $\mathcal{O}(\lambda^5)$.

Se ipotizziamo di rappresentare ogni addendo di ciascuna equazione come un vettore nel piano complesso, otteniamo dei triangoli, mostrati in figura 1.1, detti *triangoli di unitarietà*.

I loro angoli sono legati alle fasi degli elementi della matrice CKM dalle seguenti relazioni:

$$\begin{aligned}\beta &= -\arg(V_{td}) \\ \gamma &= -\arg(V_{ub}) \\ \alpha &= \pi - \beta - \gamma \\ \delta\gamma + \pi &= \arg(V_{ts})\end{aligned}$$

ed, in funzione dei parametri di Wolfenstein, si ottiene:

$$\begin{aligned}\tan\beta &= \frac{\eta}{1-\rho} \\ \tan\gamma &= \frac{\eta}{\rho} \\ \tan\delta\gamma &= \lambda^2\eta\end{aligned}$$

Lo studio della violazione di CP nei decadimenti dei mesoni B permetterà di misurare gli angoli α, β e γ e così di costruire i triangoli di unitarietà; utilizzando gli elementi di matrice per determinare i lati, potremo anche verificare la consistenza del Modello Standard nella descrizione del *mixing* dei quark nelle interazioni deboli.

Lo scopo di LHCb è quello di sovra-determinare i parametri dei Triangoli di Unitarietà del Modello Standard tramite misure, di precisione e con alta statistica, della violazione di CP nei decadimenti dei mesoni B neutri.

1.3 La violazione di CP nei mesoni B

I mesoni B sono stati legati di antiquark-quark del tipo $\bar{b}q$ in cui q può essere u, d, s o c . Delle quattro possibili particelle, riportate in tabella 1.2, due sono cariche e due sono neutre; i due mesoni neutri, \bar{B}^0 e B^0 , sono creati per interazione forte, ma decadono per

mesone	$\bar{q}q$	$M(MeV)$	$\tau(ps)$
B_u^+	$\bar{b}u$	5279.0 ± 0.5	1.653 ± 0.028
B_d^0	$\bar{b}d$	5279.4 ± 0.5	1.548 ± 0.028
B_s^0	$\bar{b}s$	5369.6 ± 2.4	1.493 ± 0.062
B_c^+	$\bar{b}c$	$(6.4 \pm 0.4) \times 10^3$	$0.46_{-0.16}^{+0.18}$

Tabella 1.2: Il sistema dei quattro mesoni B . Sono riportate la massa e la vita media di ogni particella.

interazione debole che non conserva il *beauty*. Proprio lo studio di questi decadimenti sarà la via seguita nell'esperimento LHCb.

In analogia allo studio del decadimento dei mesoni K neutri, bisogna considerare due autostati deboli B_L e B_H scritti nel seguente modo:

$$|B_L\rangle = p |B^0\rangle + q |\bar{B}^0\rangle \quad (1.8)$$

$$|B_H\rangle = p |B^0\rangle - q |\bar{B}^0\rangle \quad (1.9)$$

dove p e q sono coefficienti complessi che rispettano la condizione di normalizzazione $|p|^2 + |q|^2 = 1$. A differenza di quanto accade per i mesoni K , nel sistema così definito non è possibile osservare la violazione CP cercando direttamente decadimenti vietati: infatti non è possibile creare un fascio puro di B_L o di B_H perchè è piccola la differenza fra le rispettive vite medie e quindi fra le rispettive larghezze di decadimento Γ . In particolare si può dimostrare che $\Delta\Gamma/\Gamma \approx 4 \cdot 10^{-3}$ per B_d^0 e $\Delta\Gamma/\Gamma \approx 0.1$ per B_s^0 .

Per osservare la violazione di CP bisogna confrontare i *rate* di decadimento dello stato iniziale di B^0 e di \bar{B}^0 oppure misurarne l'andamento temporale. Si possono distinguere due classi di violazione di CP:

1) Violazione indiretta

Riguarda i decadimenti semileptonici. Nei decadimenti semileptonici B^0 decade in un leptone positivo e \bar{B}^0 in uno negativo. Il rapporto:

$$A_{sl} = \frac{\dot{N}(\bar{B}^0 \rightarrow l^- X) - \dot{N}(B^0 \rightarrow l^+ X)}{\dot{N}(\bar{B}^0 \rightarrow l^- X) + \dot{N}(B^0 \rightarrow l^+ X)} \quad (1.10)$$

è detto *asimmetria semileptonica*. Un'asimmetria non nulla testimonia che una parte dei B decade nel leptone col segno vietato dalla conservazione di CP.

Tale violazione è associata ad un cambiamento di sapore $|\Delta B| = 2$ che è previsto essere molto piccolo nei decadimenti dei B e la sua misura, resa possibile dalla gran quantità d'eventi disponibili, rappresenta un fertile campo di ricerca di nuova fisica fuori dal Modello Standard.

2) Violazione diretta

Riguarda tutti i canali in cui le ampiezze di decadimento hanno fasi deboli differenti. Infatti mentre l'interazione elettromagnetica e quella adronica possono sempre essere definite reali, per quella debole questo non è possibile.

Se si considera il decadimento di B^0 e di \bar{B}^0 verso lo stesso autostato di CP, l'asimmetria di tali canali di decadimento risulta proporzionale al seno della loro differenza di fase:

$$\mathcal{A}_f(t) = \sin[2(\phi_{mix} + \phi_w)] \sin(2\Delta Mt) \quad (1.11)$$

dove $2\phi_w$ è la differenza di fase tra il canale di decadimento del B^0 e quello del \bar{B}^0 , $2\phi_{mix}$ la fase del fenomeno del *mixing* tra B^0 e \bar{B}^0 e ΔM è la differenza di massa tra i due stati B_L e B_H , quantità tutte legate ai triangoli d'unitarietà del Modello Standard.

La misura diretta dell'asimmetria permette di determinare questi parametri da cui si può risalire agli angoli del triangoli d'unitarietà.

Per misurare accuratamente la dipendenza temporale dell'asimmetria è necessario trovare le ampiezze di decadimento in funzione del tempo del mesone B : perciò è necessaria una risoluzione spaziale che permetta di discriminare la distanza tra il vertice primario, in cui il mesone viene creato, e quello secondario, in cui decade. Per esempio, se si considera un B^0 con energia di 200GeV si trova che:

$$\gamma = \frac{E}{m} = \frac{200\text{GeV}}{5\text{GeV}} = 40$$

e la particella percorre mediamente una distanza:

$$c\tau\gamma = 464\mu\text{m} \cdot 40 \approx 2\text{mm} \quad (1.12)$$

Questo testimonia che il rivelatore di vertice dovrà avere una risoluzione spaziale eccellente ($\ll 2\text{mm}$) per effettuare la misura dell'asimmetria.

Poichè la violazione diretta di CP è legata a cambiamenti di sapore $|\Delta B| = 1$, per misurare l'asimmetria è necessario determinare il sapore del mesone B prodotto ($B = \pm 1$): il processo d'identificazione del sapore prende il nome di *tagging*.

In LHCb il *tagging* viene eseguito sfruttando i leptoni provenienti dal decadimento $b \rightarrow l + q$ e attraverso i K carichi del decadimento $b \rightarrow c \rightarrow s$.

Nelle interazioni forti vengono prodotte coppie particella-antiparticella $B^0 - \bar{B}^0$: uno dei due mesoni decade in uno dei canali di interesse per l'esperimento e ne viene completamente ricostruita l'evoluzione, mentre l'altro viene usato per il *tagging* utilizzando la correlazione tra il segno del sapore e la carica del leptone o del kaone prodotto:

$$\begin{array}{ll} B^0 \rightarrow l^+ & e \quad B^0 \rightarrow K^+ \\ \bar{B}^0 \rightarrow l^- & e \quad \bar{B}^0 \rightarrow K^- \end{array} \quad (1.13)$$

Dal punto di vista sperimentale, bisogna considerare il fatto che si possano verificare degli eventi in cui il processo di *tagging* viene sbagliato; per esempio tutti quegli eventi in cui il mesone B , prima di decadere, ha oscillato ($B^0 \leftrightarrow \bar{B}^0$). In particolare, poichè l'errore statistico sulla misura dell'asimmetria è inversamente proporzionale alla quantità $(1 - 2\omega)\sqrt{\eta N}$, dove N è il numero degli eventi, η è l'efficienza del *tagging* e ω rappresenta la frazione di eventi con il *tag* errato, i tagli del *tagging* vengono scelti in maniera tale da massimizzare la quantità $Q = \eta(1 - 2\omega)^2$, chiamata *fattore di qualità* (si veda paragrafo 2.1.2).

1.4 Il Collisore Adronico LHC (*Large Hadron Collider*)

Il collisore di particelle del CERN, LHC (*Large Hadron Collider*), rappresenta l'ultima tappa di una storia iniziata in Italia (a Frascati) quando, nel 1961, entrò in funzione il primo Anello di Accumulazione per elettroni (*AdA*). Posto ad una profondità compresa tra 50 e 175m sotto il suolo Franco-Svizzero, con una circonferenza di 26.7Km e un'energia nel CM di 14TeV, LHC è il più grande acceleratore di particelle finora costruito e sarà capace di una luminosità (grandezza che, moltiplicata per la sezione d'urto di un dato processo, fornisce il numero di eventi prodotti per unità di tempo) di $10^{34} \text{cm}^{-2} \text{s}^{-1}$.

Da quando la sopravvivenza e la riproduzione non hanno più costituito un problema da risolvere, l'uomo ha cominciato a cercare di comprendere COME fosse il mondo che lo circondava e come poterlo sfruttare al meglio; sebbene agli inizi del 1900 si credesse di aver finalmente investigato tutto lo scibile, a tutt'oggi parecchie sono le domande a cui ancora non è stata data una risposta scientifica. Scopo di LHC è quello di fornire gli strumenti per rispondere ad alcune questioni ancora irrisolte, quali *il bosone di Higgs, la GUT (Grand Unified Theory), la materia oscura, l'Antimateria*.

Attualmente sono previsti quattro grandi esperimenti lungo il tracciato di LHC:

Due di essi, ATLAS (*A Toroidal LHC Apparatus*) [15] e CMS (*Compact Muon Solenoid*) [16], riguardano due rivelatori di particelle di grandissime dimensioni e largo spettro, progettati per sfruttare al massimo la grande energia nel centro di massa che LHC può generare ($\sqrt{s} = 14\text{TeV}$). Sono realizzati per lo studio della fisica delle particelle ad alto impulso trasverso p_t e, in particolare, per la ricerca del *bosone di Higgs*. Anche se si prefiggono gli stessi obiettivi, differiscono per le tecnologie impiegate e per i metodi di misura delle traiettorie e degli impulsi delle particelle prodotte nelle collisioni.

Altri due, ALICE (*A Large Ion Collider Experiment*) [17] ed LHCb (*LHC beauty*) [18] utilizzano due grandi rivelatori per misure dedicate ad un particolare aspetto della fisica delle particelle. ALICE studierà, mediante l'esame delle collisioni tra nuclei di piombo, il problema fondamentale del confinamento della materia adronica, ovvero la formazione d'un nuovo stadio della materia, il *quark-gluon plasma*, che si suppone fosse presente nei primi istanti del *BigBang*. LHCb sarà interamente dedicato allo studio della violazione del CP nei decadimenti dei mesoni B.

1.4.1 La struttura di LHC

LHC appartiene alla famiglia dei collisori, in particolare è un collisore di protoni e di nuclei. In esso, i fasci vengono fatti circolare in versi opposti, all'interno dei due anelli circolari in modo da produrre delle collisioni in punti ben precisi [19]. L'accensione del collisore è prevista per l'inizio del 2008.

Le principali caratteristiche di LHC sono riportate nella tabella 1.3 insieme a quelle del Tevatron [20], il collisore protoni-antiprotoni del Fermilab [21], già in funzione da diversi anni.

Lungo il percorso vi sono otto stazioni lineari dove avviene l'intersezione tra i due fasci circolanti: quattro sono adibite al monitoraggio dei fasci, mentre, nelle rimanenti, verranno situati i quattro apparati sperimentali di ALICE, ATLAS, CMS e LHCb (Figura 1.2).

Per raggiungere un'alta energia nel centro di massa ($\sqrt{s} = 14\text{TeV}$) si utilizza lo stesso apparato d'iniezione usato per il LEP (*Large Electron Positron*): i protoni vengono prima accelerati tramite il *Protosincrotrone* e il *Superprotosincrotrone* fino ad un'energia di 0.45TeV, poi iniettati negli anelli dove le cavità a radiofrequenza poste lungo la traiettoria forniscono l'energia mancante per arrivare ai 7TeV per fascio.

	LHC	Tevatron
particelle collidenti	pp	$p\bar{p}$
energia nel CM	14 TeV	2 TeV
luminosità	$10^{34} cm^{-2} s^{-1}$	$2 \cdot 10^{32} cm^{-2} s^{-1}$
tempo tra due collisioni	25ns	132ns
circonferenza	26.7Km	6.28Km
sez. d'urto anelastica	80mb	50mb
sez. d'urto $b\bar{b}$	500 μb	100 μb
rapporto $b\bar{b}$ /anelastica	$6 \cdot 10^{-3}$	$2 \cdot 10^{-3}$

Tabella 1.3: Principali caratteristiche dell'acceleratore adronico *LHC*, nel caso di fasci di protoni circolanti, messe a confronto con quelle dell'acceleratore *Tevatron*.

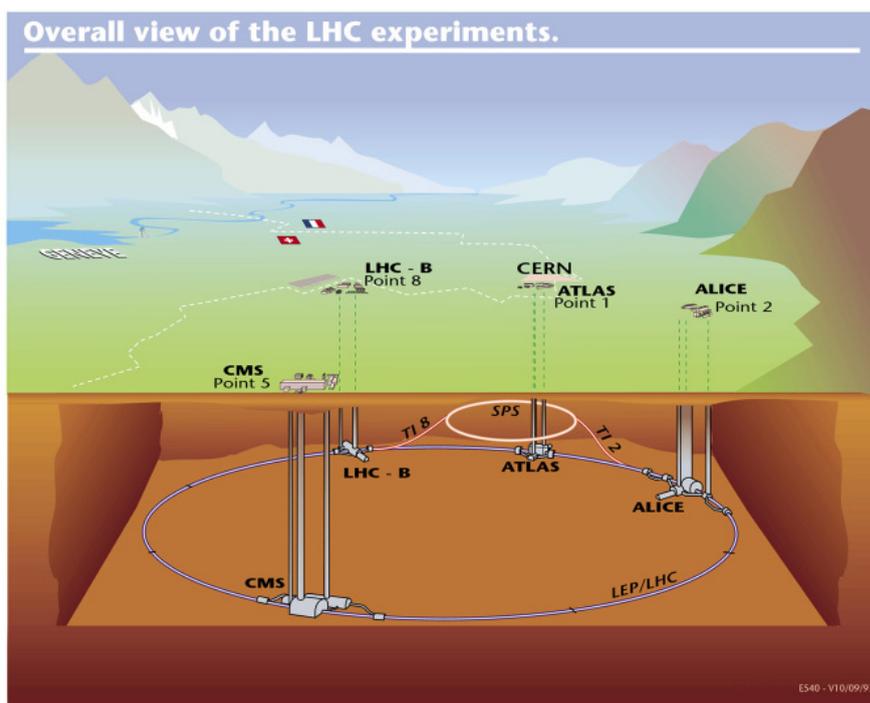


Figura 1.2: Il collisore adronico LHC (*Large Hadron Collider*); è visibile la collocazione dei quattro maggiori esperimenti: ALICE, ATLAS, CSM ed LHCb.

Un ruolo importante è svolto da grandi magneti superconduttori che forniscono un campo magnetico di $8.3T$ per mantenere i fasci in un'orbita circolare.

Il periodo di *bunch crossing* (BX), cioè la distanza temporale che intercorre tra una collisione dei fasci di particelle e la successiva, è di $25ns$.

La luminosità di un collisore si può scrivere come:

$$L = F \frac{fn_1n_2}{4\pi\sigma_x\sigma_y} \quad (1.14)$$

dove $F = 0.9$ è un fattore che tiene conto dell'angolo tra i due fasci, non esattamente paralleli, f è la frequenza di incrocio dei fasci, n_1 ed n_2 il numero di protoni, $\sigma_x\sigma_y = 15\mu m^2$ le semidispersioni gaussiane del pacchetto nella direzione di curvatura e in quella

perpendicolare al piano del collisore. Si ottiene così per LHC un valore della luminosità $L = 10^{34} \text{cm}^{-2} \text{s}^{-1}$.

Il numero d'interazioni per incrocio N che si stima verranno prodotte nel collisore è dato da:

$$N_{LHC} = \frac{\sigma_{anelastica} L_{LHC}}{f_{bunch_crossing}} = \frac{(80 \cdot 10^{-27} \text{cm}^2)(10^{34} \text{cm}^{-2} \text{s}^{-1})}{40 \text{MHz}} \approx 20 \quad (1.15)$$

Questi valori non saranno raggiunti subito, ma sarà necessaria una prima fase a *luminosità ridotta* durante la quale il valore della luminosità è previsto di un'ordine di grandezza inferiore a quello di regime.

LHCb è stato concepito per studiare eventi sotto un piccolo angolo solido, dove è massima la probabilità di trovare gli adroni- b prodotti dalle interazioni. In queste regioni si ha un elevato numero di eventi di fondo, ossia di particelle diverse dal b , che vengono comunque rilevate dagli strumenti e che, non essendo interessanti ai fini dell'esperimento, vanno filtrate; 20 eventi per incrocio è un numero troppo elevato sia per il *trigger*, sia per la ricostruzione dell'evento in LHCb.

Per ridurre gli eventi da analizzare si deve necessariamente ridurre la luminosità di progetto del collisore. Tale necessità non sussiste in ATLAS o CMS, perchè questi sono progettati per la rivelazione di eventi ad alto impulso trasverso e quindi, per costruzione, eliminano molti degli eventi che si trovano ad angoli piccoli. Per diminuire la luminosità, nel sito di LHCb sarà installato un *defocalizzatore* con il compito di aumentare il valore di $\sigma_x \sigma_y$ in modo da ottenere il giusto numero di eventi per la statistica del b .

La luminosità dell'esperimento LHCb sarà di $L_{LHCb} = 2 \cdot 10^{32} \text{cm}^{-2} \text{s}^{-1}$; questo porta ad un numero di interazioni per incrocio pari a

$$N_{LHC} = \frac{\sigma_{anelastica} L_{LHCb}}{f_{bunch_crossing}} \approx 0.4 \quad (1.16)$$

ossia, ci si aspetta un'interazione ogni 2.5 incroci.

Il numero di coppie $b\bar{b}$ prodotte in un secondo è dato dalla relazione

$$L_{LHCb} \sigma_{b\bar{b}} = 10^5 \text{s}^{-1} \quad (1.17)$$

che rappresenta un ottimo punto di partenza per lo studio dei decadimenti dei mesoni B .

Capitolo 2

I Rivelatori di LHCb

LHCb è uno spettrometro a braccio singolo con una copertura angolare in avanti che va da 10mrad a 300mrad , nel piano di curvatura del fascio, e da 10mrad a 250mrad , nel piano ad esso perpendicolare [11].

La geometria dell'apparato è giustificata dal fatto che i quark pesanti (come il b) generati nelle interazioni forti vengono prodotti in avanti ($\theta = 0, \pi$) e con rapidità fortemente correlate ($\Delta y \sim 0$); in figura 2.1 è riportato un grafico per gli angoli di produzione dei quark b e \bar{b} calcolati dal generatore di eventi PYTHIA.

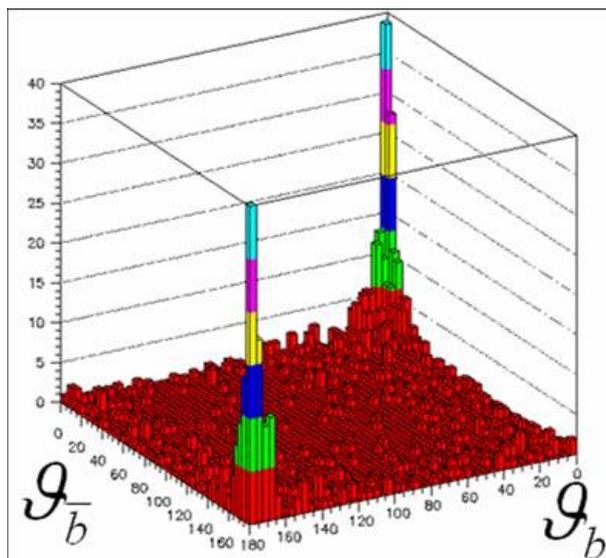


Figura 2.1: Gli angoli polari dei quark b e \bar{b} calcolati dal generatore di eventi PYTHIA. Si vede chiaramente che la distribuzione è fortemente piccata intorno all'angolo nullo ed è quindi conveniente scegliere, per il rivelatore, una geometria a braccio singolo.

L'accettanza d'un rivelatore con geometria *a braccio singolo* è, nel caso di LHCb, paragonabile a quella d'un rivelatore centrale, che coprirebbe tutto l'angolo solido, ma avrebbe costi di realizzazione più elevati. In figura 2.2 è riportata la distribuzione dell'impulso del B^0 misurata attraverso i pioni del decadimento $B_d^0 \rightarrow \pi^+\pi^-$ in tutto l'angolo solido e quella nel cono d'accettanza del rivelatore: si nota come più del 30% dei pioni prodotti finiscono nel cono d'accettanza.

Alloggiato nel pozzo di intersezione numero 8, già sede di DELPHI, LHCb sarà costituito da:

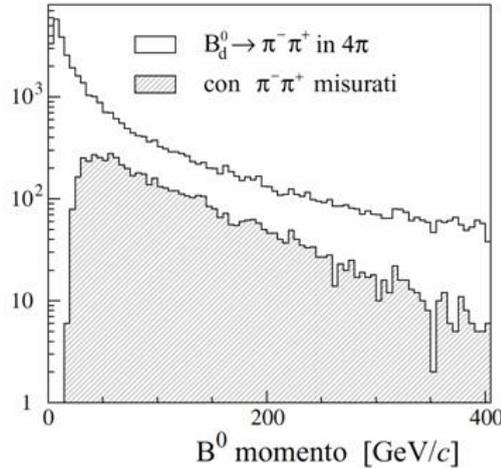


Figura 2.2: Distribuzione dell'impulso per $B_d^0 \rightarrow \pi^+ \pi^-$ in 4π e quella misurata rivelando sia π^+ che π^- nello spettrometro a braccio singolo.

- Un rivelatore di vertice (VELO), comprendente anche un contatore di vertici primari (pile-up VETO counter) (paragrafo 2.2) [23];
- Un sistema di tracciamento (paragrafo 2.3)[24];
- Due rivelatori RICH (Ring Imaging Cherenkov) (paragrafo 2.4)[25];
- Un sistema di calorimetri (paragrafo 2.5)[26];
- Un rivelatore di muoni (paragrafo 2.6)[27].

In figura 2.3 è riportato uno schema con la disposizione dei vari rivelatori di LHCb.

2.1 I processi di *Trigger* e *Tagging*

Le tappe della misura consistono nel *trigger*, nel *tagging* (ossia l'operazione di riconoscimento del sapore del quark) e nella *ricostruzione* degli eventi.

2.1.1 Il processo di *Trigger*

Il sistema di *trigger* è stato progettato per selezionare i decadimenti dei mesoni B dal fondo delle collisioni anelastiche. Come già visto LHCb disporrà di circa 10^5 coppie $b\bar{b}$ al secondo: una così alta statistica è impossibile da processare interamente. Diversamente da quanto accaduto finora nella fisica del B, in cui si avevano a disposizione pochi eventi ed era necessario un *trigger* ad altissima efficienza, si è scelto, per LHCb, di selezionare gli eventi con quantità di moto elevata.

Il sistema di trigger è diviso in quattro livelli, due *hardware* e due *software*:

livello 0 (hardware): la frequenza d'ingresso di questo livello è 40MHz (vale a dire la frequenza di *bunch crossing*), mentre quella di uscita è di 1MHz . Esso usa le informazioni provenienti dal calorimetro e dal rivelatore di muoni, per selezionare gli eventi con adroni, leptoni o fotoni con alto impulso trasverso. In funzione del tipo di particella, varia il valore della soglia di accettazione nell'intervallo tra 1GeV e 3.5GeV .

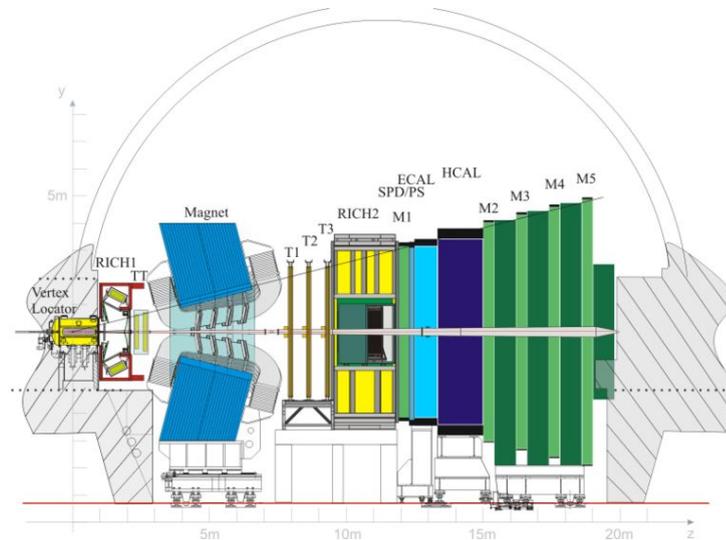


Figura 2.3: Disposizione dei rivelatori nell'esperimento LHCb.

Il contatore di vertici primari (*pile-up veto*) rigetta tutti gli eventi caratterizzati da interazioni multiple per singolo *bunch crossing*.

livello 1 (hardware): la frequenza d'ingresso è di 1MHz mentre quella di uscita di 40kHz . Il livello 1 usa le informazioni raccolte dal VELO e dalle prime stazioni di tracciamento, per identificare i vertici secondari dislocati vicino ai vertici primari, dinamica caratteristica dei decadimenti di mesoni B.

livello 2 (software): ha una frequenza d'ingresso di 40kHz ed una d'uscita di 5kHz . Avendo accesso alle informazioni dell'intero evento, permette di ridefinire i vertici secondari aggiungendo l'informazione proveniente dal sistema di tracciamento.

livello 3 (software): ha una frequenza d'ingresso di 5kHz e una d'uscita di 200Hz . Combinando le informazioni di tutti i rivelatori e facendo uso di algoritmi d'analisi, ricostruisce completamente lo stato finale.

Alla frequenza di 200Hz le informazioni vengono salvate su memorie di massa e mantenute per studi futuri. In figura 2.4 è riportato il diagramma di flusso dei quattro livelli di *trigger*.

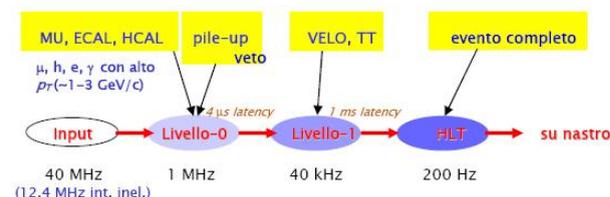


Figura 2.4: Diagramma di flusso dei quattro livelli di *trigger* dell'esperimento LHCb. Da una frequenza iniziale di 40MHz si arriva ad una frequenza di 200Hz .

2.1.2 Il processo di *Tagging*

Per misurare le asimmetrie di CP bisogna conoscere il sapore del B che si sta osservando: questa operazione è detta *tagging*. Siccome l'interazione forte porta sempre alla creazione di coppie particella-antiparticella $b\bar{b}$, il secondo B viene utilizzato per il *tagging*.

Vengono utilizzati due metodi per il riconoscimento del sapore del B: il *tagging leptonic* ed il *tagging kaonico*, a seconda della particella utilizzata per la misura.

Il *Tagging leptonic*

Nel *tagging leptonic* si cerca un leptone proveniente dal decadimento semileptonico dell'altro *quark b* dell'evento $b \rightarrow l$. Questo metodo è affetto da un numero piccolo di errori, ma ha una scarsa efficienza, dovuta alla piccola frazione di decadimento (circa il 10% degli eventi). I contributi principali ai *tag* sbagliati provengono dal *mixing* dei B^0 e dai leptoni del decadimento $b \rightarrow c \rightarrow l$. Quest'ultimi sono facilmente identificabili sperimentalmente poichè possiedono impulso trasverso basso.

Il grafico in figura 2.5 riporta l'impulso trasverso dei muoni provenienti direttamente dal b e quello dei muoni provenienti dalla cascata $b \rightarrow c \rightarrow l$. Si vede chiaramente che un taglio a $1.5\text{GeV}/c$ permette di ottenere una buona separazione per il *tagging*.

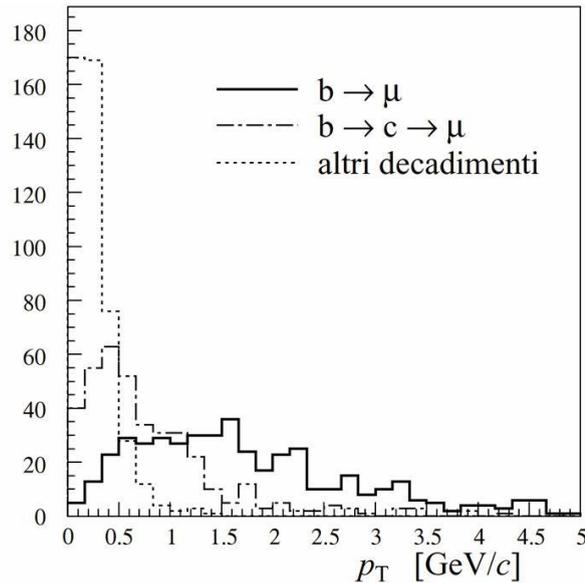


Figura 2.5: Distribuzione del p_t per muoni da $b \rightarrow \mu$, da $b \rightarrow c \rightarrow \mu$ e da altri eventi in cui il B decade in $\pi^+\pi^-$.

Il *Tagging kaonico*

Si parla di *tagging kaonico* se si usa il kaone prodotto nei decadimenti del tipo $B^0 \rightarrow K^+ X$ per l'identificazione del segno del sapore. Questa volta l'efficienza è elevata (78% degli eventi), ma bisogna riuscire a separare i K^\pm dal fondo dei pioni carichi. Per questo scopo gioca un'importanza fondamentale il RICH e la sua efficienza.

In figura 2.6 si può vedere che i K provenienti dal decadimento del b hanno un grande parametro d'impatto d_0 . Se si esegue un taglio, richiedendo che $d_0/\sigma_d > 5\text{GeV}/c$, dove σ_d

è l'errore su d_0 , si può ottenere una buona separazione dal fondo.

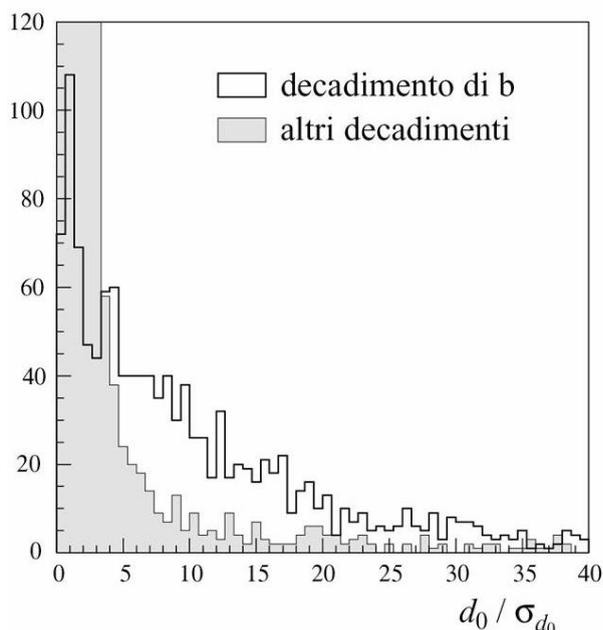


Figura 2.6: Distribuzione del parametro d'impatto per kaoni con $p_t > 0.8\text{GeV}/c$ dal decadimento del b e da altre fonti, in eventi in cui un B decade in $\pi^+\pi^-$.

Per entrambi i tagging vale la regola che il segno della carica del leptone (e o μ) o del K è opposta al segno del sapore: cioè i B^0 ($B=-1$) decadono con l^+ e K^+ e i \overline{B}^0 ($B=+1$) in l^- e K^- .

2.2 Il rivelatore di vertice (*VELO*)

Il rivelatore di vertice VELO (VERTex LOcator), è composto da 25 stazioni costituite da sensori semicircolari al silicio disposti ortogonalmente al fascio; il suo compito sarà quello di fornire precise informazioni sulla produzione e sul decadimento degli adroni- b .

Tali informazioni verranno utilizzate per ricostruire:

- la posizione del vertice primario, luogo in cui è avvenuta l'interazione;
- le tracce non originate nel vertice primario;
- la posizione del vertice secondario, luogo in cui decadono le particelle a vita media lunga.

Il VELO copre tutto l'angolo d'accettanza dell'apparato di LHCb e disporrà di una risoluzione spaziale sul vertice primario di $40\mu\text{m}$ e di una risoluzione spaziale sul tempo proprio di 40fs .

Le informazioni fornite sono utilizzate dal *trigger di Livello-1* per elaborare la ricostruzione tridimensionale dei percorsi. L'*algoritmo L1* necessita di informazioni veloci e dedicate per la ricostruzione tridimensionale delle tracce, per distinguere gli *eventi del b* da tutte quelle tracce lasciate da altre particelle e che non sono state filtrate dal *trigger di*

livello 0. Gli adroni B i cui decadimenti sono prodotti nel cono d'accettanza del rivelatore, sono tipicamente prodotti con angolo polare inferiore a 200mrad . Questo porta ad avere una proiezione del parametro d'impatto del decadimento prodotto nel vertice primario, grande nel piano rz , mentre nel piano perpendicolare all'asse d'impatto ($r-\phi$) è simile alla traccia originata nel vertice primario. L'algoritmo del *trigger L1* ricostruisce tutte le tracce nel piano rz , ma riporta in tre dimensioni solo quelle tracce che hanno un parametro d'impatto rz significativo. Per questo motivo, il sensore del VELO è composto da strisce di silicio a distanza costante, per ricostruire la traccia sul piano rz , combinate con strisce radiali ad angoli di 10° - 20° , per la combinazione delle due proiezioni.

In figura 2.7 è riportato il disegno dei sensori del VELO.

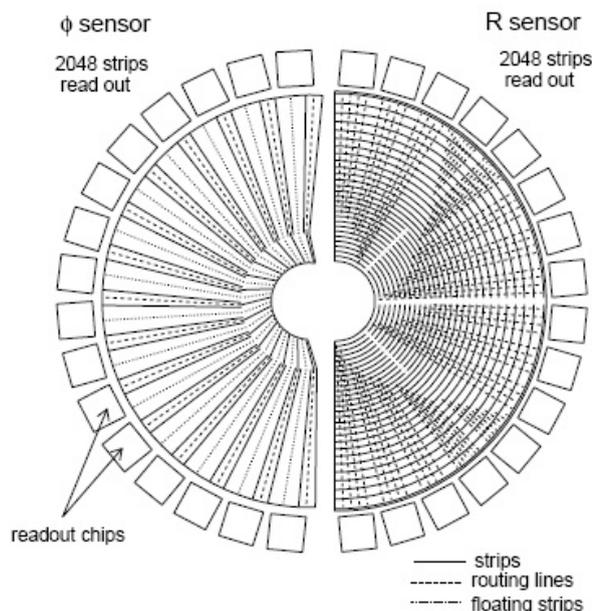


Figura 2.7: Il sensore del VELO. La parte dedicata alla misura radiale (a destra) è composta da strisce equidistanti, mentre la parte dedicata alla misura radiale (a sinistra) è composta di strisce radiali ad angoli tra 10° e 20° .

Tale geometria ha un'ulteriore vantaggio: risulta una scelta naturale utilizzare strisce di silicio più piccole nei pressi del fascio, mentre nelle regioni più remote, si possono utilizzare strisce di dimensione maggiori. Questo permette di minimizzare il numero dei canali di lettura in relazione alle probabilità di lettura del rivelatore.

Associato al rivelatore di vertice è posto un *pile-up veto counter* (VETO), che costituisce un contatore di vertici primari; questo viene utilizzato dal *trigger di Livello-0* per sopprimere gli eventi che contengono più di una interazione pp per incrocio.

2.3 Il sistema di tracciamento

Il sistema di tracciamento ha il compito di rilevare le traiettorie compiute dalle particelle prodotte e permetterne la ricostruzione.

Come già visto, le interazioni che interessano l'esperimento LHCb, producono particelle a piccoli angoli rispetto al fascio; l'accettanza del sistema di tracciamento copre un angolo

solido compreso tra i 300mrad sul piano orizzontale e 200mrad sul piano verticale, con una regione di particolare sensibilità posta intorno al percorso del fascio.

Questa regione, di circa $15\text{mrad} \times 15\text{mrad}$, è chiamata *Inner Tracker*; è la più vicina al fascio e, quindi, ricopre la zona a maggior densità di tracce; per questo motivo, è costituita da microstrisce di silicio delle dimensioni di $200\mu\text{m}$ ciascuna.

La rimanente regione è detta *Outer Tracker*. In questa zona, che copre circa il 98% della superficie del sistema di tracciamento, le stazioni traccianti sono costituite da unità modulari contenenti due piani di *tubi a deriva* con diametro di 5mm inglobati in una scatola a tenuta di gas.

Il sistema di tracciamento è composto da quattro stazioni traccianti: una stazione (TT) è localizzata fra il RICH 1 e il magnete-dipolare ed è interamente posta nella regione *Inner Tracker*; le altre tre (T1-T3) si trovano fra il magnete e il RICH 2 e sono composte sia da una regione *Inner Tracker* che da una *Outer Tracker*, a seconda della zona coperta.

Il sistema di tracciamento ha il compito di:

- rilevare le tracce delle particelle cariche, nella zona tra il VELO ed i calorimetri. La misura dell'impulso con alta precisione è essenziale per la stima della massa dei mesoni B .
- eseguire precise misure sulle direzioni delle tracce provenienti dai due rivelatori RICH. Queste informazioni sono utilizzate dall'algoritmo di identificazione delle particelle.
- eseguire misure congiunte tra il rivelatore di vertice, i calorimetri e il rivelatore di muoni.

I canali d'uscita dal sistema di tracciamento sono studiati in modo da favorire la ricostruzione della traccia, dal punto più lontano del rivelatore, seguendola indietro fino al punto d'interazione (vertice primario).

2.4 I rivelatori RICH (*Ring Imaging Cherenkov*)

La necessità di distinguere fra Pioni e Kaoni nella varietà degli stati finali è essenziale per lo studio della violazione del CP; infatti, solo riuscendo ad identificare gli adroni sarà possibile individuare i canali di reazione significativi.

L'apparato di LHCb è dotato di due rivelatori *Ring Imaging Cherenkov* (RICH 1 e RICH 2), con il compito di distinguere le particelle K da quelle π , al fine di permettere il riconoscimento del canale di decadimento e il *tagging* del sapore.

Essi coprono tutto l'angolo solido tra 10 e 330mrad che corrisponde ad un intervallo d'impulso tra 1GeV e 150GeV ($1 < p < 70\text{GeV}/c$ per il RICH1 e $20 < p < 150\text{GeV}/c$ per il RICH2).

L'ampio range d'impulsi è indispensabile per identificare sia le particelle ad alto impulso trasferito, permettendo così di sopprimere il fondo, sia gli adroni con basso impulso, necessari per il *tagging* del sapore.

In figura 2.8 è riportato uno schema con la sezione dei due rivelatori.

Il RICH1 è posto subito dopo il rivelatore di vertice ed è costituito da due radiatori, uno ad aerogel di silicio e uno a gas con C_4F_{10} , mentre il RICH2 è posto prima dei calorimetri ed è costituito da un solo radiatore con CF_4 .

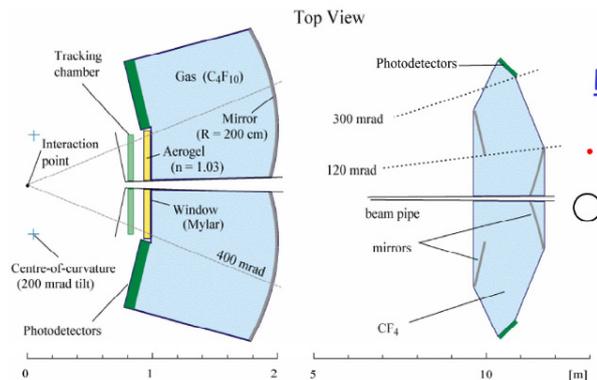


Figura 2.8: Sezione del RICH1 (a sinistra) e del RICH 2 (a destra).

2.5 Il sistema di calorimetri

Il compito del sistema di calorimetri è quello di provvedere all'identificazione di adroni, elettroni e fotoni (misurandone l'energia e la posizione) candidati per il trigger di primo livello (L0); il trigger di livello-0 deve prendere una decisione entro $4\mu\text{s}$ dall'interazione, dunque queste informazioni devono essere fornite con sufficiente selettività e in tempi molto brevi. La struttura generale è quella di un calorimetro elettromagnetico (ECAL), costituito da piani di piombo e scintillatori con fibre a spostamento di lunghezza di luce per il raccoglimento della luce, seguito da un calorimetro adronico (HCAL), composto invece da scintillatori immersi in una struttura di ferro con un sistema di lettura basato sulle fibre.

ECAL è costituito da un rivelatore di "preshower", composto da uno strato di piombo, al fine di filtrare le particelle cariche con energie più basse, da uno strato di scintillatori, e da un calorimetro con tecnologia *Shashilik*. La luce viene raccolta tramite fibre a spostamento di lunghezza d'onda WLS (*Wavelength Shifting Fibers*). Il suo scopo è identificare elettroni, fotoni e π^0 e misurarne posizione e quantità di moto. Lo stesso compito per gli adroni è svolto da HCAL, costituito da scintillatori immersi in una struttura di ferro profonda 1.5m , anch'essi letti da WLS.

2.6 Il Rivelatore di Muoni

Il rivelatore di muoni è costituito da cinque camere (M1-M5) e assolve al compito fondamentale del trigger di livello 0 e del riconoscimento dei muoni. La scelta del materiale da usare per le camere è stata condizionata sia dalla tipologia del fondo che dalle finalità dell'esperimento. In particolar modo si è tenuto conto di:

1. Alto *rate*: cioè basso *tempo morto* visto l'elevato *rate* di particelle aspettato. Vedremo che tale richiesta comporterà la scelta di diverse tecnologie per i rivelatori, dato l'incremento del flusso d'un fattore $2 \cdot 10^3$ fra la regione più esterna della stazione M5 e quella più interna di M1.
2. Resistenza all'invecchiamento: la scelta dei materiali dovrà essere tale da assicurare il funzionamento dei rivelatori almeno per i 10 anni in cui si prevede l'utilizzo del rivelatore.
3. Risoluzione temporale adatta al periodo di BX: ai fini del trigger muonico è essenziale che ad ogni particella sia assegnato il *BX-identifier* in maniera non ambigua, cioè che

venga rilevato il suo passaggio nel rivelatore in una finestra temporale non più grande del periodo di iterazione fra i fasci. In particolare si richiede che questo avvenga con un'efficienza del 95% in una finestra temporale di $20ns$.

4. Risoluzione spaziale: si richiede che la risoluzione spaziale sia tale da permettere di determinare l'impulso trasverso della particella rivelata con un'incertezza inferiore al 20%. Tale richiesta riguarda in particolare le stazioni M1 e M2. Per soddisfarla bisogna tenere conto anche del *cluster-size* geometrico del rivelatore scelto.

L'elettronica di *front-end* amplifica, forma e discrimina i segnali analogici provenienti dal rivelatore fornendo in uscita un'informazione digitale (*canali fisici*). L'esperimento necessita di un'informazione più grossolana di quella fornita a questo stadio, è per questo che è stata possibile la riduzione dei 152.832 *canali fisici* in 25.920 *canali logici*, che costituiscono gli ingressi per il *trigger* e per l'acquisizione (si veda paragrafo 3.1).

La camera M1 è posta prima dei calorimetri, in modo da non essere influenzata dallo *scattering multiplo* causato dagli assorbitori. Le restanti camere (M2-M5) costituiscono la parte finale del rivelatore e sono inframmezzate da filtri in ferro spessi $800mm$, che, assieme ai calorimetri, fungono da filtro per adroni, elettroni e fotoni; un ulteriore schermo di ferro, spesso $400mm$, è posto oltre l'ultima stazione per proteggerla da eventuali radiazioni di ritorno dalla caverna. Affinchè un evento venga preso in considerazione dovrà interessare tutte e cinque le stazioni, evidenziando un alto potere penetrativo della particella (essendo l'energia del muone superiore ai $5GeV$).

Il fascio di particelle viene notevolmente attenuato, per quanto riguarda il *rate* e l'energia delle particelle che lo compongono, sia percorrendo il cammino lungo il rivelatore, sia muovendosi trasversalmente dal centro del fascio. Per questo sono richieste caratteristiche diverse per le cinque stazioni. Inoltre, ogni singola stazione è divisa in quattro regioni (R1-R4) caratterizzate da diverse dimensioni delle *pad*; tali dimensioni sono studiate in modo che sia mantenuto pressochè costante il rapporto tra la superficie d'ogni *pad* e il *rate* di particelle aspettato.

La prima camera incontrata dal muone, M1, è fondamentale per la ricostruzione della traiettoria: è per questo motivo che è stata posta prima dei calorimetri. L'algoritmo del Processo di Tracciamento parte dal segnale proveniente dalla stazione M3 ed estrapola una retta fino al punto d'interazione; attorno a questa retta viene aperta una finestra di interesse (FOI) sulle stazioni M4, M5 e M2 nella quale cercare le altre *pad* colpite; se tutte e tre le stazioni confermano il passaggio della particella, viene estrapolata una nuova retta passante per M2 e M3; a questo punto si ricerca nella stazione M1 la *pad* più vicina alla nuova retta. Una volta selezionato l'*hit* più prossimo nella stazione M1 si utilizzano i valori veri ricavati da M1 e da M2 per ottenere il valore dell'impulso trasverso dal *trigger*. Ad analisi completata si selezionano al massimo due muoni a processo, attraverso le tracce di M1 e M2.

Le stazioni M4 e M5 non contribuiscono a alla misura di pt , ma solo a confermare la presenza di muoni penetranti con energia superiore ai $5GeV$.

In figura 2.9 è mostrato uno schema della procedura appena descritta.

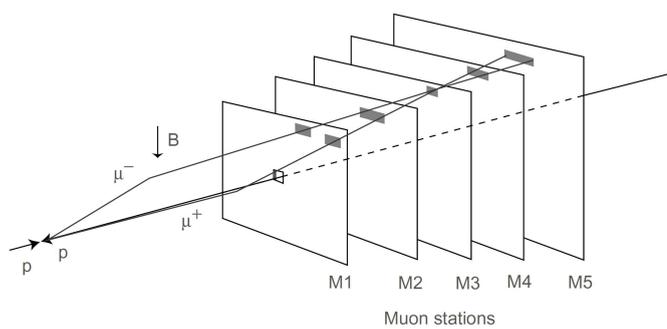


Figura 2.9: Ricostruzione della traccia di due muoni che colpiscono la stazione M3 nello stesso punto; sono evidenziate in grigio le FOI (*Field Of Interest*).

Capitolo 3

L'elettronica del rivelatore di Muoni

La funzione principale dell'elettronica del rivelatore di muoni [28] è quella di preparare l'informazione proveniente dalle stazioni M1-M5 per il *trigger L0* e *L1*. Questo corrisponde ad organizzare i 122.112 canali in uscita dalle camere, chiamati *canali fisici*, in 25.920 *canali logici* [29, 30] e marcare ogni segnale ricevuto con il proprio "*Bunch Crossing identifier*" (*BX-id*), un codice ad otto bit che lo associa al numero identificativo del *bunch* di cui fa parte.

Per avere un'idea del grado di sintesi si consideri che i 25.920 *canali logici* in ingresso al *trigger di livello zero*, costituiscono circa il 21% dei 122.112 *canali fisici* in uscita dal rivelatore di muoni.

Tale riduzione viene realizzata, in una prima elaborazione, dal *chip* DIALOG ("*DIagnostics time Alignment and LOGics*"), appositamente realizzato per l'esperimento in tecnologia *rad-hard* "*IBM 0.25 μ m*" e le cui funzioni saranno discusse nel dettaglio nel paragrafo 3.1.2.

Una seconda riduzione dei canali, provenienti dalle regioni R3 e R4 delle stazioni M2-M5 e nella regione R2 di M4 e M5, viene effettuata nell'elettronica *off-detector* dalle schede IB (*Intermediate-Board*). In questa fase si ha la necessità di unire *canali fisici*, provenienti da differenti schede di *Front-End* e da differenti camere, per formare i *canali logici* (paragrafo 3.2).

I dati raccolti nei *canali logici* vengono poi inviati all'*Unità Decisionale dell'"L0 Trigger"* per essere processati e rigettare quelli ritenuti "non significativi" per gli scopi dell'esperimento; solo i dati approvati dall'*L0 Trigger* verranno associati al giusto "*ID-bx*", compito assolto dalle schede ODE ("*Off-Detector Electronics*"), e memorizzati dal *Sistema di Acquisizione Dati (DAQ ("Data Acquisition"))*.

È importante che il primo passo del processo di sintesi dei canali, vale a dire la riduzione dei *canali fisici* in *logici*, sia realizzato nello stadio più vicino al rivelatore; questo permette infatti di ottenere subito una diminuzione del numero dei canali, riducendo notevolmente la complessità ed i costi dell'apparato. Per fare un esempio, se i *canali fisici* non fossero messi in OR nelle camere, sarebbero necessari 122.112 cavi di connessione per i segnali delle camere, in uscita dal rivelatore, complicando notevolmente la logistica dell'apparato.

Questa scelta comporta in ogni caso due svantaggi non trascurabili. In primo luogo la riduzione dei canali comporta necessariamente una perdita di ridondanza del sistema e quindi non è ovunque possibile. Inoltre, se da una parte si hanno vantaggi, sia economici che progettuali, nella riduzione dei canali, dall'altra l'elettronica sulla camera richiede l'uso di tecnologie resistenti alle radiazioni.

Nel punto di intersezione dei fasci, le collisioni $p-p$ producono un gran numero di particelle ionizzanti; queste raggiungono ed interagiscono con gli oggetti presenti nella caverna (muri, tubi, sostegni...); si vengono così a creare radiazioni di energia via via minore che non vengono assorbite del tutto dal rivelatore o dai filtri poichè prodotte fuori dalla loro portata. Queste radiazioni si propagano per diverse decine di metri intorno al punto di interazione, investendo l'ambiente circostante. A seconda della posizione occupata, le diverse schede elettroniche saranno soggette ad una diversa quantità di radiazione [35]. In figura 3.1 è riportato un disegno con la disposizione delle varie parti dell'elettronica per il controllo dei dati del rivelatore di muoni in LHCb.

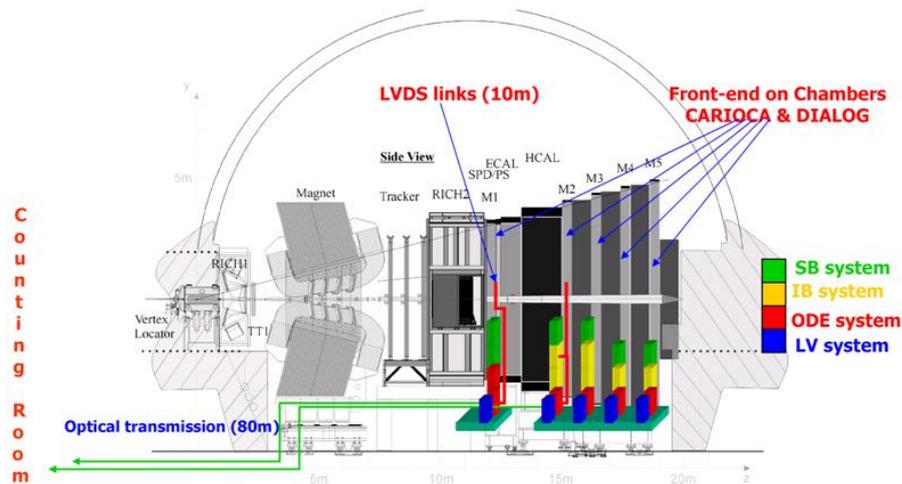


Figura 3.1: Posizione dei sottosistemi dell'elettronica in LHCb

L'elettronica preposta ad analizzare i dati provenienti dal rivelatore e prepararli per l'L0-trigger è suddivisa in *Front-End electronics* e *Off-Detector electronics*.

La prima è montata sul rivelatore di muoni, è soggetta ad alta quantità di radiazioni (circa 1Mrad in 10 anni d'esperimento) ed è composta dalle schede *CARDIAC* (paragrafo 3.1); la seconda è posta ad una distanza di circa 10m dal rivelatore ed è soggetta ad una limitata quantità di radiazioni (circa 10Krad nei dieci anni d'esperimento); essa è composta dalle schede IB, dai *crate* dell'*SB-System* e dai *crate* dell'*ODE-System* (paragrafi 3.2, 3.3.3 e 3.3.2). I calcolatori adibiti al controllo ed alla gestione del sistema, l'elettronica di *Trigger* e gli apparati di acquisizione dati, sono allestiti in una stanza separata, la "Counting Room", posta a circa 80m dal fascio e totalmente schermata dalle radiazioni. Uno schema dell'elettronica del rivelatore di muoni è indicato in figura 3.2 [38].

Elemento di controllo dell'elettronica del rivelatore è il sistema ECS (*Experiment Control System*) (paragrafo 3.3). Basato su un sistema di comunicazione CAN-bus, si possono distinguere tre parti dell'ECS: il controllo dell'elettronica, il controllo delle tensioni (HV e LV) ed il controllo delle temperature (T). In questo elaborato verrà esaminato esclusivamente il controllo dell'elettronica di *front-end*, raggiunto essenzialmente tramite 2 tipi di moduli: la *Service Board (SB)*, che permette il monitoraggio ed il controllo delle schede di FE attraverso diversi collegamenti I²C esterni, ed il *Pulse Distribution Module (PDM)*, che, nella fase di calibrazione dell'apparato, distribuisce alle schede *Service Board* ed alle schede di *Front-End* il *clock macchina* di LHC ed i segnali necessari alla sincronizzazione.

L'assegnazione del corretto numero di *bunch crossing* dei segnali provenienti dai *canali logici*, richiede un processo d'allineamento temporale e di fase sia del rivelatore di muoni

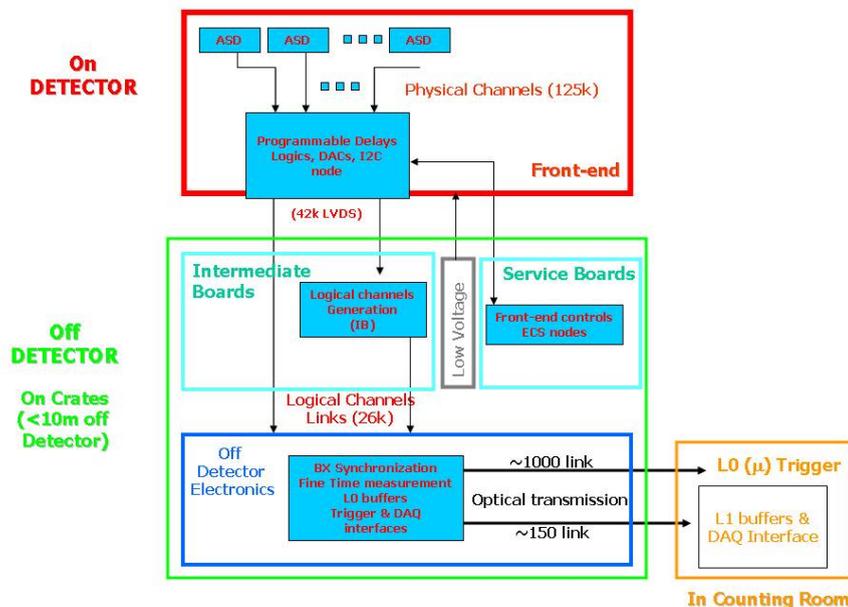


Figura 3.2: Schema dell'elettronica del rivelatore.

con i cicli dell'anello d'accumulazione, sia all'interno del *subdetector* stesso: infatti, segnali provenienti da diversi *canali fisici*, compiono percorsi diversi all'interno del rivelatore e vanno sincronizzati e messi in fase tra loro, tramite ritardi programmabili, prima di essere inviati al *trigger* (essendo il problema dell'allineamento temporale argomento di questo elaborato, verrà ampiamente discusso nel capitolo 6).

3.1 L'elettronica di *Front End*: le schede CARDIAC

Le schede dell'elettronica di *Front End* sono montate all'interno del rivelatore di muoni, collegate direttamente all'uscita delle camere; per questo, saranno soggette ad una alta dose di radiazioni, stimata in $1Mrad$ per i dieci anni di funzionamento dell'esperimento.

Componenti base di questa parte di elettronica sono due *chip* ASIC (*Application Specific Integrated Circuit*) appositamente sviluppati dall'*LHCb Muon Group* e realizzati in una tecnologia *rad-hard* per poter resistere alla forte dose di radiazioni cui saranno sottoposti: il *CARIOCA* (*Cern And RIO Current Amplifier*), ed il *DIALOG* (*DIagnostic, time Adjustment and LOGics*).

In figura 3.3 è riportato il diagramma delle schede dell'*Elettronica di Front-End*.

I due *chip* ASIC sono montati sulla stessa scheda, su un lato due *CARIOCA* e sull'altro un *DIALOG*, chiamata *CARDIAC* (*CARioca and DIALog Card*) come mostrato in figura 3.4.

Per il corretto funzionamento di LHCb, saranno utilizzate 7.536 schede *CARDIAC*.

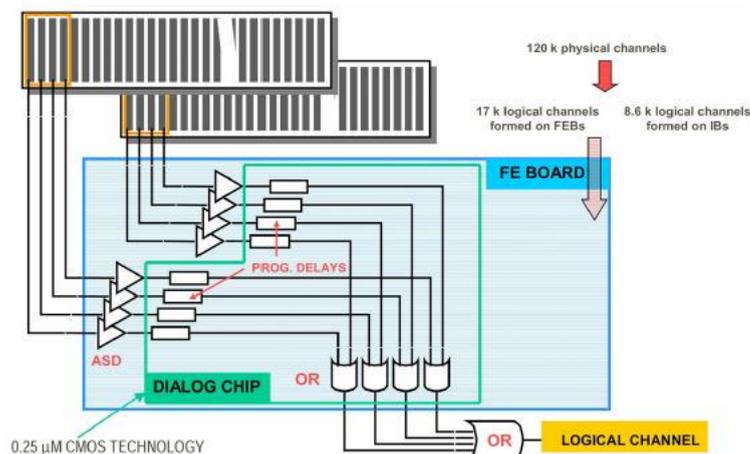


Figura 3.3: Diagramma dell'elettronica di *Front End* (FE). I chip CARIOCA ricevono l'uscita delle camere formando i *canali fisici* (digitali) che vengono inviati al chip DIALOG. Qui i *canali fisici* sono allineati temporalmente e messi in OR per andare a formare i *canali logici*.

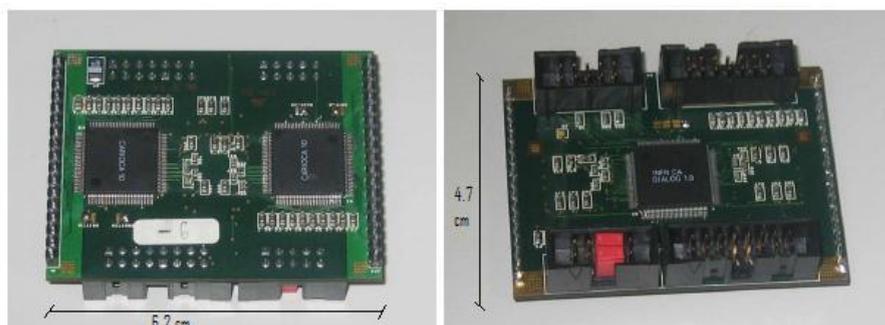


Figura 3.4: La scheda *CARDIAC*: sulla scheda sono montati, su di un lato due chip CARIOCA (foto di sinistra), sull'altro un chip DIALOG (foto di destra).

3.1.1 Il chip CARIOCA

Il primo chip, il CARIOCA (*Cern And RIO Current Amplifier*), è un ASD (*Amplifier Shaper Discriminator*). Il compito del chip CARIOCA è quello di digitalizzare il segnale analogico proveniente dalle camere del rivelatore di muoni e renderlo disponibile al resto dell'elettronica.

Il segnale analogico in uscita dalle camere, viene amplificato e filtrato per eliminare un'eventuale rumore di trasmissione; poi, grazie ad un discriminatore analogico-digitale, viene confrontato con una soglia e trasformato in segnale digitale; infine, viene reso in uscita secondo lo standard LVDS. La soglia di comparazione per il segnale in ingresso, si riferisce al valore settato in un apposito registro; l'accesso al registro di comparazione è ottenuto tramite il chip DIALOG, permettendo di impostarne il valore in ogni momento dall'apposita interfaccia utente.

Ogni chip CARIOCA accetta fino ad otto segnali analogici in ingresso e fornisce otto segnali digitali in uscita, come mostrato in figura 3.5.

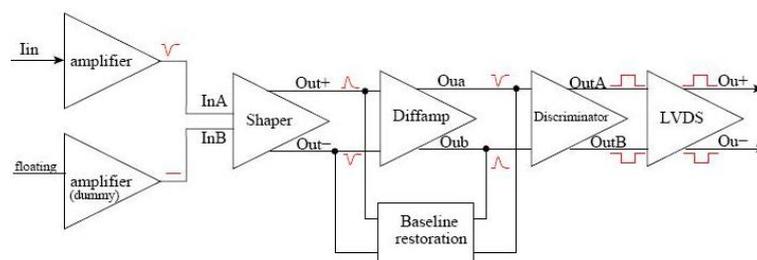


Figura 3.5: Schema del *chip* CARIOCA: i segnali analogici provenienti dal rivelatore di muoni vengono amplificati, confrontati con una soglia programmabile e resi come segnali digitali al resto dell’elettronica dell’esperimento.

3.1.2 Il *chip* DIALOG

Il secondo *chip* ASIC della *FE-Elettronics* è chiamato *DIALOG* (*DI*agnostic, *time Adjust-*ment and *LOG*ics) [39].

Come mostrato in figura 3.6, il *chip* DIALOG è provvisto di un’interfaccia *I²C* grazie la quale, l’*SB-System* accede ai suoi registri interni e gestisce la scheda CARDIAC.

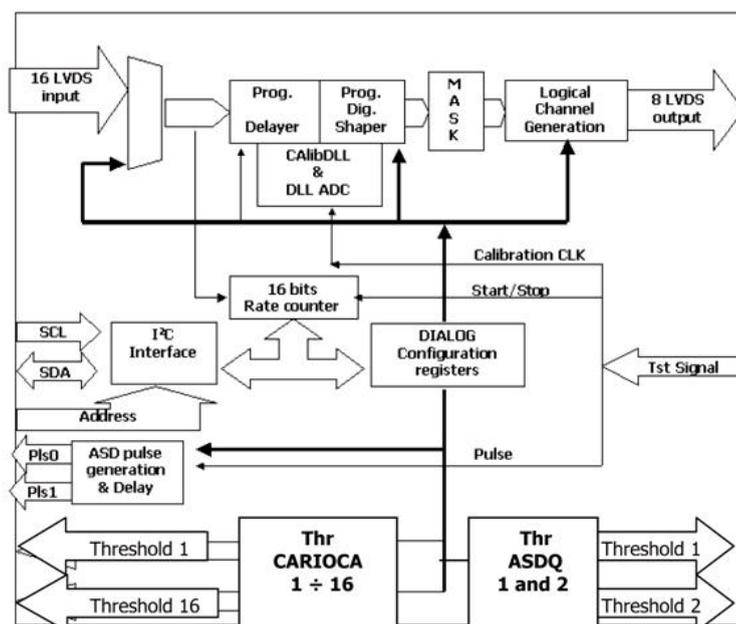


Figura 3.6: Schema del *chip* DIALOG: i segnali digitalizzati dai CARIOCA vengono allineati temporalmente per compensare i diversi “percorsi temporali” che i *canali fisici* compiono. Inoltre, viene effettuata una prima riduzione dei *canali fisici* in *canali logici*.

Le funzioni cui il *chip* DIALOG assolve sono:

- Allineamento temporale dei segnali;
- Riduzione dei *canali fisici* in *canali logici*;
- Settaggio delle soglie dei *chip* CARIOCA;
- Funzionalità di test, tramite contatore a 16 bit.

Allineamento temporale dei segnali

Ogni segnale d'ingresso è inviato ad un ritardo programmabile dove può essere ritardato fino a $50ns$ in passi da $1.5ns$, con valori programmabili canale per canale. Questa funzionalità è utilizzata per settare il *fine time delay*, Δt_F , utilizzato per compensare i vari ritardi che subiscono i canali provenienti da zone diverse del rivelatore (si veda il capitolo 6).

Ogni canale può essere mascherato singolarmente tramite un registro a 16bit, in modo da permettere di isolare un singolo *canale fisico* per individuare eventuali malfunzionamenti.

Riduzione dei canali fisici in canali logici

Ogni *chip* DIALOG è dotato di 16 ingressi LVDS ed è collegato a due *chip* CARIOCA; al suo interno, più segnali sono messi in OR ottenendo fino ad otto *Canali Logici* in uscita, che vengono inviati direttamente alle schede ODE o alle schede IB. In particolare è possibile avere otto, quattro o due *canali logici* in uscita, a seconda della zona del rivelatore in cui ci si trova.

Settaggio delle soglie dei chip CARIOCA

Tramite 16 DAC (*Digital to Analog Converter*) indipendenti, uno per canale fisico, è possibile pilotare le soglie dei due *chip* CARIOCA collegati, agendo su altrettanti registri che forniscono una granularità di 8 bit in un intervallo di tensione tra 0 e 1.5 Volt.

Funzionalità di test

Il DIALOG fornisce diverse funzionalità di test per l'apparato di lettura delle camere, come un contatore a 24-bit per canale che, pilotato dalla scheda *Service Board*, permette di contare gli *hit* ricevuti in una certa frazione di tempo, o un sistema di auto-iniezione di carica per il *chip* CARIOCA.

3.2 Le schede IB (“*Intermediate Board*”)

Le schede *Intermediate Boards (IB)* [34], mostrate in figura 3.7, sono situate ad una decina di metri dal rivelatore, e non sono quindi soggette ad una massiccia dose di radiazioni.

Il loro scopo è quello di completare la formazione dei *canali logici* fornendo un ulteriore livello di OR. Questo è necessario perchè alcuni *canali logici* sono formati con *canali fisici* provenienti da camere diverse, in particolare dalla regione R2 di M4 e M5 e dalle regioni R3 e R4 di M2-M5 del rivelatore di muoni. Nella tabella 3.1 sono riportati i rapporti tra *canali logici* e *canali fisici* per le varie regioni del rivelatore di muoni.

CH logici/ CH fisici	M1	M2	M3	M4	M5
R1	576/576	336/336	336/336	288/288	288/288
R2	576/1152	384/672	384/480	168/288	168/288
R3	576/1152	336/1152	336/1152	120/576	120/576
R4	576/1152	336/1152	336/1152	120/1152	120/1152

Tabella 3.1: Rapporto tra *canali logici* e *canali fisici* nei quadranti di ogni camera. Si noti che la riduzione maggiore dei canali avviene nelle zone interessate dalle IB.

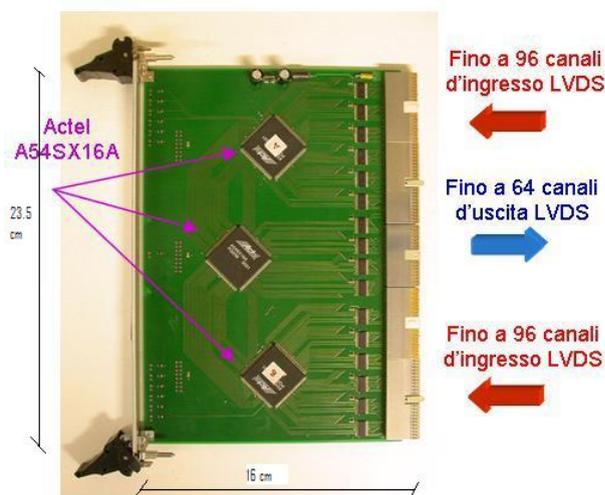


Figura 3.7: La Scheda Intermedia (*Intermediate Board, IB*).

Le schede IB sono schede molto semplici, composte di soli OR logici, e supportano fino ad un massimo di 192 ingressi e 64 uscite in standard LVDS. L'uscita delle IB, insieme ai canali logici già formati nei chip DIALOG, costituiscono l'ingresso per le schede ODE (*Off Detector electronics*).

3.3 Il Sistema di Controllo (ECS)

L'*Experiment Control System* (ECS), è un sistema di controllo che ha lo scopo di monitorare e controllare le funzioni del rivelatore di muoni. Sarà possibile avere il completo monitoraggio e controllo delle tensioni degli apparati, divise in *alta tensione (HV)* e *bassa tensione (LV)*, delle loro temperature e dell'elettronica utilizzata per analizzare ed elaborare i dati raccolti dal rivelatore [40].

L'architettura del sistema di controllo è basata su un sistema di comunicazione *CAN-bus*; la gestione di tale *bus* viene effettuata grazie ad un piccolo modulo su scheda "plug-on" realizzato appositamente per questo genere di esperimenti: l'ELMB (paragrafo 3.3.1).

In questo elaborato verrà esaminata l'ultima funzione del sistema, riferendosi con la sigla "ECS" alla parte del sistema preposta al controllo, al monitoraggio ed alla comunicazione con l'elettronica di *front-end* e le schede ODE.

L'ECS è diviso in due sottosistemi detti rispettivamente "*ECS-ODE System*" e "*Service Board System*".

Il primo, descritto nel paragrafo 3.3.2, fa uso di un modulo ELMB per ogni scheda ODE che permette di comunicare con l'esterno tramite CAN-bus. Questo sistema dispone di 4 rami CAN-bus collegati rispettivamente ai quattro *backplane* dei *crate* delle schede ODE contenenti ciascuno fino a 16 di questi moduli.

Il secondo sottosistema, descritto nel paragrafo 3.3.3 e nel capitolo 4, si occupa del monitoraggio e del controllo delle funzionalità delle schede di *Front-End* del rivelatore di muoni. Esso dispone di 40 rami CAN che, dalla "*counting room*", si diramano a gruppi di 4 fino ai *Pulse Distribution Module* dei dieci *SB-crate*: da qui il segnale CAN-bus viene distribuito a tutte le schede *Service Board* contenute nello stesso *crate* e collegate a gruppi

di 5 per ogni ramo.

Per la gestione del sistema elettronico, saranno installati 14 calcolatori; ogni calcolatore monter  un interfaccia USB-CAN (*“SysTec’s USB-CANmodul 3004006 Multiport”*) che gli permetterà di accedere fino a sedici rami CAN. Sono previsti sei calcolatori per gestire l'*SB-System*, quattro per l'*ODE-System*, due per il controllo delle tensioni (HV e LV), uno per il controllo di temperature e pressioni (T e P) ed infine, un ultimo calcolatore per il controllo generale dell'*ECS* (figura 3.8).

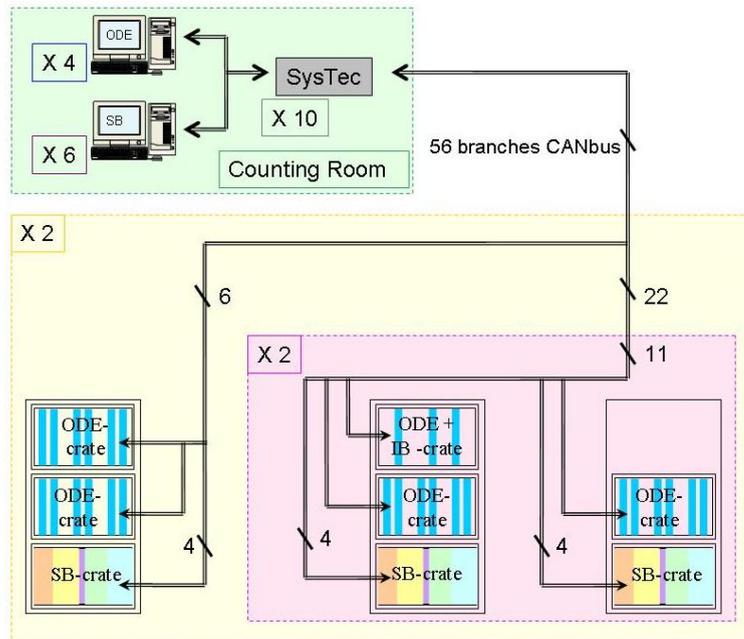


Figura 3.8: Schema logico delle connessioni del sistema ECS (Experiment Control System) per le camere per i muoni.   indicata la logica di connessione dei vari canali del *bus* CAN per l’interconnessione dei *crate* con i calcolatori. Inoltre, saranno presenti altri due calcolatori per il controllo delle tensioni, uno per temperature e pressioni ed uno per la gestione globale del rivelatore.

3.3.1 L’ELMB (*Embedded Local Monitor Board*)

L’ELMB (*Embedded Local Monitor Board*)   una scheda progettata dal gruppo DCS (*“Detector Control System”*) di ATLAS per lavorare in ambienti a moderato livello di radiazioni [41]; il suo funzionamento   basato sul microcontrollore *“ATmega128 AVR ATMEL”* in tecnologia $0.35\mu m$, ad architettura RISC, con 121 istruzioni a clock singolo e $4MHz$ di frequenza di clock [42].

La scheda, di cui si riporta lo schema a blocchi in figura 3.9,   nata come soluzione al problema di dialogare con le varie schede elettroniche utilizzate negli esperimenti in LHC.

L’esigenza era quella di avere un sistema, con logica interna, che potesse comunicare tramite un *bus* di tipo CAN, eventualmente in grado trattare segnali analogici e che fosse adatto a lavorare in ambiente soggetto a moderata quantit  di radiazioni.

Il primo requisito   assolto dal microcontrollore della ditta *“ATMEL”*; per la comunicazione,   possibile dialogare con l’ELMB grazie ad uno *“SPI CAN controller SAE81C91”*

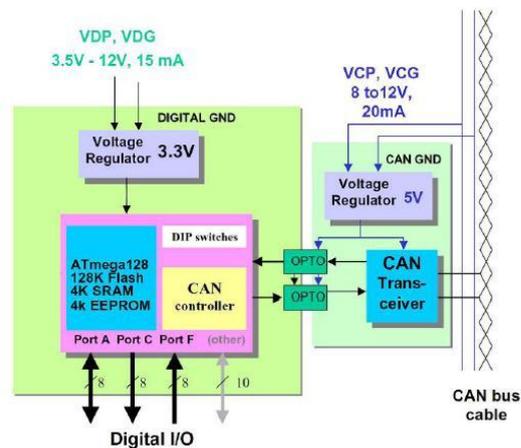


Figura 3.9: Schema a blocchi della scheda ELMB

che permette l'accesso al CAN-bus attraverso il “CANtransmitter/receivers” presente sulla scheda. L'isolamento galvanico del bus è garantito da fotoaccoppiatori.

Tutti i componenti dell'ELMB sono montati su un PCB (“Printed Circuit Board”) di dimensioni 50x67mm come mostrato in Figura 3.10 e 3.11. Sul retro del PCB sono presenti due connettori SMD ad alta densità di pin. In questo modo è possibile connettere l'ELMB alla scheda madre tramite un sistema “Plug-On”, senza doverla saldare e garantendone inoltre una rapida sostituzione all'occorrenza. È anche possibile montare, su questo lato, un ADC (*Analogic-Digital Converter*) delta-sigma a 16+7 bit con 64 ingressi e i regolatori di tensione per la sua alimentazione. Questa versione dell'ELMB è detta “analogica”.

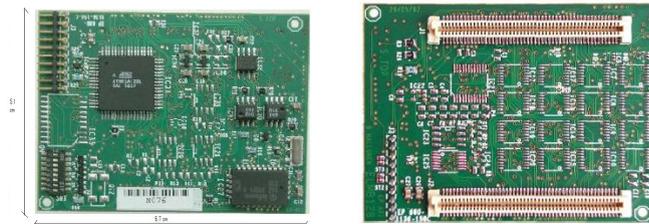


Figura 3.10: La scheda “ELMB digitale” - lato superiore (a sinistra) e lato inferiore (a destra).

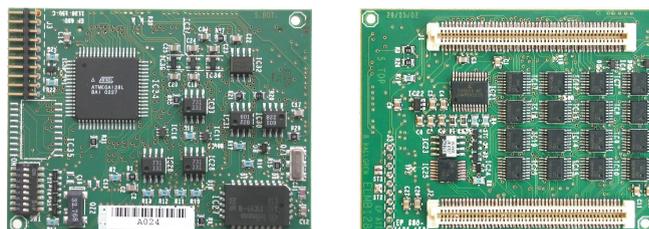


Figura 3.11: La scheda “ELMB analogica” - lato superiore (a sinistra) e lato inferiore (a destra).

Ogni *chip* montato sulla scheda dell'ELMB è certificato per poter lavorare in ambienti a moderato livello di radiazioni, conferendo all'ELMB l'ultima caratteristica richiesta e

permettendone così un largo impiego nei vari esperimenti attualmente in opera lungo LHC.

Per l'utilizzo delle schede ELMB nell'esperimento LHCb, sono stati sviluppati *software* appositi a seconda del modulo su cui le schede saranno montate. Il *firmware* implementato per l'utilizzo delle schede ELMB all'interno dell'SB-System sarà discusso nel paragrafo 4.3.1.

3.3.2 L'ODE-System: le schede *Off-Detector Electronics*

Ogni scheda *Off-Detector Electronics (ODE)* [43], è stata appositamente sviluppata per ricevere 192 *canali logici* provenienti dall'elettronica montata sul rivelatore ed inviarne i dati, sia al *trigger di livello-0*, sia al sistema di acquisizione dati di LHCb: i *DAQ* ("*Data Acquisition*") [44].

Le schede ODE sono dotate di 192 ingressi LVDS e dispongono di 13 collegamenti ottici a 1,6 Gbit/s: 12 per comunicare con il *trigger di livello 0* e uno per trasmettere i dati opportunamente impacchettati alle schede TELL1 ("*Trigger Electronics and L1 board*") [45] esterne che contengono il buffer per il *trigger di livello 1* ed il sistema di acquisizione dati DAQ.

L'ELMB presente nella scheda ODE è programmato per gestire due bus I²C, uno dedicato alle comunicazioni con i 24 *chip SYNC*, che fornisce un accesso preferenziale alle informazioni necessarie per la sincronizzazione del sistema, l'altro utilizzato per gestire e inizializzare tutti gli altri *chip* contenuti nella scheda. Esso gestisce anche il "*GLOBAL reset*" che ripristina i registri delle schede ODE nello stato di *default*. Il segnale di *reset* è generato in fase d'accensione del sistema o, grazie alle funzionalità implementate nell'ECS, ogni qualvolta sia necessario per il ripristino delle funzionalità del sistema. In figura 3.12 è riportato un diagramma con indicate le funzionalità della scheda ODE.

Al segnale in ingresso viene assegnato il giusto *BX-id* e viene inviato alle *pipeline* per L0, dette *L0 buffer*, dove attendono $4\mu\text{s}$ la risposta dell'*L0 Decision Unit*". Parallelamente il dato e i quattro bit meno significativi del *BX-id* vengono inviati al *trigger di livello 0*. I dati nelle *pipeline* attendono la decisione di quest'ultimo e, in caso di risposta affermativa, vengono trasmessi attraverso *link ottico* alle schede TELL1. Qui i dati rimangono nel *buffer* di L1 in attesa della decisione del *trigger di livello 1* e, se anche questa è affermativa, il codice viene elaborato e messo a disposizione per il DAQ.

La sincronizzazione del segnale avviene tramite 24 *chip custom* ad otto bit, detti "*SYNC chip*", che contengono dei ricevitori LVDS, le "*pipeline*" di L0 e il "*derandomizer*".

Ogni canale di ingresso del *SYNC chip* dispone di un TDC ("*Time to Digital Converter*") che permette di misurare la fase, con risoluzione di 1.5ns , del *canale logico* rispetto al periodo di interazione dei fasci. Questa misura può essere monitorata tramite un *buffer* di memoria che permette di realizzare un istogramma della distribuzione dei ritardi del canale d'ingresso. Questi dati sono accessibili al Sistema di Controllo dell'Esperimento (ECS) tramite un'interfaccia I²C e permettono di calcolare i ritardi dei *canali logici* centrando gli istogrammi, attenuando così l'effetto del *jitter temporale* da cui i segnali sono affetti. Ogni *chip SYNC* riceve il *clock macchina* di LHC attraverso un modulo TTCrx, a bordo della scheda ODE; oltre al *clock macchina*, il sistema TTC ("*Time and Trigger Control*") invia i segnali necessari alla messa in tempo del sistema attraverso il modulo TTCrx. Grazie a tali segnali, coerenti con la struttura dei "*bunch crossing*" di LHCb (cioè con la

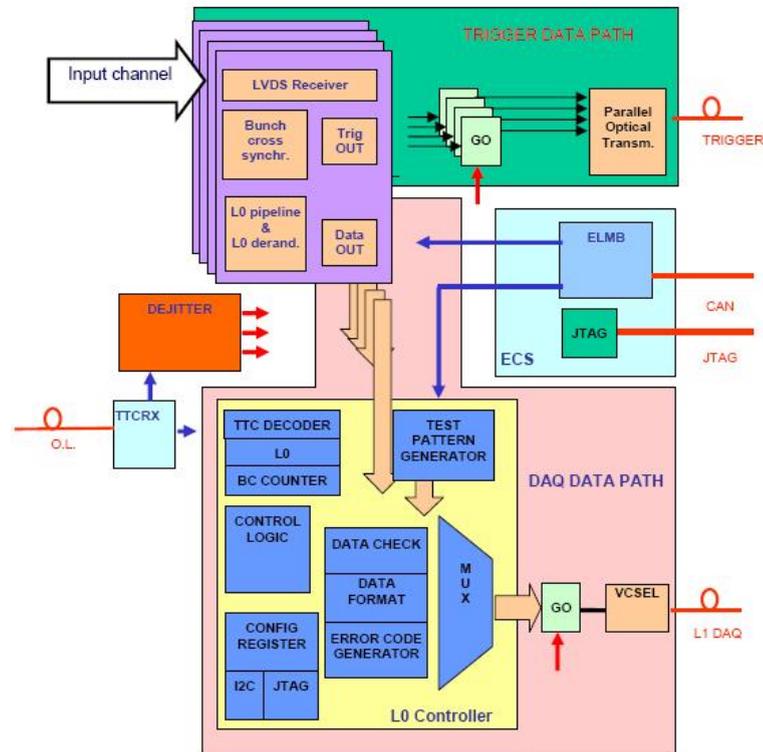


Figura 3.12: Diagramma di funzionamento delle schede ODE.

loro numerazione ciclica), esso può determinare il corretto “*BX-id*” da assegnare ai segnali ricevuti.

Tredici circuiti integrati di tipo ASIC, denominati GOL (“*Giga bit Optical transmitter*”) e progettati con tecnologia resistente alle radiazioni, gestiscono invece le comunicazioni su *link* ottico e permettono l’invio dei dati al *trigger di livello 0*.

Le schede ODE hanno le dimensioni di un modulo 6U-VME e sono contenute in dei *crate* situati nella caverna del rivelatore, negli stessi *rack* dove si trovano gli *SB-crate*: ogni *crate* contiene 18 schede.

3.3.3 L’SB-System: le schede Service Board e il modulo Pulse Distribution Module

Il *Service Board System* è quella parte dell’ECS che si occupa del monitoraggio e del controllo delle schede di *Front-End* del rivelatore di muoni [46]. Esso fa uso essenzialmente di tre unità: di un *crate* per schede nello standard 6U dotato di un *backplane* realizzato *ad-hoc* per il sistema, del *Pulse Distribution Module* [47] e del modulo *Service Board* [48], da cui prende il nome. In figura 3.13 è riportata una foto del *crate* utilizzato in laboratorio e dei primi *crate* montati in giugno.

Ogni *crate* del sistema alloggia al suo interno fino a 20 schede *Service Board* (“*SB*”) ed un modulo *Pulse Distribution Module* (“*PDM*”); quest’ultimo, si interfaccia a 4 rami CAN-bus, che interessano altrettante zone di comunicazione (Figura 3.14), e al sistema di sincronizzazione e di controllo di LHC.

Il modulo PDM, realizzato su una scheda nello standard 6U e inserito in posizione centrale, smista le informazioni, provenienti dall’esterno, ed i segnali sincroni, opportuna-

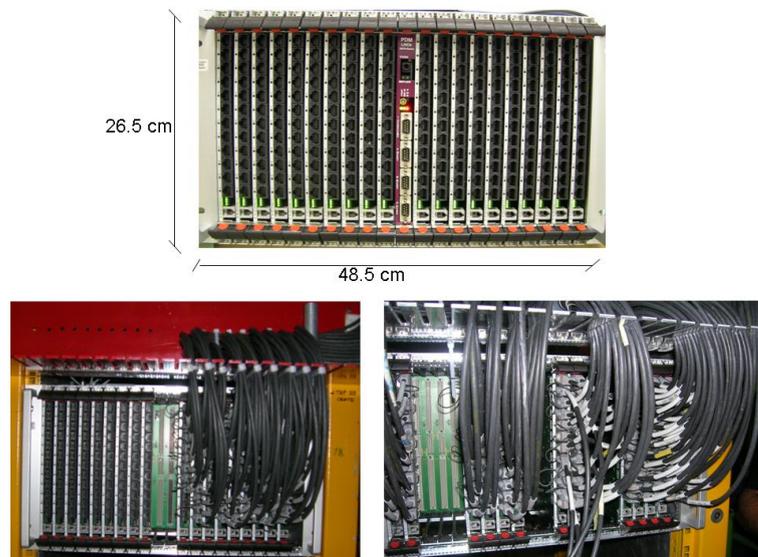


Figura 3.13: L'SB-crate: ogni crate può alloggiare fino a 20 schede SB; al centro è ben visibile la scheda PDM con la connessione per la fibra ottica e i quattro connettori per l'interfacciamento tramite altrettanti *bus* di tipo CAN. È visibile un *crate* completo di 20 schede SB e un modulo PDM, utilizzato in laboratorio, e i primi due *crate* montati a giugno 2007 nel pozzo 8.

mente generati al suo interno, lungo il *backplane*, rendendoli disponibili a tutte le schede dell'SB-crate. Essendo stato parte principale di questo lavoro di tesi, il modulo PDM verrà discusso nel capitolo 4.

La scheda SB ha le dimensioni dello standard *6U* ed alloggia 4 schede ELMB. Ogni ELMB può servire tre P^2C *bus* per lunga distanza e uno locale.

A loro volta, le schede ELMB sono connesse al *Front-End* del rivelatore, ciascuna tramite 12 cavi *twisted* a 10 fili che trasportano i segnali del bus " P^2C -like" e l'impulso di

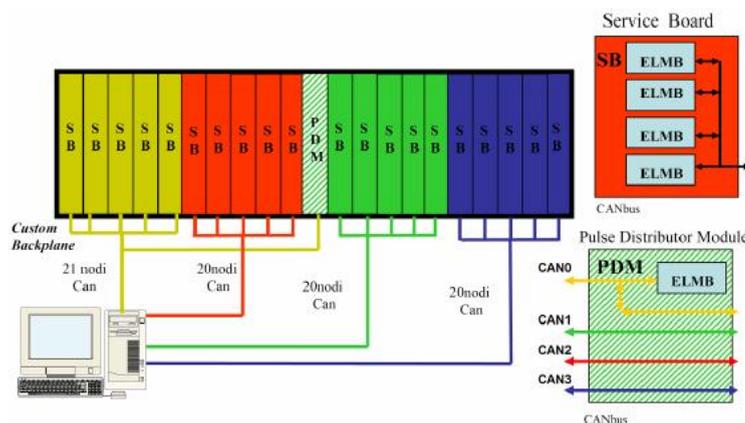


Figura 3.14: Schema dell'SB-crate: la scheda PDM permette la connessione dei calcolatori grazie a quattro *bus* di tipo CAN. Ogni unità PDM alloggia una scheda ELMB, mentre ogni unità SB ne alloggia quattro. In totale, su ogni SB-crate sono presenti fino a 81 nodi CAN.

calibrazione “*BC_pulse*” in standard LVDS.

Ognuno di questi rami collega con l’ECS fino a 8 *chip* CARDIAC e permette l’inizializzazione delle schede in fase d’accensione, la configurazione di tutti i parametri necessari al corretto funzionamento del *Front-End*, come i valori delle soglie d’acquisizione dei *canali fisici*, e la gestione della procedura di calibrazione dell’apparato di lettura delle camere, tramite opportuni segnali di test sincroni ed in fase.

Per velocizzare le operazioni di ripristino del sistema i valori dei parametri di calibrazione ottenuti, per ogni scheda CARDIAC collegata, sono memorizzati nella memoria *FLASH* a disposizione di ogni ELMB sulle schede *Service Board*: questo permette di ripristinare il corretto funzionamento del *Front-End* al riavvio od ogni volta che un errore, dovuto ad un evento di tipo SEU, corrompe il contenuto dei registri di configurazione, senza dover ripetere la lunga procedura di calibrazione.

Il “*Service Board System*” ha il compito di monitorare il corretto funzionamento delle schede CARDIAC a lui collegati e di effettuare tutte le operazioni necessarie a ristabilirne l’operatività, come il *reset* e il ripristino della configurazione. Ogni ramo di comunicazione con il *Front-End* dispone di una linea di *reset* globale, pilotata dalla scheda *Service Board* tramite un apposito registro I²C, che permette di effettuare un *reset hardware* di un intero ramo del *Front-End*. Alla scheda *Service Board* è affidato il compito di monitorare il corretto funzionamento di questo sistema e di risolvere eventuali problemi sia ripristinando il corretto funzionamento con i mezzi appena descritti sia, in caso di guasto, permettendo di individuare ed escludere i canali di lettura interessati attraverso diverse funzionalità di test. Uno schema con indicate le connessioni fra scheda SB e schede di FE è riportato in figura 3.15.

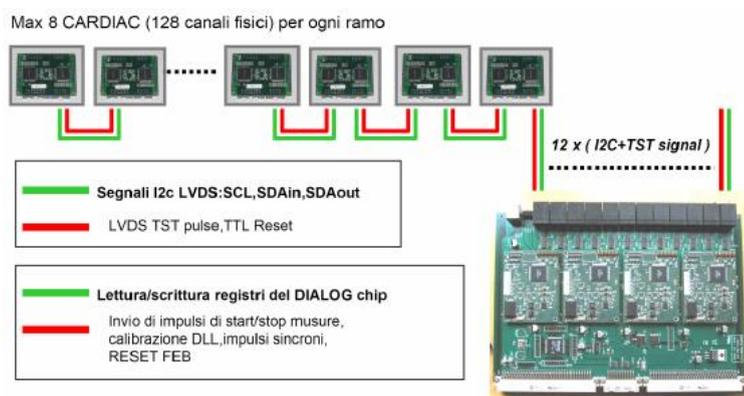


Figura 3.15: La comunicazione fra il Sistema di Controllo dell’Esperimento (ECS) ed il *front-end* (FE) del rivelatore di muoni avviene tramite la *Service Board* (SB). Ogni scheda SB dispone di 12 uscite verso altrettanti rami del FE. Ogni ramo può gestire fino a 8 schede CARDIAC

Altro importante compito del “*Service Board System*” è quello di curare la comunicazione con il sistema TFC di LHC (paragrafo 3.4) che distribuisce, a tutti i rivelatori in opera sull’anello di accumulazione LHC ed ai loro sottosistemi, i segnali necessari alla messa in tempo degli esperimenti ed i comandi *broadcast*. I segnali di sincronizzazione, come il *clock macchina* di LHC o il “*BX-id*”, vengono ricevuti tramite un connettore ottico contenuto nel modulo PDM e riallineati in tempo grazie a ritardi programmabili: questo permette di generare segnali utili per l’effettuazione di misure e per l’allineamento temporale di tutta

l'elettronica del rivelatore di muoni durante la fase di calibrazione dell'apparato.

L'SB-System è stato sviluppato dal gruppo LHCb della sezione INFN di Roma; in particolare il modulo PDM, descritto in dettaglio nel capitolo 4, e la procedura di calibrazione temporale, trattata nel capitolo 6, sono oggetto di questo elaborato di tesi.

3.4 Il sistema TFC (*Timing and Fast Control*)

Il sistema TFC (*Timing and Fast Control*) [49, 50] è preposto al controllo dell'intero processo di lettura dell'esperimento LHCb, dall'elettronica di *Front-End* fino alle *cpu* di analisi dati. L'insieme del sistema di controllo dell'elettronica ECS (*Experiment Control System*), del sistema di acquisizione dati DAQ (*Data Acquisition*) e del sistema di controllo dei tempi TFC (*Timing and Fast Control*), viene chiamato "*LHCb Online system*" [51].

In figura 3.16 è riportato uno schema del sistema "*LHCb Online system*".

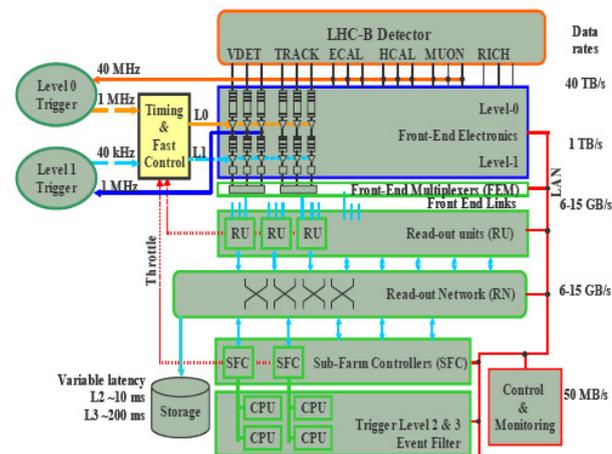


Figura 3.16: Schema del sistema "*LHCb Online system*".

L'esperimento LHCb richiede un sistema dedicato per la distribuzione di informazioni particolarmente sensibili, che necessitano di arrivare sincrone alle varie schede di elettronica sparse per l'esperimento; il TFC gestisce e distribuisce:

- Il *clock macchina* di LHC;
- I segnali con le decisioni del *trigger L0* e del *trigger L1*;
- I comandi di sincronizzazione e di *reset*;
- I contatori con il *BX-id* e con l'*EV-id*.

3.4.1 Specifiche richieste

Come detto, il compito del sistema TFC è quello di trasmettere informazioni sincrone da un punto centrale a tutti quei componenti del rivelatore di LHCb dove queste vengono richieste. In particolare, è possibile suddividere l'elettronica servita dal sistema TFC in due parti: l'"*elettronica L0*" e l'"*elettronica L1*", rispettivamente legate la *trigger L0* ed alla *trigger L1*.

In LHC è già presente un progetto con il compito di curare la trasmissione di dati in maniera sincrona su lunghe distanze: il progetto RD12-TTC, di cui si darà una breve descrizione nel paragrafo 4.2.2. Poichè le specifiche richieste dall'esperimento LHCb non sono del tutto soddisfatte da detto progetto, si è studiato ed implementato un sistema a parte, che integra le funzionalità del progetto RD12-TTC, dando vita al sistema TFC [52, 53].

In tabella 3.2 sono riportate le specifiche richieste per l'esperimento LHCb, messe a confronto con quelle fornite dal progetto RD12-TTC:

Parametro	LHCb	RD12-TTC
Frequenza di <i>clock</i>	sincrona con i fasci ($\sim 40MHz$)	sincrona con i fasci ($\sim 40MHz$)
Numero di fasi di <i>clock</i>	≥ 2	2
Intervallo di fase	$\sim 100ns$	$400ns$
Risoluzione della fase	$\sim 100ps$	$\sim 100ps$
<i>clock jitter</i>	$< 200ps$	$35ps-129ps^1$
Frequenza <i>Trigger Level-0</i>	$40MHz$	$40MHz$
Frequenza <i>Trigger Level-1</i>	$\sim 1MHz$	-

Tabella 3.2: Specifiche richiesta dall'esperimento LHCb per il sistema TFC messe a confronto con quelle offerte dal progetto RD12-TTC.

Il sistema TFC si compone di quattro moduli principali:

ODIN (Readout Supervisors) [54, 55, 56]: Riceve, sincronizza e distribuisce i segnali delle unità decisionali dei *trigger L0* ed *L1* e del *clock macchina*; inoltre è possibile inviare vari segnali di test e di calibrazione. In figura 3.17 è riportato uno schema semplificato con le principali funzionalità della scheda;

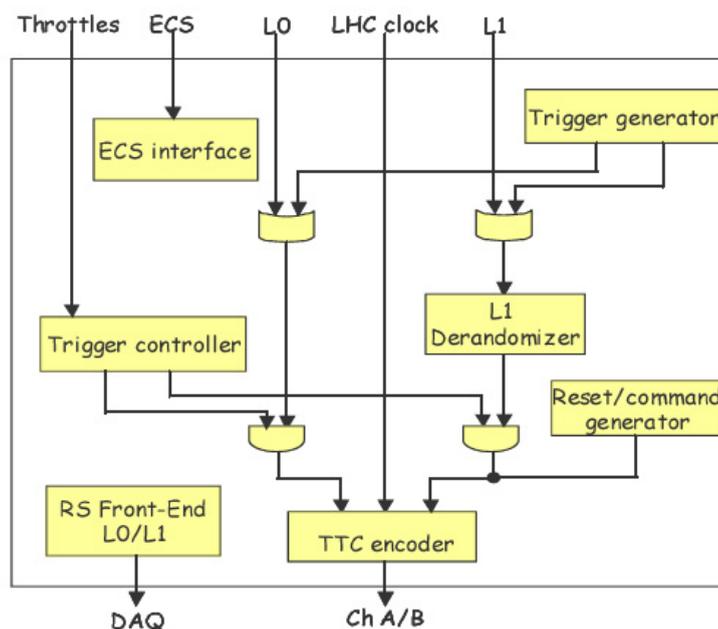


Figura 3.17: Diagramma logico delle funzioni del *TFC Readout Supervision*. Sono indicate le funzioni base della scheda ODIN.

THOR (TFC Switch) [57]: Distribuisce le informazioni ed i segnali del sistema TTC a tutta l'elettronica di *Front End*;

MUNIN (Throttle Switch): Gestisce i *throttle signal*, l'*L1-derandomizer* e i *data-driven* del sistema di *trigger L1*, inviandoli al giusto *Readout Supervisor*;

HUGIN (Throttle OR): Fornisce un livello di OR logici per i *Throttle signal* per la giusta configurazione dell'elettronica di *Front End*.

Capitolo 4

Il Pulse Distribution Module

4.1 Generalità

Il modulo PDM (*Pulse Distributor Module*), di cui si riporta una foto in figura 4.1, è realizzato su una scheda delle dimensioni nello standard $6U$ sarà alloggiato nella posizione centrale dell'*SB-crate*. Trovandosi il *Service Board System* nella prima zona esterna al rivelatore, il modulo PDM sarà soggetto ad una modesta dose di radiazioni: la quantità di particelle ionizzanti prevista nei 10 anni di funzionamento dell'esperimento è stimata in 10Krad. Questo permette di utilizzare, per i componenti elettronici della scheda, una tecnologia resistente alle radiazioni, notevolmente più economica di quella *rad-hard* utilizzata nella zona interna al rivelatore [47].



Figura 4.1: Il modulo *Pulse Distribution Module* (PDM)

4.1.1 Funzioni

Il modulo PDM, inserito all'interno dell'*SB-System*, provvede alle seguenti operazioni:

- Distribuisce il *clock macchina* di LHC a tutta l'elettronica di *Front End* (*CLK_40 signal*), rendendo possibile la procedura di calibrazione (descritta nel capitolo 6);
- Permette l'accesso al bus CAN (appendice B) che, attraverso il *Back-Plane*, raggiunge tutte le schede dell'*SB-crate*;
- Riceve da parte del Sistema *TTC* (*Time and Trigger Control*) ed elabora i segnali di temporizzazione (par 4.3.2);
- Produce e distribuisce i segnali impulsivi correlati a specifici *BX-id* (par 4.3.2);
- Tramite un bus di comunicazione dedicata, permette l'invio di comandi per il *reset* e lo *shut down* di ogni singola ELMB (*Embedded Local Monitor Board*) delle schede *Service Board* (par 4.2.4).

Occorrono circa novanta microsecondi ad ogni protone per compiere un giro completo del collisore LHC. Il fascio di protoni non è continuo, ma impulsato in “pacchetti di protoni” (chiamati *bunch*) in modo che le interazioni tra i fasci (*bunch crossing*) possano avvenire in momenti determinabili a priori. Il meccanismo è dovuto alla natura dell'acceleratore e la frequenza di impulsaggio è correlata a quella delle cavità a radiofrequenza utilizzate per accelerare il fascio di LHC. Questa struttura è indispensabile se si vuole associare un numero identificativo (*BX_id*) ad ogni evento per analizzare, in un secondo momento, i dati relativi ad una determinata interazione. Durante il funzionamento di LHC, saranno presenti 2808 pacchetti di protoni, distanziati secondo lo schema mostrato in figura 4.2; come è possibile notare, i pacchetti non sono equispaziati, ma seguono uno schema ben preciso, dovuto ai tempi di iniezione delle particelle nel percorso del collisore; in figura 4.3 è mostrato un secondo schema possibile, con maggiore simmetria, ma minor numero di *bunch* [58].

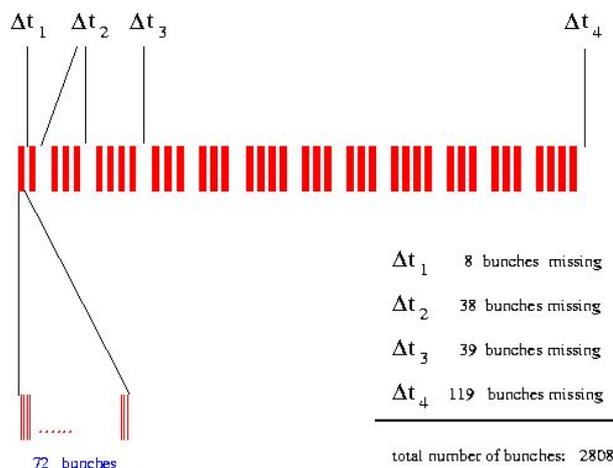


Figura 4.2: Schema d'iniezione delle particelle nel collisore LHC: sono previsti 2808 *bunch* per fascio, il cui intervallo temporale è di $25ns$ (considerando anche i *bunch* vuoti).

La risoluzione temporale richiesta all'*SB-System* è dell'ordine del nanosecondo. Questo valore è dettato, nel particolare, da un fattore pratico: l'allineamento in fase è realizzato

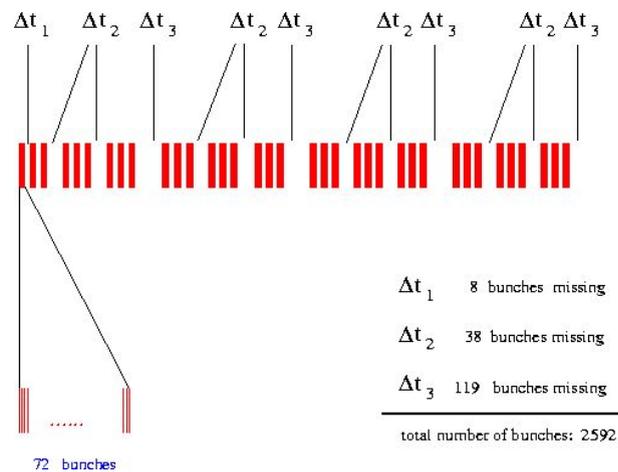


Figura 4.3: Secondo schema possibile per l'iniezione delle particelle nel collisore LHC; questo schema è estremamente simmetrico, ma prevede solo 2592 *bunch* per fascio.

con un registro a 4 bit, suddividendo quindi l'intervallo temporale di $25ns$ tra un pacchetto e l'altro in sedici parti; in questo modo si ha un valore, per ogni singolo ritardo, di $1.56ns$, facilmente ottenibile sfruttando, per la realizzazione pratica, le caratteristiche intrinseche dei ritardi delle porte logiche. Questa risoluzione è buona per gli scopi dell'esperimento e non v'è bisogno di cercarne una maggiore.

Come illustrato in figura 4.4, i quattro nodi CAN-bus, tramite cui l'interfaccia utente dialoga con tutte le schede dell'*SB-crate*, convergono nel modulo PDM.

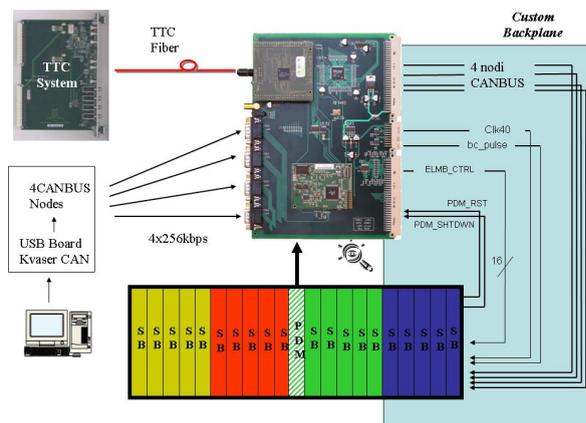


Figura 4.4: Funzioni principali del modulo PDM

I calcolatori su cui verrà caricato il programma in PVSS per la gestione dell'elettronica di FE (cap 5) saranno equipaggiati con un adattatore USB-CAN che verrà collegato direttamente al modulo PDM; di qui i segnali procederanno sul CAN-bus implementato nel *Back Plane* e raggiungeranno le schede *Service Board*.

Tramite il canale *CAN_0* è possibile accedere alla scheda del PDM; ogni canale CAN inoltre, collega fino a cinque schede SB.

Il modulo PDM ha pieno accesso ad un *bus* dedicato che, attraverso il *Back Plane*, collega ogni scheda *Service Board* e comanda un *reset* od un *power cycle* delle ELMB.

È quindi possibile, all'occorrenza, resettare una o più schede SB da remoto, inviando il corretto comando dal modulo PDM (par 4.2.4).

4.1.2 Architettura

In figura 4.5 è riportato il diagramma a blocchi con i componenti principali del modulo PDM.

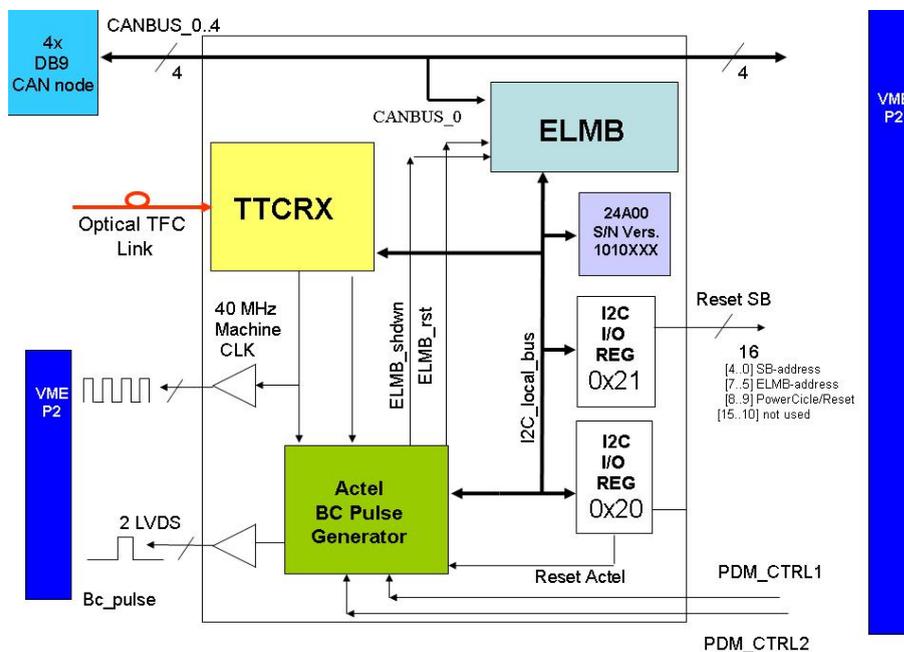


Figura 4.5: Diagramma a blocchi del modulo PDM

I componenti principali montati sulla scheda del PDM sono:

- una ELMB (*Embedded Local Monitor Board*):** [41] un microcontrollore con accesso CAN a cui è affidata la gestione dell'intera scheda (paragrafi 3.3.1, 4.3.1 e 4.2.1);
- un TTCrq (*Timing Trigger and Control*):** [59] un modulo che riceve un segnale su fibra ottica e lo rende disponibile elettronicamente alla scheda (paragrafo 4.2.2);
- una FPGA (*Field Programmable Gate Array*),** [60] a cui è affidata la logica necessaria per interpretare i comandi ricevuti dal Sistema TTC (*Time and Trigger Control*) (paragrafi 4.2.3 e 4.3.2);
- due registri a 16 bit PCF 8575,** [61] tramite cui è possibile effettuare fondamentali operazioni di ripristino del sistema in caso di errore (*reset* e *power cycle*) (paragrafo 4.2.4).

Inoltre, sono presenti un bus I²C nello standard *TTL* (*Transistor-Transistor Logic*), utilizzato per la comunicazione interna, quattro bus CAN, utilizzati per la comunicazione con il resto dell'*SB-crate*, e vari canali in standard *LVDS-bus*, utilizzati per mandare segnali *broadcast* dedicati sul *Back-Plane* dell'*SB-crate*.

Sul *Back Plane* dell'*SB-crate* corrono diverse linee per il trasferimento dei segnali dal modulo PDM alle schede *Service Board*:

Quattro canali CAN-bus, attraverso cui è possibile dialogare con le schede SB;

Il CLK_40 ed il BC_pulse, due linee differenziali dedicate, su cui passano gli omonimi segnali. Queste linee, nello standard *LVDS-bus* sono terminate ai due estremi del BP tramite resistenze da 56Ω nella configurazione “*broadcast bus*”, con un *master* al centro della linea e gli *slave* ai due lati (figura 4.6). Portano l’informazione, rispetti-



Figura 4.6: Il bus *LVDS-bus* nella configurazione con doppia terminazione

vamente, del *clock macchina* e dell’impulso di calibrazione del sistema (quest’ultimo verrà discusso in dettaglio nel paragrafo 4.3.2 e nel capitolo 6);

L’SB_control_bus, un *bus* di comunicazione parallelo a 16 bit, pilotato dal PDM, tramite cui è possibile mandare dei comandi alla FPGA di ciascuna scheda SB; in particolare è stato implementato il comando di *reset* o di *power cycle* per ogni singola scheda ELMB presente nel *crate* (paragrafo 4.2.4);

Il PDM_SHT ed il PDM_RST, due canali, pilotati da ognuna delle SB presenti nel *crate*, tramite cui è possibile comandare un *reset* od un *power cycle* dell’ELMB del PDM.

La scheda del *Pulse Distribution Module* è inserita al centro dell’*SB-Crate* (nello slot_11) e può gestire fino a 20 schede *Service Board*. Tramite la fibra ottica connessa al modulo TTCr_q, la scheda del PDM riceve il *clock* sincrono di LHC ed i messaggi che il *TTC-System* distribuisce a tutto l’esperimento.

Sulla base delle informazioni ricevute dal *TTC-System*, il modulo PDM, grazie alla logica implementata nel *chip* FPGA, genera e distribuisce due segnali principali: il clock a 40MHz (*CLK_40 signal*), in fase con il clock di LHC, e il segnale *BC_Pulse*, correlato ad un preciso *Bunch Count Identifier* o ad uno specifico comando proveniente dal TFC.

Come meglio spiegato nel capitolo 6 il segnale *BC_Pulse* viene usato nella procedura di calibrazione dell’elettronica e ricopre un ruolo importante per l’assegnazione del giusto *BX-identifier* al set di misure, essendo il segnale base per la procedura di allineamento temporale dell’elettronica. Nel paragrafo 4.3.2 verranno indicate le diverse modalità di generazione del segnale *BC_Pulse*.

4.2 Componenti

4.2.1 Personalizzazione della scheda ELMB

La descrizione *hardware* della scheda ELMB è stata esaminata nel paragrafo 3.3.1; in questo si esamineranno le potenzialità di programmazione e nel paragrafo 4.3.1 si vedrà il *software* implementato per il modulo PDM.

È possibile caricare il *software* per il microcontrollore in due modi distinti: il primo, utilizzato in laboratorio e necessario per il caricamento del *boot loader*, consiste nel caricare il programma in una memoria EEPROM; nel caso specifico ciò è stato fatto tramite

il connettore dell'interfaccia ISP (*In System Programming*), utilizzando un apposito programmatore e *software* di scrittura/lettura: il "PonyProg2000". Il secondo modo è utile per una riprogrammazione delle schede ad esperimento montato, ma sarà possibile il caricamento del solo "*firmware*"¹: non essendo possibile, all'occorrenza, accedere fisicamente al *crate* una volta acceso l'acceleratore LHC, è stata implementata la possibilità di programmare la scheda ELMB tramite l'interfaccia CAN; si può quindi sospendere l'operatività di una singola o di tutte le schede ELMB presenti in un nodo CAN e riprogrammarne il *firmware* del microcontrollore.

In pratica, grazie all'invio di un comando SDO specifico, il *boot loader* riprende il controllo del microcontrollore ed entra in uno stato di programmazione; in questo stato, il *boot loader* attende che venga inviato un codice via CAN-bus con cui programmare l'*application software*. Non appena viene fornito il codice tramite CAN, il microcontrollore viene programmato ed è necessario procedere ad un *reset* della scheda affinché il nuovo codice sia operativo.

È possibile gestire i 128Kbits della memoria flash dell'ATmega128 organizzandoli in due blocchi logici, di dimensioni variabili: il "*Boot Program*" e l'"*Application Program*". La differenza fra i due è che il primo blocco dispone di un sistema di bit di protezione che ne evita la sovrascrittura accidentale e garantisce quindi una maggiore affidabilità; viene qui caricata tutta la logica necessaria all'inizializzazione del *chip* ed il programma di avvio: il *boot loader*.

Nel secondo blocco, è scritto il programma principale: il *firmware*. È per questo motivo che, tramite CAN, è possibile riprogrammare solo il *firmware*: il *boot loader* è "protetto in scrittura" e solo un accesso fisico alla sua memoria garantisce l'intenzionalità della riprogrammazione.

All'accensione dell'*hardware* il microcontrollore esegue il *boot loader* che fra le sue funzioni ha quella di consentire la scrittura del *firmware* attraverso il protocollo CANOpen. Se nella parte di memoria flash "*Application Program*" risiede il programma principale, dopo 4 secondi il *boot loader* l'esegue. In caso contrario il microcontrollore resta in attesa finché non è attivato da un messaggio CAN o dalla scrittura del programma principale, la cui presenza viene rilevata ogni 4 secondi. In caso di chiamata da parte di una *subroutine* o di un *interrupt* i dati necessari al ripristino del programma principale vengono salvati nello *stack*. Per permettere una rapida esecuzione dei salti, lo *stack* viene allocato nella memoria SRAM a disposizione del microcontrollore, particolarmente sensibile agli effetti degli eventi dovuti alla radiazione denominati SEU (*Single Event Upset*). Un eventuale deterioramento di questi dati rende impossibile il ripristino del programma principale provocando un arresto del sistema. Per ovviare a questo inconveniente il *boot loader* fornisce anche una funzionalità denominata "*watch-dog*" per la rivelazione d'eventuali situazioni di *dead-lock* che, in caso d'errore, provvede a riavviare la scheda ripristinando nuovamente il codice dalla memoria flash.

4.2.2 Il sistema TTC

Il sistema TTC (*Timing Trigger and Control*) [62] è stato sviluppato come parte del progetto *RD12* ed è formato da una serie di schede elettroniche disegnate dal gruppo di micro-elettronica del CERN. Queste schede sono state studiate per distribuire tutti i segnali di sincronizzazione (*clock* e *reset*) e di *trigger*, lungo le grandi distanze di LHC alle

¹Per una distinzione logica tra *firmware* e *boot loader* si veda più avanti o, per lo specifico, il paragrafo 4.3.1.

varie *elettroniche di Front End*. In tabella 4.1 è possibile vedere la lista completa dei moduli supportati dal *TTC-System*.

Scheda	Descrizione
TTCvi	modulo d'interfaccia TTC - VMEbus
TTCrm	scheda di sviluppo per il TTCrx
TTCrq	nuova versione del TTCrm con QPLL
TTCvx	modulo di trasmissione per <i>low-power</i> VMEbus
TTCex	trasmettitore ottico primario
TTCtx	trasmettitore ottico secondario
TTCmx	trasmettitore ottico primario per l'uso di TTCmi
TTCmi	interfaccia TTC
TTCoc	accoppiatore ottico (1:16 e 1:32)

Tabella 4.1: Lista dei moduli TTC supportati

Il *TTC-system* distribuisce informazioni verso un grandissimo numero di ricevitori fornendo un segnale con un *jitter temporale* molto contenuto. Per trasportare il segnale da un elemento all'altro del *TTC-system* vengono usati dei cavi in fibra ottica, che rappresentano un ottimo media per i segnali digitali; nonostante l'ottima qualità del mezzo di trasporto, le grandi distanze presenti in LHC e l'impossibilità di equalizzare tutte le linee di trasmissione fanno sì che ogni apparato di ricezione (*chip* TTCrx) riceva un segnale con una fase diversa, dipendente dalla propria posizione e dalla lunghezza del cavo cui è connesso. Per ovviare a questo problema, ogni *chip* TTCrx [63, 64] ha la possibilità di settare due ritardi, con risoluzione di $0.1ns$, utilizzati per allineare il *clock locale* al *clock macchina* o per allineare i segnali d'interfaccia fra i vari moduli.

Al fine di ottimizzare il segnale e ottenere un *jitter* inferiore ai $20ps$, condizione richiesta per il funzionamento dei *link* seriali ad alta velocità, sul modulo di ricezione è stato implementato un filtro speciale, chiamato QPLL e disegnato dal gruppo di microelettronica del CERN, al quale è associato un cristallo di quarzo e che ha la funzione di stabilizzare il segnale ricevuto.

La scheda di ricezione del segnale ottico, comune a tutti gli esperimenti in LHC e riportata in foto nella figura 4.7, prende il nome di TTCrq ed è composta da un *chip* TTCrx, un filtro QPLL e un *TrueLight pin-preamplifier*.

Per la risoluzione temporale utilizzata nel modulo PDM, non è necessario che il *jitter* temporale dei segnali sia inferiore ai $20ps$ e quindi il connettore del modulo TTCrq che permette l'utilizzo del filtro QPLL non è stato montato sulla scheda del PDM.

Il principale segnale fornito dalla scheda TTCrq al modulo PDM è caratterizzato dal valore registrato nel contatore a 12 bit "*bunch ID counter*" (BX-id), ossia l'esatto numero di *bunch* cui il rivelatore è interessato. Il contatore è incrementato grazie all'invio regolare del segnale di sincronizzazione *bunch count reset*. L'esatto allineamento della fase del registro, che deve tenere conto dei ritardi del *network*, è ottenuto in due passi: l'allineamento di fase del singolo periodo di *clock* è ottenuto grazie ad uno dei due generatori di ritardo nel *chip* TTCrx; l'allineamento con il giusto periodo di *clock* è invece ottenuto grazie ad una linea di ritardo specifica del *chip* TTCrx, programmabile fino a 15 cicli di *clock*.

Il "*bunch count reset*" è inviato ai *Front End* con un comando *broadcast* sincrono ad

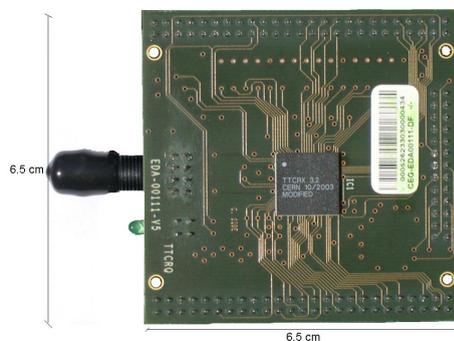


Figura 4.7: Il modulo di ricezione del segnale nel sistema TTC: il TTCr_q

otto bit, caratterizzato dal bit<0> attivo².

4.2.3 Il *chip* FPGA

Una FPGA (*Field Programmable Gate Array*) è un *chip* elettronico programmabile; il grande vantaggio nell'utilizzo di FPGA consiste nei tempi di "messa in opera": infatti sono notevolmente ridotti i tempi di sviluppo di una FPGA rispetto a quelli di un *chip custom* e soprattutto, nel caso di una produzione limitata, i costi sono notevolmente più contenuti.

L'architettura di una FPGA è rappresentata da una matrice di macrocelle, costituite da una parte di logica combinatoria programmabile e da elementi di memoria, *latch* o *flip flop* che comunicano fra loro e con i segnali di ingresso e uscita; queste comunicazioni avvengono mediante un insieme di collegamenti disposti orizzontalmente e verticalmente che è possibile programmare.

In pratica il progettista sceglie tra una libreria di elementi disponibili, sfruttando una disposizione comune, ma personalizzandone il collegamento e quindi l'architettura.

Esistono vari tipi di *chip* FPGA, a seconda della tecnologia usata e della casa produttrice:

- Case come la ALTERA e la XILINX producono *chip* basati su memorie RAM (SRAM o simile); questi *chip* caricano il codice programmato sulla memoria ogni volta che viene effettuato il *power up* attraverso la lettura di una PROM. Questa tecnologia è semplice e ben collaudata; di contro ha un elevato consumo energetico e il fatto che nelle memorie di tipo RAM sono frequenti errori SEU (*Single Event Upset*), in quanto molto sensibili a particelle ionizzanti.
- La ACTEL utilizza metodi di programmazione resistenti alle radiazioni; in particolare i *chip* della serie *ProAsic* si avvalgono di memorie FLASH³: si è così ottenuto un *chip* riprogrammabile, la cui logica è pronta all'uso ad ogni accensione e con il minor dispendio di energia della categoria; inoltre si ha una ottima tolleranza alle radiazioni.

Nella zona in cui saranno presenti gli *SB-crate* è previsto un livello di radiazioni di circa 10Krad nell'arco dei 10 anni di esperimento: è quindi necessario tenere nella giusta considerazione il problema.

²Per una trattazione dei segnali *broadcast* si vedano il paragrafo 4.3.2 e la tabella 4.2

³Fondamentalmente la differenza tra una memoria di tipo RAM (*random-access memory*) e una di tipo FLASH consiste nel fatto che le prime hanno bisogno di un'alimentazione per il mantenimento dei dati, mentre le seconde no; di contro le RAM presentano un tempo di accesso in lettura e scrittura inferiore delle FLASH

Sebbene non sarà possibile ad essere vivente stazionare nelle vicinanze dei *crate*, per l'elettronica risulta comunque modesta la quantità di particelle ionizzanti prevista; è stato quindi possibile trovare un compromesso tra adeguata resistenza alle radiazioni e produzione commerciale: per questo è stata scelta una "Actel Pro Asic^{plus} APA150" come FPGA. La serie "Pro Asic^{plus}" della ACTEL è una famiglia di FPGA basate su tecnologia "Flash 220nm".

L'architettura ProAsic^{plus} [60] è provvista di una struttura di tipo granulare dove ogni elemento base è chiamato "tebola"; il cuore del *chip* è provvisto di un così elevato numero di elementi, che l'FPGA viene chiamata letteralmente "mare di tegole" (*Sea-of-Tiles*TM). Ogni "tebola" è configurabile come *Flip-Flop*, *Latch* o come un elemento logico a tre ingressi ed una uscita, programmando lo *switch* costituito dalle singole celle Flash.

La struttura delle FPGA ProAsic^{plus} è stata studiata per fornire altissime prestazioni attraverso una struttura gerarchica a quattro livelli:

1. Le linee di percorso *Ultra Fast* locali (fig 4.8) che collegano l'uscita d'ogni "tebola" direttamente all'ingresso di una delle otto "tebole" che la circondano.

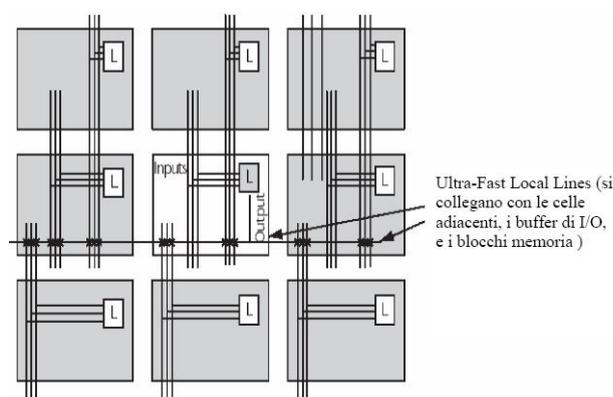


Figura 4.8: Struttura delle interconnessioni locali delle FPGA *Pro Asic^{plus}* della ditta ACTEL; è visibile il sistema denominato "*Ultra Fast local line*".

2. Le linee di risorsa *Long-Line resources* (fig 4.9) che permettono l'indirizzamento a lunghe distanze con connessioni ad alto numero d'uscite. Questa connessione varia in lunghezza e può collegare fino a 4 "tebole" percorrendo l'intero *chip* in verticale o in orizzontale.
3. *L'High Speed very long-line* (fig 4.10) che si estende sull'intero *chip* con il minor ritardo possibile e che è usata per linee molto lunghe e con molte uscite.
4. *L'high performance global network* (fig 4.11) è una rete a bassa distorsione per reti a molte uscite; i suoi segnali sono accessibili dai *pin* esterni o dalla logica interna. Questa rete è tipicamente usata per distribuire i segnali di *clock*, di *reset* e altri segnali che hanno necessità di essere guidati con la minima distorsione possibile a più elementi. La Rete Globale di distribuzione del segnale di *Clock* è implementata con un collegamento ad albero; in questo modo i segnali possono essere introdotti in ogni elemento. Questi possono essere impiegati in modo gerarchico con ogni segnale arrivando a collegare ogni "tebola".

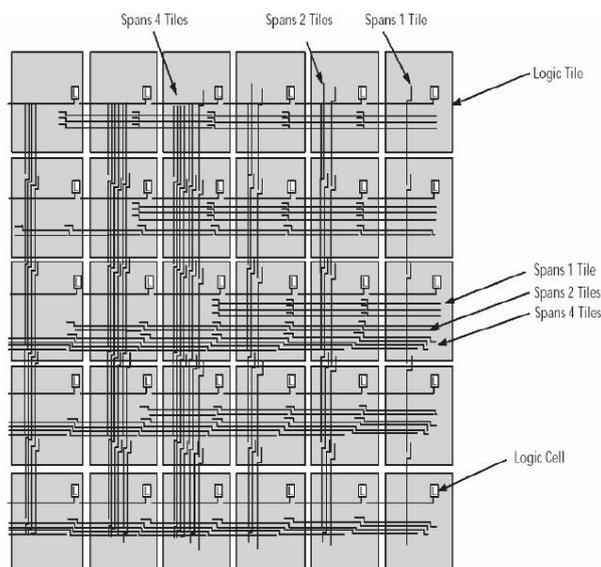


Figura 4.9: Struttura delle interconnessioni locali delle FPGA *Pro Asic^{plus}* della ditta ACTEL; è visibile il sistema denominato “*Long Line*”.

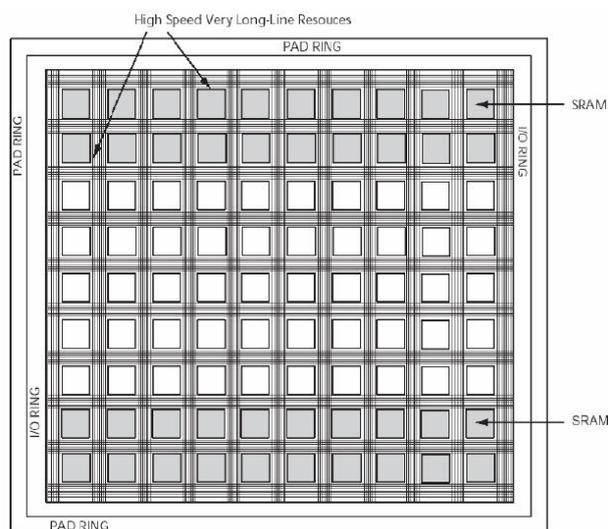


Figura 4.10: Struttura delle interconnessioni locali delle FPGA *Pro Asic^{plus}* della ditta ACTEL; è visibile il sistema denominato “*High Speed very long line*”.

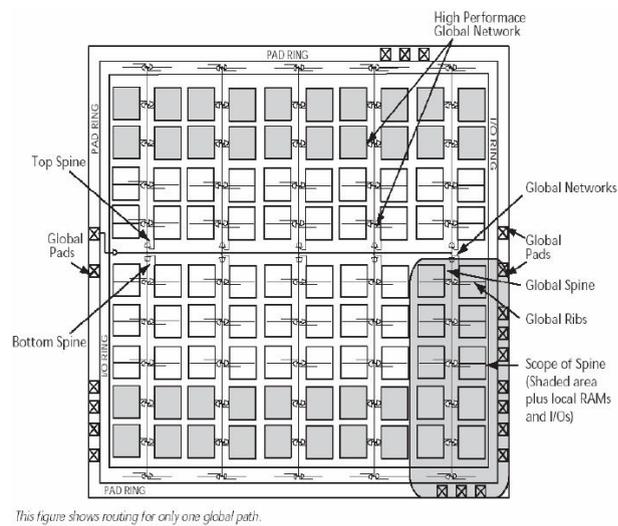


Figura 4.11: Struttura delle interconnessioni locali delle FPGA *Pro Asic^{plus}* della ditta ACTEL; è visibile il sistema denominato “*High Performance Global Network*”.

Ogni cella logica è caratterizzata da tre ingressi, ognuno dei quali può essere invertito, e un'uscita che può essere collegata ad una linea *Ultra-Fast* o ad una *Long-Line*. Ognuno dei moduli a tre ingressi e un'uscita può essere configurato come una "tegola" eccetto per la porta XOR a tre ingressi. La tegola, il cui schema è riportato in figura 4.12, può essere configurata come *Latch* o come *flip flop* con le funzioni *clear* e *set*. La cella dunque consente di essere usata in modo molto flessibile come cella logica o porta sequenziale.

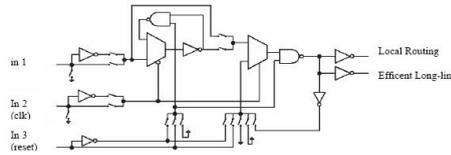


Figura 4.12: Schema della *tegola logica* di una ACTEL Pro Asic^{plus}

Diversamente dalle FPGA con tecnologia SRAM, le logiche ProAsic^{plus} utilizzano una tecnologia pronta a funzionare non appena si accende l'alimentazione; questo grazie all'implementazione degli *switch ISP Flash* come elemento di programmazione. Gli *switch* di questo tipo, di cui si riporta uno schema in figura 4.13, sono costituiti da due transistori che possono configurare una porta in una delle seguenti condizioni:

sensing transistor, che è utilizzata essenzialmente per scrivere e verificare la tensione della porta variabile.

switching transistor, che può essere usata per connettere o separare collegamenti nella rete o per configurare la logica, inoltre, può essere usata per cancellare l'impostazione della porta variabile.

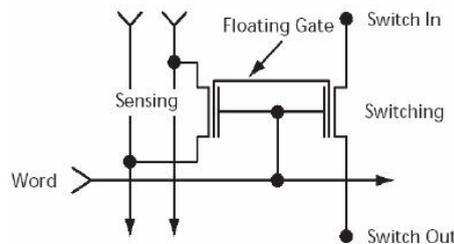


Figura 4.13: schema dello *switching gate* di una ACTEL Pro Asic^{plus}

Nella FPGA è implementata tutta la logica necessaria alla codifica e all'esecuzione dei messaggi *broadcast* di calibrazione di LHC oltre che tutti quei segnali necessari al controllo delle *Service Board* e all'emissione d'impulsi di *test* e segnali diretti all'elettronica di *Front End* (vedi paragrafo 4.3.2).

4.2.4 Registri I²C: i PCF

Sulla scheda PDM sono infine presenti due registri I²C "Philips PCF 8575" a 16 bit.

Agendo sul bit <15> del PCF all'indirizzo 0x20, è possibile effettuare un *power cycle* del sottosistema APA-TTC; infatti, il bit comanda il segnale "RST_ACTEL" che, attivo basso, comanda un *reset* dell'ACTEL e toglie l'alimentazione al TTCrq.

Attualmente i restanti bit del registro sono riservati per applicazioni future.

L'altro registro, raggiungibile all'indirizzo 0x21, è strettamente correlato al bus "SB_control_bus" e permette il *reset* o il *power cycle* delle ELMB montate sulle *Service Board* presenti nel *Crate*. In figura 4.14 è mostrato uno schema delle connessioni al bus "SB_control_bus" della scheda SB e della scheda PDM.

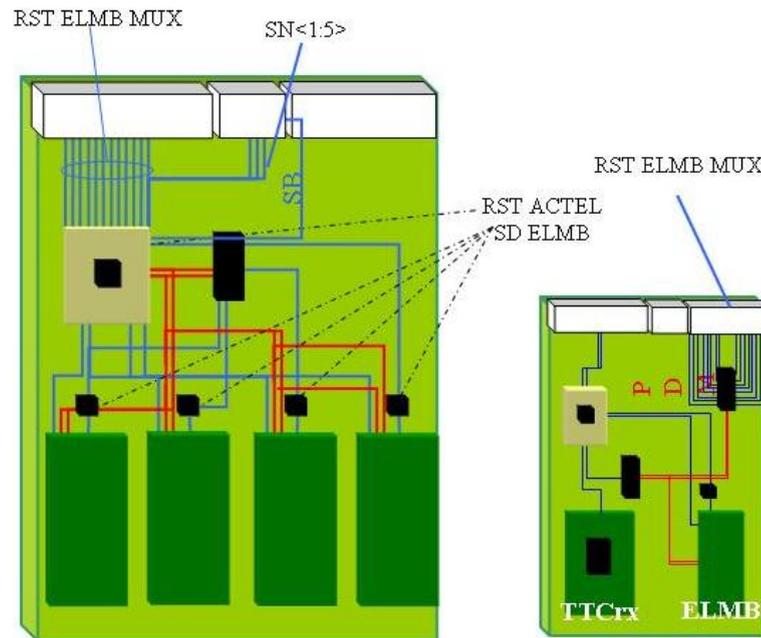


Figura 4.14: Schema di reset delle schede ELMB installate sulle *Service Board* (SB): grazie alla logica implementata nelle FPGA dei moduli SB, è possibile eseguire un *reset* od un *power cycle* delle schede ELMB presenti sulle SB inviando il giusto segnale dal modulo PDM. La figura di sinistra rappresenta le connessioni nella scheda SB, mentre quella di destra le connessioni nella scheda del PDM.

Il bus SB_control_bus viaggia sul *Back-Plane* e arriva ai *chip* FPGA delle *Service Board*; a seconda dei valori dei 16 bit del bus, i *chip* FPGA pilotano o meno il *reset* o il *power cycle* delle relative ELMB. La codifica del bus è stata implementata secondo lo schema seguente:

- i bit [4..0] indicano l'indirizzo della scheda SB nel crate; da 1 a 21 (attenzione: nello slot_11 è sito il modulo PDM, quindi a questo indirizzo non risponderà nessuna SB ed il comando verrà ignorato).
- i bit [7..5] indicano l'indirizzo della scheda ELMB; da 0 a 3 per la singola scheda, 5 per indicarle tutte; il valore 4 non è utilizzato.
- i bit [9..8] indicano se operare uno *shutdown* o un *reset*; 01 *reset*, 10 *shutdown* (per ottenere un *power cycle*, si pilota uno spegnimento seguito dal rilascio della linea).
- i bit [15..10] sono attualmente riservati per applicazioni future.

Se si invia un comando di *Shut Down* ed il PCF non viene successivamente sovrascritto con un valore diverso, la/le relative schede ELMB rimarranno non alimentate. Questa opzione è stata introdotta per poter spegnere una o più schede ELMB di un *crate* in caso di necessità.

4.3 Funzionalità ed Utilizzo

Il modulo *PDM* ha molte potenzialità e si presta bene per un gran numero di funzioni che vanno ben oltre la semplice distribuzione dei segnali ricevuti dal sistema TTC; infatti è possibile eseguire una prima calibrazione dell'elettronica anche in assenza del fascio di protoni di LHC, sfruttando i segnali generati dal modulo PDM stesso (si veda il capitolo 6).

Due sono i protocolli di trasmissione utilizzati dal modulo PDM: le comunicazioni interne fra i vari componenti della scheda avvengono su *bus* I²C secondo un protocollo I²C-like (appendice A), mentre le comunicazioni tra la scheda e l'interfaccia utente viaggiano su *bus* CAN secondo il protocollo CANOpen (appendice B).

4.3.1 Il *firmware* implementato nella scheda ELMB

Il codice implementato nella scheda ELMB è suddiviso in due blocchi logici: il *boot loader* ed il *firmware* vero e proprio.

Il primo è scritto nel “*Boot Program*” della memoria *FLASH* e gode di un sistema di bit di protezione che ne evita la sovrascrittura accidentale; il *boot loader* è incaricato di inizializzare la scheda e gestire tutte le operazioni di avvio del microcontrollore. Praticamente, può essere visto come un microcontrollore virtuale, con il compito di riprogrammare il microcontrollore vero e proprio; di fatto è questa l'operazione che svolge in caso di arresto del microcontrollore dovuto a corruzioni dovute ad eventi di tipo SEU.

Ciò che veramente differenzia il codice delle schede ELMB destinate ai moduli PDM da quelle montate sulle schede SB è rappresentato dal *firmware*. Scritto nell’“*Application Program*” il *firmware* non gode di alcun sistema di protezione da scritture remote ed è possibile aggiornarlo in qualsiasi momento tramite il *bus* CAN.

Come visto, il modulo PDM può funzionare correttamente solo se viene inserito al centro del *crate*, nello *slot_11*; siccome non è banale la possibilità di errore nell'inserimento della scheda, è stata implementata una *routine* che, a conclusione delle operazioni di avvio ed inizializzazione della scheda, legge dal *Back Plane* l'indirizzo dello *slot* di inserimento e comunica visivamente all'operatore la posizione del modulo: se l'indirizzo d'inserimento corrisponde allo *slot_11*, allora si accenderanno entrambi i led posti sul pannello verticale mentre, se l'indirizzo indica uno *slot* alla destra od alla sinistra di quello centrale, lampeggerà rispettivamente il led giallo o quello verde e l'altro rimarrà spento.

L'accensione dei due led, oltre ad indicare il corretto inserimento del modulo PDM nel *crate*, indica anche l'avvenuto *reset* dell'intera scheda.

Infatti, l'operazione di controllo della posizione di inserimento nel *crate* è successiva ad un *power cycle* ed alla completa inizializzazione dei componenti della scheda.

Come spiegato nel prossimo paragrafo, le linee di alimentazione e settaggio della scheda TTCrq transitano nel *chip* ACTEL e quindi, effettuando un *power cycle* su questo, si effettua automaticamente un *power cycle* anche sul modulo del TTC. Con la riscrittura dei registri il *reset* della scheda del PDM è completo.

All'interno del *firmware* dell’*Embedded Local Monitor Board* sono implementati inoltre tutti gli oggetti necessari al funzionamento del protocollo CANOpen e tutti gli indirizzamenti dei registri e delle periferiche I²C (si veda l'appendice C)

- 10 : A seguito di un comando diretto dell'utente, una macchina a stati sincrona implementata nella logica dell'FPGA, genera un impulso singolo; questa modalità, insieme con la precedente, è stata molto utilizzata nella fase di test della scheda.
- 11 : È infine possibile generare il segnale *BC_pulse* in maniera sincrona e in fase con il *clock macchina* di LHC, attivato dall'apposito comando *broadcast* di sincronizzazione proveniente dal sistema TTC; sarà questa la normale modalità di funzionamento. In figura 4.16 è riportato un esempio della generazione dei vari impulsi a seguito del comando *broadcast* da parte del sistema TTC.

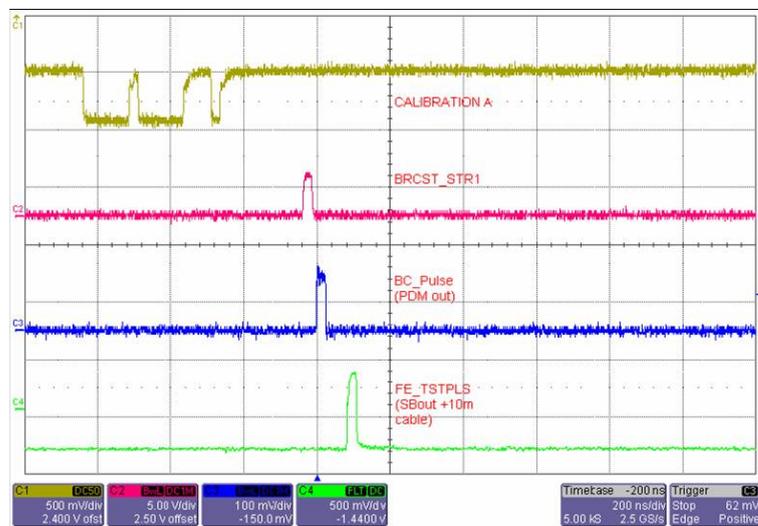


Figura 4.16: Misura all'oscilloscopio del segnale *BC_Pulse*: si vede la generazione del segnale *broadcast* da parte del sistema TTC (*CALIBRATION_A*), la successiva ricezione del segnale da parte del *chip* TTCrq (*BRCST_STR1*) e generazione dell'impulso *BC_Pulse* da parte del *chip* FPGA del modulo PDM; è visibile anche il segnale rilevato all'uscita dei cavi delle schede SB e destinate alle schede CARDIAC (*FE_TSTPLS*)

Tutta la logica necessaria alla codifica ed alla giusta esecuzione dei messaggi *broadcast* dell'esperimento è implementata nel *chip* FPGA: i segnali *broadcast* giungono all'*SB-System* sulla fibra ottica connessa al modulo TTCrq; tramite un *bus* parallelo nello standard TTL i segnali arrivano al *chip* della ACTEL che li interpreta ed esegue i relativi comandi. in tabella 4.2 è indicato l'elenco attuale dei *broadcast* usati nell'esperimento.

LHCb command	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NOOP	0	0	0	0	0	0	0	0
B-ID reset	0	0	0	0	0	0	0	1
L0-ID reset	0	0	0	0	0	0	1	0
ROT	1	Readout type			L0-ID[1:0]		L0-ID reset	B-ID reset
Reset	0	1	reserved		L0 FE reset		L0-ID reset	B-ID reset
Calibration Pulse	0	0	0	1	Pulse type		L0-ID reset	B-ID reset
CMD1 (reserved)	0	0	0	0	x	x	L0-ID reset	B-ID reset
Snapshot	0	0	1	0	0	0	L0-ID reset	B-ID reset
CMD2 (reserved)	0	0	1	0	1	0	L0-ID reset	B-ID reset
CMD3 (reserved)	0	0	1	0	1	1	L0-ID reset	B-ID reset
CMD4 (reserved)	0	0	1	1	x	x	L0-ID reset	B-ID reset

Tabella 4.2: Elenco dei comandi *broadcast* usati nel sistema TTC.

In caso di blocco del microcontrollore è possibile eseguire un *reset* della scheda pilotando le linee di alimentazione della scheda ELMB; questa operazione è possibile grazie a due registri implementati all'interno del *chip* FPGA. Esistono due modi di eseguire il *reset* della scheda ELMB del modulo PDM: il primo grazie all'interfaccia I²C, il secondo utilizzando le due linee dedicate, "*PDM_reset*" e "*PDM_shutdown*", che, attraverso il *Back Plane*, connettono direttamente il modulo a tutte le schede *Service Board*. È così possibile eseguire le operazioni necessarie al ripristino accedendo ad una qualunque delle altre schede ELMB presenti sui nodi CAN-bus.

4.4 Primo sistema di test: TTCvi e TTCvx

Prima che fosse disponibile in laboratorio una scheda ODIN configurata e funzionante, è stato montato un banco di prova per il modulo PDM, simulando la presenza del *TTC-System* grazie ad una scheda di interfaccia USB-VME della Caen (la *V1718*) [65]. Questa scheda permette l'accesso al *bus* VME grazie all'interfaccia utente *software* fornita dalla casa madre (l'applicativo "*CAENVMEdemo*").

Come si vede in figura 4.17, tramite la connessione USB del computer, è possibile accedere alla scheda *V1718*, che permette l'accesso ai registri del *TTCvi*. Se correttamente configurato, il modulo *TTCvi* [66] è in grado di generare i segnali *broadcast* del sistema TTC, in modo da simulare l'esistenza dell'intero apparato. Il compito di trasformare il segnale elettrico in ottico è affidato al modulo *TTCvx* [67] che riceve i segnali generati dal *TTCvi* grazie a due connettori "lemo" posti sul pannello frontale. Dal modulo *TTCvx*, tramite una fibra ottica, il segnale giunge al modulo *TTCrq* della scheda PDM, esattamente come accadrà al *pozzo 8* del CERN, una volta montato l'esperimento.

Il modulo *TTCvi* permette di codificare sui due connettori lemo *channel_A* e *channel_B* il valore caricato all'interno di quattro FIFO secondo due modalità principali: è possibile caricare la FIFO e poi far uscire il dato corrispondente ad un determinato istante, oppure è possibile far uscire il dato non appena scritto nella memoria FIFO. Questa seconda modalità è quella che è stata adoperata in laboratorio.

Con questo primo sistema sono state testate le linee di propagazione dei vari segnali ed il giusto funzionamento della logica implementata sul modulo PDM; mentre si procedeva con questi primi test, è stata configurata e messa in opera, una scheda ODIN con relativa

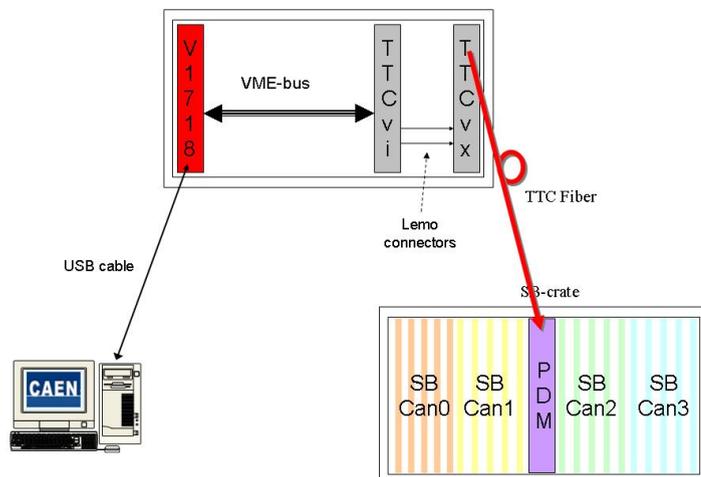


Figura 4.17: Schema dei collegamenti del banco di test: tramite l'applicativo “CAENV-MEdemo” è possibile interfacciare il computer al modulo *V1718*; tramite il *bus* VME questi è in grado di pilotare il modulo *TTCvi*; i segnali generati dal *TTCvi* sono codificati sulla fibra ottica dal *TTCvx* e ricevuti dal *TTCrq*, che provvede a decodificarli e renderli disponibili al modulo *PDM*.

interfaccia *software* e piena possibilità di funzionamento. In tutte le misure sui ritardi e sui *jitter* temporali del segnale “*BC_Pulse*”, l'impulso è stato generato grazie all'invio del segnale di test dell'ODIN (si veda il capitolo 6).

4.5 Secondo sistema di test: ODIN

Sarà la scheda ODIN, cuore del sistema TFC, a generare ed inviare a tutta l'elettronica del rivelatore di muoni i segnali *broadcast* di calibrazione e di test.

In laboratorio è stata installata una scheda ODIN ed un computer con la relativa interfaccia utente in PVSS, da adoperarsi per la sua gestione.

La connessione col computer avviene tramite interfaccia LAN e, una volta lanciato il sistema PVSS, è possibile accedere alle funzionalità della scheda ODIN.

I segnali generati dalla scheda ODIN, vengono inviati tramite due canali in logica *ecL* ad un modulo *TTCvi* che provvede a codificare le informazioni ricevute e ritrasmetterle su fibra ottica. Tramite fibra, le informazioni raggiungono i *crate* e vengono ricevute dalle varie schede elettroniche di destinazione.

Nello specifico, le schede ODIN e *TTCvi*, sono state collegate tramite due connettori “lemo” da 6.0ns: rispettivamente, uno per il *channel_A* ed uno per il *channel_B*. Il modulo *TTCvi* è stato collegato al modulo *TTCrq* della scheda *PDM* tramite una fibra ottica da 4m. In figura 4.18 è mostrato uno schema delle connessioni utilizzate in laboratorio.

Inizialmente l'impulsaggio dalla scheda ODIN è stato effettuato manualmente, facendo generare il corretto segnale di calibrazione (*CALIBRATION_A*) alla pressione di un pulsante dell'interfaccia utente; in questa modalità si è testato il giusto funzionamento della

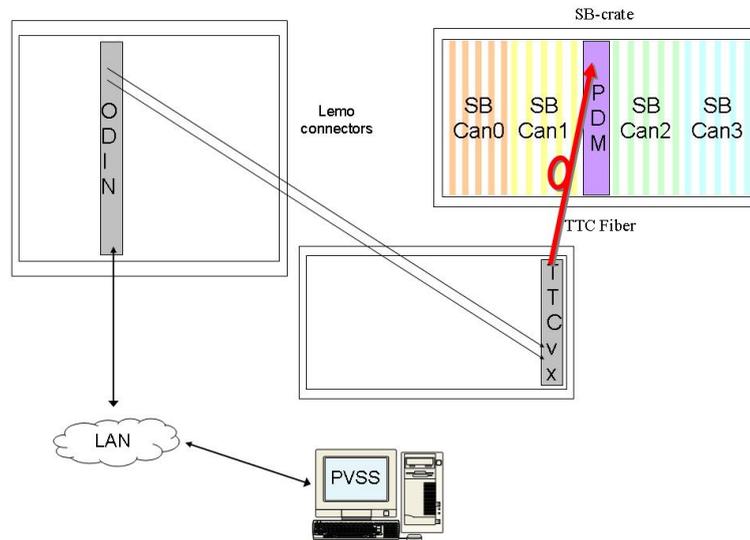


Figura 4.18: Schema dei collegamenti tra la scheda ODIN e l'SB-crate, nei test svolti in laboratorio. Il sistema PVSS sviluppato per ODIN, vi si interfaccia tramite rete LAN; i segnali generati dalla scheda ODIN sono codificati sulla fibra ottica dal TTCvx e ricevuti dal TTCrq, che provvede a decodificarli e renderli disponibili al modulo PDM.

logica, sia del modulo PDM che delle schede SB, potendo vedere la risposta al comando di calibrazione in uscita dall'SB-crate verso le schede di FE.

Per le misure di tempo, dovendo collezionare un numero quanto più grande possibile di valori, è stata sfruttata la possibilità di simulare il funzionamento di LHC nella scheda ODIN e, quindi, di impulsare continuamente ad una determinata frequenza (si veda 6.2.2 per un maggiore dettaglio).

Capitolo 5

Il PVSS (“*Prozess Visualisierungs- und Steuerungs-System II*”)

La realizzazione di un sistema telematico, che consenta in remoto il controllo degli apparati ed il monitoraggio delle grandezze da essi misurati, tramite calcolatore o rete di calcolatori, richiede l’impiego di alcuni *software*.

Come visto nel capitolo 3, la comunicazione tra le varie schede di elettronica presenti nelle camere a muoni di LHCb è basata sulla scheda ELMB (*Embedded Local Monitor Board*) (paragrafi 3.3.1 e 4.2.1) e su di un *bus* di tipo CAN (*Controller Area Network*) (appendice B). La funzione primaria che i *software* d’interfaccia devono svolgere è consentire la comunicazione bidirezionale con una o più schede ELMB. In termini *software*, la comunicazione tra calcolatori e schede di elettronica, viene gestita dal *driver* del modulo di interfaccia USB-CAN *SysTec*, utilizzato per il collegamento del calcolatore al *bus*. Non sarebbe pratico il dialogo con le schede ELMB direttamente nel loro *standard* di comunicazione, costituito dai messaggi sul *bus* in formato esadecimale. A risolvere questo problema interviene l’*OPC Server* (si veda l’appendice D) che traduce i messaggi standard del CAN in una sorta di variabili di facile lettura e che permette una facile gestione degli oggetti implementati nel *firmware* dell’ELMB. Inoltre, esso ha delle funzioni molto utili per l’eventuale distribuzione dei dati provenienti dalle schede ELMB in una rete di calcolatori. Per completare il sistema e renderlo “*user friendly*” si è scelto PVSS II (paragrafo 5.1) e le sue estensioni FRAMEWORK e FWELMB (paragrafo 5.2.1). Con questi *software* si ha la possibilità di comunicare con le schede ELMB e archiviare e gestire i dati attraverso dei pannelli interattivi, rendendo in tal modo il sistema versatile e di facile utilizzo.

I *software* impiegati costituiscono una sorta di struttura gerarchica, schematizzata nella figura 5.1, in cui ognuno ha bisogno dei sottostanti per poter operare correttamente. Infatti, l’*OPC Server* necessita del *driver* di interfaccia per poter comunicare attraverso il *bus*, il PVSS opera come *client*, utilizzando per l’acquisizione dei dati l’*OPC Server* come tramite, e i *software* rimanenti non sono altro che dei complementi al PVSS stesso.

5.1 Generalità e funzioni del programma PVSS

PVSS II (“*Prozess Visualisierungs- und Steuerungs-System II*”) è un pacchetto *software* per “il controllo dell’automazione ingegneristica”. È usato per connettere dispositivi *hardware* o *software*, acquisire i dati che essi producono e usare tali dati per la loro supervisione: quello che viene chiamato un sistema *SCADA* (“*Supervisory Control And Data Acquisition*”) [68].

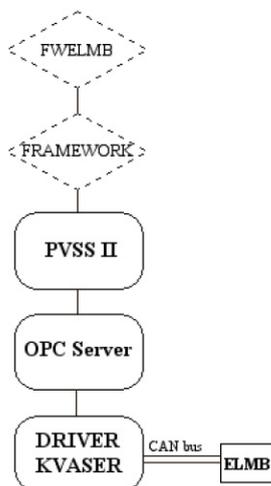


Figura 5.1: Struttura gerarchica del *software* impiegato; si noti la centralità del programma PVSS II.

L'architettura di questo programma è distribuita. Esso è costituito da più processi denominati *manager*, ciascuno con delle specifiche funzionalità, che possono essere lanciati indipendentemente. È possibile, tramite un pannello di controllo, decidere quanti e quali processi eseguire e con quali opzioni. Le istanze di PVSS vengono chiamate *progetti*. Ad ognuna di esse può corrispondere un diverso insieme di *manager* da lanciare, un *database*, un insieme di pannelli *Human Interface*, che è possibile predefinire. Nella figura 5.2 è visualizzata la struttura dei moduli del *software*.

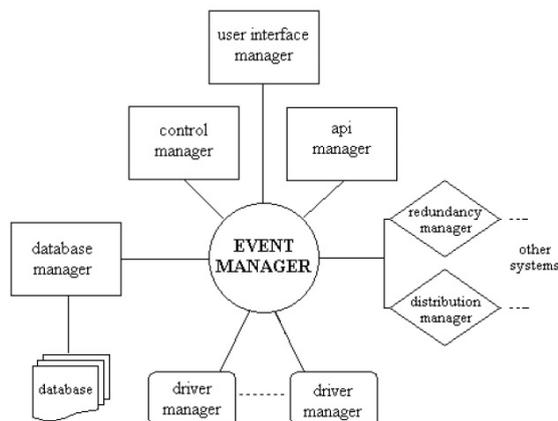


Figura 5.2: Schema logico della struttura del programma PVSS II. Sono indicati i vari moduli che ogni progetto PVSS può utilizzare.

L'insieme dei processi lanciati, connessi tra di loro, viene denominato *PVSS system*. Diversi *PVSS system*, implementati su diversi calcolatori, possono essere collegati tra di loro tramite i *distribution managers* utilizzando un protocollo basato sul TCP/IP. In tal modo il *software* può essere impiegato all'interno di una rete di calcolatori, permettendo la gestione dei dati e la loro distribuzione differenziata sui PC collegati.

Esistono anche dei *redundancy manager* grazie ai quali è possibile collegare l'interfaccia

utente a due *PVSS system*. Per cui se uno dovesse malfunzionare si ha la possibilità in poche decine di secondi di passare all'altro, mantenendo dati e configurazioni preesistenti. Il PVSS opera su piattaforme *Windows NT* e *Linux* o anche su reti miste. Un esempio è riportato in figura 5.3

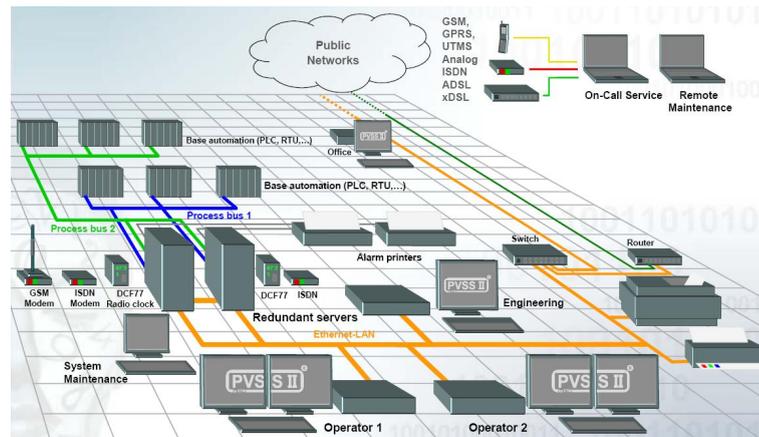


Figura 5.3: Il programma PVSS è in grado di gestire sistemi complessi, dotati di stazione di controllo e sviluppo anche connessi in remoto secondo qualsivoglia tecnologia.

E' possibile definire degli intervalli di variabilità per un certo elemento, in modo che siano generate delle condizioni di allarme quando il valore oscilla al di fuori degli stessi. Gli allarmi possono essere opzionalmente visualizzati e/o archiviati indipendentemente dai dati, garantendo una gestione appropriata del sistema.

In quest'ottica, il programma PVSS II si propone come *"supervisory software"* per il centro di controllo dell'apparato di automazione di un qualsivoglia processo complesso, purchè dotato di un'interfaccia di calcolatori elettronici per il proprio controllo e per la gestione. (figura 5.4)

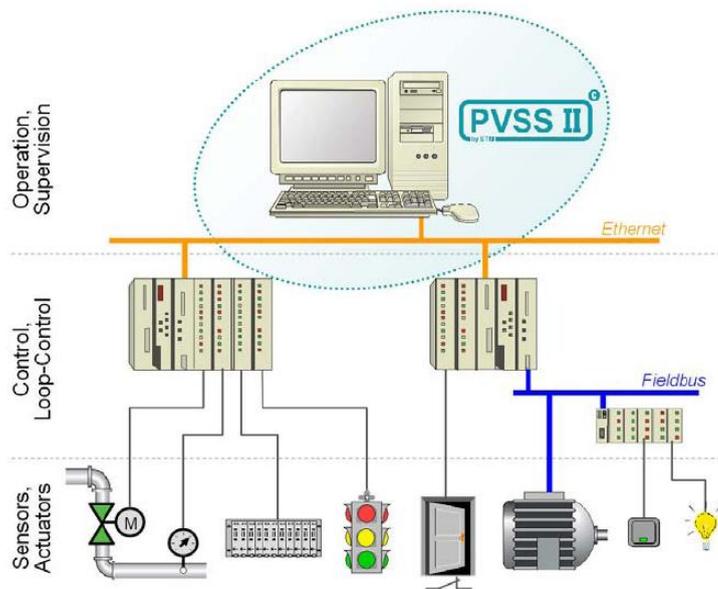


Figura 5.4: Ruolo del PVSS in un sistema automatizzato.

Di seguito sono indicate le principali potenzialità degli applicativi del pacchetto PVSS:

- Ogni dato proveniente dalla sensoristica telemetrica, viene registrato in tempo reale in un *database* dinamico; questo *database* è accessibile da tutte le altre applicazioni, per effettuare la visualizzazione a schermo, l’invio remoto ad altre stazioni o per essere processate in altro modo;
- I dati immagazzinati nel *database* possono essere memorizzati sul calcolatore per poter essere richiamati in un secondo momento dall’interfaccia utente o da qualsiasi altro programma;
- È possibile generare allarmi a seconda dei valori dei dati provenienti dalla sensoristica. Inoltre, gli allarmi sono memorizzati in un apposito *database* e possono essere richiamati successivamente sia per la visualizzazione che per operazioni di statistica;
- L’interfaccia con l’operatore viene resa possibile dagli *User Interface manager* i quali consentono di operare tramite pannelli completamente editabili dall’utente con degli elementi grafici. Come avviene nei *visual language*, un evento, come un *click* del mouse, può essere associato ad una funzione, rendendo così il pannello interattivo. Esistono elementi grafici predefiniti rappresentativi dei dispositivi reali: interruttori, valvole, cisterne e altri è possibile crearne all’occorrenza. L’interfaccia grafica risulta così molto pratica e intuitiva.
- Il programma PVSS permette la scrittura e l’esecuzione di *script* secondo la sintassi del linguaggio C con l’aggiunta di specifici parametri propri dei sistemi SCADA; tramite il *control manager* è possibile eseguire degli *script* che possono operare in *background* contemporaneamente ai pannelli grafici; tramite gli *API manager* è consentito, invece, a *software* indipendenti, realizzati dall’utente in linguaggio C, accedere alle risorse del *database* di PVSS.
- È disponibile un *tool* di parametrizzazione grafica (*PARA*), tramite cui è possibile configurare tutte le variabili del progetto PVSS come:

- Definire la struttura dei *database*;
 - Definire quali dati debbano essere archiviati;
 - Definire quali dati e in quali condizioni, debbano generare allarmi;
 - etc...
- È possibile interfacciare il programma PVSS con l'*hardware* tramite una nutrita libreria di *drivers* come: OPC, ProfiBus, CanBus, Modbus, TCP/IP e Applicom.

Nel progetto descritto in questa tesi il PVSS è stato utilizzato come OPC Client, in modo da ottenere la comunicazione con schede ELMB. La funzione di OPC Client è contenuta in uno dei *driver manager* di PVSS. Per il corretto funzionamento del sistema bisogna configurare una serie di *DataPoint* in modo da associarli agli oggetti di OPC e quindi alle informazioni provenienti dalla scheda ELMB.

5.2 Il programma PVSS al Cern

Nell'ottica di LHC, il programma PVSS permette sia un'interfaccia unica tra le varie parti dell'esperimento, sia la possibilità di intervenire con una conoscenza base dell'elettronica e non dovendo avere necessariamente un conoscenza approfondita di tutte le varie schede e parti dell'esperimento stesso.

Si consideri l'esperimento LHCb: la complessità delle schede elettroniche, sia per il numero di funzioni svolte dalla singola scheda, che per le diverse tipologie presenti, fanno sì che sia impossibile avere un operatore che conosca nel particolare i settaggi di tutti i vari apparati e possa, all'occorrenza, agire in tempi brevi sul settaggio dell'esperimento [70, 72, 71].

Se inoltre si considera il fatto che LHCb è una collaborazione internazionale e che quindi i vari gruppi che hanno lavorato al disegno ed alla programmazione delle varie schede, provengono da scuole molto differenti e posseggono metodologie di sviluppo differenti, il quadro si complica ancora.

5.2.1 FRAMEWORK e FWELMB

Il pacchetto *software* PVSS viene frequentemente adoperato negli esperimenti condotti presso il CERN come *SCADA system*. Per questa ragione i ricercatori del CERN hanno realizzato FRAMEWORK [69], che è una estensione al PVSS. Questa contiene file di configurazione, *script*, pannelli *user interface* e tutto il necessario per ottenere un *PVSS system* completo, subito in grado di operare con un insieme di dispositivi fisici. Questi dispositivi sono stati scelti tra quelli più frequentemente adoperati durante le sperimentazioni condotte al CERN.

FWELMB costituisce un ulteriore complemento di FRAMEWORK che, in particolare, contiene gli elementi necessari per gestire schede ELMB tramite il PVSS. Una volta installati i due *software*, si ha a disposizione una configurazione precostituita del sistema e una serie di pannelli di interfaccia.

Esistono due categorie di pannelli. I pannelli *editor* consentono di completare la configurazione del sistema. I pannelli *navigation* permettono invece di visualizzare i dati provenienti da schede ELMB e inviare messaggi di comando e controllo sul *bus*. Per il corretto funzionamento del sistema è importante che sia impostata la giusta corrispondenza tra i *DataPoint* e gli *item* di OPC, definiti nel suo file di configurazione.

5.3 Configurazioni ed interfaccia per il *Pulse Distribution Module*: il progetto “PDM”

Per la gestione dell’“elettronica dei muoni”, si è scelto di installare il programma PVSS su calcolatori aventi *MS Windows XP* come sistema operativo e configurare un OPC Server (“*CANOpenOPC*”¹) come driver di interfaccia tra il programma PVSS e il *bus* CAN.

In questo modo i “*DataPoint*” del PVSS mappano gli “*OPC items*” che, a loro volta, sono in corrispondenza biunivoca con l’“*Object Dictionary*” del CANOpen²

5.3.1 L’OPC CANOpen Server

La prima operazione effettuata per il progetto “PDM” è stata la configurazione del *server OPC*; in vero questa è stata un’operazione abbastanza veloce e semplice, come spiegato in appendice D, dove è anche riportato il listato del file di configurazione dell’OPC Server.

5.3.2 Lo schema del progetto “PDM”

Il progetto “PDM” è stato un progetto transitorio, praticamente utilizzato solo in fase di *debug* della scheda e subito inserito ed adattato al progetto generale di gestione dell’ECS (si veda 5.4).

Nel progetto sono stati implementati quattro “*DataPoint type*” (figura 5.5):

PDM che contiene due solo *DataPoint* con informazioni sulla versione dell’ELMB;

PDM_ACTEL dove sono implementati tutti i registri e le funzionalità dell’ACTEL;

PDM_PCF da cui è possibile gestire i due registri PCF;

PDM_TTCrx in cui sono inseriti tutti i registri del TTC.

La creazione dei *DataPoint* e la loro corrispondenza con gli *opc-item* è stata fatta “a mano”, creandoli e collegandoli uno per uno; questo perchè, essendo in numero limitato (circa una cinquantina) e non conoscendo ancora l’esatto standard per i nomi, non ho ritenuto necessario l’implementazione di un automatismo per la creazione e la corrispondenza di tali elementi.

I pannelli grafici (per la cui utilizzazione si rimanda al paragrafo 5.3.3) sono stati creati inserendo direttamente negli stessi il codice necessario a leggere e scrivere i *DataPoint*, rendendo leggero e di facile analisi il codice degli stessi.

5.3.3 L’interfaccia utente del progetto “PDM”

A causa della posizione chiave del modulo PDM nell’ambito dell’SB-System, non tutte le funzioni implementate nel modulo sono adatte ad una gestione ordinaria e non esperta della sua interfaccia. Per questo motivo, sono state sviluppate due distinte categorie di pannelli grafici: una per l’operatore incaricato della normale gestione del sistema ed una per utente “esperto”, con pieno controllo della scheda (figura 5.6).

L’amministratore ordinario ha a disposizione due pannelli: il “*PULSE*” ed il “*CRATE*”. Come già discusso nel paragrafo 4.3.2 (a cui si rimanda per una trattazione delle diverse

¹In appendice D è riportato il listato del file di configurazione dell’OPC Server.

²In appendice C è riportato l’OD per il modulo PDM.

5.3. CONFIGURAZIONI ED INTERFACCIA PER IL PULSE DISTRIBUTION MODULE: IL PROGETTO



Figura 5.5: I DataPoint type del progetto PDM

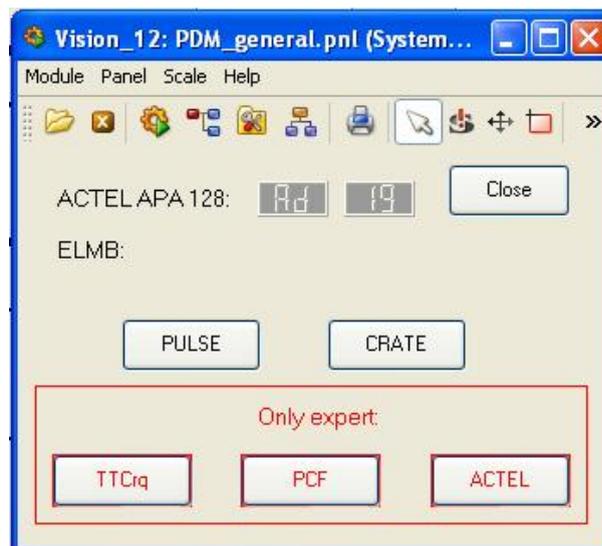


Figura 5.6: Il pannello di accesso per la gestione del modulo PDM: è possibile raggiungere due categorie di pannelli, una per l'amministrazione ordinaria e l'altra per una gestione "esperta" della scheda.

sorgenti del segnale), è possibile settare il *chip* ACTEL del modulo PDM per selezionare quattro diverse sorgenti del segnale *BC_Pulse*; la gestione di questa operazione è svolta attraverso il pannello "PULSE" (figura 5.7). Come prima cosa, è presente un visualizzatore dello stato del TTCrq, verde se il TTC funziona correttamente, lampeggiante rosso se è necessario un *reset* del modulo. Tramite un menù a discesa, è possibile selezionare le quattro differenti sorgenti per il segnale *BC_Pulse*:

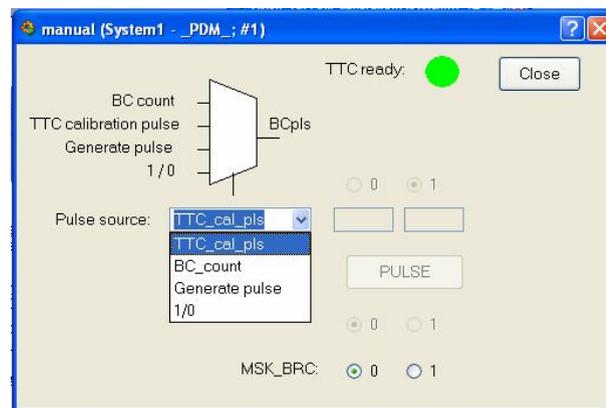


Figura 5.7: Il pannello “PULSE” tramite cui l’utente ordinario può settare la sorgente del segnale *BC_Pulse*.

TTC_cal_pls : seleziona come sorgente il segnale *broadcast* proveniente dal TTC. Questa è la modalità di *default* con cui lavora il modulo PDM; con questa selezione, si abilita inoltre la possibilità di scegliere se mascherare il segnale (e quindi non generare nessun impulso) o meno.

BC_count : Genera un impulso, sincrono col *clock* macchina di LHC, abilitato da un comparatore ogni volta che il valore del “*BX-identifier*” dell’evento che interessa il rivelatore in quel momento diventa uguale al valore scritto nell’apposito registro (“*Cmp_pdm*”). Con questa selezione si abilita un campo dove inserire il valore del registro comparatore;

Generate Pulse : Si abilita un pulsante alla cui pressione viene generato l’impulso dalla scheda;

1/0 : la linea del *BC_Pulse* viene settata ad un valore logico fisso.

Il secondo pannello per la gestione ordinaria (figura 5.8),permette la gestione della manutenzione dell’SB-crate.

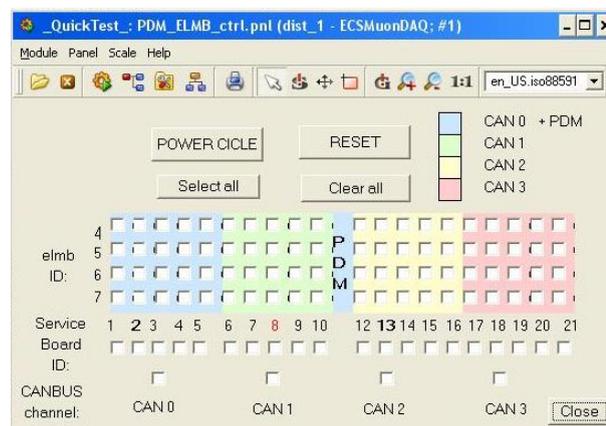


Figura 5.8: Il pannello “CRATE” tramite cui l’utente ordinario può eseguire un “reset” o un “power cycle” delle schede ELMB del *crate*; è possibile selezionare una sola scheda, le schede di un’intero modulo SB, di un intero canale CAN o di tutto il *crate*.

5.3. CONFIGURAZIONI ED INTERFACCIA PER IL PULSE DISTRIBUTION MODULE: IL PROGETTO

Tramite questo pannello è possibile eseguire il “reset” o il “power cycle” delle schede ELMB (*Embedded Local Monitor Board*) presenti nel crate.

Nel caso di *reset* o *power cycle* su più schede contemporaneamente, saranno inviati più comandi sequenziali, partendo dalla scheda ELMB con indirizzo minore (dall’alto al basso e da sinistra a destra); nel caso non sia presente una scheda nel *crate*, il corrispondente *box* non risulterà abilitato.

È possibile selezionare singole schede del *crate*, anche distribuite in punti diversi del *crate*, tramite il *box* “elmb ID”; si possono facilmente settare tutte le schede presenti su di un modulo *Service Board* (“Service Board ID”) o quelle collegate ad uno stesso canale CAN (“CANBUS channel”); è infine presente un pulsante per selezionare tutte le schede ELMB presenti nel *crate*. Naturalmente non è possibile resettare o spegnere la scheda ELMB installata sul modulo PDM dal modulo stesso poichè si perderebbe la comunicazione e la gestione della scheda; all’occorrenza è possibile eseguire questa operazione da una qualunque delle schede *Service Board*.

L’utente esperto avrà accesso diretto ai registri dei vari *chip* presenti sul *Pulse Distribution Module* tramite tre pannelli distinti:

Il primo darà accesso, in lettura e scrittura, ai registri del modulo TTCrq (pannello *TTCrx*, in figura 5.9), il secondo ai due registri PCF del modulo PDM (pannello *PCF*, in figura 5.10) e l’ultimo consentirà il settaggio dei registri del *chip* FPGA (pannello *ACTEL* in figura 5.11).

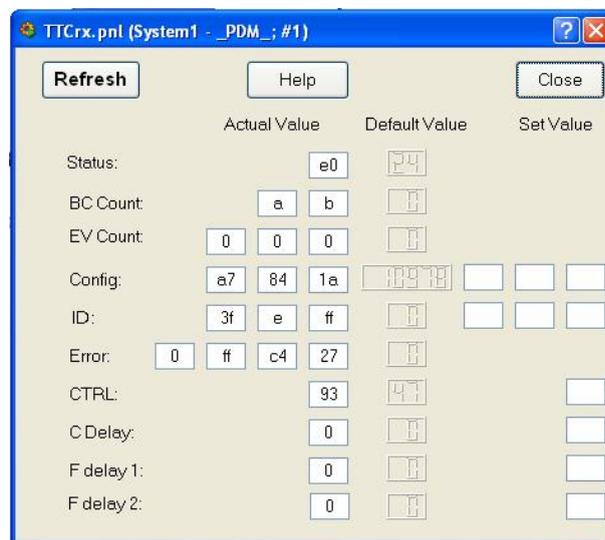


Figura 5.9: Il pannello “*TTCrx*” tramite cui l’utente esperto ha accesso completo ai registri del modulo TTCrq.

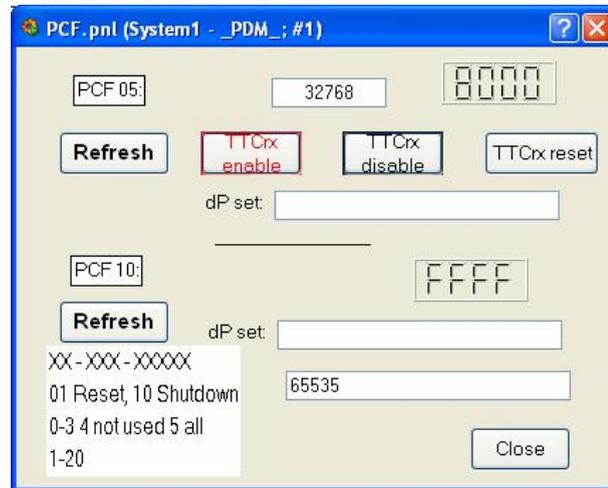


Figura 5.10: Il pannello “*PCF*” tramite cui l’utente esperto può settare il valore dei due registri PCF del modulo PDM; tramite il primo si abilita o meno l’alimentazione del TTCrq, il secondo è usato per la manutenzione delle schede ELMB presenti nel *crate*.

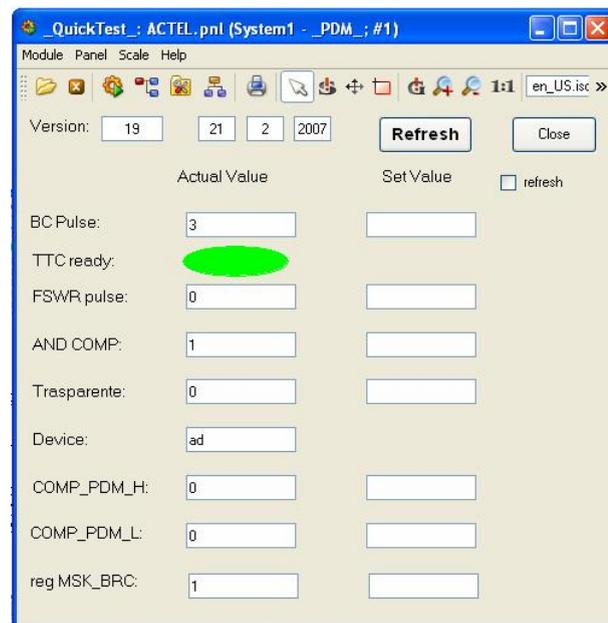


Figura 5.11: Il pannello “*ACTEL*” tramite cui l’utente esperto può accedere ai registri implementati nella logica del *chip* FPGA del modulo PDM.

5.4 Il progetto finale per l'ECS

Al momento dell'avvio dell'esperimento dovrà essere operativo un unico sistema PVSS in grado di gestire tutto l'apparato di LHCb. Perchè ciò sia possibile è necessario che i vari "sottosistemi" rispondano a determinate specifiche ed automatismi comuni.

Il primo passo verso questo obiettivo è stato l'inserimento del progetto di interfaccia per il modulo PDM all'interno del progetto di gestione dell'intero ECS.

Le principali modifiche apportate, hanno interessato l'introduzione di automatismi nella generazione del sistema PVSS; in particolare:

- Sono stati automatizzati i processi:
 - di creazione del file *opcconfig*;
 - di creazione dei *DataPoint type*;
 - di creazione dei *DataPoint*;
 - di corrispondenza tra *DataPoint* e *opc-items*.
- È stata modificata la struttura dei pannelli, introducendo l'utilizzo di *script* per il richiamo delle funzioni;
- Si sono integrati i pannelli nel progetto globale.

Automazione dei processi di generazione del sistema PVSS

Come indicato nel capitolo 3, saranno presenti diversi calcolatori adibiti al controllo dell'elettronica dell'esperimento; ognuno di questi sarà collegato ad una parte diversa dell'esperimento ed ad un numero diverso di schede elettroniche. Per ovviare a questa disomogeneità, è stata implementata una procedura di generazione automatica dei *PVSS system* installati sui singoli calcolatori; questa procedura si basa su un file nel quale sono indicate tutte le connessioni dei vari canali CAN. In questo modo, basandosi sull'indirizzo del canale CAN al quale ogni singolo calcolatore è connesso, è possibile risalire a tutta l'elettronica collegata e, di conseguenza, generare la logica necessaria a supervisionare solo quella parte di elettronica e non tutto l'esperimento.

La prima operazione da compiere, sarà la generazione del file necessario all'OPC Server per la gestione dell'*hardware*.

Il secondo passo è relativo alla configurazione del programma PVSS II: la creazione dei *DataPoint type* e dei *DataPoint*. Si è deciso di creare un tipo di *DataPoint* apposito per il PDM, per poter avere una distinzione logica della gestione del modulo rispetto al resto dell'elettronica (ogni scheda avrà i suoi *DataPoint type*). A differenza del progetto "PDM", dove per i nomi dei *DataPoint* si era privilegiato l'aspetto pratico della lettura, si è utilizzata la notazione ufficiale dettata dagli standard dell'esperimento: anche se questo rende macchinosa la lettura del nome del *DataPoint*, in esso sono contenute tutte le informazioni necessarie ad identificare di cosa si stia parlando e del relativo componente cui esso si riferisce in maniera univoca per tutto LHCb.

L'ultimo passo del processo di automazione è l'assegnazione della corrispondenza biunivoca tra *OPC-item* e *PVSS-DataPoint*.

Modifica dei pannelli

Il nome dei vari *DataPoint* contiene tutte le informazioni necessarie ad individuare univocamente il componente cui esso si riferisce; quindi, per fare un esempio, il *DataPoint*

che permette la scrittura sul registro PCF del modulo PDM installato nel *crate* collegato alla stazione *M4M5*, si chiama *MUONC_DAQ_Q4_M45C_PDM.PCF.PCF_05*, mentre quello relativo allo stesso componente, ma nella stazione *M2M3* (entrambi dal lato Criogenia) si chiama *MUONC_DAQ_Q4_M23C_PDM.PCF.PCF_05*; come si vede nel nome cambia l’informazione della posizione in cui si trovano i due registri.

È necessario che i vari pannelli siano in grado di recepire i nomi dei *DataPoint* inserendovi un parametro che li possa distinguere; questo perchè, nella gerarchia delle macchine preposte al controllo, l’operatore dovrà poter accedere a tutti i componenti dell’elettronica dell’esperimento dalla stessa postazione. Questa caratteristica è ottenuta grazie al parametro *\$*; inserito nel nome del *DataPoint*, permette di essere di essere sostituito con quanto indicato nell’apposito codice. È necessario che il pannello superiore, passi al pannello figlio il valore del parametro, altrimenti il codice non funzionerà. In figura 5.12 è possibile vedere l’utilizzo di questo parametro per uno dei pannelli.

Script in parent-panel

```
string baseDP;
sprintf(baseDP, "MUONC_DAQ_Q4_M45C_PDM");
ChildPanelOnCentral("PDM_general.pnl",
    "",
    makeDynString("$2:" + baseDP));
```

Script in children-panel

```
sDp = dpSubStr($2-"PCF.PCF_05: original.. value", DPSUB_SYS_DP_EL);
```

Figura 5.12: Esempio di utilizzo del parametro *\$* nel “pannello padre”(figura superiore) e nel “pannello figlio” (figura inferiore). Grazie a questa sintassi, è possibile passare un parametro da un pannello padre ad uno figlio e generare il nome dei *DataPoint* in maniera standardizzata per tutti gli apparati dell’esperimento.

La seconda modifica alla struttura dei pannelli riguarda la modalità di esecuzione del codice in essi contenuto: le specifiche fornite per l’esperimento vietano l’accesso diretto ai *DataPoint* da parte dei pannelli e richiedono l’utilizzo di funzioni specifiche per fare ciò. Questa direttiva è motivata dall’esigenza di vietare la possibilità di modificare il codice di un determinato sistema PVSS accedendo, da un altro sistema, direttamente alla struttura dei *DataPoint*. A differenza del progetto “PVSS”, in cui il codice per l’accesso ai *DataPoint* era scritto direttamente nei pannelli, si è quindi provveduto a modificare la struttura, con l’inserimento di apposite funzioni per l’accesso ai *DataPoint* e rispettando quindi le specifiche richieste.

Integrazione dei pannelli nel progetto finale

L’ultimo passo è stato quello dell’inserimento dei vari pannelli per la gestione del modulo PDM all’interno dei pannelli globali, inserendo i pulsanti per permetterne l’apertura e omogenizzando la grafica e lo stile.

Capitolo 6

L'allineamento temporale

In una logica sequenziale, un sistema sincrono è caratterizzato dalla presenza di un segnale ad onda quadra, diffuso a tutti i componenti e a cui si riferiscono gli altri segnali: *il clock*. Questo significa che i componenti elettronici del sistema attenderanno il fronte di salita (o di discesa, a seconda della convenzione usata e delle specifiche) del segnale di *clock* per abilitare le proprie porte di ingresso e di uscita.

Tuttavia, il tempo di propagazione dei segnali è finito qualsiasi sia il mezzo entro cui essi transitano; questo fa sì che se i *“percorsi temporali”* non sono identici, l'informazione raggiunge i vari componenti del sistema elettronico in tempi diversi. Se i segnali elettrici saranno soggetti ad un *ritardo* differente l'uno dall'altro, il sistema perderà la qualità di essere sincrono; per evitare ciò, bisogna stimare i vari ritardi cui sono soggetti i diversi canali e compensarli, al fine di allineare temporalmente tutto il sistema.

Esistono fondamentalmente due tipi di ritardi intrinseci ai circuiti elettrici: il *“ritardo di propagazione”* ed il *“ritardo di elaborazione”*. Il primo è dovuto alla velocità con cui un segnale si propaga dal punto in cui viene generato a quello in cui viene utilizzato, comprendendo in questo sia il tempo impiegato per la propagazione all'interno della pista elettrica, dei cavi e dei connettori, sia il tempo necessario a transitare all'interno di componenti attivi; il secondo dipende dalle operazioni logiche che i vari componenti attivi eseguono per elaborare, ritrasmettere o generare un segnale.

All'accensione del rivelatore di muoni, i segnali generati dallo stesso evento (stesso *BX-id*), ma provenienti da diversi *canali fisici*, arriveranno al sistema di memorizzazione dati in tempi differenti [22]. Infatti, i cavi di connessione tra le varie schede dell'esperimento non hanno tutti la stessa lunghezza, nè le piste all'interno delle varie schede sono equalizzate; inoltre, segnali provenienti da stazioni diverse del rivelatore, vengono processati da un numero diverso di *chip*.

È quindi necessario provvedere ad una operazione di allineamento temporale del sistema, effettuando una stima dei diversi ritardi cui i segnali saranno sottoposti. Infatti, un passo indispensabile per l'elaborazione dei dati è quello di poter identificare l'evento: senza sapere a quale interazione corrispondano i dati memorizzati, sarebbe vana ogni analisi e ogni studio del fenomeno.

I contributi più significativi al ritardo dei vari segnali sono:

- il “tempo di volo” delle particelle;
- la distribuzione del tempo di risposta delle diverse camere del rivelatore di muoni;
- la lunghezza dei percorsi elettronici (cavi e piste);

- il numero degli stadi logici a cui i diversi segnali sono sottoposti.

In generale, come è facile intuire, i diversi canali risulteranno avere diversi tempi di ritardo, caratteristici da canale a canale.

Per poter assegnare il giusto *BX-id* ad ogni segnale è di fondamentale importanza che questi siano “allineati in tempo”.

Per poter eseguire un corretto “allineamento temporale” del rivelatore di muoni è necessario:

1. misurare il ritardo totale (Δt) per ogni *canale fisico* con una risoluzione dell'ordine del nanosecondo;
2. equalizzare i ritardi di tutti i *canali fisici* che compongono uno stesso *canale logico*;
3. sincronizzare ogni *canale logico* rispetto alla frequenza di *bunch crossing*, assegnandogli il giusto “*BX-id*”.

La strategia seguita consiste nel dividere il ritardo totale Δt che ogni segnale matura nel rivelatore di muoni in due parti:

il **Coarse Delay**, $\Delta t_c = INT(\frac{\Delta t}{25ns})$, che esprime il numero intero di colpi di *clock* di cui è composto l'intero ritardo;

il **Fine Delay**, $\Delta t_f = \Delta t - \Delta t_c$, che misura la fase del ritardo rispetto al *clock macchina* in passi da $1.56ns$.

Il compito di allineare i vari segnali è assegnato alle schede ODE (*Off-Detector Electronics*, paragrafo 3.3.2) dove, ai dati provenienti dai *canali logici*, viene assegnato il corrispondente “*BX-id*”.

Se dividiamo il ritardo che ogni segnale matura nel rivelatore di muoni in due parti, una dal punto di rilevazione fino ai *chip* CARIOCA, che generano i *canali logici*, e l'altra da qui fino all'elettronica di *trigger*, capiamo che la seconda porzione del ritardo è introdotta esclusivamente dalla struttura dell'elettronica ed è possibile valutarne i ritardi senza aver bisogno della fisica dei fasci di protoni.

Nella logica implementata sul modulo PDM (capitolo 4) sono state inserite particolari funzionalità in grado di velocizzare la procedura di sincronizzazione dell'elettronica del rivelatore. In figura 6.1 è riportato lo schema logico della procedura di allineamento temporale: è possibile lanciare una procedura di test che invia, al posto dei segnali provenienti dalle camere per i muoni, un opportuno *pattern* nei *canali logici* simulando il passaggio di una particella nel rivelatore di muoni. Contemporaneamente l'ODIN emula il funzionamento reale dell'apparato inviando il segnale di *trigger di livello zero* come se un evento fosse stato realmente rilevato al momento dell'invio del *pattern*.

Questo permetterà di abbreviare notevolmente il tempo necessario per la calibrazione del sistema, durante il funzionamento dell'anello d'accumulazione, giungendo in tempi più brevi alla fase d'acquisizione dati; e, inoltre, di monitorare l'andamento temporale dell'esperimento ed eventualmente ricalibrare l'apparato ai giusti valori.

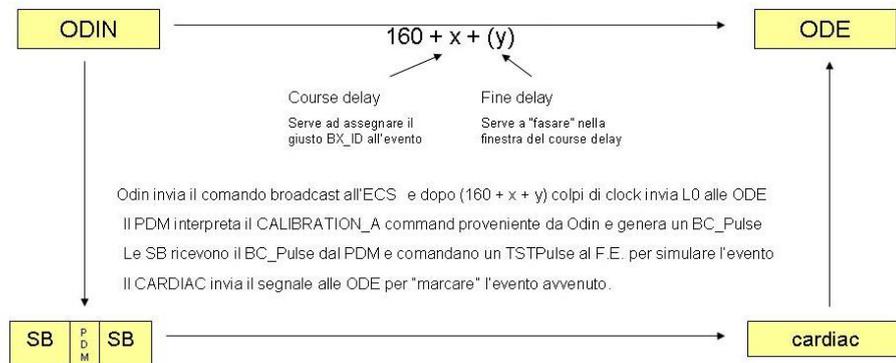


Figura 6.1: Schema logico della procedura di allineamento temporale in assenza di fisica.

6.1 il “Coarse Delay” ed il “Fine Delay”

In ogni scheda ODE è presente un *chip* appositamente progettato per effettuare misure di ritardo sui vari canali: il “*SYNC*”. Grazie ad un TDC appositamente realizzato, il *chip SYNC* è in grado di istogrammare i tempi di arrivo di ogni singolo segnale e di ricavare quindi, misurandone i valori, il “*Coarse Delay*” ed il “*Fine Delay*”. È così possibile utilizzare i ritardi programmabili dei *chip DIALOG* (paragrafo 3.1) per centrare l’istogramma del “*Coarse Delay*” all’interno dell’intervallo di $25ns$ tra un *bunch crossing* e l’altro e compensare così le differenze di ritardo dei vari canali. L’allineamento della fase è ottenuto all’interno degli stessi *SYNC*, compensando il “*Fine Delay*” dei vari *canali logici* e sincronizzando le fasi dei segnali che trasportano i dati con il segnale dell’orbita di LHC.

Una volta allineato e assegnato il giusto *BX-id*, il segnale è pronto per essere inviato all’unità decisionale dell’*L0-trigger* ed eventualmente, memorizzata nei moduli DAQ.

SYNC time measurements

Ogni *chip SYNC* riceve fino ad otto *canali logici*; ogni canale d’ingresso è collegato ad un TDC (“*Time to Digital Converter*”) costruito appositamente e basato su un DLL (“*Delay Locked Loop*”) il cui schema è riportato in figura 6.2.

Il *clock* di riferimento (*Ref Clock*) usato per la calibrazione del DLL è il *clock macchina* di LHC. Le 16 uscite del blocco VCDL (“*Voltage Controlled Delay Line*”) costituiscono 16 diversi segnali di *clock*, ognuno sfasato rispetto all’altro di un sedicesimo del periodo di *clock* principale ($\frac{25ns}{16} = 1.56ns$). Questi sedici *clock* sono collegati all’ingresso di altrettanti *flip flop D*. Il *canale logico* è connesso invece all’ingresso del *clock* di tutti i *flip flop*. Non appena giunge un segnale sul *canale logico*, i *flip flop D* forniscono un codice a 16 bit, il cui valore indica lo sfasamento del segnale rispetto al *clock macchina*. Questa informazione, viene codificata in una parola a 4 bit il cui valore fornisce il *fine delay* Δt_f del *canale logico*.

Fine Time Histogram

Analizzando un *canale logico* alla volta, è possibile registrare più valori del *Fine Delay* in un istogramma, chiamato *Fine Time Histogram*. In figura 6.3 è mostrato lo schema della logica per il *Fine Time Histogram*.

L’istogrammatore è basato su sedici contatori a 24bit. I 4bit in uscita dal TDC abilitato, vengono inviati, ad ogni fronte di salita del *clock macchina* ($25ns$), ad un *decodificatore di fase*. Se il canale è interessato da un dato, il contatore corrispondente al ritardo di fase

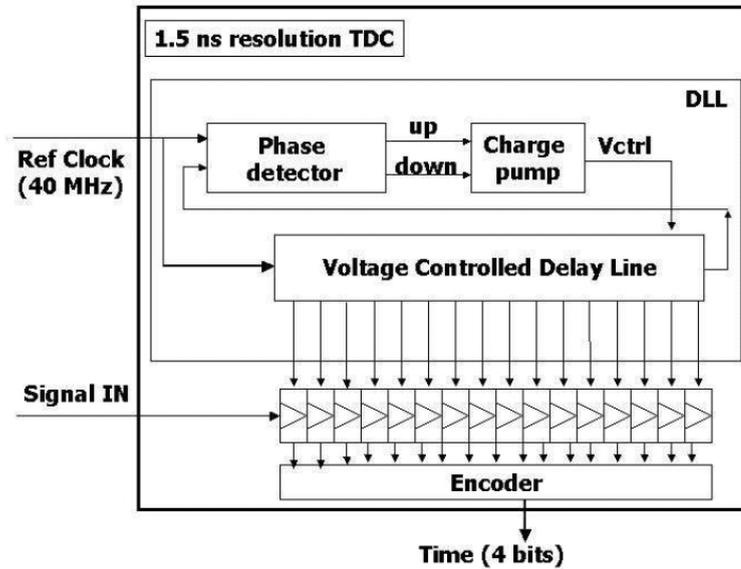


Figura 6.2: Schema della cella TDC implementata nel *chip* SYNC.

portato verrà incrementato di una unità nel colpo di *clock* successivo. Quando uno dei contatori a 24 bit è saturo, la procedura di istogrammazione viene interrotta e rimane in attesa di una lettura dei dati tramite l'interfaccia I²C. Prima che sia possibile effettuare una nuova misura, è necessario *resettare* l'istogrammatore. Selezionando un diverso *canale logico*, l'operazione di *reset* è automatica.

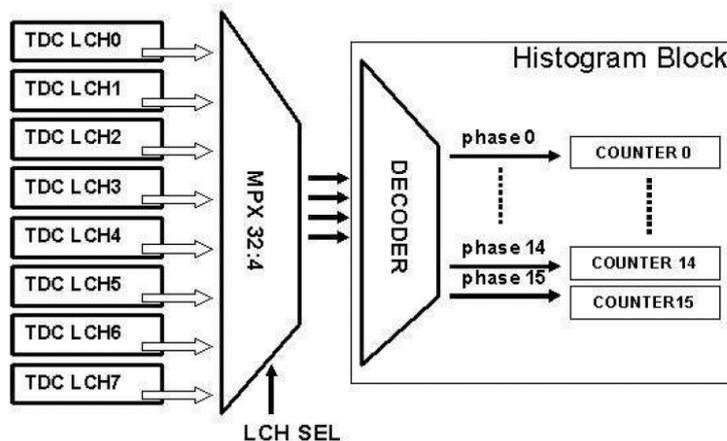


Figura 6.3: Schema dell'istogrammatore del *chip* SYNC per l'istogramma del *Fine Time Histogram*: il range temporale dell'istogramma è di 25ns e sarà diviso in 16 parti, una per contatore. La risoluzione dell'istogramma corrisponde così alla risoluzione del TDC: 1.56ns .

Il *Fine Time Histogram* rende possibile la ricostruzione della distribuzione temporale del *canale logico* selezionato, canale per canale. Se la distribuzione non è centrata nell'intervallo temporale dei 25ns , l'istogramma può essere usato per calcolare il giusto valore di Δt_f da usarsi nei *chip* DIALOG per allineare il *canale logico*.

Coarse Time Histogram

L'istogrammatore del *chip SYNC* può essere utilizzato anche per effettuare una misura del *Coarse Delay* Δt_c . Infatti, *canale logico* per canale, il TDC può essere usato per avere l'informazione se quel particolare canale è stato interessato da un evento.

Il “*Coarse Time Histogram*” è composto dividendo il periodo totale di $25ns$ in 16 parti, una corrispondente a ciascun contatore. L'istogramma acquisisce dati ogni $89.1\mu s$ per 16 fasci consecutivi. Il segnale di riferimento per l'acquisizione dei dati nell'istogrammatore è l'*orbit signal*, ossia il segnale che indica il passaggio di un *bunch* in LHC (ossia il quarto contatore verrà incrementato se il corrispondente *canale logico* viene interessato da un evento quattro cicli dopo lo *start* dell'orbita).

6.2 la procedura di test del PDM

Come già accennato, tramite la logica implementata nel *Pulse Distribution Module*, è possibile effettuare una procedura di *test* finalizzata alla stima del ritardo imputabile all'elettronica dell'ECS.

Come mostrato in figura 6.4, il modulo PDM riceve dalla scheda ODIN, tramite la fibra ottica connessa al TTCrq, un segnale *broadcast* di calibrazione (il *CALIBRATION_A*); il segnale viene “raccolto” dal *chip* FPGA che lo analizza e, decodificatolo, genera un impulso: il *BC_PULSE*. Questo viene inviato come segnale *broadcast* attraverso il *backplane* e raggiunge le schede *Service Board* dell'*SB-crate*. In ogni scheda SB, la locale FPGA analizza e decodifica il messaggio, generando a sua volta un nuovo segnale (il *TSTPLS*); quest'ultimo segnale viene inviato, tramite i dodici canali *long-P* in uscita dalle schede SB, all'elettronica di *Front-End*. Viene così avviato il processo di “auto-impulsaggio” dei *chip* DIALOG ed è possibile eseguire la misura dei canali in assenza della fisica dei fasci.

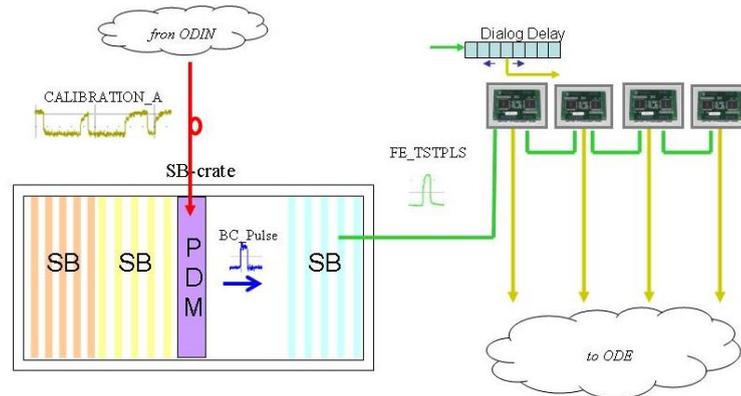


Figura 6.4: Schema logico della procedura di test per l'allineamento temporale dell'elettronica in assenza della fisica dei fasci.

6.2.1 le misure del jitter

Requisito fondamentale perchè sia effettivamente possibile utilizzare questa procedura è che il segnale *BC_pulse* non influisca in modo significativo sul *jitter* temporale del sistema. Per la verifica di questa caratteristica è stata effettuata l'analisi del *jitter* temporale che affligge la generazione del segnale emulato, studiando l'andamento dei ritardi tra l'invio del comando *broadcast* di calibrazione e la generazione del *pattern* nel *Front-End*.

Innanzitutto è stata eseguita la caratterizzazione della propagazione del segnale *BC_Pulse* lungo il *BackPlane* (BP), misurando il *tempo di propagazione* ed il relativo *jitter* del segnale dalla generazione da parte del *chip* FPGA del modulo PDM fino al connettore sul BP. In figura 6.5 è riportato il grafico relativo.

In accordo con quanto aspettato, il *jitter* temporale introdotto dal sistema *PDM-crate* è dell'ordine dei *100ps* e quindi è accettabile ai fini della risoluzione dell'esperimento.

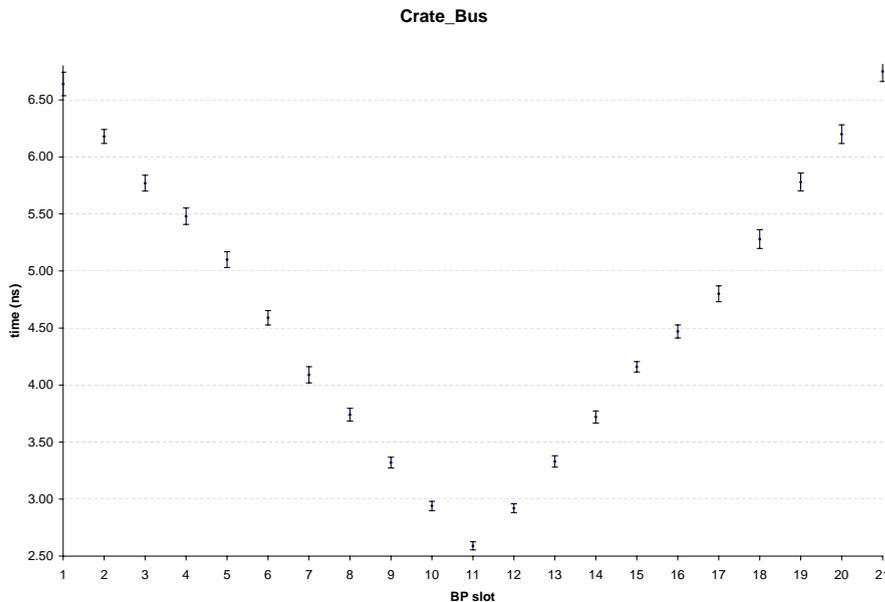


Figura 6.5: Tempi di propagazione del segnale *BC_Pulse* dalla sua generazione nel modulo PDM ai connettori del BP: il *jitter* temporale dei segnali è inferiore ai *100ps*.

Il passo successivo è stato quello inerente allo studio del *disturbo* introdotto dalle schede SB. Si è quindi esaminata la propagazione del comando di *test* dalla generazione del *BC_Pulse* sul modulo PDM, alla generazione del *TSTPLS* sulla scheda SB, fino all'invio del segnale sui canali *I²C*; il segnale di riferimento è stato preso sulla scheda del PDM, mentre quello di *stop* dopo un cavo lungo un metro.

Una delle informazioni cercate dallo studio di queste misure è la conferma del fatto che le schede SB forniscono prestazioni simili per quanto riguarda il ritardo ed il *jitter* introdotto; ossia il fatto che i vari *chip* della ditta ACTEL montate sulle schede SB non introducano un ritardo intrinseco, dovute a differenze strutturali di fabbricazione, superiore alla sensibilità richiesta (dell'ordine del nanosecondo).

Sono stati eseguiti due tipi di misure:

la distribuzione "in Y": ossia la misura dei segnali su tutti i canali per le schede *Service Board* inserite nelle posizioni *slot_01*, *slot_10*, *slot_12* e *slot_21*; la scelta delle posizioni delle schede per queste misure è stata fatta considerando le due più vicine e le due più lontane al modulo PDM, dai due lati del *crate*.

la distribuzione "in X": ossia la misura dei segnali su tutte le schede SB per i canali *CH_03*, *CH_07* e *CH_11*; la scelta dei canali è stata effettuata considerando quello con risposta più veloce e i due più lenti (in alcune schede i due canali risultano sovrapposti o scambiati in ordine).

Come si vede dalla figura 6.6, in cui a titolo d'esempio è riportato il grafico dei valori di un *set* di misure per la distribuzione "in Y", le schede *Service Board* presentano distribuzioni del ritardo dei segnali con uno stesso andamento, ad indicare che la programmazione dei *chip* FPGA montati sulle schede SB in esame fornisce una distribuzione del ritardo per canale simile per ogni scheda.

Le misure evidenziano un andamento quasi lineare per i ch_00-ch_07 per poi presentare un andamento monotono decrescente nei restanti canali. Il *jitter* temporale dei segnali risulta dell'ordine del centinaio di *ps*.

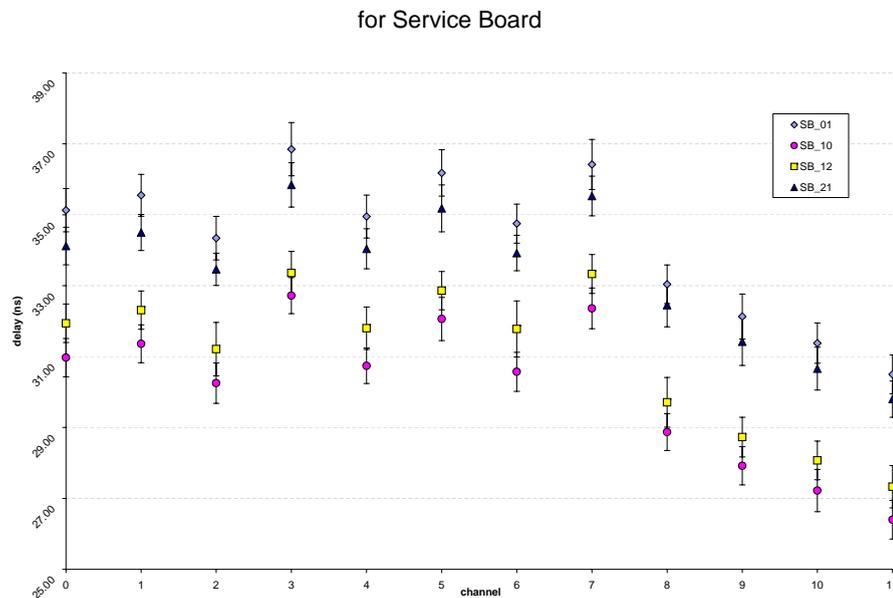


Figura 6.6: Tempi di propagazione del segnale *BC_Pulse-TSTPLS* dalla generazione nel modulo PDM fino ad un metro di cavo dopo le schede SB. Distribuzione "in Y" per SB. Il *jitter* temporale dei segnali è dell'ordine dei 100ps.

Il secondo tipo di misure, di cui si riporta un grafico d'esempio in figura 6.7, è stato effettuato facendo "scorrere" la stessa scheda in tutte le posizioni del *crate* ed eseguendo la misura sui tre canali prima indicati. La distribuzione ottenuta è in linea con la distribuzione trovata per il BP e quindi conferma la stabilità del ritardo introdotto dai *chip* della ACTEL montati sulle schede.

Tuttavia, questo *set* di misure evidenzia una non completa simmetria tra le due metà dell'*SB-crate*: mentre la parte "destra" (SB_12-SB_21) presenta un andamento quasi monotono, la parte "sinistra" (SB_01-SB_11) presenta un andamento oscillante per le schede SB_01-SB_04, per poi avere un andamento monotono nelle restanti schede.

Un ultimo tipo di misure è stato effettuato per avere conferma dell'"intercambiabilità" delle schede *Service Board*; si sono misurati tutti i canali della scheda SB, nello stesso *slot*, ma variando ogni volta la scheda. Queste misure, di cui si riporta un esempio in figura 6.8, hanno evidenziato che le varie schede hanno una distribuzione dei ritardi simile l'una dall'altra, con uno scostamento massimo inferiore ai 2ns tra la più reattiva e la più lenta, in accordo con la risoluzione richiesta. Inoltre, il *jitter* temporale introdotto sulla generazione

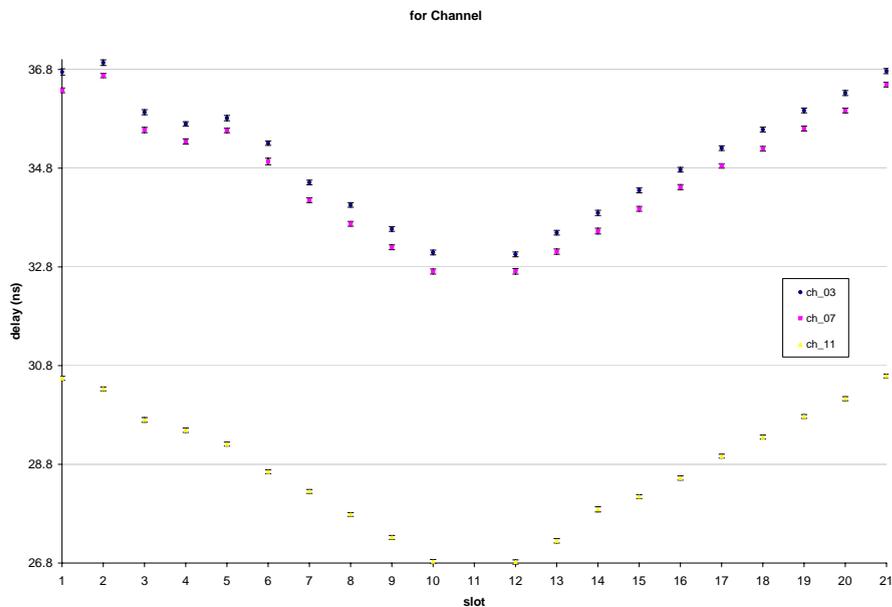


Figura 6.7: Tempi di propagazione del segnale *BC_Pulse-TSTPLS* dalla generazione nel modulo PDM fino ad un metro di cavo dopo le schede SB. Distribuzione “in X” per canale.

e trasmissione del segnale dall'*SB-System* è sempre risultato dell'ordine dei $100ps$, in pieno accordo con le specifiche.

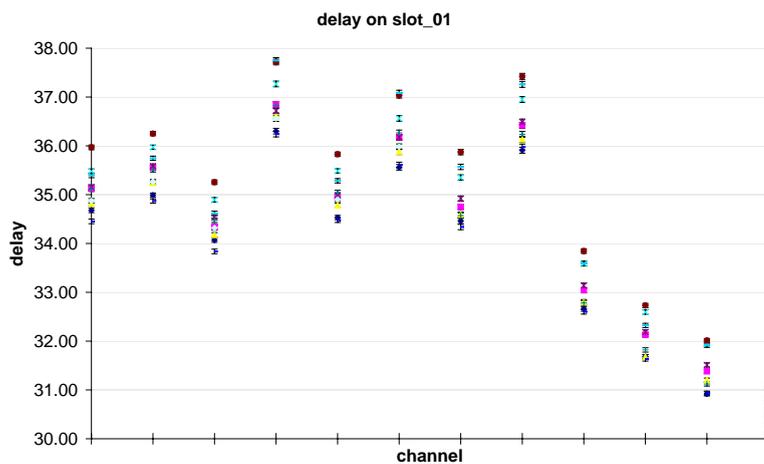


Figura 6.8: Tempi di propagazione del segnale *BC_Pulse-TSTPLS* dalla generazione nel modulo PDM fino ad un metro di cavo dopo le schede SB. Distribuzione di più schede a parità di posizione all'interno dell'*SB-crate*.

6.2.2 Il sistema di misura

Il sistema utilizzato in laboratorio durante la fase di misura, mostrato schematicamente in figura 6.9, è composta da:

- una scheda ODIN;
- un calcolatore per l'interfaccia PVSS;
- un modulo TTCvx;
- un SB-crate;
- una macchina *LeCroy WR104Xi*;
- una sonda differenziale *LeCroy AP034*;
- una sonda *LeCroy PP007*.

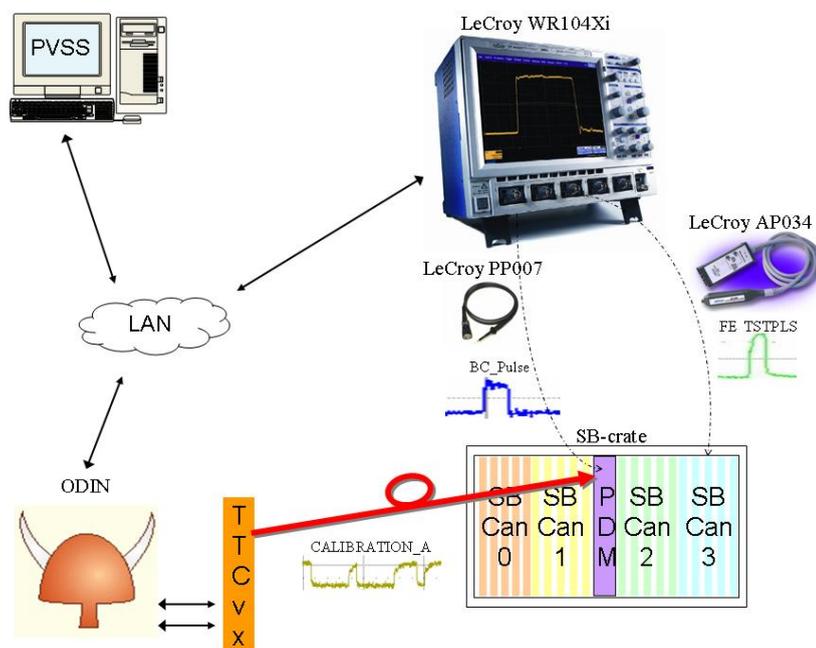


Figura 6.9: Apparato di misura utilizzato in laboratorio. È stato utilizzato un *LeCroy WR104Xi* equipaggiato con una sonda *AP034* ed una *PP007*, una scheda ODIN completa di interfaccia PVSS, una scheda TTCvx ed un SB-crate completo (20 schede SB e 1 modulo PDM).

Nella tabella 6.1, sono riportate le caratteristiche salienti della macchina LeCroy utilizzata; la grande potenzialità delle LeCroy WaveRunner consiste nel fatto che sono composte da due moduli distinti e interagenti: uno relativo all'acquisizione dati e gestito da *software proprietario LeCroy* e un'altro relativa all'elaborazione dei dati acquisiti e composta da un *Personal Computer* con sistema operativo *MS Windows*.

Lo *strobe* dell'oscilloscopio è stato sempre riferito al fronte di salita del segnale presente sul connettore J2-5 della scheda PDM, utilizzando la sonda *AP034*; su tale connettore, è possibile rilevare il segnale *BC_Pulse* generato dalla FPGA montata sul modulo PDM.

modulo oscilloscopio	
<i>banda passante</i>	1GHz
tempo di salita	400ps
impedenza d'ingresso	1MΩ/20pF
isolamento tra i canali	> 40dB@ < 100MHz (> 30dB@1GHz)
risoluzione verticale	8bit
<i>jitter</i>	2ps rms (@100mV/div)
<i>single-shot sample rate/ch</i>	5GS/s (10GS/s su due soli canali)
rumore introdotto	non superiore a 35nV/ ÷ Hz (> 10.000 : 1@60Hz)
modulo calcolatore	
processore	Intel 2.0GHz o superiore
S.O.	MS Windows XP
RAM	512MB
sonda differenziale AP034	
<i>banda passante</i>	1GHz
resistenza d'ingresso	2MΩ
capacità d'ingresso	0.9pF
rumore introdotto	non superiore a 35nV/ ÷ Hz (> 10.000 : 1@60Hz)
sonda PP007	
<i>banda passante</i>	non inferiore a 500MHz
resistenza d'ingresso	10MΩ
capacità d'ingresso	9.5pF

Tabella 6.1: Principali caratteristiche della macchina *LeCroy WR104Xi*.

La misura effettuata è stata eseguita dalla macchina della LeCroy tramite la funzione *ddelay* e consiste nel ritardo tra il primo fronte di salita (con riferimento a metà altezza) del segnale misurato con la sonda *AP034* ed il primo fronte di salita del segnale misurato con la sonda differenziale. I valori ottenuti, sono stati istogrammati (tramite la funzione *hist*) grazie al *software* residente sulla macchina LeCroy stessa.

Sull'istogramma ottenuto, sono stati calcolati i seguenti valori:

hmean: la media aritmetica dei valori istogrammati;

hmin: il più piccolo valore istogrammato;

hmax: il più grande valore istogrammato;

range: la differenza tra il maggior valore ed il minor valore istogrammato;

fwhm: l'ampiezza, a metà altezza, dell'istogramma dei valori.

I grafici riportati come esempio nel paragrafo precedente, riportano i valori di *hmean* su istogrammi di non meno di 10.000K misure, considerando come errore (*jitter temporale*) il valore corrispondente di *fwhm*.

Per forzare la generazione dell'impulso, si è utilizzata la scheda ODIN disponibile in laboratorio; grazie al progetto PVSS implementato, è possibile generare il segnale *broadcast* "CALIBRATION_A" in due modi:

1. su comando diretto dell'operatore, tramite il corrispondente pannello e la pressione del relativo pulsante "single shot"(come mostrato in figura 6.10);

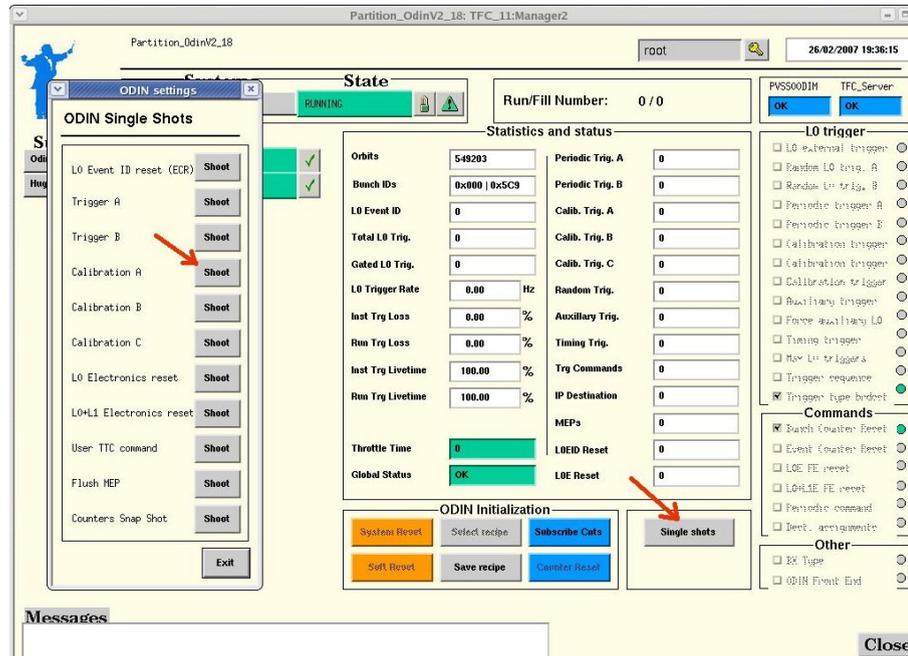


Figura 6.10: Sistema PVSS per la gestione della scheda ODIN: è indicata la procedure per la generazione manuale dei segnali *broadcast*

2. in maniera automatica, simulando sulla scheda il funzionamento del collisore.

Nell'utilizzo della modalità automatica, è necessario richiamare i parametri dell'acceleratore, precedentemente impostati nei pannelli di configurazione (figura 6.11).

In tabella 6.2 sono riportati i parametri per la simulazione del collisore LHC, utilizzati nelle misure effettuate.

Parametro	Valore
<i>orbit</i>	1
<i>offset A</i>	256
<i>trigger delay A</i>	176
<i>trigger windows A</i>	3

Tabella 6.2: Parametri relativi al segnale *CALIBRATION_A*, utilizzati per la simulazione del funzionamento del collisore LHC durante le misure effettuate in laboratorio sull'SB-crate.

In questo modo è stato possibile ottenere una situazione molto prossima a quella che si avrà una volta in funzione l'esperimento. Per ovviare alla dipendenza dalla temperatura, che incide in maniera percettibile sui tempi di propagazione dei segnali, è stata chiusa la bocca d'areazione del condizionatore, posta nei pressi dell'apparato. Nonostante ciò, è stato possibile notare una leggera discrepanza tra le misure effettuate in pieno giorno e quelle effettuate la notte.

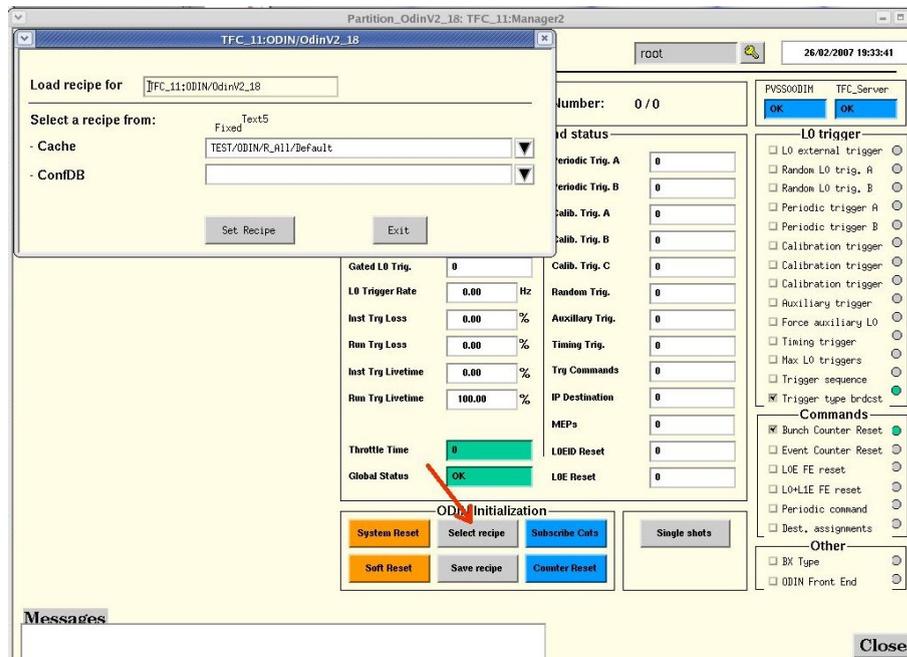


Figura 6.11: Sistema PVSS per la gestione della scheda ODIN: è indicata la procedura per richiamare la configurazione pre-impostata nel sistema.

Questo tipo di problema, non si avrà durante il funzionamento dell'esperimento, poiché l'inerzia termica della caverna di LHC garantirà il perdurare di una temperatura costante per tutto l'apparato elettronico installato.

conclusioni

Durante il primo periodo di tesi, ho lavorato al completamento del lato *software* del *Pulse Distribution Module*, partecipando attivamente alla messa in opera finale del codice; parallelamente ho testato il corretto funzionamento delle schede PDM, verificandone la corretta funzionalità. L'ultimo periodo di tesi, l'ho dedicato all'implementazione dell'interfaccia utente per la gestione del modulo PDM e all'inserimento del sistema PVSS all'interno del più generale sistema per la gestione dell'ECS; parallelamente ho partecipato ai test delle schede SB da inviare al CERN per l'installazione nel rivelatore di muoni dell'esperimento LHCb.

Le misure eseguite hanno caratterizzato sia le schede *Service Board* che il modulo PDM, sia per quanto riguarda la progettazione e realizzazione *hardware* che per la logica implementata, confermandone l'idoneità all'utilizzo nell'esperimento. A causa di ritardi tecnici, non è stato possibile eseguire in laboratorio un test completo della procedura di calibrazione e vedere così l'istogramma del *fine delay* composto da una sola riga; tuttavia, i test in corso al CERN stanno dando ottimi risultati e ci si aspetta un pieno successo non appena sarà possibile eseguire il primo test sull'apparato montato.

La logica implementata sulle ELMB delle schede prodotte dal gruppo LHCb di Roma è aggiornabile via CAN-bus ed è quindi possibile sviluppare ulteriori funzioni e aggiornamenti per il *software* installato all'interno degli SB-crate.

Al momento, il *software* installato sul modulo PDM è alla versione 1EPT per il *firmware* dell'ELMB e alla versione AD19 per quello dell'FPGA; quello della scheda SB alla versione R003 per l'ELMB ed alla AB27 per l'FPGA; il sistema PVSS è operativo e lo si sta installando nell'apparato definitivo al CERN.

Fino ad oggi (6 settembre 2007), sono state testate e rese operative 15 schede PDM e 118 schede SB, di cui 79 già posizionate all'interno degli SB-crate.

Durante il mio lavoro di tesi, ho avuto la possibilità di accostarmi al mondo sia degli esperimenti di fisica delle alte energie, sia al mondo dell'elettronica.

Lavorare nel gruppo LHCb della sezione di Roma dell'INFN, mi ha dato l'occasione di "vedere" e "toccare" quelle nozioni teoriche apprese durante il corso di studi facendomi ritrovare il gusto per la Fisica e la ricerca in generale.

Ho anche studiato ed applicato nozioni riguardo la trasmissione dati su *bus* VME, I^2C , CAN (quest'ultimo secondo il protocollo CANOpen).

Ho avuto occasione di studiare e programmare con il *software* SCADA PVSS.

Ho acquisito competenze nella programmazione nei linguaggi C, C++, VisualC, PVSS, Verilog.

Appendice A

Il Protocollo I^2C

L' I^2C (*Inter-Integrated Circuit*) è un protocollo di trasmissione seriale, inizialmente sviluppato dalla Philips, che fa uso di 2 sole linee di trasmissione, denominate “*SDA*” (*Serial Data*) ed “*SCL*” (*Serial Clock*); la configurazione tra trasmettitore e ricevitore è di tipo *Master-Slave*. L'algoritmo di codifica prevede che la linea *SCL*, che trasmette gli *strobe* sui quali campionare i dati sull'altra linea, sia comandata sempre dal dispositivo *Master*. La linea *SDA*, che contiene i dati, può invece essere comandata sia dal *Master* che dallo *Slave*, perchè deve consentire la trasmissione in entrambi i versi.

Come mostrato nello schema in figura A.1, quando la linea *SCL* è alta significa che il dato è disponibile alla lettura sulla linea *SDA*, quando la linea *SCL* è bassa, la linea *SDA* può transire allo stato del nuovo dato.

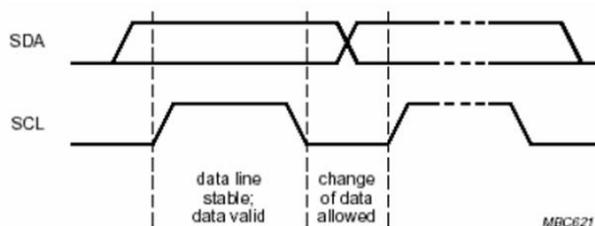


Figura A.1: Schema del funzionamento del protocollo I^2C

I dati vengono trasmessi 8 bit alla volta, alla fine di ogni dato viene generato un *acknowledge (ACK)*, ossia un *comando* di controllo. Il pacchetto inizia con un segnale di *Start*, seguono i dati con i rispettivi *ACK*; il pacchetto finisce con un segnale di *Stop*. In figura A.2 è riportato lo schema di invio di un pacchetto di dati secondo il protocollo I^2C .

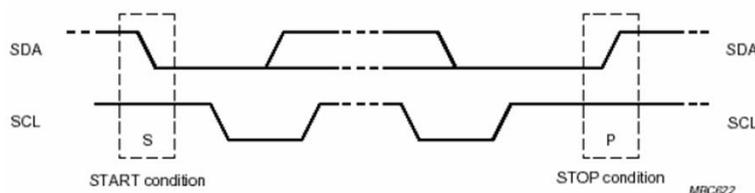


Figura A.2: il *Schema di invio di un pacchetto di dati secondo il protocollo I^2C*

I segnali di *Start* e *Stop* vengono sempre generati dal dispositivo *Master*, che comanda la trasmissione. Il segnale di *Start* è rappresentato dalla commutazione da livello alto a basso della linea *SDA*, mentre *SCL* è tenuta a livello alto. Lo *Stop* invece dalla commu-

tazione dell'SDA da livello basso ad alto, sempre con l'SCL a livello alto. Alla fine di un trasferimento può anche essere generato un *repeated-Start* al posto di uno *Stop*, in caso si voglia trasmettere subito un nuovo pacchetto. La condizione del *repeated-Start* è identica a quella dello *Start*.

La trasmissione di un dato tra un dispositivo ed il *Master* è, come già detto, seriale con 8 bit trasmessi alla volta, seguiti da un *acknowledge (ACK)*. Questo segnale è di solito generato dal *Master* con una transizione dell'SCL, durante la quale il dispositivo ricevitore (che può essere sia il *Master*, che lo *Slave*), deve tenere l'SDA a livello basso. Se il ricevitore, invece, non riceve bene il segnale, allora il canale SDA rimane a livello alto, durante la transizione di ACK. Questo verrà interpretato dal *Master* come un *not acknowledge* e la trasmissione verrà terminata con un segnale di *Stop*. Nel caso in cui si stia effettuando una trasmissione dallo *Slave* verso il *Master*, arrivati all'ultimo dato, per avvertire lo *Slave* che deve finire la trasmissione, il *Master* non deve trasmettere l'*acknowledge* (ovvero deve tenere l'SDA a livello alto); lo *Slave* interpreta il comando e rilascia l'SDA, in modo che il *Master* possa generare successivamente lo *Stop*. In figura A.3 è mostrato uno schema di generazione del segnale di *Acknowledge*.

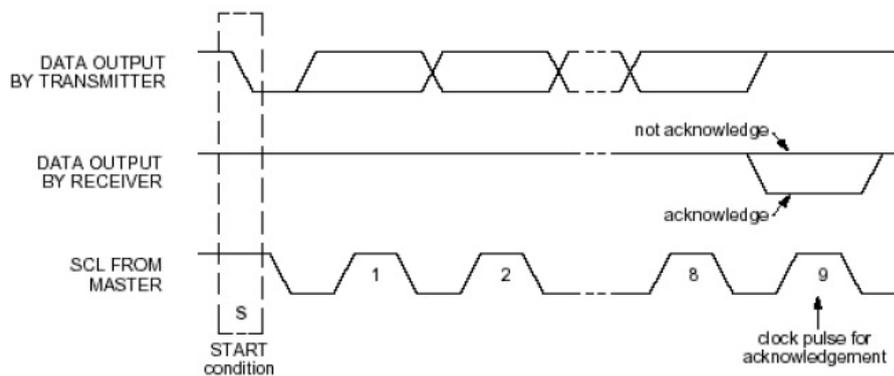


Figura A.3: Generazione del segnale di *Acknowledge* nel protocollo I²C.

Mentre i parametri principali della trasmissione su I²C-bus sono definiti in maniera generale, il formato dei pacchetti di dati può variare a seconda dell'utenza. In generale si usa un pacchetto così strutturato: all'inizio si genera uno *Start*, poi il primo pacchetto di dati, *Address + R/W*, che indica l'indirizzo di 7 bit con il quale si vuole comunicare, più 1 bit per segnalare se si vuole effettuare una lettura, od una scrittura. I successivi *frame Data*, sono 8 bit di dati che si vogliono trasmettere. Il numero di pacchetto di dati può essere grande a piacere, e alla fine viene generato uno *Stop*. Naturalmente, alla fine di ogni pacchetto trasmesso si deve sempre aspettare il segnale di *acknowledge*. In figura A.4 è mostrato un esempio di trasmissione di un pacchetto di dati secondo il protocollo I²C.

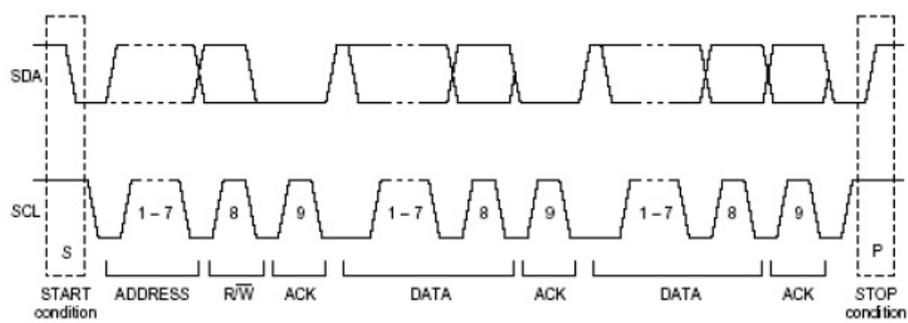


Figura A.4: Trasferimento completo di un pacchetto nel protocollo I^2C .

Appendice B

Il *CAN-bus* ed il protocollo *CANOpen*

B.1 Il CAN-bus

Il CAN (*Controlled Area Network*) è un modello di rete di comunicazione digitale appositamente studiato per applicazioni di controllo dei processi industriali, dove è necessario comunicare a grandi distanze in un ambiente tipicamente rumoroso. Il CAN-bus si distingue particolarmente, infatti, per un complesso sistema di verifica dell'integrità dell'informazione. Tuttavia, esso si basa su un metodo di comunicazione piuttosto elementare che semplifica notevolmente l'allestimento della rete.

Il suo funzionamento è basato su soli 3 dei 7 livelli, del normale modello di rete OSI (*Open System Interconnect*):

- Il *Physical Layer*, ovvero il *buffer* di I/O che connette il cavo con il dispositivo CAN. Vengono usate tre linee di segnale: SDI, SDO, SCLK, più le alimentazioni (a 12 V);
- Il *Data Link Layer*, ovvero il livello in cui viene definito il formato dei dati di indirizzamento e di trasmissione;
- L'*Application Layer*, ovvero l'interfaccia tra il controllore CAN ed i dispositivi finali con i quali si vuole comunicare. Si basa su una serie di comandi standard per servizi utili per tutti i dispositivi collegati. Fornisce le diverse configurazioni di comunicazione per ogni dispositivo, e definisce il modo in cui i dati vengono trasmessi tra i dispositivi.

I primi due livelli definiscono la rete CAN-bus, mentre il terzo livello è utilizzabile con diversi tipi di protocollo. In figura B.1 è riportato uno schema del sistema di comunicazione CAN. In particolare, per l'esperimento LHCb è stato scelto il protocollo di comunicazione di alto livello "CANOpen" (descritto in B.2).

La trasmissione dei dati avviene con messaggi inviati in maniera *broadcast*. Questo significa che ogni dispositivo collegato al bus riceve ogni messaggio presente sulla rete. Le informazioni necessarie a identificare il contenuto, la priorità nonché la provenienza sono codificate per *default* nella parte iniziale del messaggio. Il dispositivo ricevente può, accedendo a tali informazioni, eventualmente ignorare il messaggio recepito, se opportuno. Questo tipo di comunicazione, non effettuata come consuetudine *da punto a punto*, ha grossi vantaggi pratici:

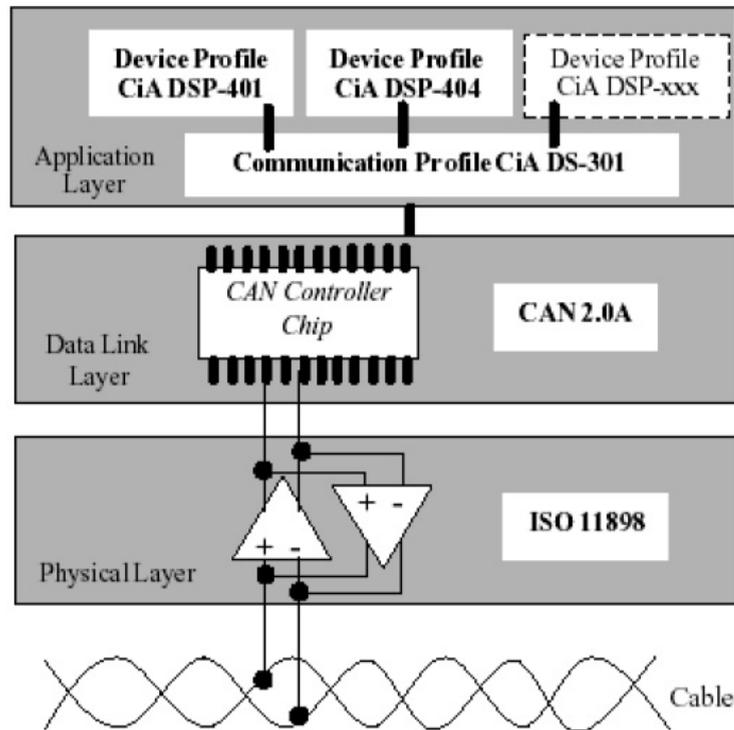


Figura B.1: Schema generale del sistema di comunicazione CAN

- Non è necessario specificare alcun tipo di indirizzamento;
- I dispositivi collegati al bus, detti *nod*i, sono identificati da un solo numero intero denominato *node-id*.
- Collegare o scollegare un nodo dal bus non comporta alcuna riconfigurazione *hardware* o *software* del sistema.
- È sufficiente che ogni dispositivo abbia un *node-id* differente.

Teoricamente il CAN-bus potrebbe collegare un numero molto alto di nodi. Insorgono delle limitazioni per ragioni legate all'*hardware*: quali l'impedenza di carico del bus e i ritardi di propagazione. Inoltre, il numero di bit dedicato al *node-id* non può essere illimitato. Il *node-id* è codificato generalmente con 7 bit, quindi è possibile collegare fino a 128 nodi a meno di id riservati secondo protocollo, il che, comunque, costituisce un numero considerevole di dispositivi collegabili.

Tutti i nodi collegati al bus possiedono la cosiddetta abilità di *bus master*, il che significa che possono prendere autonomamente il controllo del bus e iniziare l'invio di un messaggio.

Nel CAN-bus la codifica del bit utilizza livelli di tensione con convenzione NRZ (*Non Return Zero*). La tensione che corrisponde fisicamente ai bit trasmessi è, diversa da zero per entrambi i livelli e non sono previsti intervalli a tensione nulla di separazione tra due bit consecutivi. Inoltre, il *bus* è realizzato in *hardware* in modo che un livello di tensione sia dominante rispetto all'altro, detto *recessivo*. Il livello *dominante* si sovrappone a quello recessivo, per cui la cifra binaria associata a questo, non importa quale, sovrascrive l'altra

sul bus.

In figura B.2 è riportato il formato standard dei messaggi CAN; questo si compone di



Figura B.2: Il formato standard per i messaggi su bus di tipo CAN.

vari campi, con funzioni specifiche:

SOF (*start of frame*): è un bit che identifica l'inizio della comunicazione: sempre uguale a un livello dominante;

identifier : contiene le informazioni relative al nodo di provenienza, al contenuto e alla priorità del messaggio. Questo campo è fondamentale in caso di accesso contemporaneo di più nodi al bus. Ogni dispositivo è abilitato all'accesso al bus solo quando rileva una condizione di bus inutilizzato (*IDLE*). Può accadere, tuttavia, che più nodi contemporaneamente cerchino l'accesso al bus, entrando in competizione. In questo caso, avviene semplicemente che un nodo interrompe l'invio del messaggio se si accorge che un altro lo sta sovrascrivendo. Il meccanismo descritto permette di effettuare un corretto arbitraggio dell'accesso al bus senza alcuna perdita di banda. Difatti nessuna ritrasmissione o informazione aggiuntiva viene utilizzata. Dato che il livello dominante sovrascrive il recessivo, l'accesso al bus sarà guadagnato dal messaggio con *identifier* codificato da livelli dominanti nei primi bit. Ad esempio, se il livello dominante corrisponde allo zero, come avviene di solito, gli *identifier* più piccoli saranno quelli corrispondenti a una priorità superiore perchè, avendo livelli dominanti zero nei bit più significativi, sovrascriveranno gli altri;

RTR (*Remote Transmission Request*): identifica il messaggio come una richiesta di trasmissione dati. Con essa si notifica al nodo ricevente la richiesta di invio di un messaggio avente uguale *identifier* ma contenente le informazioni richieste. Una *transmission request* sarà sempre priva del campo *DATA*;

IDE (*IDentifier Extended*): indica se il messaggio ha un formato esteso, con *identifier* a 18 bit anzichè 11. Il supporto ai messaggi estesi non è sempre disponibile in tutti i sistemi;

r0 : è riservato al *controller* ed è costituito da due soli bit;

DLC (*Data Length Code*): definisce la lunghezza del messaggio, consistente nel numero di *byte* di dati contenuti nella sezione *DATA* fino a un massimo di otto. Nel caso di messaggio di richiesta di trasmissione dati, il campo *DLC* contiene il numero di *byte* richiesti;

CRC e ACK : sono utilizzati per il controllo di errore che è particolarmente articolato.

Tutti i nodi collegati al bus verificano l'integrità del messaggio ricevuto tramite il campo *CRC* (*Cyclic Redundancy Code*), che è una somma dei bit effettuata al momento della trasmissione. Se la stessa somma, effettuata in ricezione, non corrisponde

a quella inviata, sarà segnalato un *errore di CRC*. Il controllo di errore viene effettuato anche sul formato del messaggio che, se non rispetta lo standard, genera un errore di formato.

Nel campo *ACK (ACKnowledgement)* il nodo trasmittente invia un bit recessivo, che viene sovrascritto con uno dominante da ogni nodo ricevente che non abbia riscontrato errori. Se ciò non avviene, viene notificata un'altra condizione di errore da parte del nodo trasmittente, detta di *acknowledgement*.

I campi del messaggio fino a parte del *CRC* sono codificati in modo che una sequenza di più di 5 bit uguali venga automaticamente interrotta da un bit di valore opposto. Questa tecnica è denominata *bit stuffing* e permette un ulteriore controllo, che genera un *errore di stuff*, qualora non sia rispettata in un qualsiasi messaggio ricevuto.

EOF (*End Of File*): serve a indicare la conclusione del messaggio. Consiste in una sequenza fissa di sette livelli recessivi. Le condizioni di errore vengono trasmesse tramite dei messaggi costituiti da sequenze di bit uguali consecutivi che, violando la regola del *bit stuff*, possono essere distinti facilmente dai normali messaggi.

Per evitare che l'invio di ripetute segnalazioni di errore da parte di nodi malfunzionanti o malcollegati possa disturbare tutta la rete, esiste un meccanismo di conteggio degli errori. Tale procedura è definita di *fault confinement*. I nodi che non comunicano correttamente vengono individuati e disabilitati o inibiti, mantenendo la rete in condizioni operative. Ogni nodo è dotato di due contatori di errore in ricezione e in trasmissione. In sintesi la procedura consiste nell'incrementarli di otto unità ogniqualvolta si verificano degli errori e decrementarli, se la trasmissione o la ricezione dei messaggi viene portata a termine con successo, di una unità. In base al valore di questi contatori il nodo assume degli stati differenti. Normalmente è *error active* ma diviene *error passive* quando viene superato 128 o *bus off* oltre il 256. I nodi *error active* effettuano appena possibile la ritrasmissione del messaggio errato, quelli *error passive* attendono, trasmettendo una sequenza fissa di bit recessivi, prima di provare a ritrasmettere. I nodi *bus off*, infine, sono disabilitati a trasmettere fino a un comando di *reset* o alla ricezione di undici bit recessivi consecutivi.

In realtà il sistema è più complesso: definisce delle eccezioni e distingue gli errori segnalati a seconda dello stato del nodo che li rileva. Questo per una gestione ottimale delle condizioni di errore ripetuto. Ad esempio, se sul bus ci fosse un solo nodo, passerebbe in breve tempo allo stato *bus off*, a causa dei continui *acknowledgement errors* rilevati per l'assenza di nodi in ricezione. Poiché questa non corrisponde a una condizione di cattivo funzionamento di quel nodo, deve essere tollerata, come infatti accade. Solitamente i nodi sono predisposti per inviare un messaggio di *warning* quando il valore dei contatori di errore raggiunge la soglia delle 96 unità, per notificare una condizione di funzionamento non ottimale della rete.

IFS (*Inter Frame Space*): costituisce una sequenza fissa di separazione tra messaggi consecutivi, necessaria al corretto funzionamento del sistema. La separazione non viene adottata per i messaggi di errore, che vengono inviati non appena esso viene rilevato. Se la corretta separazione tra i messaggi non viene rispettata, allora un nodo può inviare un messaggio di *Overload* che ha semplicemente l'effetto di occupare il bus e ritardare le successive trasmissioni. Questo messaggio può essere inviato anche in seguito ad una particolare condizione interna del nodo che richieda una separazione temporale superiore a un *IFS* con il messaggio successivo. Anche il messaggio

di *overflow* non deve essere necessariamente preceduto da *IFS*, così come quelli di errore.

Il terzo livello del CAN, il *CAN Application Layer (CAL)* si occupa di fornire 4 servizi tramite quattro tipi di dati diversi:

CMS: (CAN Message Specification), specifica in che modo l'interfaccia CAN possa accedere ad un dispositivo, che in generale viene chiamato un *nodo* della rete, ed in quale modalità, che può essere del tipo *Variabile*, *Evento* o *Dominio*.

NMT: (Network Management), si occupa del supporto dei servizi della rete, come *inizializzazione*, *start* e *stop* dei nodi, e controllo che i vari nodi rispondano. E' concepito secondo una logica di tipo *master-slave*.

DBT: (DistriBuTor), si occupa della distribuzione dinamica degli identificatori dei nodi, i *COB-ID (Communication Object Identifier)*.

LMT: (Layer Management), abilita il cambiamento dei parametri dell'*Application Layer*, come l'indirizzo dei nodi, o l'interfaccia tra un nodo ed il controllore CAN.

Il protocollo standard del CAN-bus non definisce il formato dei dati nè i contenuti possibili per i messaggi. Il CAN 2.0 si limita a costituire un protocollo di trasferimento dei dati con le direttive base già descritte.

Per costituire uno schema di comunicazione completo è necessario adoperare un cosiddetto *higher layer protocol* che definisca sintatticamente i messaggi da adoperare e la loro corrispondenza con le informazioni disponibili al nodo.

Il protocollo di alto livello adoperato nella scheda ELMB è denominato *CANOpen*. Mentre il trasferimento dei dati è gestito fisicamente dal controllore del bus, è principalmente il *master processor* della scheda ad implementare nella pratica la gestione della comunicazione ad alto livello.

B.2 Il protocollo CANOpen

Il *CANOpen* si occupa dell'implementazione di un sistema di controllo distribuito utilizzando i 4 servizi dell'*Application Layer* della rete CAN. La trasmissione viene effettuata alla velocità di *500KBit/s*.

Si può immaginare come una sorta di interfaccia che traduce le informazioni che il dispositivo fisico deve scambiare, in messaggi codificati in formato compatibile con il CAN 2.0. Il *CANOpen* definisce, innanzitutto, delle tipologie di messaggio standard. A queste è associato un *COB-ID (Communication Object Identifier)* corrispondente all'*identifier* nel formato standard CAN 2.0. Il *COB-ID* viene ottenuto accodando ad un *function-code* a 4 bit, associato ad ogni tipo di messaggio, l'identificativo di nodo a 7 bit.

Essendo lo zero definito come livello recessivo, i *function-code* più bassi e, a parità di questi, i *node-id* più bassi corrispondono a maggiore priorità del messaggio.

L'elemento base del *CANOpen* è il *device Object Dictionary (OD)*, ovvero, una libreria di oggetti con la quale si definiscono tutti i parametri e le informazioni necessarie per la comunicazione tra rete e dispositivo. Ogni oggetto è indirizzato con un indice a 16 bit, e un sottoindice ad 8 bit. Per ogni nodo della rete esiste un OD che contiene tutte le informazioni che riguardano quel particolare dispositivo. Ogni nodo è quindi completamente caratterizzato dal suo *Object Dictionary*.

Ogni oggetto all'interno dell'OD, è caratterizzato da un nome, dal tipo (numero, vettore....) e dalla categoria (obbligatorio, opzionale o condizionale).

In figura B.3 è riportato uno schema del protocollo di trasmissione CANOpen.

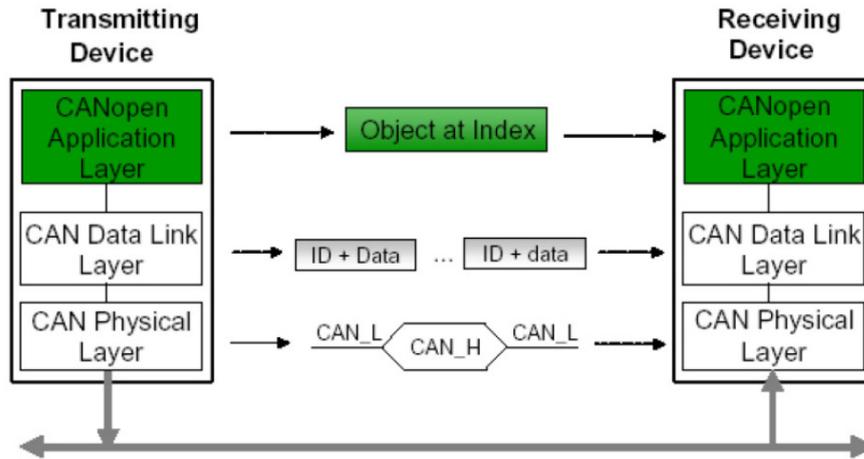


Figura B.3: Schema del protocollo CANOpen

Oltre agli OD, che descrivono i nodi ed i dispositivi collegati, il CANOpen trasmette i dati veri e propri, sotto forma di oggetti chiamati *Communication Object (COB)*. Questi sono divisi in 4 tipi:

Administrative Message: sono messaggi di gestione e controllo della rete e dell'*Application Layer* come inizializzazione e configurazione. Sono la gestione CANOpen dei segnali NMT, DBT LMT.

Gli stati operativi possibili di un nodo CANOpen sono quattro, come vediamo in figura B.4.

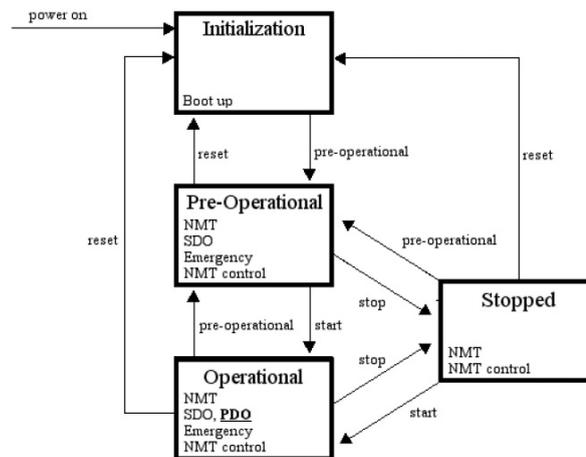


Figura B.4: Diagramma degli stati di un nodo CANOpen.

Nello schema raffigurato si può comprendere il significato dei comandi di NMT elencati, che impongono uno specifico cambiamento di stato.

Il primo stato è l'inizializzazione, in cui il dispositivo entra all'accensione o dopo ogni *reset hardware* o *software*. Il secondo è lo stato *pre - operational* che il nodo raggiunge

autonomamente, dopo una breve fase di *start - up*, dopo quello di inizializzazione. Lo stato *operational* è lo stato di completa attivazione di tutte le funzionalità comunicative del nodo mentre in quello *pre - operational* non tutti i messaggi possono essere inviati.

Allo stato *stopped* corrisponde una condizione di inibizione della comunicazione: solo le funzioni di controllo vengono mantenute attive. Il messaggio *NMT* può essere inviato al singolo nodo o a tutti se si specifica un *node - id* zero nel *COB - ID*, quindi il valore zero non è assegnabile nel CANOpen a nessun nodo.

Service Data Object (SDO): è usato per trasferire i dati da un dispositivo *client* ad un dispositivo *server*. La trasmissione è sempre doppia, poichè ogni volta che un dispositivo trasmette un dato, il ricevitore gli deve rispondere generando un *acknowledge* che indica che la trasmissione è avvenuta con successo. Il dato che si vuole trasferire non ha limiti di lunghezza, nè è soggetto ad alcun particolare protocollo, ma devono essere il *client* ed il *server* a sapere come codificare e decodificare il dato trasmesso. Il pacchetto di trasmissione è diviso in segmenti. Alla fine di ogni segmento si aspetta l'*acknowledge* e l'ultimo segmento contiene un indicatore di fine pacchetto. Ogni segmento è formato da 4 blocchi:

- il COB-ID, che indica verso quale nodo si vuole trasmettere.
- Il comando, che indica se si vuole effettuare un *download/upload*, una richiesta/risposta, l'indicatore di fine pacchetto, la richiesta di interruzione della trasmissione e la lunghezza del dato che si vuole trasmettere.
- La mappatura del dato sull'OD, che non è altro che l'indirizzo ed il sottoindirizzo dell'OD relativo al tipo di oggetto che si sta trasmettendo.
- Infine viene trasmesso il dato stesso.

Process Data Object (PDO): è usato per trasferire i dati in tempo reale da un dispositivo che li produce verso uno o più dispositivi, che li leggono. Può trasmettere da 1 fino a 8 byte e non prevede alcun protocollo di trasmissione particolare, per cui anche in questo caso, basta che il trasmettitore ed il ricevitore usino lo stesso. Il PDO è un pacchetto composto da due diversi blocchi più il dato da trasmettere: il *Communication Parameter*, che contiene l'identificatore del nodo (COB ID) ed il tipo di trasmissione che si vuole effettuare. Il messaggio può essere trasmesso in quattro modi diversi: sincrono ad un oggetto esterno chiamato *SYNC*, e aperiodico, oppure sincrono e periodico, oppure asincrono e comandato da una richiesta di un altro dispositivo, ed infine asincrono e comandato da un particolare oggetto specificato nell'OD. Successivamente viene trasmesso il *Mapping Parameter* che contiene, oltre alla mappatura sull'OD dell'oggetto da trasmettere, anche la lunghezza del dato stesso (nel caso esso sia più corto di 8 bit). Alla fine viene trasmessa la parola ad 8 bit contenente il dato. La trasmissione PDO a differenza di quella SDO non è controllata da nessun messaggio di riuscita trasmissione.

Special Function Object: oggetti con funzione particolari, come il *SYNC*, oppure segnali di temporizzazione tra i vari dispositivi, o oggetti di emergenza (*Emergency Object*), in caso di errori interni, o oggetti di controllo dei nodi (*Node Guardian Identifier*) in accordo con i segnali NMT dell'*Application Layer*.

In particolare, l'*Emergency Object* è costituito da un unico messaggio contenente fino ad otto *byte* di dati; viene usato per notificare una condizione particolare del nodo o,

spesso, l'occorrenza di un errore od un malfunzionamento. L'*emergency object* è un messaggio sempre associato ad un unico nodo ed è ad alta priorità. Il dato contenuto in questo tipo di messaggio è codificato nell'OD in modo caratteristico a seconda del dispositivo, per notificare un particolare tipo di evento occorso ogni volta con un diverso messaggio.

Infine, il messaggio con la più bassa priorità, che viene denominato *NMT error control*, contiene nel campo dati la codifica dello stato in cui il nodo si trova. Tale messaggio viene utilizzato per l'implementazione di schemi di verifica del funzionamento del nodo. Una prima possibilità è il cosiddetto *node guarding* in cui a un messaggio *RTR* del *NMT master*, segue la risposta del nodo, contenente l'informazione sul suo stato attuale. Questo permette di monitorare il corretto funzionamento del nodo ed eventuali modifiche indesiderate occorse nello stato dello stesso. Altra possibilità è costituita dall'*heartbeat*, in cui un messaggio di controllo viene inviato dal nodo sul bus a intervalli di tempo predefiniti. La cessazione di questo segnale di *battito cardiaco* rilevato dal *NMT master* sarà indice di disfunzioni nel nodo. Lo stesso tipo di messaggio ma privo del campo dati è il cosiddetto *boot - up message*, che viene inviato per notifica da ogni nodo, al passaggio dallo stato di inizializzazione a quello *pre - operativo*.

In figura B.5 è riportato uno schema indicante le relazioni tra CAN-bus, *Object Dictionary* e applicazioni *software*.

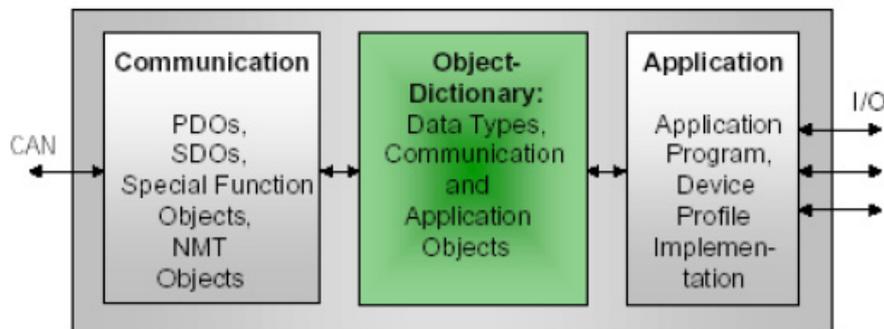


Figura B.5: Relazione tra CANbus, *Object Dictionary* e applicazioni *software*.

Appendice C

Gli objects del PDM

Index (hex)	Sub Index	Name	Data Object	Attr	Default	Comment
4300		I2C bus3(G)	Record		Default	
	0	I2C ADDRESS	U8	RW	0xA3	
	1	I2C number of bytes	U8	RW	0xB3	
	2	I2C data bytes	U32	RW		
	3	I2C SubAddr	U8	RW	0xC3	
	4	I2C subaddr data bytes	U32	RW		
4700		Actel PDM Registers	Record		Default	
	00	REG_SEL_BC_PULSE [1:0]	U8	RW	0x00	
	01	TTCrdy	U8	R	0x00	
	03	REG_SFWRPULSE	U8	RW	0x00	
	04	REG_AND_COMP	U8	RW	0x00	
	05	REG_TRASPARENTE	U8	RW	0x00	
	06	REG_BCNTRES				
	09	DEVICE	U8	R	0x00	
	0A	VERSION	U8	R	0x00	
	0B	Day	U8	R	0x00	
	0C	Month	U8	R	0x00	
	0D	Year	U8	R	0x00	
	0E	REG_COMP_PDM_L [7:0]	U8	RW	0x00	
	0F	REG_COMP_PDM_H [3:0]	U8	RW	0x00	
	10	REG_MSK_BRC		RW		
	11	REG_TTC_SUBADDR [7:0]		R		
	12	reset_b		RW		

Index (hex)	Sub Index	Name	Data Object	Attr	Default	Comment
5700		TTCRx Register	Record		Default	
	0	Fine Delay 1	U8	RW	0x00	
	1	Fine Delay 2	U8	RW	0x00	
	2	Coarse Delay	U8	RW	0x00	
	3	Control	U8	RW	0x93	
	8	Single error count 7-0	U8	RW	0x00	
	9	Single error count 15-8	U8	RW	0x00	
	10	Double error count 7-0	U8	RW	0x00	
	11	SEU error count 7-0	U8	RW	0x00	
	16	ID 7-0	U8	RW	0x00	
	17	MasterModeA	U8	RW	0x00	
	18	MasterModeB	U8	RW	0x00	
	19	Config1	U8	RW	0x1A	
	20	Config2	U8	RW	0x84	
	21	Config3	U8	RW	0xA7	
	22	Status	U8	RW	0xE0	
	24	Bunch counter 7-0	U8	RW	0x00	
	25	Bunch counter 15-8	U8	RW	0x00	
	26	Event counter 7-0	U8	RW	0x00	
	27	Event counter 15-8	U8	RW	0x00	
	28	Event counter 23-16	U8	RW	0x00	
5800		PDM Slot	Record		Default	
		Read slot address of PDM and turn on yellow and green LED if right	U8	R		
5701		Reset Board	Record		Default	
		Reset Actel and (then) TTCrq set TTCrx register to default value				
5900		PCF U5 Register	Record		Default	
	15	RST_ACTEL		RW		
5901		PCF U10 Register	Record		Default	
	<15..0>	ELMB_CTRL<15..0>		RW		
5902		SB's ELMB Control by PCF U10				
	0	Service Board Address (1-20)		W		
	1	ELMB Address (0-3 or 5 for all)		W		
	2	Reset or Shutdown (1 RST - 2 SHTD)		W		

Appendice D

L'OPC (*OLE for Process Control*)

OPC (*OLE for Process Control*) è un *software* frequentemente utilizzato nel campo del controllo e della automazione dei processi industriali. Esso ha come funzione primaria quella di costituire un'interfaccia di comunicazione con le apparecchiature utilizzate nel campo dei controlli automatici, quali controllori, attuatori, PLC.

OPC viene generalmente fornito, esattamente come un *driver*, dalle case produttrici ed ha funzioni simili. Il *driver* vero e proprio di un dispositivo si occupa, infatti, di effettuare un interfacciamento diretto con la periferica, in modo da consentire fisicamente la comunicazione. Se esso non effettua nessuna elaborazione dei dati provenienti dall'apparecchiatura, questi vengono resi in un formato che sarà tipico del protocollo di comunicazione utilizzato. L'utente o, più propriamente, un sistema *software* per il controllo dei processi deve essere in grado di adoperare queste informazioni e interpretarle correttamente. La soluzione a questo problema costringerebbe ogni *pacchetto software* a comprendere *driver* adatti per tutte le diverse tipologie di periferiche o le case produttrici a distribuire *driver* compatibili con ogni *software* per l'automazione.

Operare in questo modo sarebbe dispendioso e inefficiente: è questa la ragione per cui si adopera OPC, che presenta il grosso vantaggio di definire uno standard, ormai universalmente riconosciuto.

Se un dispositivo è dotato di un *OPC Server* in grado di comunicare con esso e capace di organizzare i dati secondo il suo standard, qualsiasi sistema *software* dotato della funzione di *OPC Client* potrà operare correttamente con quel dispositivo.

OPC è un *software* che opera su sistemi *Windows NT* e *Linux* ed è basato, innanzitutto, sul sistema di gestione dati della *Microsoft OLE (Object Linking and Embedding)*.

Con questa tecnica è possibile definire degli oggetti che un programma, denominato *server*, è in grado di creare ed editare. Qualsiasi applicazione, denominata *client*, può accedere a questi oggetti attivando il *server*. Se un *client* modifica tramite il *server* un oggetto questo verrà aggiornato su tutti i *client*. OPC Server, in particolare, crea uno spazio di indirizzamento (*Address Space*) suddiviso in *gruppi* e *item*. Questi oggetti sono associati ai dati provenienti in remoto dall'apparecchiatura con un'organizzazione standardizzata o configurabile dall'utente.

Ogni *item* contiene delle informazioni accessibili in lettura o scrittura con il sistema tipico della tecnologia OLE. Una applicazione *client* sarà quindi in grado di visualizzare e adoperare i dati contenuti negli *item* e modificarli. OPC, mantenendo il collegamento con il dispositivo, potrà inviare le informazioni allo stesso, consentendo così la comunicazione.

Un'altra funzione importante e molto utile di OPC è quella di consentire l'accesso agli *item* attraverso rete locale. Utilizzando le tecnologie *Microsoft COM / DCOM (Distributed*

Component Object Model) i *client* possono accedere al *server* e quindi agli *item* in remoto, attraverso la rete. È sufficiente che solo un calcolatore sia dotato dell'*hardware* necessario al collegamento con un dispositivo. I dati provenienti dall'apparecchiatura saranno comunque accessibili a tutti i calcolatori in rete tramite l'OPC Server. La configurazione delle autorizzazioni di accesso al *server OPC*, definibile nel sistema operativo, consente, opzionalmente, di limitare la divulgazione delle informazioni. Queste funzioni sono utilissime nei sistemi di automazione, in quanto consentono la distribuzione mirata dei dati provenienti dalle apparecchiature sui diversi terminali della rete. Inoltre, permettono di ottenere un certo livello di protezione e sicurezza: in un sistema di controllo di un processo industriale la modifica indesiderata di alcuni parametri potrebbe essere molto dannosa, se non pericolosa.

L'OPC Server utilizzato con la scheda ELMB è stato elaborato dagli stessi gruppi di ricerca che l'hanno progettata. Esso è costituito da un programma principale e due estensioni (*DLL*) che consentono di operare con due possibili schede di interfaccia per il CAN-bus. Il *software* consente di associare i dati provenienti dalla scheda ELMB, nei messaggi standard CANOpen, con gli *item* definiti dall'OPC standard.

Per consentire la corretta comunicazione con la scheda ELMB, tramite un file di configurazione, bisogna settare opportunamente gli *item* e definire le corrispondenze con i messaggi provenienti dal bus. Di seguito viene riportato il listato del file di configurazione implementato per il modulo PDM.

D.1 Il file di configurazione per l'OPC Server

[CANBUS]

CAN_BUS_0 = KVCANserver+ 0 125000

[DEVICE]

ELMBPDM = CAN_BUS_0 03 HB

[SDOItem]

ELMBPDM_VERSION = ELMBPDM 1009 0 IO VT_UI4
 ELMBPDM_SUBVERSION = ELMBPDM 100a 0 IO VT_UI4
 ELMBPDM_ACTEL_SEL_BC_PULSE = ELMBPDM 4700 0 IO VT_UI1
 ELMBPDM_ACTEL_TTC_RDR = ELMBPDM 4700 1 IO VT_UI1
 ELMBPDM_ACTEL_REG02 = ELMBPDM 4700 2 IO VT_UI1
 ELMBPDM_ACTEL_SFWRPULSE = ELMBPDM 4700 3 IO VT_UI1
 ELMBPDM_ACTEL_AND_COMP = ELMBPDM 4700 4 IO VT_UI1
 ELMBPDM_ACTEL_TRASP = ELMBPDM 4700 5 IO VT_UI1
 ELMBPDM_ACTEL_BCNTRES = ELMBPDM 4700 6 IO VT_UI1
 ELMBPDM_ACTEL_REG07 = ELMBPDM 4700 7 IO VT_UI1
 ELMBPDM_ACTEL_REG08 = ELMBPDM 4700 8 IO VT_UI1
 ELMBPDM_ACTEL_DEV = ELMBPDM 4700 9 IO VT_UI1
 ELMBPDM_ACTEL_VERS = ELMBPDM 4700 A IO VT_UI1
 ELMBPDM_ACTEL_DAY = ELMBPDM 4700 B IO VT_UI1
 ELMBPDM_ACTEL_MONTH = ELMBPDM 4700 C IO VT_UI1
 ELMBPDM_ACTEL_YEAR = ELMBPDM 4700 D IO VT_UI1
 ELMBPDM_ACTEL_COMP_PDM_L = ELMBPDM 4700 E IO VT_UI1
 ELMBPDM_ACTEL_COMP_PDM_H = ELMBPDM 4700 F IO VT_UI1
 ELMBPDM_ACTEL_REG_MSK_BRC = ELMBPDM 4700 10 IO VT_UI1

```
ELMBPDM_ACTEL_REG_TTC_SUBADDR = ELMBPDM 4700 11 IO VT_UI1
ELMBPDM_ACTEL_RESET_B = ELMBPDM 4700 12 IO VT_UI1
ELMBPDM_TTCRX_FDELAY_1 = ELMBPDM 5700 0 IO VT_UI1
ELMBPDM_TTCRX_FDELAY_2 = ELMBPDM 5700 1 IO VT_UI1
ELMBPDM_TTCRX_CDELAY = ELMBPDM 5700 2 IO VT_UI1
ELMBPDM_TTCRX_CTRL = ELMBPDM 5700 3 IO VT_UI1
ELMBPDM_TTCRX_ERR_0 = ELMBPDM 5700 8 IO VT_UI1
ELMBPDM_TTCRX_ERR_1 = ELMBPDM 5700 9 IO VT_UI1
ELMBPDM_TTCRX_ERR_2 = ELMBPDM 5700 A IO VT_UI1
ELMBPDM_TTCRX_ERR_3 = ELMBPDM 5700 B IO VT_UI1
ELMBPDM_TTCRX_ID_0 = ELMBPDM 5700 10 IO VT_UI1
ELMBPDM_TTCRX_ID_1 = ELMBPDM 5700 11 IO VT_UI1
ELMBPDM_TTCRX_ID_2 = ELMBPDM 5700 12 IO VT_UI1
ELMBPDM_TTCRX_CFG_1 = ELMBPDM 5700 13 IO VT_UI1
ELMBPDM_TTCRX_CFG_2 = ELMBPDM 5700 14 IO VT_UI1
ELMBPDM_TTCRX_CFG_3 = ELMBPDM 5700 15 IO VT_UI1
ELMBPDM_TTCRX_STATUS = ELMBPDM 5700 16 IO VT_UI1
ELMBPDM_TTCRX_BC_COUNT_0 = ELMBPDM 5700 18 IO VT_UI1
ELMBPDM_TTCRX_BC_COUNT_1 = ELMBPDM 5700 19 IO VT_UI1
ELMBPDM_TTCRX_EV_COUNT_0 = ELMBPDM 5700 1A IO VT_UI1
ELMBPDM_TTCRX_EV_COUNT_1 = ELMBPDM 5700 1B IO VT_UI1
ELMBPDM_TTCRX_EV_COUNT_2 = ELMBPDM 5700 1C IO VT_UI1
ELMBPDM_RSTB = ELMBPDM 5701 0 IO VT_UI1
ELMBPDM_PDM_SLOT = ELMBPDM 5800 0 IO VT_UI1
ELMBPDM_PCF_U5 = ELMBPDM 5900 0 IO VT_UI2
ELMBPDM_PCF_U10 = ELMBPDM 5901 0 IO VT_UI2
ELMBPDM_PCF_SBELB = ELMBPDM 5902 0 IO VT_UI2
```


Glossario

ASIC : I *Circuiti Integrati per Applicazioni Specifiche* sono circuiti integrati costruiti su misura per gli scopi di un particolare progetto. Per la propria natura rappresentano quanto di più performante è possibile realizzare rispetto alle specifiche di utilizzo, ma non permettono nessun aggiornamento, hanno un alto costo iniziale e tempi di realizzazione relativamente lunghi.

Broadcast : un segnale “*broadcast*” è un comando che viene inviato senza specificare l’indirizzo del componente di destinazione: esso ha effetto su tutta l’elettronica del sistema (si pensi ad esempio ad un comando di *reset* dell’intero sistema).

Bunch Crossing : un *bunch crossing* indica l’evento in cui i due fasci $p-p$ si intersecano in uno degli otto punti d’intersezione dell’acceleratore. Se si considera la velocità del fascio, il diametro dell’anello e la distanza fra due intersezioni dei fasci, si ricava che la frequenza propria di *bunch crossing* in LHC è di 40MHz, ossia 1 evento ogni $25ns$.

BX-id : *Bunch Crossing identifier*. Ad ogni intersezione dei fasci viene assegnato all’evento un numero ad otto bit, il (*BX-id*), che l’identifica; lo stesso codice identificativo deve essere applicato anche alle misure che il rivelatore di muoni compie e che, dopo aver transitato per l’elettronica dell’esperimento, verranno memorizzate sul supporto di memoria di massa.

FIFO : una memoria *FIFO (First In First Out)* è una memoria “orizzontale” (contrapposta allo *stack* che può essere considerato “verticale”); ossia è un *buffer* nel quale è possibile immagazzinare, uno alla volta, diversi dati a seconda della quantità di memoria disponibile. La particolarità della FIFO è che i *bit* “transitano” nella memoria e ne riescono nello stesso ordine in cui vi sono stati immagazzinati, dopo un numero di colpi di *clock* pari alla profondità della memoria; per questa particolarità una FIFO può essere impiegata come una rudimentale linea di ritardo.

Flip Flop : sono dei dispositivi elettronici, utilizzati nell’elettronica digitale, come elementi sincroni di memoria ad un bit. Essi prevedono due soli stati logici possibili, pilotati da ingressi sincroni, ovvero regolati con un ingresso dinamico detto *clock*.

Il nome Flip-Flop deriva dal rumore che facevano i primi circuiti di questo tipo costruiti con dei relè, quando cambiavano di stato.

Flip Flop D : i “*Data Flip-Flop*” mantengono la porta d’uscita allo stato logico della porta d’ingresso; tale stato viene aggiornato esclusivamente in presenza di un fronte di salita nel segnale di *clock*.

Isolamento Galvanico : l’isolamento galvanico garantisce che fra due circuiti elettrici passi l’informazione portata dal segnale elettrico ma non ci sia contatto ohmico; questo serve ad evitare che variazioni di potenziale sul piano di massa si ripercuotano

come rumore sul segnale di ingresso. Solitamente l'isolamento galvanico è ottenuto grazie a componenti ottici: il segnale elettrico viene trasformato in segnale luminoso, garantendo così il passaggio dell'informazione. In un sistema digitale ottico, il livello basso è rappresentato dall'assenza di luce, mentre quello alto è dato dalla presenza di luce, indipendentemente dalla sua intensità. In questo modo è possibile evitare il propagarsi di sbalzi di tensione e/o corrente da un circuito all'altro, pulendo il segnale da eventuale rumore.

LVDS : i segnali *LVDS (Low-Voltage Differential Signaling)* sono segnali differenziali in grado di essere trasmessi a grande velocità e basso rumore; infatti la piccola ampiezza del segnale elettrico garantisce un basso accoppiamento elettrico e magnetico riducendo notevolmente il rumore sul cavo.

Il trasmettitore inietta una piccola corrente, nominalmente di $3.5mA$, in una linea o nell'altra a seconda dello stato logico da rappresentare; la corrente percorre la linea di trasmissione fino a transitare nella resistenza di adattamento, per poi tornare al trasmettitore sull'altra linea. Il ricevitore deve essere in grado di apprezzare la polarità della corrente per poter determinare lo stato logico. Segnali di questo tipo vengono chiamati *current loop*.

Il protocollo LVDS viene utilizzato quando si ha necessità di cavi di trasmissione superiori al metro di lunghezza e segnali con fronti di commutazione dell'ordine del nanosecondo.

Power Cycle : Un "*power cycle*" è l'operazione per cui si toglie corrente ad un *chip* elettronico, per provocarne uno spegnimento, con la successiva cancellazione di tutte le memorie volatili presenti; alla riaccensione si ottiene una completa reinizializzazione come da specifiche tecniche.

SEU : *Single Event Upset* rappresenta un fenomeno di natura fisica per cui un determinato evento, quale il passaggio di una particella ionizzante attraverso il componente, cambia lo stato logico di una o più celle di memoria. Questo tipo di errore, presente in componenti di tipo semiconduttore, in genere non è distruttivo per l'elemento in sé; tuttavia è necessario dotare il circuito di una logica di correzione, *software* od *hardware*, per evitare che un SEU produca un errore statisticamente rilevante.

Stack : uno *stack* è una memoria "a pila"; ossia è un *buffer* nel quale è possibile immagazzinare, uno alla volta, diversi dati a seconda della quantità di memoria disponibile. La particolarità dello *stack* è che i dati vengono richiamati in ordine inverso a come sono stati immagazzinati; questo permette di poter immagazzinare dati senza necessariamente saturare la capacità di memoria disponibile per richiamarli.

TTL : gli elementi della classe di circuiti digitali integrati Transistor-Transistor Logic sono costruiti con l'utilizzo di transistor a giunzione bipolare (BJT) e resistori. È chiamata logica transistore-transistore perchè sia la funzione di porta logica (ad esempio OR) che quella di amplificazione, sono svolte da transistori.

VME : Il *bus* VME è un *bus* standard per l'utilizzo nei calcolatori; originariamente era stato implementato per le linee dei processori Motorola 68000 ma trova oggi largo impiego per usi in molte applicazioni e standard nelle specifiche IEC dell'ANSI/IEEE 1014-1987.

Watchdog : Il *watchdog* o *watchdog timer* (in italiano: temporizzazione di supervisore - letteralmente: cane da guardia) è un sistema di temporizzazione *hardware* che

permette ad un microcontrollore la rilevazione di un *“loop”* infinito di programma o di una situazione di *“deadlock”*.

Tale rilevazione può consentire di prendere dei provvedimenti per correggere la situazione, generalmente effettuando un *reset* del sistema e la sua reinizializzazione.

Acronimi

acronimo	per esteso	
ACK	ACKnowledge	Condizione/segnale di conferma
ADC	Analog-Digital Converter	Convertitore Analogico-Digitale
ALICE	A Large Ion Collider Experiment	Un esperimento di LHC per lo studio di ioni pesanti
ASD	Amplifier Shaper Discriminator	Filtro amplificatore, formatore e discriminatore
ASIC	Application Specific Integrated Circuit	Circuito Integrato per Applicazione Specifica
ATLAS	A Toroidal LHC ApparatuS	Un esperimento di LHC per lo studio del bosone di Higgs
CAN	Controller Area Network	<i>Bus</i> di comunicazione
CARDIAC	CARioca and DIAlOG Card	Scheda per chip Carioca e chip Dialog nell'esperimento LHCb
CARIOCA	Cern And RIO Current Amplifier	Scheda di <i>Front End</i> nell'esperimento LHCb
CERN	Conseil Européen pour la Recherche Nucléaire	Consiglio Europeo per la Ricerca Nucleare
CMS	Compact Muon Solenoid	Un esperimento di LHC per lo studio del bosone di Higgs
COB	Communication OBJect	Elemento del protocollo CAN
DAC	Digital to Analog Converter	Convertitore Digitale-Analogico
DAQ	Data AcQuisition	Sistema di acquisizione dati
DIALOG	DIagnostic time Alignment and LOGics	Scheda di <i>Front End</i> nell'esperimento LHCb
DLL	Delay Locked Loop	Circuito elettronico per la sincronizzazione dei segnali
ECS	Experiment Control System	Sistema di Controllo dell'Esperimento LHCb
ELMB	Embedded Local Monitor Board	Scheda elettronica utilizzata al CERN per la comunicazione su CAN-bus
FIFO	First In First Out	Tipo di memoria digitale
FOI	Field Of Interest	finestra di interesse
FPGA	Field Programmable Gate Array	Circuito di elettronica programmabile
GOL	Giga-bit Optical transmitter	Modulo per la trasmissione dati ad alta velocità tramite <i>link</i> ottico
GUT	Grand Unified Theory	Grande Teoria Unificata
I ² C	Inter-Integrated Circuit	<i>Bus</i> di comunicazione
IB	Intermediate Board	Scheda elettronica dell'esperimento LHCb
INFN	Istituto Nazionale di Fisica Nucleare	Istituto Nazionale di Fisica Nucleare
ISP	In System Programming	Connettore per la programmazione residente sulla scheda
LHC	Large Hadron Collider	Grande Collisore per Adroni
LHCb	LHC beauty experiment	Un esperimento di LHC per lo studio della violazione di CP
LNF	Laboratori Nazionali di Frascati	Laboratori Nazionali di Frascati
LVDS	Low-Voltage Differential Signaling	Segnale digitale differenziale a basso voltaggio
NMT	Network Management	Elemento del protocollo CAN
OD	Object Dictionary	Dizionario Oggetti
ODE	Off Detector Elettronics	Scheda di controllo dell'elettronica dell'esperimento LHCb
OPC	Ole for Process Control	Protocollo di comunicazione
PDM	Pulse Distribution Module	Scheda di controllo dell'elettronica dell'esperimento LHCb
PDO	Process Data Object	Elemento del protocollo CAN
PLL	Phase Locked Loop	Circuito elettronico per la sincronizzazione di fase
PVSS	ProzessVisualisierungs- und Steuerungs-System	Sistema SCADA
SB	Service Board	Scheda di controllo dell'elettronica dell'esperimento LHCb
SB-System	Service Board System	Sistema composto da schede SB e un modulo PDM
SCADA	Supervisory Control And Data Acquisition	<i>Software</i> per la gestione di sistemi ingegneristici
SDO	Service Data Object	Elemento del protocollo CAN
TDC	Time to Digital Converter	Convertitore Tempo-Digitale
TFC	Timing and Fast Control system	Sistema di controllo dei segnali
TTC	Time and Trigger Control	Sistema di controllo dei segnali
TTL	Transistor-Transistor Logic	Logica Transistore-Transistore

Bibliografia

- [1] Richard P. Feynman, *The Character of Physical Law*, MIT Press, Cambridge 1965.
- [2] F. Mandl, G. Shaw, *Quantum Field Theory*, John Wiley & Sons, 1993.
- [3] S. Weinberg, *A model of Leptons*, Phys. Rev. Lett. **19** (1967) 1264.
- [4] A. Salam, J.C. Ward, *Electromagnetic and weak interactions*, Phys. Lett. **13** (1964) 168.
- [5] H.Fritzsch, M. Gell-Mann, H. Leutwyler, Phys. Lett. **47** (1973) 365.
- [6] J. Christenson et al., Phys. Rev. Lett. **13** (1964) 138.
- [7] A.D. Sakharov, JEPT Lett. **6** (1967) 21.
- [8] M.B. Gavela et al., Modern Phys. Lett. **9A** (1994) 795.
- [9] D.E. Groom et al. (Particle Data Group), The European Physical Journal **C15**, 1 (2000), URL: <http://pdg.lbl.gov/>.
- [10] G. Passaleva, *Violazione di CP nei mesoni B*, 5/6/2001
- [11] LHCb: Technical Proposal, *A Large Hadron Collider Beauty Experiment for Precision Measurements of CP Violation and Rare Decays*, CERN-LHCC-98-004, LHCC-P-4, CERN 1998 <http://lhcb-tp.web.cern.ch/lhcb-tp/>
- [12] M. Kobayashi & K. Maskawa, Prog. Theor Phys. **49** (1973) 652.
- [13] L. Wolfenstein, Phys. Rev. Lett. **51** (1983) 1945.
- [14] N. Cabibbo, Phys. Rev. Lett. **10** (1963) 531.
- [15] *The Atlas Experiment*, <http://pdg.lbl.gov/atlas/atlas.html>
- [16] *The CMS Experiment*, <http://www.phys.ufl.edu/hee/cms/>
- [17] *The Alice Experiment*, <http://map.web.cern.ch/ALICE/welcome.html>
- [18] *The Large Hadron Collider beauty experiment* , <http://lhcb.web.cern.ch/lhcb/>
- [19] *LHC Design Report*, <http://ab-div.web.cern.ch/ab-div/Publications/LHC-DesignReport.html>
- [20] *Tevatron Department Homepage*, <http://www-bdnew.fnal.gov/tevatron/>
- [21] *Fermilab Homepage*, <http://www.fnal.gov/>

- [22] *LHCb Computing Technical Design Report, 11*, 77th LHCC Meeting, 29 June 2005, CERN/LHCC 2005-019; LHCb TDR 11, LAL 05-229
- [23] *LHCb VELO (VERtEX LOcator) : Technical Design Report*, CERN-LHCC-2001-0011 [sic!], CERN-LHCC-2001-011, LHCb-TDR-5, CERN, 2001, *The Vertex Detector Home Page* <http://lhcb-vd.web.cern.ch/lhcb-vd/Default.htm>
- [24] *LHCb Silicon Tracker - Technical Design Reports*, CERN-LHCC-2002-029, CERN 2002, <http://www.physik.uzh.ch/groups/lhcb/public/tdr/index.html>
- [25] *LHCb RICH : Technical Design Report*, CERN-LHCC-2000-037, LHCb-TDR-3, CERN, 2000, *RICH detectors Home Page* <http://lhcb-rich.web.cern.ch/lhcb-rich/>
- [26] *LHCb calorimeters Technical Design Report*, CERN-LHCC-2000-0036-[sic!], CERN-LHCC-2000-036, LHCb-TDR-2, CERN, 2000 <http://lhcb-calo.web.cern.ch/lhcb-calo/>
- [27] *LHCb muon system : Technical Design Report*, CERN-LHCC-2001-010, LHCb-TDR-4, CERN, 2001, *LHCb Muon Group Home Page* <http://lhcb-muon.web.cern.ch/lhcb-muon/>
- [28] *LHCb Detector Paper*, CERN LHCC 2006-xxx, ISBN 92-9083-180-4, CERN, 2007, <http://lhcb-muon.web.cern.ch/lhcb-muon/muondraft.pdf>
- [29] A.Lai, *Muon architecture review*, CERN 27-3-2003
- [30] A.Lai et al., LHCb note 2000-50
- [31] W. Bonivento et al., *Development of the CARIOCA front-end chip for the LHCb Muon Detector*, Nucl. Instrum. Meth. A491:233-243, 2002 URL: <http://riegler.home.cern.ch/riegler/>
- [32] S. Cadeddu, C. Deplano and A. Lai, *The DIALOG chip in the front-end electronics of the LHCb Muon Detector*, 2004 IEEE Nuclear Science Symposium 0-783-8701-5. IEEE Trans.Nucl.Sci. 2004 in press.
- [33] *Muon Off-Detector electronics: The IB system*, LHCb Technical Note. URL: <http://agenda.cern.ch/askArchive.php?base=agenda&categ=a054562&id=a054562s-1t13%2Fdocument%2Flhcb-2003-023rev2.doc>
- [34] LHCb LNF Group, *Muon Off-Detector electronics: The IB system*, Muon Electronics Production Readiness Review , 15 July 2005 at Roma - La Sapienza, <http://indico.cern.ch/getFile.py/access?contribId=s1t13&resId=1&materialId=0&confId=a054562>
- [35] *Radiation studies with FLUKA*, <http://lhcb-background.web.cern.ch/lhcb-background/Radiation/RadLevels.htm>
- [36] V.Bocci et al., *Radiation test and application of FPGAs in the Atlas Level 1 Trigger*, 7th Workshop on Electronics for LHCExperiments, Stockholm, Sweden, 10-14 September 2001.
- [37] V.Bocci et al., *Time Resolution of the Muon Detector: Consequences for Trigger Performance and Detector Synchronization*, LHCb-2001-058.

- [38] V. Bocci, G. Martellotti, A. Lai, A. Sciubba, *Muon Detector Front-end Architecture: an update*, LHCb 2001-030.
- [39] S. Cadeddu and A. Lai, *DIALOG- β DATA SHEET*, I.N.F.N. Sezione di Cagliari, Cagliari, Italy; LHCb 2003-016 MUON
CERN 27/3/03
- [40] *Guide for ECS FSM design in LHCb sub-detectors*, LHCb Technical Note, EDMS 655828, CERN, 2005.
- [41] *ELMB technical documentation*,
<http://atlasinfo.cern.ch/ATLAS/GROUPS/DAQTRIG/DCS/ELMB/SB/index.html>
- [42] *Data sheet ATmega128*,
http://www.atmel.com/dyn/resources/prod_documents/2467s.pdf
- [43] *Muon Off_ Detector Electronics Board*, LHCb Technical Note,
<http://agenda.cern.ch/askArchive.php?base=agenda&categ=a054562&id=a054562s-1t15/document>
- [44] Balla, P. Ciambrone, G. Felici, *ODE Board L0 front-end for muon system*.
- [45] Guido Haefeli et al., *TELL1 Specification for a common read out board for LHCb*, LHCb 2003-007, IPHE 2003-02, July 23, 2005
http://lphe1dell1.epfl.ch/ghaefeli/specification_and_documents/TELL1.pdf
- [46] V.Bocci, G.Chiodi, F.Iacoangeli, R.Nobrega, D.Pinci, W.Rinaldi, *A Prototype of the LHCb Muon Detector Control System*, ICALEPCS 2005, Geneva, Switzerland, 10-14 Oct. 2005
- [47] V.Bocci, G.Chiodi, F.Iacoangeli, W.Rinaldi, *Data Sheet Pulse Distribution Module*, Muon Electronics Production Readiness Review, 15 July 2005, CERN
- [48] V.Bocci, G.Chiodi, F.Iacoangeli, W.Rinaldi, *Data Sheet Services Board*, Muon Electronics Production Readiness Review, 15 July 2005, CERN
- [49] R. Jacobsson et al., *The LHCb Timing and Fast Control System*, LHC Electronics Workshop, 2001.
http://lhcb-online.web.cern.ch/lhcb-online/TFC/documents/LEB2001_paper.pdf
- [50] R. Jacobsson et al., *TFC Switch Specifications*, LHCb Technical Note LHCb Technical Note, LHCb DAQ 2001-018
- [51] URL: <http://lhcb-online.web.cern.ch/lhcb-online/>
- [52] R.Jacobsson, *Driving the LHCb Front-End Readout*, Real Time Conference, Montreal, 18-23/05/2003
- [53] URL: <http://lhcb-online.web.cern.ch/lhcb-online/TFC/default.html>
- [54] URL: <http://lhcb-online.web.cern.ch/lhcb-online/TFC/html/Odin.html>
- [55] Z. Guzik and R. Jacobsson, *LHCb Readout Supervisor 'ODIN' - Technical reference*, CERN Feb. 2006

- [56] R. Jacobsson et al., *The Final LHCb Readout Supervisor "ODIN"*, CERN, LHC Electronics Workshop 2002,
<http://lhcb-online.web.cern.ch/lhcb-online/TFC/documents/LEB2002.pdf>
- [57] R. Jacobsson, B. Jost and Z. Guzik, *TFC Switch Specifications*, LHCb 2001-18, May 2000, <http://lhcb-online.web.cern.ch/lhcb-online/TFC/html/Thor.html>
- [58] URL: <http://lhc-beam-beam.web.cern.ch/lhc-beam-beam/>
- [59] P. Moreira, *TTCRq module*, CERN - EP/MIC, Geneva Switzerland November 2004,
<http://proj-qpll.web.cern.ch/proj-qpll/images/manualTTCrqp.pdf>
- [60] *ProASICPLUS® Flash Family FPGAs DataSheet*,
http://www.actel.com/documents/ProASICPlus_DS.pdf
- [61] *DataSheet PCF8575 Remote 16-bit I/O expander for I2C-bus*, Feb1999
http://agenda.cern.ch/askArchive.php?base=agenda&categ=a03650&id=a03650s1t7-%2Ftransparencies%2FODE_Overview.ppt
- [62] *Timing, Trigger and Control (TTC) Systems for the LHC*
<http://ttc.web.cern.ch/TTC/intro.html>
- [63] P. Moreira et al., *A Timing, Trigger and Control Receiver ASIC for LHC Detectors*, CERN - EP/MIC, Geneva Switzerland January 2003,
http://ttc.web.cern.ch/TTC/TTCrx_manual3.8.pdf
- [64] *TTCRx ASIC*, <http://ttc.web.cern.ch/TTC/TTCmain.html#TTCrx>
- [65] *Technical Information Manual*, mod V1718-VX118, NPO:00106/03:V1718.MUTx/06,
<http://www.caen.it/getattach.php?mod=V1718&obj=mn&id=991>
- [66] Ph.Farthouat, P.Gällnö CERN EP - ATE, *TTC-VMEbus INTERFACE, TTCvi - MkII*, EP 680-1128-050-C CERN, Rev1.6 May 2000,
<http://ttc.web.cern.ch/TTC/TTCviSpec.pdf>
- [67] P.Gällnö CERN EP - ATE, *"TTCvx" Technical description and users manual*, CERN, May 1999, <http://ttc.web.cern.ch/TTC/TTCvxManual1a.pdf>
- [68] *PVSS II Getting Started - Basics*, URL:
http://itcobe.web.cern.ch/itcobe/Services/Pvss/GettingStarted/PVSS_II_GettingStarted_Basics_en.pdf
- [69] JCOP Framework Team, *JOINT CONTROLS PROJECT (JCOP) FRAMEWORK SUB-PROJECT GUIDELINES AND CONVENTIONS*, CERN-JCOP-2000-008, CERN, Vers 1.2 July 2007,
<http://itcobe.web.cern.ch/itcobe/Projects/Framework/Documentation/guidelinesDocument.pdf>
- [70] V.Bocci, G.Cernicchiaro, G.Chiodi, C.Gobel, F.Iacoangeli, R.Nobrega, W.Rinaldi, *Control System for the LHCb Muon Detector Front-end using SCADA PVSS*, 10th workshop on Electronics for LHC Experiments and Future Experiments, 13 - 17 September 2004, BOSTON, USA.
- [71] V.Bocci, G.Chiodi, F.Iacoangeli, R.Nobrega, D.Pinci, W.Rinaldi, *Implementation of the Control System for the LHCb Muon Detector*, 12th WORKSHOP on Electronics for LHC and future Experiments, 25-29 September 2006, Valencia, Spain.

- [72] V.Bocci, G.Chiodi, F.Iacoangeli, R.Nobrega, D.Pinci, W.Rinaldi, *Performance Evaluation of the Control System for the LHCb Muon Detector*, Nuclear Science Symposium & Medical Imaging Conference (NSS/MIC) 2005, October 23 - 29, 2005, Puerto Rico.
- [73] F.Iacoangeli, *Studio del sistema di test e di controllo delle camere per I muoni dell'esperimento LHCb*, Tesi di laurea , Università La Sapienza, Roma, 2003
- [74] F.Iacoangeli, *Studio e realizzazione di elettronica resistente a radiazione ionizzante per il controllo e la calibrazione del rivelatore di muoni dell'esperimento LHCb*, Tesi di dottorato, Università Tor Vergata, Roma, 2007

Ringraziamenti

All'Ignoranza che ha la sua residenza nel Castello, perchè mi ha scosso dal torpore, facendomi riprendere gli studi e dandomi nuove motivazioni nell'approfondimento della Cultura e delle Scienze.

Innanzitutto vorrei ringraziare i miei genitori: per avermi fatto condurre “a modo mio” questo percorso di studio, senza mai farmi mancare appoggio e sostegno fino a giungere a scrivere questa pagina.

Un sincero ringraziamento ai miei relatori:

Valerio Bocci, per la fiducia accordatami, la guida ed i numerosi insegnamenti ricevuti durante tutto il lavoro di tesi;
Franco Meddi per la disponibilità con cui mi ha seguito e i consigli ricevuti durante il lavoro di stesura della tesi.

Non posso non ringraziare di cuore Francesco Iacoangeli, per la pazienza con cui ha saputo consigliarmi e la collaborazione ricevuta durante il lavoro di tesi.

Desidero inoltre ringraziare:

Federico Spila, Rafael Nobrega Antunes e Davide Pinci, per i numerosi consigli ricevuti durante il mio lavoro;
Giacomo Chiodi, Antonio Rossi, Daniele Ruggieri, Felice Citronelli per il supporto tecnico e la disponibilità.

Un caloroso ringraziamento va a tutti i componenti del laboratorio di elettronica della sezione di Roma dell'INFN.

Un ultimo ringraziamento a tutti coloro che hanno reso piacevoli e divertenti questi anni passati a Fisica: Betta, Alberto, Federico, Valeria, Silvia, Alessio, Andrea, Giuseppe, Vania, Carla, ...