Università di Pisa

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Magistrale in Fisica

Tesi di Laurea Magistrale

Ricostruzione di tracce in tempo reale su FPGA a LHC

Candidato:
Daniele Ninci

Relatori:
Prof. Giovanni Punzi
Dott. Franco Spinella
Dott. Riccardo Cenci

Anno Accademico 2014–2015

Indice

Introduzione

Nei moderni esperimenti situati agli acceleratori adronici i sistemi di tracciatura in tempo reale e di trigger rivestono un'importanza fondamentale per la discriminare gli eventi interessanti dal fondo. Inoltre gli elevati valori di energia e luminosità raggiunti attualmente dagli acceleratori richiedono agli esperimenti lo sviluppo di tecniche avanzate e innovative per effettuare una tracciatura in tempo reale in modo efficiente.

In questa tesi abbiamo studiato l'implementazione su dispositivi a logica programmabile (FPGA) di un nuovo algoritmo di tracciatura che trae ispirazione dal funzionamento dell'apparato visivo dei mammiferi, chiamato Retina Artificiale. Questo algoritmo sfrutta il calcolo parallelo della risposta di una matrice di celle, che contengono una banca dati di tracce memorizzate, coprendo tutto lo spazio dei parametri in cui le tracce sono definite. Interpolando la risposta delle celle adiacenti, è possibile ottenere un'alta efficienza mantenendo limitato il numero di celle usate. Descriveremo in particolare il progetto di una unità di processamento di tracce (Track Processing Unit, TPU), un sistema che implementa l'algoritmo Retina Artificiale realizzato su FPGA. La TPU ha come obiettivo finale quello di ricostruire tracce con alta efficienza alla frequenza di 40 MHz, che è anche la frequenza delle collisioni raggiunta negli odierni acceleratori adronici.

L'utilizzo di questo algoritmo è in particolarmente importante negli esperimenti dedicati a misure di precisione, in cui eseguire una efficiente selezione degli eventi richiede una ricostruzione accurata delle tracce dell'evento stesso, in particolar modo quelle che coinvolgono i quark pesanti charm e bottom. Infatti, gli eventi che contengono i quark b e c, sono privi di una segnatura caratteristica (come ad esempio l'energia totale trasversa, l'energia trasversa mancante o la presenza di leptoni ad alto impulso trasverso) utile per preselezionare gli eventi. Un esempio di questo tipo di esperimenti è LHCb, situato lungo l'acceleratore LHC presso i laboratori del CERN di Ginevra, che ha come scopo lo studio della fisica dei quark pesanti. In questo caso I sistemi di tracciatura in tempo reale sono indispensabili per effettuare le misure previste dall'esperimento stesso.

Le basi di partenza per questo lavoro sono state le simulazioni delle prestazioni della TPU sia nella configurazione di LHCb prevista per il 2015, in cui la frequenza di lettura degli eventi è pari a 1 MHz, sia nella configurazione prevista per il 2020, in cui la frequenza sarà di 40 MHz. Nel primo caso si è assunto di inviare alla TPU i dati provenienti dal rivelatore Inner Tracker (IT), mentre nel secondo dal rivelatore di vertice (VELO) e dal rivelatore Upstream Detector (UT). Il lavoro di tesi consiste nell'implementazione dell'algoritmo con i parametri definiti nello studi precendente utilizzando il linguaggio di descrizione della logica di alto livello VHDL e nella successiva simulazione della logica. Infatti la simulazione della logica, anche ad uno stadio iniziale, risulta fondamentale per dimostrare la fattibilità tecnica in termini di velocità, dimensioni e costi dell'apparato.

Per l'implementazione della TPU, abbiamo utilizzato due diversi dispositivi a logica programmabile prodotti dalla ditta Altera. Nel caso del rivelatore IT abbiamo utilizzato un dispositivo di media grandezza, appartenente alla famiglia Altera Stratix III, mentre nel caso dei rivelatori VELO e UT, che costituiscono un tracciatore in 3D assai più complesso, abbiamo usato un dispositivo altamente performante, appartenente alla famiglia Altera Stratix V. La prima scelta è stata guidata dal fatto che lo Stratix III è usato in una scheda elettronica sviluppata dalla Sezione di Pisa dell'Istituto Nazionale di Fisica Nucleare. Questo permette di eseguire facilmente e in tempi brevi dei test di laboratorio su schede complete già disponibili, senza doverne progettare e costruire di nuove. Inoltre, anche se progettata per altri scopi, la scheda in questione mantiene una completa compatibilità con il sistema di acquisizione dati di LHCb e questo apre la possibilità di effettuare in futuro un test parassitico della TPU direttamente sui rivelatori di LHCb. Dall'altro lato, la famiglia Altera Stratix V è stata scelta perché comprende dispositivi all'avanguardia come quantità di celle logiche contenute e l'applicazione dell'algoritmo ai rivelatori VELO e UT ne richiede una grande quantità.

Nel primo Capitolo si discutono i benefici di un sistema di tracciatura in tempo reale, riferendoci ad alcuni esempi di sistemi di tracciatura implementati in esperimenti passati e attuali installati ai collisionatori adronici. Nel secondo Capitolo si descrive l'attuale esperimento LHCb e la configurazione prevista per l'upgrade del 2020, focalizzandoci sul sistema di trigger e dei rivelatori di traccia. Descriviamo quindi in dettaglio l'algoritmo della retina artificiale e la TPU nel Capitolo 3. Nel Capitolo 4 si introducono i dispositivi a logica programmabile, motivando la scelta dei dispositivi FPGA, e vengono descritti gli FPGA di Altera e i software utilizzati per la progettazione e la simulazione degli stessi. Nel quinto Capitolo si descrivono l'implementazione della TPU applicata all'IT, le sue prestazioni e i risultati della simulazione logica. Infine, nel Capitolo 6 si descrivono ancora l'implementazione, le prestazione e i risultati della simulazione logica ma questa volta per l'applicazione della TPU ai rivelatori VELO e UT nella configurazione di LHCb del 2020.

1

Trigger di traccia per collisionatori adronici

1.1 Motivazioni sperimentali

Nel 1964, durante lo studio dei decadimenti dei mesoni K neutri a vita media lunga in stati finali di due e tre pioni [1], è stata osservata per la prima volta l'evidenza indiretta della violazione della simmetria CP, cioè la simmetria dei processi fisici sotto inversione delle coordinate spaziali (trasformazione di parità, P) e di tutti i numeri quantici intrinsechi delle particelle considerate (trasformazione di coniugazione di carica, C). Nel caso dei mesoni K, la violazione di CP riguarda i quark di tipo s. Sin da allora l'asimmetria di CP è stata ampiamente studiata nell'ambito della fisica della alte energie e molto importanti sono stati gli studi effettuati sulla fisica del *flavour* alle macchine acceleratrici, in particolare nei settori del *charm* e del *beauty*.

Infatti gli adroni formati da quark di tipo b e c, rappresentano due importanti sistemi per lo studio della violazione di CP. Gli adroni contenenti il quark b appartengono alla terza famiglia di quark e possono quindi decadere in quark appartenenti alla prima ed alla seconda famiglia. In questo modo, si possono misurare effetti maggiori della violazione di CP rispetto al sistema contenente i K. Inoltre sono accessibili cinematicamente molti più canali di decadimento, in quanto il quark b ha una massa circa 20 volte maggiore rispetto a quella del quark s. Gli adroni con il quark c sono gli unici sistemi nei quali è possibile studiare le interazioni con i quark u, che, almeno in linea di principio, possono avere una dinamica differente rispetto ai quark d. Tuttavia, la presenza di molti canali di decadimento disponibili risulta in una piccola frazione di decadimento (*branching ratio*, B.R.) dei processi individuali, richiedendo così di acquisire campioni con un elevato numero di eventi per avere alta statistica.

La fisica dei quark pesanti può essere studiata con l'utilizzo di due tipi di versi di macchine acceleratrici: le *B-factories* e gli acceleratori adronici.

Le B-factories sono collisionatori e^+e^- in cui i due fasci hanno un'energia differente, e collidendo producono la $\Upsilon(4S)$, che decade con un B.R. del 96% in coppie $B\bar{B}$ [2] (dove $B = B^0$ or B^+) con un vertice secondario distante tipicamente $200 - 300\,\mu$m

dal punto di interazione elettrone-positrone. Utilizzando un'energia nel centro di massa leggermente al di sopra della massa della $\Upsilon(4S)$, ovvero della soglia di produzione del quark b, si evita la presenza di prodotti di frammentazione, imponendo delle restrizioni cinematiche che si riflettono in una riduzione degli eventi di fondo. Interazioni primarie multiple in una singola collisione dei fasci (*pile-up*) sono generalmente assenti e la molteplicità delle tracce è tipicamente di ~ 5 tracce per evento. Tuttavia la sezione d'urto di produzione di una coppia $B\bar{B}$ è limitata a $\sigma(b\bar{b}) \sim 1$ nb. Esperimenti installati alle B-factories sono stati BABAR[3] al *Linear Accelerator Center* di Stanford (SLAC) e BELLE [4] ai laboratori KEK in Giappone.

Ai collisionatori adronici invece la sezione d'urto di produzione dei quark b e c è molto maggiore rispetto alle B-factories e il processo di produzione dominante è la produzione inclusiva non risonante $b\bar{b}$:

$$\begin{array}{c} \sigma(p\bar{p} \to b\bar{b}X, \sqrt{s} = 1.96\ \text{TeV}) \sim 80\ \mu\text{b}, \\ \sigma(p\bar{p} \to b\bar{b}X, \sqrt{s} = 7\ \text{TeV}) \sim 250\ \mu\text{b}, \end{array} \tag{1.1}$$

dove $\sqrt{s}$ è l'energia nel centro di massa della collisione integrata su tutto l'angolo solido. In Figura 1.1 si mostra la sezione d'urto per processi $pp(\bar{p})$ in funzione dell'energia nel centro di massa. L'energia disponibile nel centro di massa $\sqrt{s}$, permette la produzione di tutte le specie di adroni contenenti b: i mesoni B^0 e B^+, ma anche B_s^0, B_c^+ insieme ad altri barioni. Tuttavia, la sezione d'urto di produzione $b\bar{b}$ ai collisionatori adronici è circa 3 ordini di grandezza più piccola della sezione d'urto totale inelastica protone-protone $\sigma(p\bar{p}\,\text{inelastic},\ \sqrt{\text{s}} = 7\,\text{TeV}) \sim 70\,\text{mb}$ [5], col risultato di un rapporto segnale rumore tipicamente molto piccola per i canali di maggiore interesse; ad esempio per il processo $B^0 \to K\pi$ si ha un rapporto segnale rumore dell'ordine di $\mathcal{O}(10^{-9})$. Simili argomentazioni sono valide anche per gli adroni c, sebbene siano caratterizzati da una sezione d'urto di produzione maggiore $\sigma(p\bar{p} \to c\bar{c}X,\ \sqrt{s} = 7\ \text{TeV}) \sim 6\,\text{mb}$ [6]. Dal 2015, l'energia nel centro di massa disponibile ad LHC aumenterà a $\sqrt{s} = 14\,\text{TeV}$, aumentando ulteriormente le sezioni d'urto di produzione [7, 8]:

$$\begin{array}{c} \sigma(p\bar{p} \to b\bar{b}X, \sqrt{s} = 14\ \text{TeV}) \sim 500\ \mu\text{b}, \\ \sigma(pp \to c\bar{c}X, \sqrt{s} = 14\ \text{TeV}) \sim 10\,\text{mb}, \\ \sigma(pp\,\text{inelastic}, \sqrt{s} = 14\ \text{TeV}) \sim 100\,\text{mb}. \end{array} \tag{1.2}$$

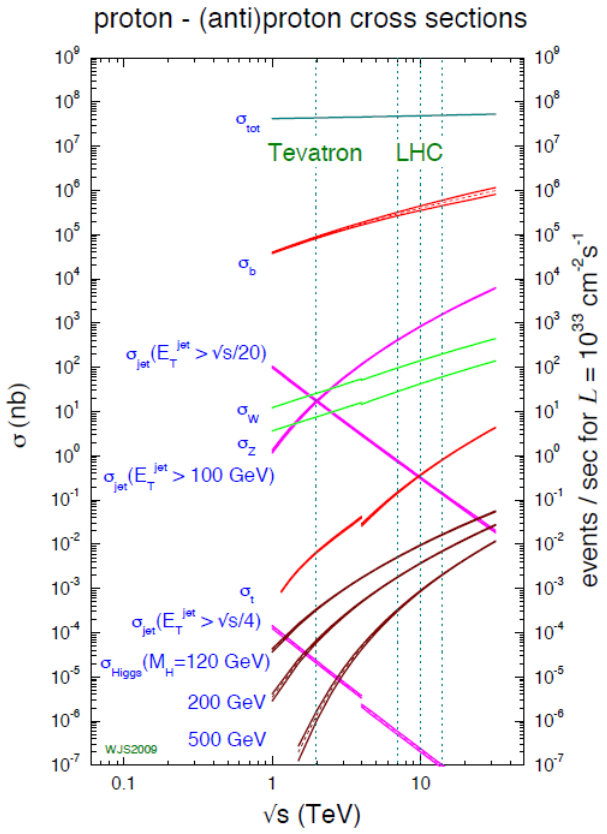


Figura 1.1: La sezione d'urto per $p\bar{p}$ and pp come funzione dell'energia nel centro di massa. Le discontinuità rappresentano le transizioni tra le sezioni d'urto misurate per processi $p\bar{p}$ e pp.

1.2 Il trigger e la tracciatura di quark pesanti alle macchine adroniche

Agli acceleratori adroni la complessità tipica di un evento e l'alta molteplicità di traccia rappresentano le difficoltà principali, specie quando vengono effettuate le misure di precisione richieste dalla fisica del flavour. Essendo la larghezza di banda del sistema di acquisizione finita, non è possibile memorizzare tutti i dati di ogni collisione, ma è necessario selezionare gli eventi ritenuti interessanti, in modo da sfruttare al meglio le risorse di calcolo e di memoria.

I decadimenti di adroni composti da quark pesanti, possono essere selezionati applicando specifiche richieste sulle variabili cinematiche, poiché risultano avere una distribuzione diversa rispetto al segnale di fondo. Il momento trasverso p_t degli adroni che contengono b è generalmente dell'ordine di $< p > \approx 5$ GeV/c, che risulta maggiore dei valori tipici del fondo dovuto ai quark leggeri. Tuttavia, la distribuzione di p_t per gli adroni con b decresce rapidamente, con il risultato che molti adroni

hanno un basso momento trasverso e i prodotti di decadimento hanno spesso $p_t < 1\,\mathrm{GeV/c}$. Quindi la selezione di particelle di basso momento risulta particolarmente importante. È possibile distinguere tra i differenti decadimenti degli adroni b tramite la loro vita media relativamente lunga $\tau \approx 1.5\,\mathrm{ps}$. Questo corrisponde a un vertice di decadimento (vertice secondario) spostato dal vertice primario di circa $c\tau \approx 500\,\mu\mathrm{m}$.

Argomenti simili risultano validi anche per gli adroni c. I mesoni neutri contenenti il quark c hanno vite medie $\tau \approx 0.41\,\mathrm{ps}$, $c\tau \approx 120\,\mu\mathrm{m}$, mentre i mesoni carichi hanno $\tau \approx 1\,\mathrm{ps}$, $c\tau \approx 310\,\mu\mathrm{m}$. Tuttavia, avendo questa famiglia di adroni una massa più piccola, i loro prodotti di decadimento hanno un p_t minore rispetto ai prodotti degli adroni b.

Il sistema utilizzato per la selezione in tempo reale degli eventi è generalmente chiamato *trigger*. Sistemi di trigger basati sulla ricostruzione di tracce in tempo reale sono necessari per effettuare studi di fisica del flavour ai collisionatori adronici, perché è necessario acquisire eventi molto rari immersi in un fondo prodotto con sezioni d'urto maggiori di alcuni ordini di grandezza.

Il principio fondamentale di funzionamento di un sistema di trigger e di tracciatura in tempo reale riguarda la possibilità di poter suddividere in un numero finito di celle la regione di interesse dello spazio dei parametri. La configurazione degli hit sui rivelatori dovuti ad una tipica traccia può essere precalcolata e memorizzata in una banca dati (*pattern bank*). Quindi gli hit provenienti dai rivelatori sono confrontati in tempo reale con i valori memorizzati nella pattern bank. Vari autori hanno proposto soluzioni diverse riguardo ai metodi usati per la pattern bank dati e per eseguire i confronti.

1.3 Sistemi di trigger basati su tracce a CDF

Il Collider Detector al Fermilab (CDF) [9] è stato un rivelatore costruito per lo studio delle interazioni $p\bar{p}$ ad alta energia, al collisionatore Tevatron nei pressi di Chicago. Nell'ultimo Run di presa dati, cominciato nel 2001, la macchina ha eseguito collisioni con energia nel centro di massa di $1.96\,\mathrm{TeV}$, con una luminosità di $2 \times 10^{32}\, cm^{-2} s^{-1}$. L'acceleratore è stato operativo per circa 30 anni, dal 1983 al 2011, quando ha terminato le operazioni a causa dei tagli del budget e la competizione dovuta ad LHC, che ha raggiunto una maggiore energia ed una più alta luminosità .

Il rivelatore di CDF aveva una struttura concentrica, a strati successivi in cui erano installati i rivelatori di traccia al silicio, una camera a fili, i calorimetri e i rivelatori a muoni, partendo dalla regione piu' vicina al punto di interazione e andando verso l'esterno.

Gran parte dei successi di fisica raggiunti da CDF sono stati possibili grazie alle migliorie apportate ai sistemi di tracciatura e di trigger. Infatti, oltre a migliorare le prestazioni dei singoli rivelatori, sono stati implementati algoritmi di ricostruzione di tracce in tempo reale per l'elaborazione delle decisioni di trigger. Inoltre lo stesso sistema di trigger venne riprogettato suddividendolo in 3 livelli: il Livello 1 e 2 erano completamente implementati in hardware, mentre il Livello 3 era gestito da software

eseguiti su una farm di PC. Gli algoritmi di tracciatura realtime sviluppati sono stati XFT (*eXtremely Fast Trigger*) per la camera a fili, ed SVT (*Silicon Vertex Trigger*) per il tracciatore al silicio.

1.3.1 Il Silicon Vertex Tracker

Il sistema di ricostruzione di tracce in tempo reale, applicato al tracciatore al silicio, era il *Silicon Vertex Tracker*, SVT [10]. L'idea originale è stata sviluppata all'inizio degli anni '90 [11], per poi essere implementata all'interno di CDF a partire dal 2001. Questo algoritmo ha permesso di ricostruire tracce sul piano trasverso alla linea di fascio, usando la terna di parametri composta da (p_t, ϕ, d), dove p_t è l'impulso trasverso, ϕ è l'angolo azimutale della traccia e d è il parametro d'impatto, la minima distanza di avvicinamento della traccia alla linea di fascio.

L'algoritmo era implementato in due fasi, una di confronto tra le tracce e i modelli (*pattern*) precalcolati, ed uno di *fitting* delle tracce, per il calcolo dei parametri. La prima fase era implementata utilizzando le Memorie Associative, un sistema di calcolo altamente parallelizzato. Lo schema di funzionamento della memoria associativa è mostrato in Figura 1.2. Al suo interno, sono memorizzate tutte le possibili tracce derivanti da eventi di fisica interessanti (*Patt0-Patt1...*), calcolate in precedenza attraverso una simulazione di alto livello. Ogni pattern è composto da una *word* ogni piano, cioè le coordinate di intersezione della traccia sul piano. La risoluzione di queste tracce è minore rispetto a quella offerta dal rivelatore.

Gli hit provenienti dal rivelatore sono divisi per piano di appartenenza e vengono confrontati in parallelo con tutti i possibili patterns, identificati nella figura su righe orizzontali (*Patt 0-Patt 3*). Se un hit risulta essere uguale alla corrispondente word del pattern, si alza un segnale, settando un flip-flop ad 1. L'AND logico di tali flip-flop indica se la traccia è compatibile con una di quelle precalcolate . Gli hit vengono successivamente passati allo stadio di fitting. Il calcolo dei parametri viene eseguito con un fit linearizzato sugli hit [12], utilizzando il metodo della minimizzazione del χ^2.

La selezione del pattern tramite l'AND logico dei flip-flop è stata modificato negli anni, per evitare di perdere eventi a causa dell'inefficienza del rivelatore. Infatti si è passati ad eseguire una logica maggioritaria, in cui si accetta una traccia anche se manca una corrispondenza.

Il sistema SVT lavorava suddividendo il rivelatore in 12 settori ϕ, ed eseguendo calcoli su ogni settore completamente in parallelo.

SVT è stato pensato per poter generare il trigger di secondo livello, sia a causa della velocità di lettura dei silici a CDF, sia perché il processo di estrazione dei parametri risulta non essere compatibile con i tempi richiesti dal trigger di primo livello. Questo sistema, è stato implementato totalmente via hardware, utilizzando per le memorie associative un chip costruito appositamente per questa applicazione, in tecnologia *full-custom ASIC* [13] (si veda la sezione 4.2), dato l'alto grado di parallelismo che non poteva essere supportato da altri dispositivi.

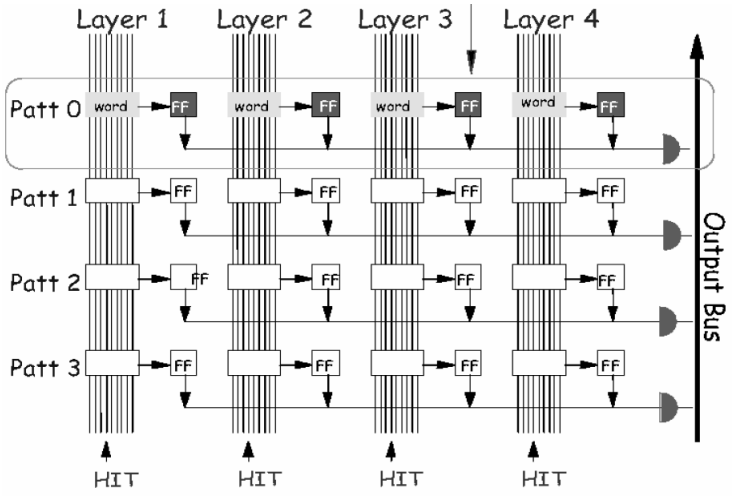
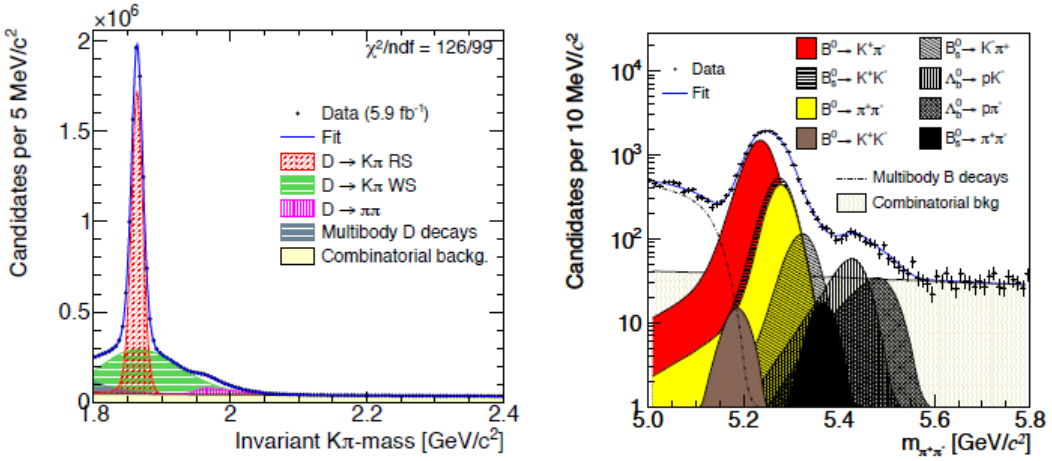


Figura 1.2: Schema di funzionamento di una memoria associativa.

Risultati ottenuti con SVT Il sistema SVT è stato capace di fornire misure dei parametri di impatto con una precisione ed un efficienza paragonabile all'analisi offline, permettendo di eseguire misure sulla fisica dei quark b e c con una precisione molto elevata, per un ambiente adronico. I campioni di adroni B e D raccolti sono stati molto numerosi, ad esempio per i canali $D^0 \to K\pi$ and $B^0 \to K\pi$ [14, 15] come si vede in Figura 1.3.



(a) (b)

Figura 1.3: Distribuzione di massa invariante per i candidati $k\pi$ derivanti dal decadimento del D^0 (a) e del B^0 (b), misurati a CDF.

Gli eventi raccolti hanno permesso di effettuare per la prima volta in un ambiente adronico misure di precisione essenziali per la fisica dei quark pesanti. Ad esempio CDF ha effettuato la prima osservazione della violazione di CP diretta nei decadimenti di mesoni B_s e di barioni che contengono un quark b [16]. Inoltre è stata

misurata l'asimmetria di CP in importanti decadimenti come $B_s^0 \to K^-\pi^+$, la cui rilevanza è data dalla possibilità di verificare la presenza di fisica oltre il modello standard senza una forte dipendenza dai modelli teorici. Recentemente e' stata anche riportata la prima evidenza per il decadimento $B_s^0 \to \pi^+\pi^-$ [17], che rappresenta una classe di decadimenti interessanti per la mancanza di valide predizioni teoriche quantitative. In aggiunta CDF ha effettuato anche la misura dell'angolo gamma del triangolo CKM usando per la prima volta decadimenti $B^\pm \to DK^\pm$ [18] raccolti ad un acceleratore adronico. Infine l'esperimento ha misurato per la prima volta l'ampiezza delle oscillazioni del B_s e la relativa frequenza [19], come mostrato in Figura 1.4, che possono essere usati per estrarre il modulo dell'elemento V_{ts} della matrice CKM.

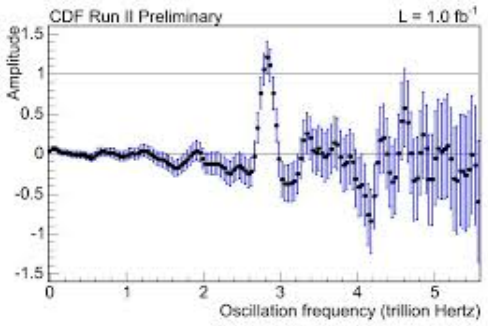
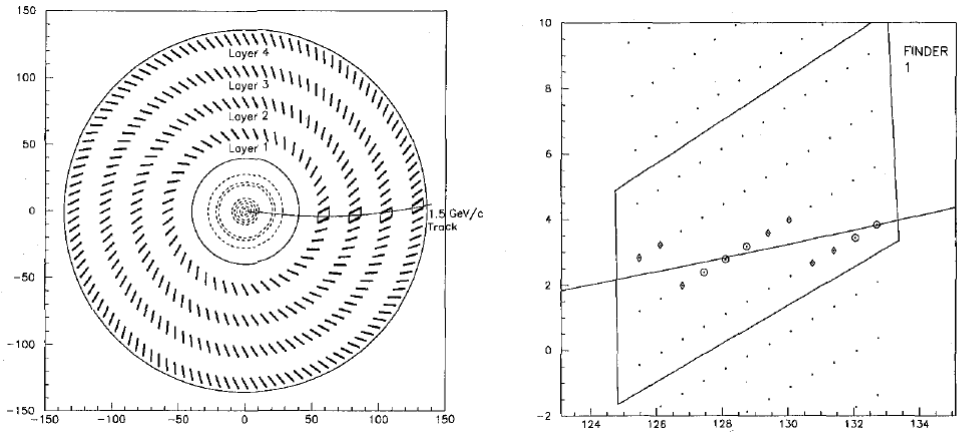


Figura 1.4: Valori dell'ampiezza di $B_s^0 - \bar{B}_s^0$, in funzione della frequenza di oscillazione, misurati a CDF. Il grafico è stato ottenuto combinando le misure derivate dai decadimenti adronici e semileptonici.

1.3.2 L'eXtremely Fast Trigger

L'eXtremely Fast Tracker (XFT) [20] è stato un sistema utilizzato per ricostruire le tracce cariche nella camera a fili di CDF, il *Central Outer Tracker*, COT (Figura 1.5(a)), utilizzato per la decisione del primo livello di trigger. Il rivelatore a fili era formato da 4 strati, composti da celle di 12 fili, inclinate di 35° rispetto alla direzione radiale. La ricostruzione delle tracce veniva effettuata unendo 4 celle adiacenti. XFT era in grado di misurare, con risoluzione sufficiente per il trigger di livello 1, il momento trasverso e l'angolo azimutale ϕ di particelle con alto impulso trasverso.

L'algoritmo di ricostruzione era diviso in due fasi distinte, la fase di ricerca (*Finder*) e di combinazione (*Linker*). Nella fase di Finder si va a confrontare la traccia trovata in ogni gruppo di 4 celle, per ciascuno strato, con un modello delle tracce interessanti precedentemente calcolato chiamato *mask*. Le mask sono formate da una combinazione di 12 fili, come si vede in Figura 1.5(b). Ogni Finder passa le informazioni trovate ad un sistema chiamata Linker, che cerca le tracce interessanti confrontando le diverse maschere individuate con una delle combinazioni calcolata



(a) (b)

Figura 1.5: (a) il rivelatore COT, attraversato da una traccia con impulso di $P_t = 1.5$ GeV/c. (b) un ingrandimento della traccia che attraverso un piano, identificando il gruppo di 4 celle su cui il Finder ricerca i segmenti di traccia. I punti marcati con i rombi, rappresentano quei fili che hanno generato un segnale ritardato rispetto a quelli cerchiati, a causa della diversa distanza dalla traccia. L'unione di questi punti crea una mask.

precedentemente. L'algoritmo ricercava le tracce confrontando le combinazioni di 4 maschere su 4 o di 3 su 3, per coprire eventuali inefficienze del rivelatore.

L'intero processo era eseguito in in 5.5 μs. Le informazioni ricavate da XFT erano poi combinate insieme a quelle provenienti dai calorimetri e dalle camere a muoni per elaborare il trigger di livello 1.

1.4 Il Fast TracK di Atlas a LHC

I buoni risultati ottenuti utilizzando l'algoritmo di SVT in un ambiente adronico, hanno portato a cercare di ricreare una soluzione simile utilizzabile ad LHC. Attualmente, l'esperimento ATLAS sta sviluppando un nuovo sistema di tracciatura in tempo reale, il processore *Fast Track* (FTK) [21], partendo dallo stesso principio di funzionamento di SVT. Questo dispositivo sarà utilizzato nel Run di LHC previsto per il 2015 per ricostruire le tracce nel tracciatore al silicio interno, per il secondo livello della catena di trigger, in un ambiente più estremo rispetto a quello di SVT. Questo è possibile usando le nuove tecnologie a disposizione che hanno permesso di migliorare il sistema SVT, in modo che possa lavorare con prestazione più elevate, in termini di velocità e quantità di dati da elaborare, a causa dell'ambiente più estremo rispetto a quello in cui lavorava CDF. Inoltre il chip che agisce da memoria associativa è stato ridisegnato, dando la possibilità di memorizzare un numero di pattern maggiore rispetto ai chip usati a CDF.

L'utilizzo di FTK può risultare necessario per la misura dei parametri di impatto a livello del trigger, uno strumento fondamentale per identificare i decadimenti provenienti dal quark b. Un altro contributo riguarderà i canali di fisica in cui particelle decadono in leptoni τ [22]. In questo modo sarà possibile, grazie ad FTK, la misura di importanti canali di decadimento, come $H \to b\bar{b}$ e $H \to \tau\tau$, oltre a migliorare la precisione nelle misure delle caratteristiche della particella di Higgs.

2

L'ambiente sperimentale al Large Hadron Collider

Il settore della fisica dei quark pesanti rappresenta ancora oggi una delle principali linee di ricerca per lo studio della fisica oltre il modello standard. L'esperimento LHCb al *Large Hadron Collider* (LHC), è stato progettato appositamente per lo studio della fisica dei quark pesanti (*bottom* e *charm*), in particolare per avere l'opportunità di effettuare misure precise dell'asimmetria CP in un ambiente adronico.

In questo Capitolo descriveremo in dettaglio le condizioni sperimentali, le caratteristiche dell'esperimento attuale e le prospettive di potenziamento per il futuro.

2.1 Il Large Hadron Collider

Il Large Hadron Collider (LHC) [23] è un acceleratore destinato allo studio delle collisioni tra protone-protone e tra ioni pesanti, situato ai laboratori del CERN di Ginevra, sul confine tra Svizzera e Francia. LHC è stato installato in un tunnel circolare lungo 27 Km, a circa 100 m di profondità, nella stessa cavità dove prima era alloggiato il *Large Electron Positron* (LEP). I protoni sono estratti da idrogeno gassoso e la loro energia viene gradualmente incrementata da una serie di macchine acceleratrici, di cui mostriamo uno schema in Figura 2.1. I protoni estratti vengono accelerati dal Linac 2 fino ad un'energia di 50 MeV, successivamente dal Booster fino ad un'energia di 1.4 GeV. Quindi i protoni entrano prima nel *Proton Synchrotron*(PS), raggiungendo un'energia 25 GeV, e poi nel *Super Proton Synchrotron* (SPS), raggiungendo i 450 GeV. Infine i protoni vengono immessi in LHC.

In LHC, i due fasci di protoni o di ioni, circolano in direzioni opposte in due cavità separate. I fasci vengono curvati utilizzando 1200 dipoli magnetici superconduttori lunghi 15 m, raffreddati a temperature di 1.9 K tramite 120 tonnellate di elio superfluido, i quali generano un campo magnetico di 8.3 T. Come mostrato in Figura 2.2, per accelerare i due fasci sono richieste due cavità separate, poste all'interno della stessa struttura di ferro. I dipoli magnetici sono costituiti da spire superconduttrici di Niobio-Titanio (NbTi) di diametro $d = 6 \div 7\,\mu$m, distanziate di 1 μm. Per realizzare tutti i dipoli magnetici sono stati impiegati 300000 Km di filo superconduttore.

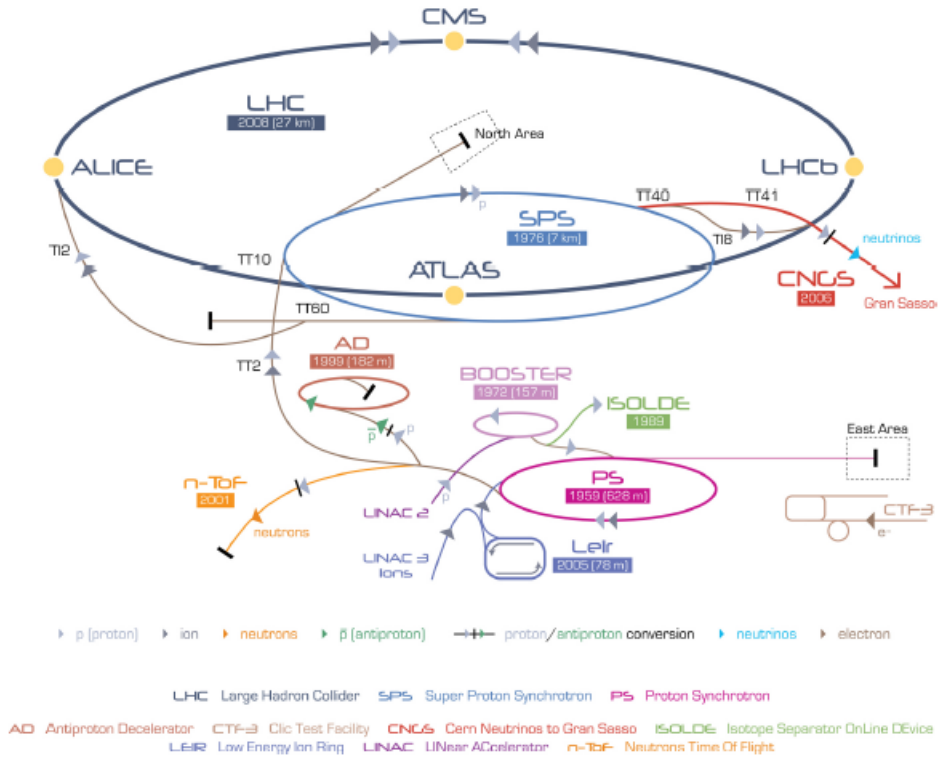


Figura 2.1: Schema del complesso degli acceleratori del CERN.

Usando il NbTi si riescono ad ottenere delle spire con concentrazioni di metallo omogenee, essenziali per non compromettere le qualità superconduttrici. Infatti disomogenità superiori all'1 % provocherebbero una non perfetta superconduttività, che può rovinare il magnete.

I fasci collidono in quattro punti lungo l'anello, dove sono installati i quattro maggiori esperimenti di LHC. Gli esperimenti ATLAS e CMS sono classificati come *general porpouse*, ovvero non hanno un obiettivo di fisica specifico, mentre ALICE ed LHCb sono stati costruiti rispettivamente per lo studio delle interazioni tra ioni pesanti e per la fisica dei quark pesanti. Altri due esperimenti minori sono stati installati lungo l'acceleratore: TOTEM, destinato per la misura della sezione d'urto totale pp e per lo studio della fisica in "avanti" ed LHCf per lo studio di particelle di interesse astrofisico.

I fasci di protoni non sono continui, ma suddivisi in pacchetti di protoni detti *bunch*. LHC prevede la possibilità di avere bunch distanti 25 ns l'uno dall'altro, il che corrisponde ad una frequenza di interazione massima di $1/(25\text{ ns}) = 40$ MHz. La configurazione dei bunch all'interno dell'anello può essere modificata a seconda dell'energia e della luminosità delle collisioni richieste [24]. La massima luminosità raggiungibile, secondo progetto, è di $\mathcal{L} = 10^{34}\,\text{cm}^{-2}\text{s}^{-1}$, ad un'energia nel centro di massa di $E_{cm} = 14\,\text{TeV}$. Tuttavia fino ai Run del 2012, le interazioni sono avvenute

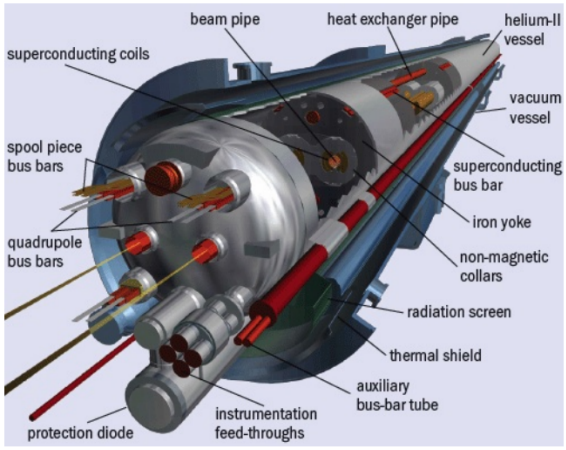


Figura 2.2: Rappresentazione di una sezione di un dipolo magnetico ad LHC.

ad un'energia di 8 TeV. Si prevede di raggiungere l'energia massima di progetto con l'upgrade di LHC previsto per il 2015. Un riassunto dei principali parametri di LHC utilizzati per le interazioni *pp* fino al 2012 viene mostrato in Tabella 2.1.

Parametri	**Progetto**	**2010**	**2011**	**2012**
Luminosità di picco ($\mathrm{cm^{-2}s^{-1}}$)	$1 \cdot 10^{34}$	$2.1 \cdot 10^{32}$	$3.7 \cdot 10^{33}$	$7.7 \cdot 10^{33}$
Energia protoni (TeV)	7	3.5	3.5	4
bunch per fascio	2808	368	1308	1380
Intensità dei bunch	$1.15 \cdot 10^{11}$	$1.2 \cdot 10^{11}$	$1.5 \cdot 10^{11}$	$1.6 \cdot 10^{11}$
Distanza tra i bunch (ns)	25	150	50	25-50

Tabella 2.1: Principali parametri di LHC durante i run svolti nel 2010, 2011 e 2012, confrontati con i valori definiti dal progetto.

2.2 Il rivelatore di LHCb

LHCb è un esperimento dedicato allo studio della fisica dei quark pesanti [25]. Lo scopo principale è quello di ricercare evidenze indirette della fisica oltre il modello standard attraverso lo studio delle transizioni con quark pesanti e attraverso lo studio della violazione di CP e dei decadimenti rari degli adroni b e c.

LHCb è costruito come uno spettrometro a singolo braccio, con una copertura angolare intorno al fascio da 10 mrad a 300 (250) mrad sul piano di curvatura (non curvatura), corrispondente ad un intervallo di pseudorapidità[1] η compresa tra 1.8

[1]La pseudorapidità è definita come $\eta = -ln(tan(\frac{\theta}{2}))$, dove θ è un angolo che si misura a partire dalla linea di fascio.

e 4.9. La copertura angolare dell'esperimento è giustificata dal fatto che la sezione d'urto di produzione per coppie di quark $b\bar{b}$ è una funzione dell'angolo di produzione dei due quark. La sezione d'urto decresce rapidamente all'aumentare dell'angolo di produzione, il che risulta in coppie $b\bar{b}$ prevalentemente prodotte lungo la linea del fascio, come si può vedere dalla Figura 2.3 [26, 27].

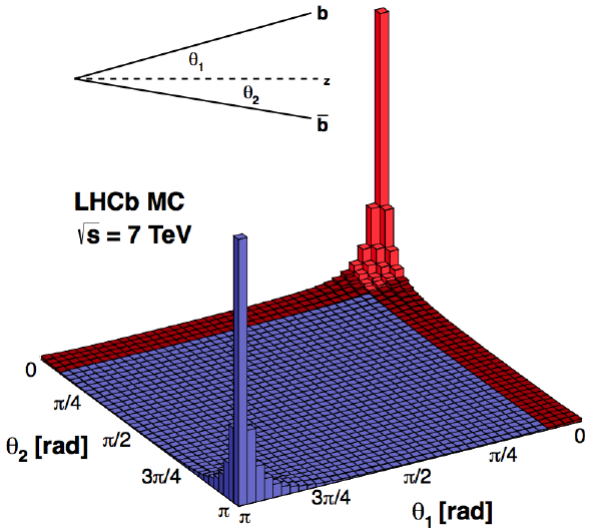


Figura 2.3: Correlazione angolare tra i quark b e $\bar{b}$, nel processo di produzione di coppie, simulata utilizzando il generatore di eventi PYTHIA.

La configurazione dei rivelatori in LHCb è mostrata in Figura 2.4. Il sistema di coordinate adottato è destrorso, in cui l'asse x è diretto verso il centro dell'acceleratore, l'asse y verso l'alto e l'asse z lungo la direzione del fascio.

LHCb è formato da un sistema di tracciatura per particelle cariche e da un sistema di identificazione di particelle. Il sistema di tracciatura include un dipolo magnetico non superconduttore, un rivelatore di vertice, e due serie di tracciatori posti subito prima e subito dopo il campo magnetico (TT e T1-T2-T3). Il sistema di identificazione di particelle è formato da due rivelatori Cherenkov (*Ring Imagine Cherenkov*, RICH), un calorimetro elettromagnetico ed uno adronico, ed un sistema di rivelazione dei muoni.

Ad ogni collisione dei fasci, il rivelatore è interessato da un alto numero di particelle provenienti dalle molteplici interazioni primarie che possono avvenire. Queste rendono difficile il compito dei sistemi di processamento degli eventi che lavorano in tempo reale, oltre a causare danni ai rivelatori dovuti alla radiazione. Nella configurazione attuale i dati provenienti dai rivelatori possono essere letti ad una frequenza massima di 1 MHz. Quindi, per diminuire la probabilità di avere un alto numero di interazioni ad ogni collisione, è stato implementato un metodo che riduce la luminosità nominale dei fasci di LHC ad un valore di $\mathcal{L} = 4 \cdot 10^{32}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$, nel punto di intersezione di LHCb [28]. Questo valore massimizza la probabilità di avere un

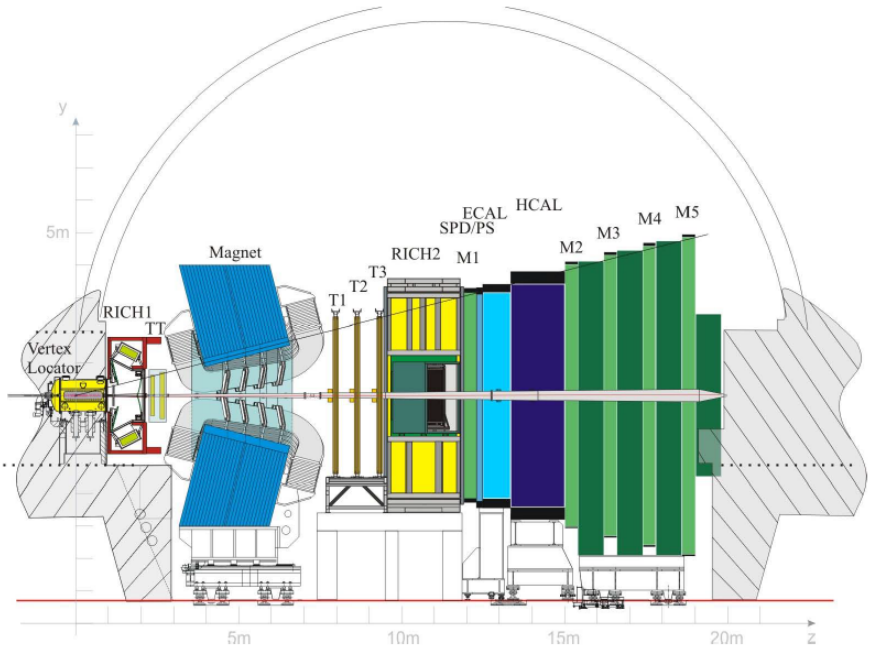


Figura 2.4: Vista trasverale dell'esperimento LHCb. La linea di fascio è lungo l'asse z.

unico vertice primario per ogni collisione dei bunch, come riportato in Figura 2.5. La luminosità viene regolata modificando la distanza dei fasci sul piano orizzontale. La corrente dei fasci diminuisce in modo esponenziale durante le collisioni a causa dei protoni che finiscono fuori dall'orbita nominale ad ogni giro e vengono persi all'interno dell'anello. Quindi, per mantenere una luminosità il più possibile costante, la distanza dei fasci viene progressivamente ridotta. La dimensione trasversa dei fasci al punto di interazione è di circa 160 μm e la distanza tra i centri varia da 100 μm a 40 μm. In Figura 2.6 mostriamo gli effetti del controllo della luminosità durante una tipica acquisizione dati effettuata fino al 2012. In queste condizioni, il numero medio di vertici primari delle interazioni pp per ogni collisione di fasci si riduce ad 1 [29].

A causa della geometria in avanti dei rivelatori in LHCb, solo le particelle prodotte ad alta pseudorapidità entrano nell'accettanza dell'esperimento. Le particelle prodotte all'indietro attraversano solamente il primo rivelatore di vertice, e sono utilizzate nel sistema di trigger per determinare la posizione longitudinale ed il numero dei vertici primari delle interazioni.

2.2.1 I rivelatori di traccia

Il sistema di tracciatura fornisce una ricostruzione della geometria delle tracce cariche, utilizzata ad esempio per il calcolo della carica e del momento delle particelle,

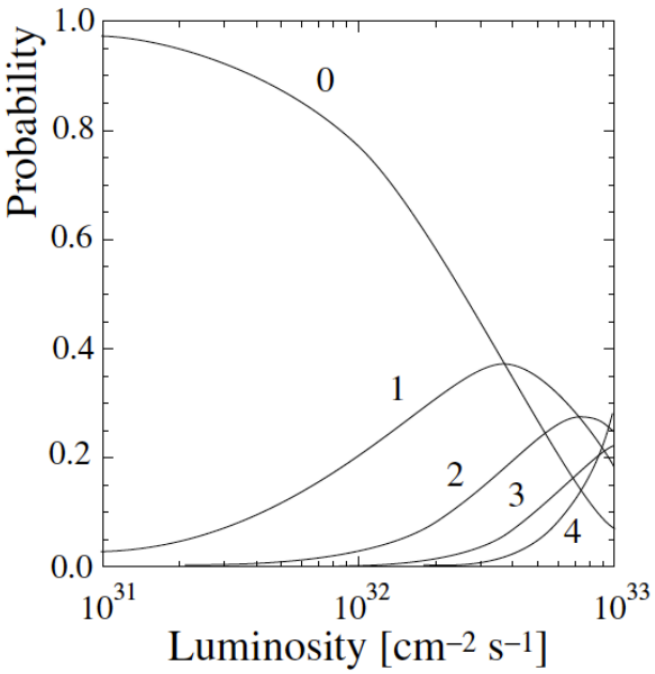


Figura 2.5: Probabilità di avere 0, 1, 2, 3 e 4 vertici primari in una collisione tra bunch, in funzione della luminosità istantanea. Questo grafico è stato ottenuto mediante una simulazione, in cui sono stati assunti bunch separati di 25 ns e una sezione d'urto *pp* inelastica di 80 mb. Il valore di luminosità istantanea che massimizza la probabilità di avere un unico vertice primario per ogni collisione tra bunch è di $4 \cdot 10^3 2\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$.

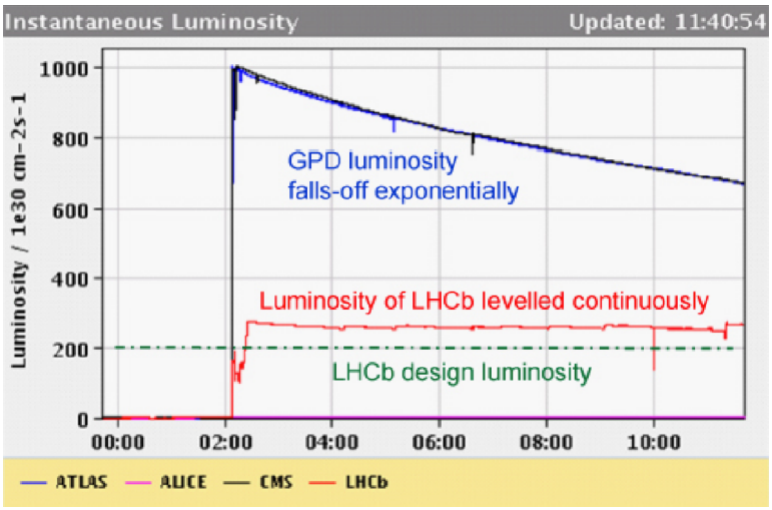


Figura 2.6: Confronto tra la luminosità istantanea di LHC e di LHCb durante la presa dati del Run tra il 2010 e il 2012.

misurando la posizione del vertice e la curvatura causata dal campo del dipolo magnetico.

Il dipolo magnetico

Il campo magnetico di LHCb è generato da un dipolo magnetico non superconduttore, prevalentemente diretto lungo l'asse y. Il campo magnetico integrato generato è di circa 4 T·m, nella regione compresa tra $0 < z < 10$m, dove sono disposti i rivelatori di traccia. Il campo magnetico permette la misura del momento delle particelle con una risoluzione del $0.4 \div 0.6$ % su un intervallo di momento compreso tra $5 - 100$ GeV/c. Le disomogenità del campo magnetico sul piano xy interessano un'area di circa 1 m^2 e sono misurate con la precisione dell' 1%. Il campo magnetico integrato nella regione centrale tra $z = 3$ m e $z = 8$ m vale circa $\int Bdl = 3.615 \cdot$Tm, mentre nella regione del VELO $\int Bdl = 0.1159$ Tm. A causa dell'elevata sensibilità al campo magnetico, i rivelatori RICH sono schermati, riducendo il campo magnetico interno ad un valore di $20 \cdot 10^{-4}$ T.

Il dipolo magnetico è formato da due avvolgimenti identici, ognuno formato da 15 piatti di alluminio carbonato spesso 10 cm. Le spire, pesanti in totale 54 tonnellate, sono installate simmetricamente intorno alla linea di fascio. Una vista del dipolo magnetico è mostrata in Figura 2.7(a). Il Dipolo ha dimensioni di 11 m $\times$ 8 m $\times$ 5 m. Il magnete dissipa una potenza elettrica di 4.2 MW, con una corrente nominale di 5.85 kA, supportando correnti fino 6.6 kA. Attualmente il campo magnetico viene periodicamente invertito, per ridurre gli effetti di sistematica nelle misure dell'asimmetria CP.

Per fornire una risoluzione elevata nella ricostruzione del momento delle particelle, l'intensità del campo magnetico deve essere conosciuta con grande precisione. Per questo motivo, la calibrazione è avvenuta con l'utilizzo di 180 sonde Hall, che hanno fornito una precisione relativa di $4 \cdot 10^{-4}$ nell'intero volume della tracciatura. In Figura 2.7(b) viene mostrata la misura della componente B_y del campo magnetico.

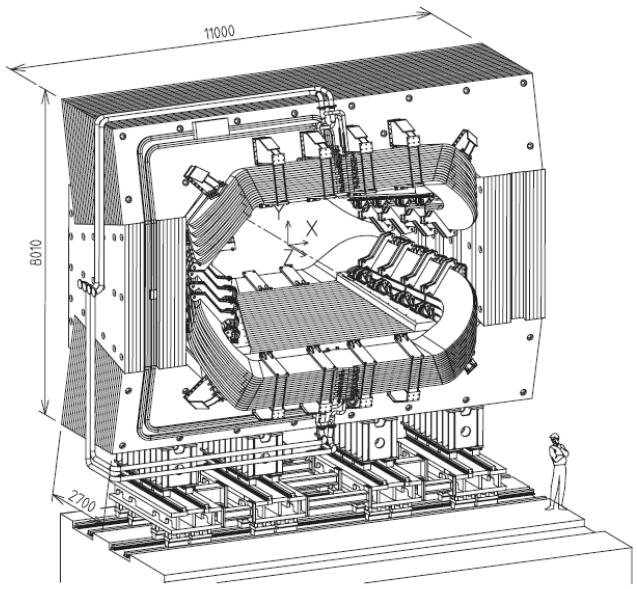
Vertex locator detector

Il rivelatore di vertice *Vertex Locator* (VELO) [30], misura la traiettoria delle particelle cariche nella regione più vicina al punto di interazione. Il suo scopo è di misurare i vertici primari e lo spostamento dei vertici secondari, che rappresentano una segnatura dei decadimenti di quark pesanti, con una risoluzione spaziale più elevata della tipica lunghezza di decadimento degli adroni b e c in LHCb ($c\tau \sim 0.01 \div 1\,\text{cm}$). Il VELO ricopre quindi un ruolo fondamentale nella discriminazione del segnale derivante dai quark pesanti dagli eventi di fondo, specialmente ai livelli superiori del trigger.

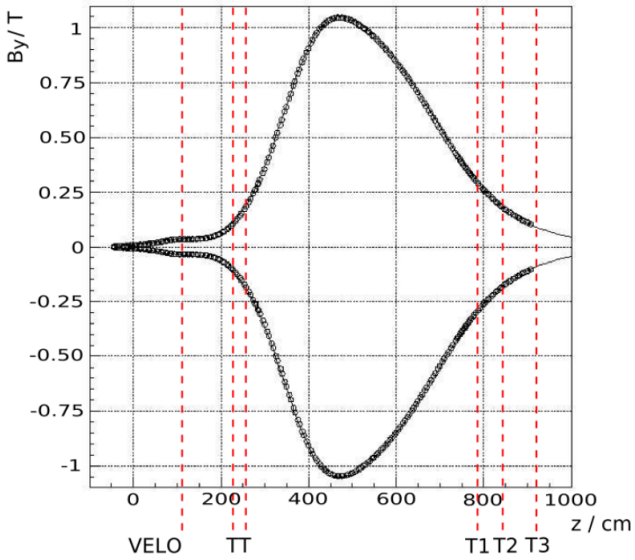
Il VELO è formato da 21 stazioni di forma circolare, istallate lungo l'asse del fascio all'interno della *beam pipe*, da entrambi i lati del del punto di interazione nominale lungo l'asse z. Le stazioni poste nella regione $z > 0$ forniscono la misura ad alta precisione dei vertici, mentre le stazioni a $z < 0$ sono utilizzate come sistema di veto per gli eventi multipli di $pile - up$, che comporta la misura della posizione dei vertici lungo l'asse z e la misura delle particelle totali prodotte all'indietro. Ogni piano del VELO è costituito da due tipi di sensori al silicio, ovvero strip radiali r e azimutali ϕ, per la misura delle coordinate delle particelle. Ogni stazione è divisa in due metà mobili, chiamate moduli, come possiamo vedere in Figura 2.9. Questo permette di regolare la distanza dei sensori di silicio dal fascio, affinché non si danneggino nella fase di iniezione dei fasci. In questa fase, in cui il VELO viene detto aperto, la distanza tra i moduli è di 30 mm dall'asse del fascio, mentre durante la presa dati con i fasci stabili, in configurazione chiusa, la distanza si riduce a 5 mm. Ogni piano nella parte delle z positive è composto da sensori di tipo r e di tipo ϕ, mentre le stazioni a z negativo sono costituite solamente da sensori di tipo r.

Entrambe le strip r e ϕ sono centrate intorno alla posizione nominale del fascio, coprendo una regione in r compresa tra $8\,\text{mm} \div 14\,\text{mm}$, suddivisi in 4 settori da 45°, per ridurre l'occupazione. La geometria dei sensori è rappresenta in Figura 2.8. I sensori r consistono in strip concentriche semicircolari, con passo che si incrementa da $38\,\mu\text{m}$ fino a $102\,\mu\text{m}$. I sensori ϕ sono suddivisi in due regioni concentriche: quella interna copre una regione $r = 8 \div 17.25\text{mm}$ e quella esterna $r = 17.25 \div 42\text{mm}$ con un passo che si incrementa linearmente dal centro. Questo design è stato progettato per migliorare la risoluzione sulle tracce e per distinguere meglio il segnale dal rumore.

Per minimizzare la quantità di materiale incontrata dalle particelle che attraversano il rivelatore di veritce, il sensore deve operare nel vuoto. Questo è separato dalla beam pipe attraverso uno strato di alluminio, in modo da prevenire eventuali contaminazioni della beam-pipe dovute ai moduli. Ogni modulo del VELO è schermato dalle radiofrequenze, utilizzando una copertura chiamata *RF-foil*, come si vede



(a) Dipolo Magnetico



(b) Componente B_y del campo magnetico di LHCb, in funzione della coordinata z.

Figura 2.7: (a) una vista del dipolo magnetico utilizzato in LHCb. (b) la misura effettuata per la componente B_y del campo magnetico del dipolo, nei due stati di polarizzazione.

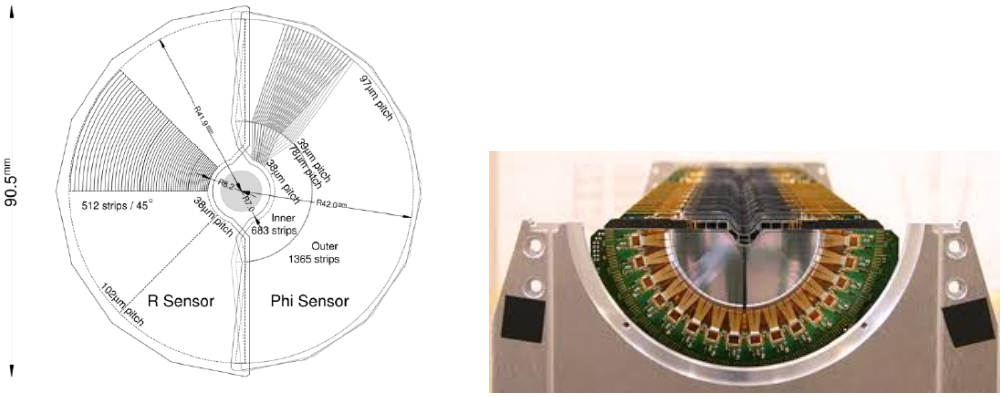


Figura 2.8: A sinistra: una vista della geometria del VELO e dei sensori r e ϕ. Sono riportati i sensori ϕ appartenenti a due moduli adiacenti, così da rappresentare due moduli stereo. A destra: i rivelatori al silicio del VELO.

in Figura 2.10. La struttura è corrugata, così da permettere la sovrapposizione dei moduli del VELO nella configurazione chiusa.

Le prestazioni del VELO sono state studiate tramite test su fascio. La risoluzione su un singolo hit è fortemente correlata con il passo dei sensori e l'angolo della traccia rispetto alla linea di fascio. La risoluzione sugli hit, varia tra $\approx 10\,\mu$m e $\approx 25\,\mu$m a seconda del passo delle strip.

Tracciatori al silicio: Tracker Turicensis (TT) e Inner Tracker (IT)

I tracciatori al silicio (ST) [31] consistono in un Tracciatore Turicense (TT) e il rivelatore Inner Tracker (IT). Entrambi utilizzano sensori a microstrip di silicio, distanziate di $\approx 200\,\mu$m. Tali rivelatori sono utilizzati per la misura della curvatura della traccia.

Le stazioni TT, situate nella regione antestante il dipolo magnetico, hanno un'accettanza di $\sim$ 150-200 mrad sul piano di curvatura, e di $\sim$ 40-60 mrad sul piano yz. Il rivelatore è stato progettato per la ricostruzione delle tracce a basso momento che vengono deviate dal campo magnetico al di fuori dell'accettanza del rivelatore interno, mentre l'IT ricostruisce le tracce che oltrepassano il campo magnetico mantenendosi vicine alla linea di fascio. Il TT è formato da un'unica stazione di rivelatori, mentre l'IT ne ha 3. Ciascuna stazione degli ST è suddivisa in 4 piani, in cui il primo e l'ultimo hanno le strips verticali, e quelli centrali hanno le strips orientate di +5° e -5° rispetto alla verticale. Le coordinate misurate sono x-u-v-x. In Figura 2.11 vediamo due rappresentazioni del TT e dell'IT sul piano xy. Ogni piano del TT è suddiviso in due moduli verticali, ognuno formato da 7 sensori al silicio, raggruppati in 3 settori di lettura (K, L, M).
Ogni piano dell'IT è formato da 4 moduli posizionati intorno alla linea del fascio, due laterali e due assiali. I moduli assiali sono formati da una singola unità di sen-

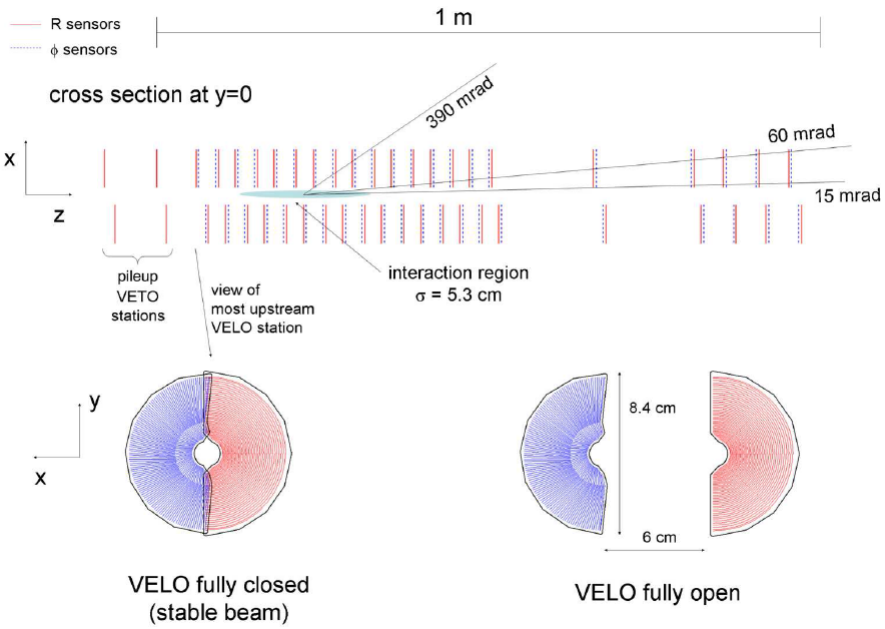


Figura 2.9: Struttura del rivelatore VELO e di una stazione di tracciatura, nella configurazione aperta e chiusa.

sori, mentre quelle laterali sono formate da due unità ciascuna. La risoluzione su un singolo hit è di $\approx 50\ \mu$m. Il TT ha una copertura in accettanza pari a quella massima dell'esperimento, mentre l'IT ne copre solamente il 2%, che equivale al 20% del totale delle tracce in LHCb.

L'Outer Tracker

L'*Outer Tracker*, OT [32], è l'ultimo rivelatore di traccia di LHCb, e ricopre una regione di accettanza fino a 300 (250) mrad nel piano di curvatura (non curvatura). L'OT è formato da camere a *straw*, grazie al quale le tracce vengono ricostruite con una risoluzione spaziale di $\approx 200\ \mu$m in un intervallo di momento molto ampio, in quanto l'OT è usato per la ricostruzione di tracce per la regione di accettanza non coperta dall'IT. L'OT ha la stessa struttura dell'IT, come mostrato in Figura 2.12, ed è posto intorno allo stesso IT. È quindi formato da 3 stazioni di 4 piani, con gli straw tube orientati secondo la configurazione di coordinate *(x,u,v,x)*. Ogni piano del rivelatore consiste in una matrice di moduli ognuno contenente due piani da 64 tubi, disposti in modo da garantire la massima sovrapposizione tra due piani adiacenti. Le camere a straw sono riempite con una mistura di $ArCO_2$ in un rapporto di 70 : 30, che assicura un tempo di deriva al di sotto di 50 ns, corrispondente a due intersezioni dei fasci.

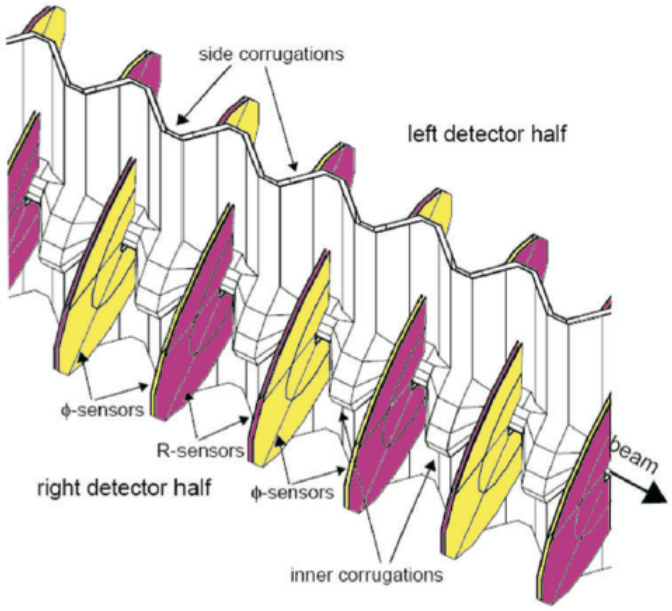


Figura 2.10: Struttura dei RF foil nela configurazione col VELO chiuso.

Beetle Chip

Tutti i rivelatori al silicio utilizzano lo stesso chip di *front-end*. Il chip, chiamato Beetle Chip [33], è un chip custom, resistente alle radiazioni, progettato in tecnologia CMOS a 0.12 μm con un front end analogico. Le specifiche temporali, come la frequenza di campionamento, sono state progettate per essere compatibili con la frequenza delle collisioni di 40 MHz di LHC. Ciascun Beetle chip ha 128 canali, ognuno possiede un preamplificatore, un formatore ed una pipeline analogica di 160 stadi, disegnata per la latenza di 4 μs del Livello 0 di trigger. I dati vengono fatti uscire attraverso 4 uscite comandate da multiplexer a 32 ingressi, con una frequenza di 40 MHz. Con queste specifiche, un evento può essere letto in 900 ns. Il funzionamento del chip in ambiente altamente radioattivo è garantito per almeno 5 anni, sottoposto ad una dose annuale di 2 MRad. La protezione contro il cambiamento dei registri interni dovuti alle radiazioni (*single event upset*) è implementata attraverso una logica di ridondanza. La programmazione dei registri interni e il controllo delle impostazioni avviene tramite il protocollo I2C. Le funzioni interne possono essere testate tramite un impulsatore interno, di cui può essere regolata l'altezza dell'impulso.

2.2.2 Rivelatori per riconoscimento di particelle

L'identificazione di particelle ricopre un ruolo importante nello studio dei decadimenti della fisica del flavour. In particolare i rivelatori Cherenkov devono essere

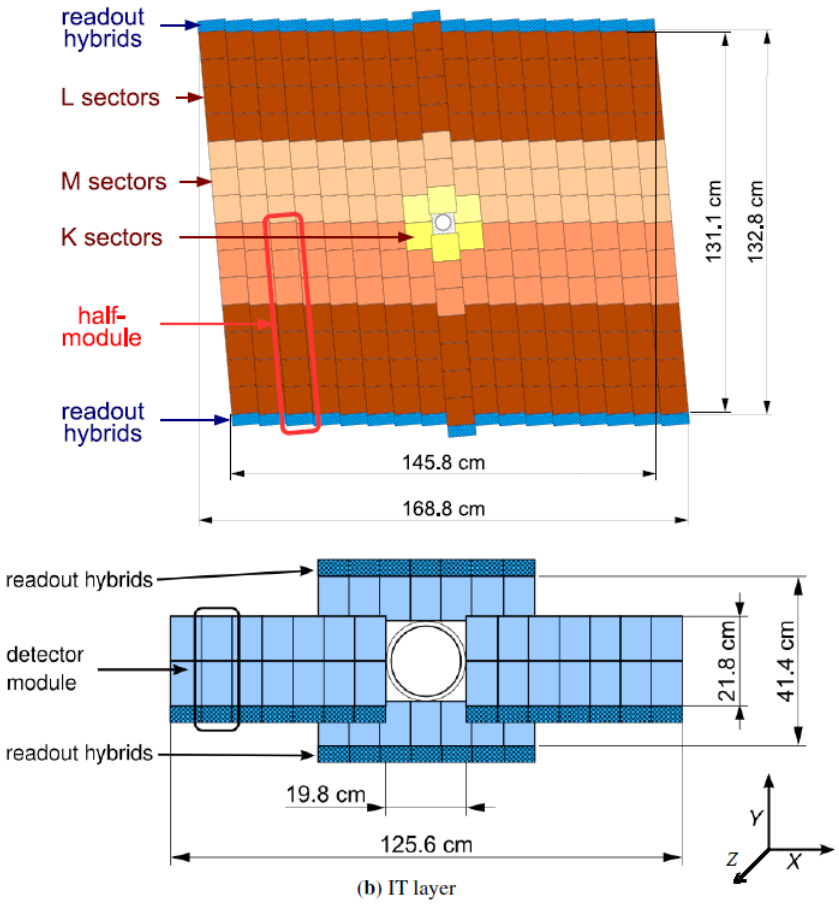
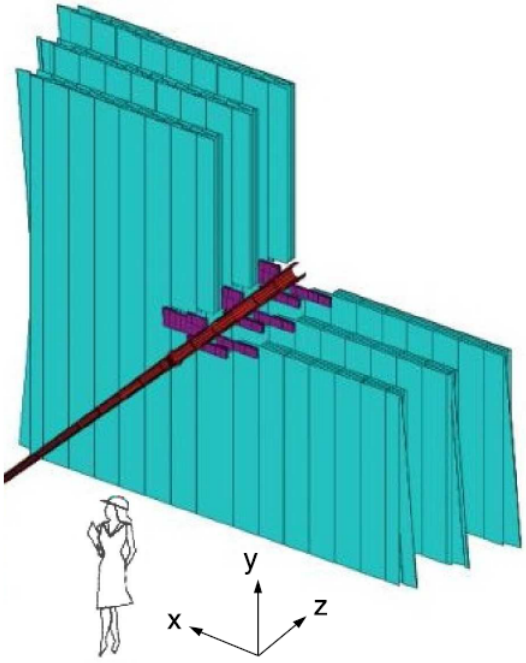


Figura 2.11: Vista di un piano di coordinata v per il TT e di un piano x dell'IT.

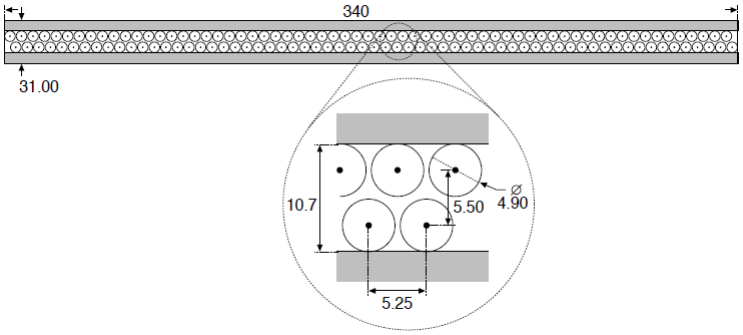
capaci di separare i kaoni dai pioni, in modo da aiutare nell'identificazione dei processi di interesse fisico dal fondo. I calorimetri permettono invece l'identificazione di elettroni, fotoni e adroni, mentre i muoni sono riconosciuti dalle camere a muoni.

Rivelatori Cherenkov

Due rivelatori ad anelli Cherenkov, il RICH1 e il RICH2 [34], permettono l'identificazione di particelle cariche con un momento compreso tra 1 e 100 GeV/c. In particolare, il RICH1 è sensibile ad un intervallo di momento tra 1 e 60 GeV/c, mentre il RICH2 è ottimizzato per valori di momento tra 15 e 100 GeV/c. Questa differenza è dovuta ai diversi radiatori utilizzati: aerogel e C_4F_{10} per il RICH1 e CF_4 per il RICH2. La geometria dei RICH è mostrata in Figura 2.14. Ciascun rivelatore è dotato di due tipi di specchio: uno specchio sferico per identificare l'anello Cherenkov, e una serie di specchi piatti per guidare i fotoni sugli *Hybrid Photon Detector* (HPD), posti fuori dall'accettanza dei rivelatori che sono sensibili a fotoni con lun-



(a)



(b)

Figura 2.12: Vista di una stazione dell'OT lungo l'asse del fascio (a) e di un modulo di straw tube (b).

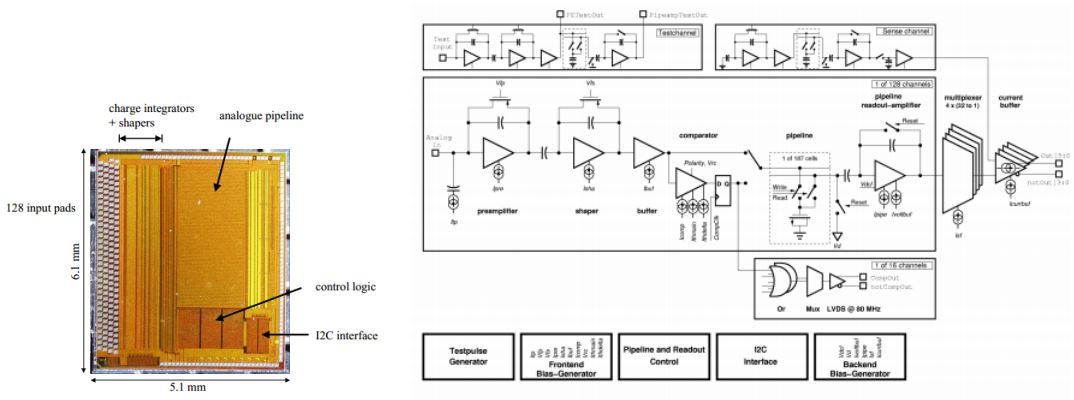


Figura 2.13: Il Beetle Chip: una vista del chip e uno schema a blocchi delle principali operazioni.

ghezze d'onda compresa tra 200 e 600 nm. I due RICH sono entrambi schermati dal campo magnetico per permettere il corretto funzionamento degli HPD. Il RICH1 è posto nella regione che precede il magnete, ricoprendo l'intera accettanza dei LHCb, mentre il RICH2 è situato oltre le stazioni di tracciatura dell'OT, con un'acettanza compresa tra 10 mrad e ≈ 110 mrad. L'efficienza di separazione di π-K è del 90%, per particelle di momento superiore a 30 GeV/c.

Calorimetri

Il sistema di calorimetri è utilizzato per fornire informazioni per l'elaborazione del Livello 0 di trigger, distinguendo elettroni, fotoni e adroni e fornendo insieme una rozza misura dell'energia e della posizione.

Il sistema di calorimetria è composto da un calorimetro elettromagnetico (ECAL) [35] ed uno adronico (HCAL) [36]. Sono posizionati tra le prime due camere a muoni, coprendo una regione angolare tra 25 e 300 (250) mrad sul piano di curvatura (non curvatura). Di fronte all'ECAL sono posti due ulterioi sottorivelatori: un *preshower* e una matrice di scintillatori, separati da uno spessore di piombo per la conversione di fotoni-elettroni. Questi due sono usati per rigettare pioni carichi e neutri nel trigger di elettroni di Livello 0, in modo da diminuire la contaminazione [37]. I pioni carichi vengono identificati guardando lo sviluppo longitudinale dello sciame nel preshower. Lo strato di piombo è spesso 15 mm, equivalenti a circa 2.5 lunghezze di radiazione per gli elettroni, che generano un segnale molto superiore rispetto a quello dei pioni carichi. Gli scintillatori sono usati per identificare e rigettare i pioni neutri.

I calorimetri sono divisi in 4 quadranti disposti intorno alla linea di fascio. La segmentazione laterale varia a seconda della distanza dal fascio, e risulta più grande nel calorimetro adronico, come mostrato in Figura 2.15. Lo spessore totale dell'ECAL corrisponde a 25 lunghezze di radiazione, garantendo il contenimento quasi totale

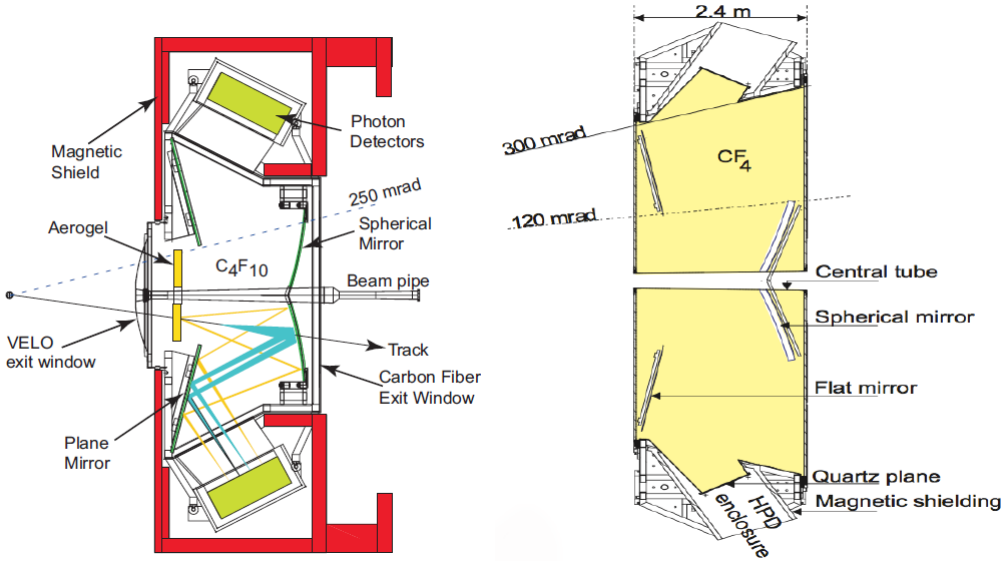


Figura 2.14: Geometria dei rivelatori Cherenkov, RICH1 a sinistra e RICH2 a destra. Nel primo le ottiche sono disposte verticalmente, nel secondo orizzontalmente.

dello sciame. Lo spessore del HCAL risulta invece essere di circa 5.6 lunghezze di interazione. I due calorimetri vengono letti allo stesso modo: la luce di scintillazione viene trasmessa a fotomoltiplicatori con fibre ottiche di tipo *wavelenght-shifter*. L'ECAL è composto da un alternanza di strati di materiale scintillante spessi 4 mm e fogli di piombo da 2 mm, mentre l'HCAL ha il materiale scintillante spesso 4 mm alternato a spessori di ferro di 16 mm.

Le risoluzioni in energia sono di $\sigma_E/E(\text{GeV}) \approx 10\%/\sqrt{E(\text{GeV})}$ per l'ECAL e di $\sigma_E/E(\text{GeV}) \approx 70\%/\sqrt{E(\text{GeV})}$ per l'HCAL.

Rivelatori di muoni

I rivelatori di muoni [38] riescono ad identificare e a misurare il momento trasverso dei muoni penetranti, per le decisioni del trigger di basso ed alto livello, con una risoluzione paragonabile a quella della ricostruzione offline. Sono formati da 5 stazioni rettangolari, identificate con M1÷M5, posizionate lungo la linea di fascio, ricoprendo un accettanza angolare da 20 (16) a 306 (258) mrad nel piano di curvatura (non curvatura). La stazione M1 situata tra il RICH2 e i calorimetri, serve per migliorare la misura del momento trasverso dei muoni rivelati nelle stazioni successive. Le stazioni M2-M5 sono situate dopo i calorimetri. Tra una stazione e l'altra è presente uno spessore di ferro spesso 80 cm, equivalente a circa 20 lunghezze di interazione, per selezionare i muoni più penetranti. Per attraversare tutte le stazioni, un muone deve avere un'energia minima di 6 GeV/c. Le stazioni sono state costruite con una geometria proiettiva che parte dal nominale punto di interazione. Ogni stazione è

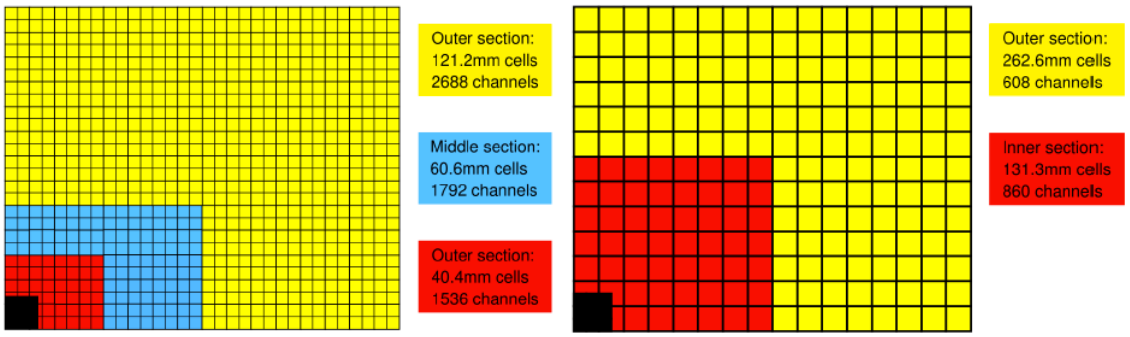


Figura 2.15: Segmentazione laterale del preshower, della matrice di scintillatori e dell'ECAL a sinistra, e dell' HCAL a destra. Viene mostrato solamente un quadrante dei calorimetri. L'area in nero rappresenta la distanza del rivelatore dalla linea di fascio.

suddivisa in quattro quadranti, disposti intorno alla linea del fascio. Come mostrato in Figura 2.16, ciascun quadrante è diviso in quattro regioni, chiamate R1-R4, identificate da uno spessore crescente allontanandosi dalla linea di fascio. La rivelazione dei muoni è effettuata usando due diverse tecnologie: rivelatori *triple gas electron multiplier* e camere proporzionali multifilo. La prima è utilizzata nella regione più interna (R1) della prima stazione, dove l'alta densità di particelle richiede un rivelatore con un elevata tolleranza alle radiazioni; le seconda è usata nel resto delle camere. Il gas usato è una mistura di Ar, CO_2, e CF_4 in proporzioni diverse. Le prime tre stazioni (M1-M3) contribuiscono alla misura del momento trasverso, mentre le ultime due (M4-M5) mostrano la presenza di particelle che hanno superato gli spessori di materiale assorbente. La risoluzione media sul momento trasverso è del 20%, calcolata con la ricostruzione di un muone isolato.

2.2.3 Il trigger di LHCb

Il trigger di LHCb [39] è stato concepito per effettuare una selezione efficiente dei decadimenti dei quark pesanti, rispetto al grande fondo dovuto ai quark leggeri, passando dai 40 MHz delle collisioni a 5kHz, che e' la frequenza massima con cui e' possibile memorizzare gli eventi. Solamente una piccola frazione degli eventi, circa 15 kHz, contiene decadimenti derivanti dagli adroni b dentro l'accettanza di LHCb. La frequenza di decadimenti di adroni b interessanti è più bassa, equivalente a pochi Hz. Il corrispondente valore per adroni c è circa 20 volte maggiore. Diventa quindi cruciale per il trigger riuscire ad eliminare il fondo fin dai primi stadi dell'acquisizione.

Il trigger di LHCb è suddiviso in due stadi: il Livello 0 (L0) e il trigger di alto livello (*High Level Trigger*, HLT). La struttura a doppio livello permette una ricostruzione veloce e parziale degli eventi al livello 0, ed una ricostruzione più complessa ad alto livello. Il livello L0 e' completamente implementato su elettronica dedicata,

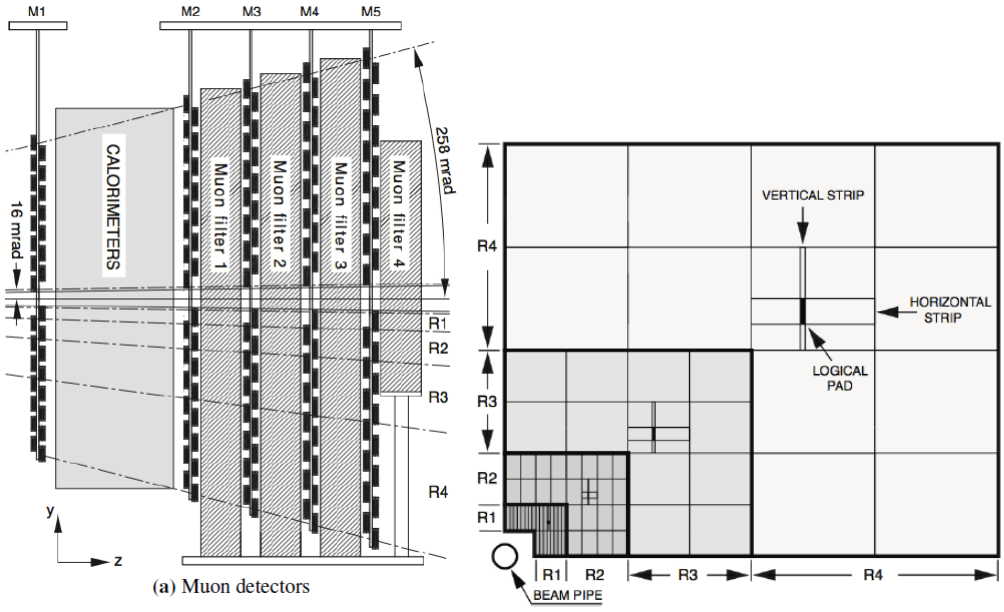


Figura 2.16: A sinistra: una vista laterale del rivelatore di muoni. A destra: la geometria di un quadrante delle camere, in cui ogni rettangolo rappresenta una camera. Le dimensioni lineari e la segmentaione di R1, R2, R3 ed R4 stanno nel rapporto 1:2:4:8.

lavorando in modo sincrono con la frequenza delle collisioni. Tramite l'L0, utilizzando le informazioni dei calorimetri e delle camere a muoni, si riesce abbattere il flusso di dati da 40 MHz fino a 1.1 MHz, che risulta essere la massima velocità a cui gli eventi possono essere letti dai rivelatori. HLT invece viene implementato in un sistema software, in cui si esegue una selezione più fine degli eventi, basata sull'utilizzo di tutti i rivelatori e riducendo il flusso di dati a 5 kHz, la massima frequenza a cui gli eventi possono essere memorizzati. In Figura 2.17 mostriamo il flusso del trigger di LHCb.

Il trigger di Livello-0

Il trigger L0 è effettuato combinando canali indipendenti, *L0 pile-up*, *L0 muon*, *L0 calorimeter* ed *L0 hadron*. La decisione di livello 0 viene effettuata attraverso l'OR logico del risultato dei tra rami, con il risultato di ridurre il flusso dati da 40 MHz ad 1.1 MHz.

L'unità di decisione del L0 fornisce la decisione globale di livello 0, distribuendola a tutte le schede del readout, e successivamente alle schede di front-end. Questo è necessario in quanto le informazioni di alcuni rivelatori vengono fornite solamente dopo la generazione dell'L0. I dati di tutti i rivelatori vengono memorizzati in registri formati da una pipeline analogica, letta con una latenza fissata a $4\,\mu$s, in cui viene presa la decisione del trigger di livello 0. Queste specifiche sono state raggiunte

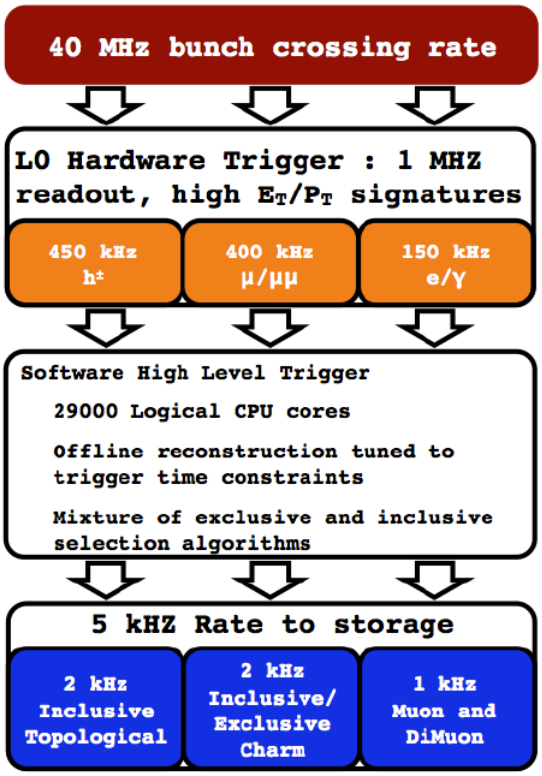


Figura 2.17: Il flusso del trigger di LHCb, con riportate le selezioni e la frequenza degli eventi all'ingresso di ogni stadio.

costruendo schede elettroniche *custom* per l'intero sistema L0, facendo largo uso di strutture parallele ed a pipeline per concludere il processamento entro la latenza fissata. A questo stadio, sono processate le informazioni derivate dai calorimetri e dai rivelatori a muoni.

Il trigger Level 0 pile-up contribuisce alla misura di luminosità e non è utilizzato per la selezione di eventi interessanti. Le informazioni sono quelle derivanti dalle stazioni di veto del VELO, con le quali vengono misurati il pile-up e la molteplicità delle tracce.

Il trigger Level 0-muon utilizza le informazioni delle 5 stazioni del rivelatore di muoni, per identificare i muoni più energetici. Una volta identificati due candidati muoni con alto momento trasverso in un quadrante, la decisione del trigger dipende da due soglie: una sul valore più elevato del momento trasverso (Level 0 muon) ed una sui due valori più alti di momento trasverso (Level 0 dimuon).

Il trigger Level 0-calorimeter viene elaborato grazie alle informazioni derivanti dal sistema di calorimetri, inclusi il preshower e gli scintillatori. Viene calcolata l'energia trasversa depositata in un cluster di 2×2 celle della stessa dimensione, sia per l'ECAL che per l'HCAL. L'energia trasversa è combinata con il numero di hit presenti nel preshower e nello scintillatore, in modo da definire un trigger specifico per i fotoni, gli elettroni e gli adroni.

Il trigger Level 0-hadrons ha lo scopo di collezionare campioni numerosi di particelle provenienti dai decadimenti di adroni c e b. Tali particelle hanno in media un momento trasverso maggiore rispetto alle particelle provenienti dai dedacimenti dei quark leggeri, il che aiuta nella discriminazione rispetto al fondo.

Il trigger di alto livello

Gli eventi selezionati dal Livello 0 di trigger vengono inviati ad un centro di selezione, composto da una farm di 29000 processori commerciali per PC, per la decisione dello stadio di HLT. L'algoritmo di HLT è implementato tramite un programma scritto in C++, che viene eseuito da ogni processore, ricostruendo e selezionando gli eventi nel modo più simile possibile all'algoritmo usato per le ricostruzioni offline. Una sostanziale differenza tra gli algoritmi online ed offline è il tempo disponibile per ricostruire un singolo evento. La ricostruzione offline richiede 2 s per evento, mentre l'algoritmo online deve completare la ricostruzione di un evento in circa 50 ms, tempo che dipende sia dalla frequenza di arrivo degli L0 e dalla potenza di calcolo dei processori.

Le selezioni usate all'interno di HLT sono molteplici e sono specifiche per gli eventi di interesse. in particolare decadimenti di adroni c e b. Ogni selezione è specificata da un certo algoritmo di ricostruzione e da criteri di selezione che riguardano la cinematica delle particelle, la topologia del decadimento e l'identificazione delle particelle. Il tempo di processamento di HLT è diviso tra due livelli, chiamati HLT1 ed HLT2. La principale differenza tra i due riguarda la complessità delle informazioni che possono essere utilizzate ed il tempo che hanno a disposizione. Viene quindi

effettuata una ricostruzione parziale, in HLT1, per diminuire il flusso a 30 kHz, per poi procedere ad una ricostruzione complessa in HLT2.

In HLT1, vengono ricostruite le tracce nel VELO e selezionate quelle candidate ad essere provenienti dal decadimento di quark pesanti andando a valutare il parametro di impatto della traccia rispetto al vertice primario più vicino alla traccia. In HLT2, si procede ad una ricostruzione completa in tutti i tracciatori dei candidati trovati nel VELO. Si ricostruiscono quindi i vertici secondari, imponendo delle condizioni sulle lunghezze di decadimento e sulle masse, così da ridurre il flusso dei dati a 5 kHz.

2.2.4 Il sistema di readout

Una volta che il segnale L0 è stato generato, i dati vengono trasferiti dalle schede di frontend alla *counting room*, situata ad un piano sopra i detector, tramite fibre ottiche o linee di rame, dove le schede di readout, chiamate TELL1, interfacciano i rivelatori con il sistema di acquisizione dati.

La TELL1 [40] è una scheda generica su cui possono essere montate piccole schede aggiuntive, chiamate mezzanini, per ricevere in ingresso dati da differenti rivelatori. Esistono due tipi di mezzanini: uno con interfaccia analogica (A-Rx) per i dati del VELO, i quali vengono digitalizzati in loco, ed uno con interfaccia ottica (O-Rx) per gli altri rivelatori. Ciascuna scheda A-Rx monta un ADC con 16 canali a 10 bit, campionati a 40 MHz; ogni scheda O-Rx usa due connettori con ricevitori ottici a 12 vie che sostengono un flusso dati di ~ 1.3 Gb/s, per un totale 24 canali per scheda. La banda totale di ingresso risulta quindi essere $16 \times 4 \times 10 \times 40\text{MHz} = 25.6$ Gb/s per i canali analogici e di $24 \times 1.28\text{Gb/s} = 30.7$ Gb/s per quelli ottici.

I dati ricevuti dai mezzanini sono processati da 5 FPGA Altera, della famiglia Stratix, 4 di pre-processamento (PP) ed uno per la formattazione dei dati (*Synk Link*, SL).

Ciascuna PP riceve dati che hanno superato il livello 0 di trigger, da una delle schede Rx, ad una frequenza di 1.1MHz. Le operazioni svolte riguardano a riduzione del rumore, la *zero suppression* e il trasferimento dei dati alla SL.

La SL distribuisce alle PP i segnali di sincronizzazione, come il clock, il trigger, e l'identificatore degli eventi. In più ha il compito di formattare i dati provenienti dalle PP, e di trasmettere tali pacchetti al resto della catena di readout via Ethernet.

La TELL1 è equipaggiata anche con:

- una *credit card PC* (CCPC) commerciale, che utilizza linux come sistema operativo, il quale gestisce gli *slow control* della scheda e l'ECS (vedi sotto);
- una scheda (TTCrx) che riceve i segnali di controllo, come il clock, il reset e il trigger L0;
- una scheda Ethernet con 4 connessioni da 1 Gb/s in uscita;

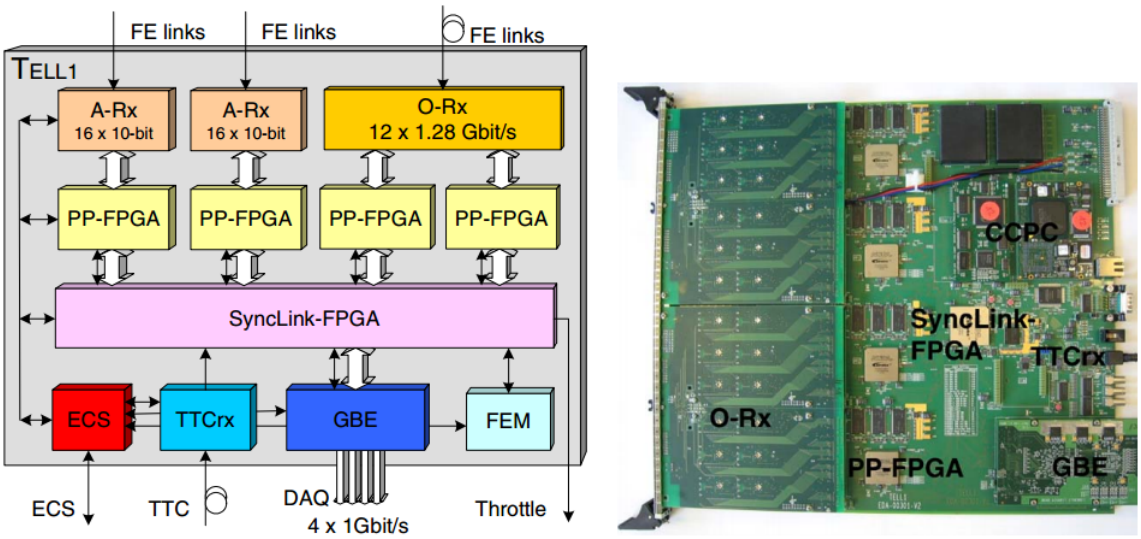


Figura 2.18: A sinistra: un diagramma a blocchi della TELL1, con entrambi i tipi di mezzanini per la ricezione dei dati. A destra: una foto della scheda equipaggiata con i ricevitori ottici.

- una mezzanina che simula alcuni segnali provenienti dai Beetle chip, in particolare il segnale di validazione dei dati (*data valid*), poiché questo non viene inviato alle schede insieme ai dati.

La sincronizzazione con il resto del sistema di DAQ è garantita dalla rete di *Timing and Fast Control* (TFC), che genera il clock globale e gestisce l'invio dei segnali veloci e dell'*Experiment Control System* (ECS).

2.3 Modifiche al rivelatore di LHCb per l'upgrade del 2020

A partire dal 2020, l'esperimento LHCb sarà oggetto di sostanziali miglioramenti, riguardanti sia il rivelatore che i sistemi di trigger e di acquisizione dati [41]. Fra i più importanti cambiamenti, ci sarà quello del sistema di acquisizione dati, capace di leggere l'intero rivelatore a 40 MHz invece dell'attuale frequenza di 1.1 MHz, e lo sviluppo di un trigger puramente basato su software. Inoltre molti dei rivelatori dell'esperimento saranno migliorati per rimanere pienamente efficienti anche nelle nuove condizioni sperimentali: energia nel centro di massa $\sqrt{s} = 14\,\text{TeV}$ e un importante aumento nella luminosità, fino a $\mathcal{L} = 2 \cdot 10^{33}\,\text{cm}^{-2}\text{s}^{-1}$. Questo porterà ad una più alta molteplicità di traccia rispetto alla configurazione attuale, e ad un numero di interazioni pp primarie per collisione pari a $\mu = 7.6$.

2.3.1 I nuovi rivelatori per LHCb

I più importanti miglioramenti nel rivelatore riguarderanno i rivelatori di traccia, che saranno totalmente riprogettati per ottenere misure più precise. I rivelatori per l'identificazione di particelle saranno migliorati per permettere una lettura a 40 MHz readout, e per meglio sostenere il nuovo ambiente sperimentale caratterizzato dalla maggiore molteplicità di tracce.

Il rivelatore VELOPIX

Il rivelatore di vertice sarà radicalmente migliorato [42], con una migrazione alla tecnologia a pixel (VELOPIX) invece dell'attuale tecnologia a microchip, comportando la sostituzione di tutti i sensori al silicio e dell'elettronica. Il nuovo VELO è composto da 26 tracking layers, due dei quali sono pile-up stations usati per misurare la molteplicità di tracce nella regione a z negativo. Ogni stazione è divisa in due moduli, con la possibilità di distanziarli dall'asse del fascio come nell'attuale VELO. Ogni modulo contiene 4 sensori al silicio con un'area attiva di $42.46 \times 14.08\ \mathrm{mm}^2$. L'intero rivelatore VELOPIX detector conta 41 M pixels, con una dimensione di 55 x 55 $\mu\mathrm{m}^2$ in un piano trasverso all'asse del fascio. Il raggio interno dell'area sensibile sarà ridotto da $r = 8.2$ mm a meno di $r = 5.1$ mm, per migliorare la risoluzione dei parametri di impatto. La risoluzione prevista per un singolo impatto è $\approx$ 12-15 μm per le coordinate x ed y. In Figura 2.19(a) sono riportate le caratteristiche principali del nuovo VELO.

Upstream Tracker

L'attuale tracciatore Turicense [43] sarà rimpiazzato dal rivelatore *Upstream Tracker* (UT), un rivelatore formato da 4 piani di sensori a micro-strip di silicio. Rispetto al TT, il nuovo UT avrà dei sensori più sottili, con una segmentazione più fine, e coprirà una regione più grande in accettanza. Le strip nei piani saranno orientate secondo la configurazione x-u-v-x, con le strip verticali nel primo e nell'ultimo piano, mentre i piani intermedi avranno le strip direzionate con un angolo rispetto alla verticale di $-5°$ e $+5°$. La dimensione ed il passo dei sensori dipendono dalla loro posizione sul piano del rivelatore. Vicino alla linea del fascio i sensori hanno uno spessore di 95 μm, con un passo di 5 cm, mentre nell'area più lontana, le strip sono spesse 190 μm e distanti 10 cm. La copertura angolare dell'UT è di 314 (248) mrad nel piano di curvatura (non curvatura). In Figura 2.19(b) vediamo è riportata la struttura dell'UT.

Tracciatore a fibre scintillanti

Il complesso di rivelatori OT ed IT saranno rimpiazzati da un tracciatore a fibre scintillanti (SFT). Ogni piano dell'SFT sarà formato da fibre scintillanti lunghe 2.5 m lette da fotomoltiplicatori posti al di fuori dell'accettanza del rivelatore. Il nuovo tracciatore avrà 3 stazioni, ognuna formata da 4 piani, in cui le fibre sono

disposte secondo la configurazione di coordinate x-u-v-x, già descritta nei precedenti paragrafi. Le fibre avranno un diametro di 0.25 mm, formata internamente da un polimero ricoperto di materiale organico scintillante. La luce viene prodotta dall'eccitazione del polimero, e viene propagata lungo la fibra tramite riflessione multiple. La luce si propaga all'interno della fibra con un tempo di 6 ns/m, con una lunghezza di attenuazione tipica di ~ 4 m ed un tempo di decadimento di ~ 3 ns.

Rivelatore Cherenkov

Nell'Upgrade del rivelatore RICH1 sarà mantenuto il radiatore con il C_4F_{10}, mentre sarà rimosso il modulo con l'aereogel. Il RICH2 manterrà la stessa struttura con il radiatore di CF_4 [44]. L'attuale sistema di lettura sarà sostituito con fotomoltiplicatori multi-anodo con un elettronica di lettura esterna che può operare a 40 MHz. Tutti i componenti ottici saranno riutilizzati il più possibile, ricalibrando la posizione e l'orientamento.

Calorimetri

Il sistema di calorimetri sarà equipaggiato con un'elettronica completamente nuova, poiché saranno usati dei differenti fotomoltiplicatori per la lettura. L'apparato di reiezione del fondo, il preshower e gli scintillatori, saranno rimossi perché non verranno più utilizzati nella decisione del trigger di livello 0. La struttura del calorimetro elettromagnetico e adronico rimarrà invariata.

Camere a muoni

Nella nuova configurazione del rivelatore di muoni, la stazione M1 sarà rimossa, a causa della grande occupazione attesa di questa stazione con la luminosità attesa nell'upgrade, che renderebbe difficile l'associazione degli hit in M1 con le tracce dei muoni nelle altre camere. La configurazione delle stazioni M2-M5 rimarrà invariata, ma sarà aggiunto uno spessore ulteriore intorno alla linea di fascio prima dell'HCAL, per migliorare l'assorbimento degli sciami e ridurre il flusso di particelle nella zona più vicina al fascio della stazione M2.

2.3.2 Il nuovo sistema di acquisizione dati e di trigger per LHCb

Con l'upgrade di LHC del 2020, la luminosità crescerà fino ad un valore di $10^{34}\mathrm{cm}^{-2}\mathrm{s}^{-1}$ ($2 \cdot 10^{33}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ per LHCb) gli attuali sistemi di trigger non saranno più in grado di selezionare dati in modo efficiente, a causa dell'elevata luminosità prevista. Quindi, in LHCb, le informazioni provenienti dai calorimetri e dalle camere a muoni non saranno più adatte per elaborare il trigger di livello 0. Quindi, il nuovo sistema di trigger dovrà essere fortemente basato sulle informazioni dei rivelatori di traccia, che diventeranno la base fondamentale per le decisioni del trigger. Requisito di questo nuovo sistema sarà la capacità di elaborare gli eventi dai tracciatori ad una frequenza di 40 MHz, rispetto a quella attuale di 1 MHz.

L'interfaccia tra i rivelatori e il sistema di acquisizione dati, attualmente implementato nella scheda TELL1, sarà implementata in una nuova scheda di lettura, denominata TELL40. La scheda sarà progettata mantenendo la struttura base della TELL1 e utilizzando dispositivi più performanti, come gli FPGA della famiglia Altera Stratix V. Questa è la nuova famiglia di FPGA prodotta da Altera, ed offre molte risorse per il processamento locale dei dati, sia in termini di blocchi logici sia come numero di connessioni disponibili. La scheda riceverà gli hit dai sottorivelatori ad una frequenza di 40 MHz per evento, formattandoli in pacchetti da distribuire alla farm di processamento degli eventi. I pacchetti sono inviati attraverso una una rete di connessione veloce basata su protocolli standard di comunicazione, per la quale si pensa di utilizzare reti Ethernet a 10 Gigabit. La scheda sarà compatibile con lo standard hardware ATCA (*Advanced Telecomunication Computing Architecture*), che sarà utilizzato in altri esperimenti del CERN. Inoltre la scheda dispone di un interfaccia per la ricezione dei segnali di temporizzazione e di TFC ed ECS).

All'interno di questo sistema di FPGA è possibile inserire un dispositivo come quello che sarà discusso in questa tesi

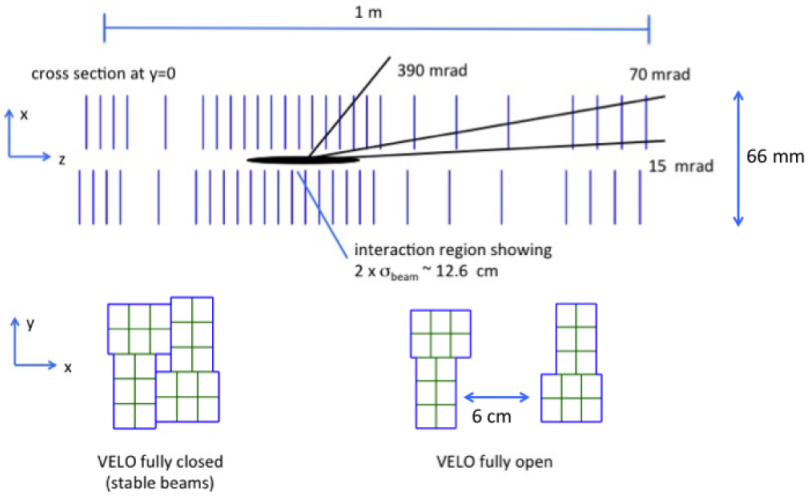
Il nuovo trigger sarà implementato su una farm di PC, dove un software dedicato analizzerà le informazioni provenienti da tutti i rivelatori per ottenere la decisione finale [45]. Tuttavia, la fattibilità di tale sistema necessita ulteriori studi, poiché l'implementazione richiede un numero molto elevato di CPU per processare tutti gli eventi. Quindi, poiché nel primo periodo il numero di CPU disponibili potrebbe non essere sufficiente, si pensa di mantenere un trigger di livello 0 che consenta di variare il rate di ingresso alla farm tra 1 e 40 MHz. I parametri principali per la progettazione del sistema di trigger riguardano la frequenza di collisione dei pacchetti con almeno una interazione in ingresso alla farm (parametro stimato ad un valore $\sim 10\,\text{MHz}$), la frequenza di uscita delle informazioni dalla farm, stimata a $\sim 20\,\text{kHz}$, e la dimensione dei dati da processare ($\sim 100\,\text{KB}$). Il software per l'elaborazione del trigger deve poter eseguire l'algoritmo di tracciatura, la ricostruzione delle tracce e la selezione degli eventi in tempo reale per un gran numero di canali.

2.4 Ricostruzione di tracce in tempo reale ad LHC

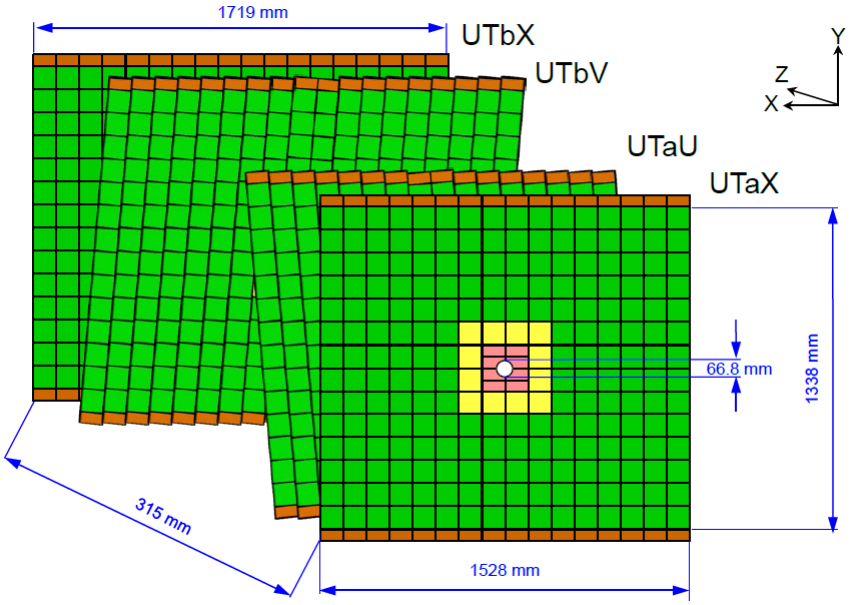
Nella prospettiva di aumentare la luminosità di LHC nei Run futuri, negli esperimenti CMS ed ATLAS si stanno progettando dei sistemi hardware dedicati alla ricostruzione di almeno una parte delle tracce alla frequenza di 40 MHz a cui avverranno le interazioni. Anche in LHCb, in parallelo alla progettazione del sistema descritto nella sezione precedente, che è considerato l'opzione di base, sono stati effettuati studi di soluzioni alternative basate non solo su CPU e sulla loro rete di interconnessione, e che fin dall'inizio possano processare gli eventi alla frequenza richiesta di 40 MHz senza dover ridurre il flusso di dati in ingresso.

Questa è basata su un nuovo algoritmo chiamato "Retina Artificiale" e punta a ricostruire le tracce con semplici operazioni che possono essere implementate su di-

spositivi elettronici riprogrammabili (FPGA). Questo algoritmo e la sua applicazione a casi reali verranno descritti nei capitoli seguenti.



(a)



(b)

Figura 2.19: Layout dell'upgrade del VELO (a) e dei rivelatori UT (b).

3

Un Processore di tracce basato sull'algoritmo della Retina Artificiale

In questo capitolo presentiamo un nuovo algoritmo specializzato per la ricostruzione di tracce in tempo reale. L'unità di processamento degli hit è chiamata Track Processing Unit (TPU), che ha come obiettivo quello di ricostruire tracce a livello 0 del trigger di LHC, con una frequenza di eventi in ingresso di 40 MHz ed alcuni μs di latenza. Questa unità è basata su di un innovativo algoritmo di tracciatura, detto *retina artificiale*, che mira che simula alcune delle caratteristiche di organizzazione della retina dei mammiferi.

3.1 L'apparato visivo nei mammiferi

L'algoritmo della retina artificiale trae ispirazione dalla biologia, cioè dall'apparato visivo mammiferi. Questo apparato è in grado di processare una grande quantità di segnali provenienti dall'ambiente esterno, utilizzando neuroni specializzati nel riconoscere determinati schemi nelle immagini analizzate. Esperimenti recenti hanno mostrato che la corteccia visiva elabora una prima bozza dell'immagine, contenente le informazioni spaziali della stessa, in circa 30 ms con una frequenza massima di circa 1 kHz, impiegando quindi circa 30 cicli per processare ciascuna immagine [46].

L'immagine in ingresso all'occhio, impressiona la retina, che codifica l'immagine e la invia tramite il nervo ottico alla corteccia visiva, dove viene processata. La codifica spaziale avviene nei *campi recettivi* [47], strutture formate da un disco centrale e un anello concentrico all'esterno. Ad ogni campo recettivo è associato una cellula *centro-on* o *centro-off*. Le cellule centro-on rispondono massivamente se lo stimolo luminoso è localizzato nel centro del loro campo recettivo, invece hanno una risposta inibitoria se lo stimolo interessa la periferia del campo. Le cellule centro-off rispondono in modo opposto agli stessi stimoli; dunque un'illuminazione diffusa produrrà una debole risposta da ambedue le cellule. La risposta di queste cellule può essere parametrizzata come una gaussiana, cioè funzione continua e non a scalino. Il comportamento di queste cellule fa si che i campi recettivi siano degli ottimi stru-

(a) (b)

Figura 3.1: Codifica di un immaigne effettuata della retina: (a) l'immagine reale che viene percepita dalla retina; (b) la codifica che ne viene fatta. Si nota che vengono messi in risalto solamente i contorni delle figure, senza nessuna informazione sul colore. Queste vengono elaborate successivamente a livello della corteccia visiva.

menti per identificare i contorni degli oggetti, grazie alla loro capacità di percepire differenze di illuminazione. In Figura 3.1 vediamo una simulazione della retina, in cui è messa in risalto la capacità di distinguere i contorni delle figure.

I campi recettivi sono organizzati in macrostrutture di diverso tipo, con diversi orientamenti. Questo è fondamentale in quanto permette alla retina di riconoscere l'orientamento dei contorni; per esempio, campi recettivi disposti verticalmente risponderanno a linee orizzontali con minore intensità, rispetto ad una quasi verticale, come è mostrato in Figura 3.2.

L'immagine viene quindi passata alla corteccia visiva, che elaborerà tutte le informazioni provenienti dalla retina per creare l'immagine finale da noi percepita [48].

Il sistema così descritto è in parte simile agli algoritmi di tracciatura in tempo reale utilizzati negli esperimenti di della fisica delle alte energie [49].

3.2 L'algoritmo della retina artificiale

L'algoritmo della retina artificiale è un algoritmo estremamente parallelizzato per la ricostruzione di tracce, pensato per gli esperimenti della fisica delle alte energie e proposto per la prima volta nel 1999 [50]. Lo scopo è quello di ricostruire tracce nei rivelatori imitando la capacità del sistema visivo dei mammiferi, descritto in 3.1, di riconoscere specifici modelli, e compiere una somma pesata dei segnali da una varietà di recettori.

Andiamo ora ad illustrare il principio di funzionamento dell'algoritmo, considerando un modello semplificato in cui un rivelatore a piani paralleli venga attraversato

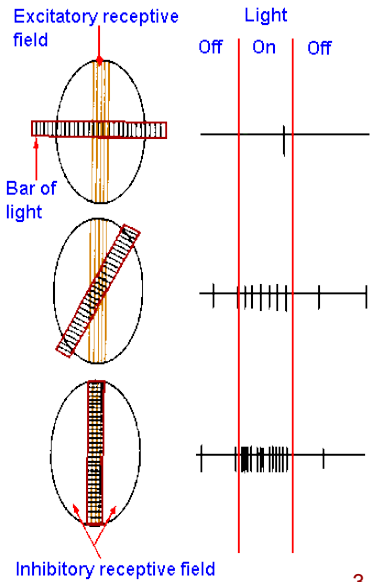


Figura 3.2: Risposta di cellule centro-on disposte verticalmente, in base alla corrispondenza tra l'orientamento dei campi recettivi (*receptive field*) e l'immagine in ingresso (*bar of light*). A sinistra sono mostrati i campi recettivi, a destra l'intensità della risposta degli stessi.

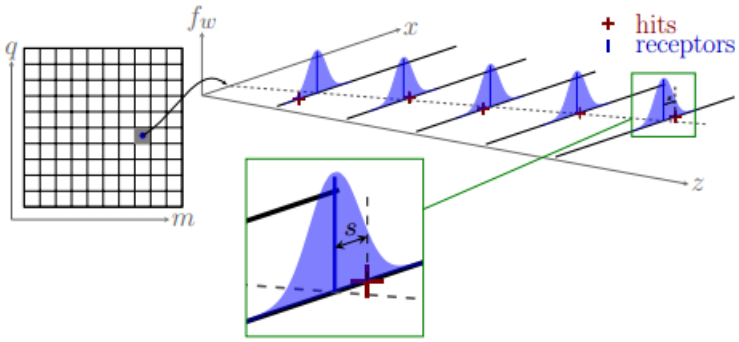


Figura 3.3: Rappresentazione schematica della corrispondenza tra una traccia nello spazio dei parametri (a sinistra) e nello spazio reale (a destra). Viene mostrato anche il peso di un hit in base alla sua distanza s dal recettore.

da tracce rettilinee. Date le coordinate delle intersezioni delle tracce con il rivelatore (ci riferiremo a tali intersezioni con il termine *hit*), siamo interessati a calcolare i parametri delle tracce vere che li hanno generati. Se ci limitiamo ad una vista trasversale del nostro sistema, le tracce sono univocamente determinate da due parametri, che chiameremo (q, m). Possiamo quindi discretizzare lo spazio dei parametri, costruendo una griglia bidimensionale, in cui ogni elemento della stessa, detto *cella*, corrisponde ad una coppia di parametri (q_i, m_j). Ogni cella $(i,\ j)$ corrisponde quindi ad una *traccia mappata*, le cui intersezioni con il k-esimo piano del rivelatore sono chiamate recettori $\{x_k^{i,j}\}$; quindi una traccia mappata ha un recettore su ogni piano. In Figura 3.3 si mostra il concetto mappatura di una traccia nello spazio dei parametri. Consideriamo adesso una traccia passante per il nostro rivelatore: per ogni hit calcoliamo la distanza s_{ijkr} con tutti i recettori del piano considerato ed usiamola per calcolare il peso w dei recettori utilizzando una funzione peso f_w, ad esempio una funzione gaussiana. Così ad ogni recettore (q_i, m_j) viene associato un peso:

$$w_{ijkr} = exp(-\frac{(s_{ijkr})^2}{2\sigma^2}) \tag{3.1}$$

dove la distanza vale:

$$s_{ijkr} = \bar{x}_{k,r} - x_k^{ij}, \tag{3.2}$$

dove $\bar{x}_{k,r}^{(r)}$ è l'r-esimo hit sul piano k e σ è la varianza della funzione peso, che rappresenta uno dei parametri della retina. Calcoliamo infine il *livello di eccitazione* della cella R_i come:

$$R_{i,j} = \sum_{k,r} w_{ijkr}. \tag{3.3}$$

Questa procedura viene eseguita per tutte le celle completamente in parallelo.

In Figura 3.4 si mostra la simulazione dell'algoritmo descritto per un semplice evento. Si parlerà più approfonditamente della simulazione ad alto livello nella sezione 3.4. Possiamo notare che ogni traccia dello spazio fisico reale, corrisponde ad un *cluster*, ovvero un gruppo di celle nello spazio dei parametri, con un livello di eccitazione più alto rispetto alle celle vicine. Il valore dell'eccitazione viene poi paragonato con una soglia fissata, in modo da discriminare il contributo delle tracce vere da quello del rumore di fondo.

I parametri delle tracce vengono estratti andando a cercare il baricentro del cluster. In questo modo si raggiungono precisioni superiori rispetto al passo della griglia, che può essere lasciato sufficientemente maggiore rispetto alla tipica risoluzione spaziale del rivelatore, ponendo così un limite sul numero massimo di celle utili senza perdere

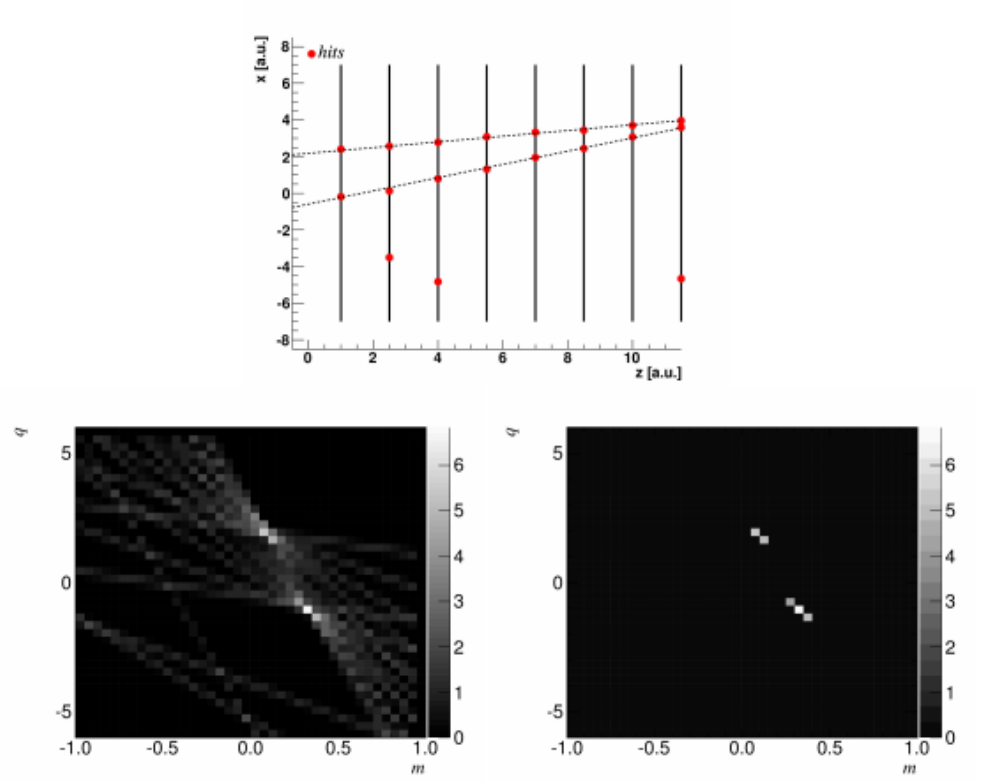


Figura 3.4: Risposta della retina (in basso a sinistra) ad un particolare evento (in alto). In basso a destra sono presenti i cluster sopra soglia ricostruiti dalla retina.

efficienza. Tuttavia, il numero di celle è limitato inferiormente dalla capacità della retina di separare due tracce, e quindi è correlato con il grado di occupazione del rivelatore. Infatti se quest'ultimo è molto affollato, ci sarà bisogno di una segmentazione più fitta della retina così da non perdere efficienza e risoluzione sui parametri della traccia.

L'implementazione della retina in un caso bidimensionale, come quello appena descritto, ha molte analogie con la trasformata di Hough [51]. Questa tecnica è stata proposta per la prima volta nel 1959 per l'analisi computerizzata delle fotografie nelle camere a bolle, per poi essere brevettata nel 1962, come strumento alla base dell'elaborazione artificiale delle immagini e della *computer vision*. Tuttavia l'alto grado di parallelismo e la risposta analogica agli stimoli, rende l'algoritmo retina più vantaggioso, soprattutto se il problema presenta un grado di complessità elevato.

3.2.1 Differenze con altri metodi di tracciatura

L'algoritmo retina presenta alcune novità rispetto ai metodi di tracciatura basati sulle memorie associative, tipo SVT. Infatti, mentre gli altri sistemi forniscono una

risposta binaria come risultato del confronto, il sistema retina risponde in modo continuo, in funzione della distanza tra gli hit del rivelatore e i valori memorizzati, imitando la risposta continua dei fotorecettori ad un'eccitazione luminosa. Questo permette di gestire meglio le inefficienze del rivelatore applicando una soglia ai valori di eccitazione della retina, in quanto una traccia buona non è identificata con il numero di hit che le appartengono (come in SVT), ma con la risposta che i recettori hanno agli stimoli esterni.

Notiamo inoltre che l'estrazione dei parametri della traccia viene effettuata andando a cercare il centro di massa di un cluster. Rispetto ad altri metodi come SVT, in cui si effettuava un fit linearizzato per estrarre i parametri della traccia, utilizzando una logica dedicata per questa funzione (*track fitter logic*, implementata su chip programmabili), c'è un notevole risparmio sia in termini di latenza e che di hardware dedicato. Questo è un accorgimento importante per poter utilizzare l'algoritmo retina al livello 0 della catena di trigger, in cui la comune elettronica di front-end degli esperimenti permette una latenza tipica nella lettura dell'ordine di alcuni μs. Difatti la ricerca del baricentro di un cluster risulta un processo estremamente più veloce da implementare, rispetto al calcolo di un fit.

3.3 Architettura della TPU

L'architettura proposta per la TPU [52] rispecchia la descrizione del paragrafo precedente, in cui i recettori sono mappati all'interno di una matrice di celle, che eseguono l'algoritmo in parallelo.

L'implementazione è stata pensata suddividendo la TPU in 3 stadi principali, come viene mostrato in figura Figura 3.5. Gli hit, provenienti dai piani del rivelatore, vengono inviati in parallelo alle celle attraverso una rete di indirizzamento (*switching network*), suddivisa in due parti principali, il *pre-switch* e lo *switch*. Ogni cella di recettori rappresenta l'unità fondamentale di processamento, chiamata *Engine*, implementata come un blocco logico indipendente, la quale realizza le operazioni fondamentali di confronto e di ricerca del cluster. Le informazioni utili vengono per calcolare i parametri delle tracce inviate ad ultima unità (*fitter*).

Questa architettura è stata concepita per poter processare gli eventi in un tempo sufficiente corto da rientrare nel bilancio temporale utile per un il livello-0 di trigger.

3.3.1 La rete di switch

Una parte fondamentale della TPU è il sistema di smistamento dei dati, grazie al quale gli hit vengono distribuiti in parallelo a tutti gli engine in tempo reale. In un ambiente come LHC, questa operazione e' complicata, a causa dell'abbondante flusso di dati in uscita dai rivelatori, dell'ordine di alcuni Tbit/s ad una frequenza di 40 MHz. È necessario quindi avere a disposizione una rete intelligente che invii gli hit al minimo numero di engine ai quali questi hit contribuiscono con un peso significativo. Dal punto di vista di una futura implementazione, questa rete è realizzabile se viene

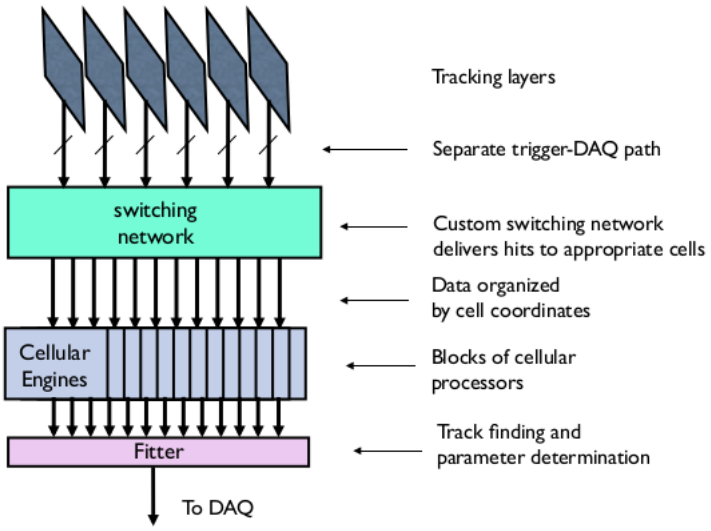


Figura 3.5: L'architettura della TPU, divisa in tre stadi principali.

suddivisa in 2 fasi: una ridistribuzione fisica degli hit e un indirizzamento più fine, effettuato su chip a monte degli engine.

Introduciamo adesso due definizioni, illustrate in Figura 3.6, utili seguire la descrizione della rete di switch:

- un *gruppo* è un'area fisica dei piani del rivelatore. Ogni hit appartiene ad unico gruppo, e viene identificato attraverso il numero di gruppo (*group number*), assegnato in base alle coordinate dell'hit sul piano del rivelatore;
- una *regione* è l'unione di tutte le celle nello spazio dei parametri che hanno una risposta significativa se interessate da hit appartenenti ad uno stesso gruppo.

È evidente che ci sia una certa sovrapposizione tra le regioni, poiché un hit potrà interessare più di una regione, nonostante la traccia vera appartenga ad una regione soltanto. Una buona definizione di gruppo sarà quella che minimizza la sovrapposizione tra le regioni.

Il pre-switch

Il pre-switch identifica la prima fase di distribuzione degli hit ai vari moduli della TPU. In questa fase si indirizzano fisicamente gli hit ridistribuendo i canali in uscita da ogni rivelatore, duplicandoli in caso di necessità. Questo accorgimento è di vitale importanza per diminuire il flusso di dati in ingresso ai singoli moduli.

Lo switch

Una seconda rete di indirizzamento è presente su ogni modulo della TPU. Tale dispositivo è in grado di inviare un hit in ingresso a tutti gli engine che sono sensibili

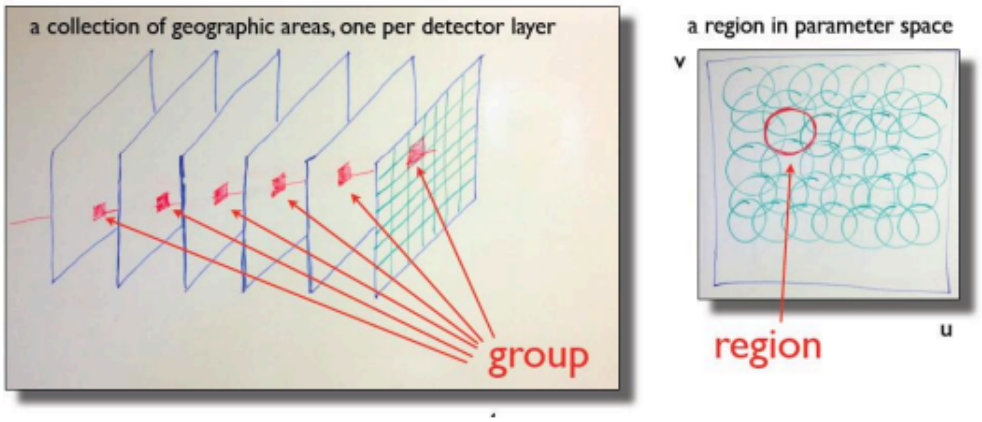


Figura 3.6: Rappresentazione dei concetti di gruppo e regione.

ad esso. L'instradamento viene effettuato memorizzando in ciascun nodo della rete una tabella (*look-up table*) che indirizza l'hit in base al proprio group number. Un esempio di questa struttura è mostrata in Figura 3.7. Da notare che, anche se sono presenti le connessioni per inviare gli hit a tutti gli engine, questa condizione non si verifica mai.

3.3.2 Gli engine

Il blocco di logica che implementa la cella della retina è chiamato *Engine*. Al suo interno si effettuano il confronto tra dati in ingresso e i recettori e la ricerca dei massimi locali. Ogni engine riceve in ingresso gli hit dei sottorivelatori, definiti in una stringa di bit contenente le corrispondenti coordinate geometriche sul detector, il piano a cui appartengono e un'etichetta che identifichi a quale evento l'hit appartenga (detta *timestamp*), in caso di necessità.

L'engine implementa le operazioni in sequenza. Le coordinate degli hit in ingresso vengono sottratte ai valori $x_i(k)$ dei recettori precalcolati e memorizzati all'interno dell'engine stesso. I recettori giusti da sottrarre vengono scelti in base al piano di appartenenza dell'hit. Le differenze vengono quindi sommate in quadratura, così da avere la distanza euclidea tra i recettori e gli hit. Tale distanza viene poi usata per estrarre da una memoria il valore del peso corrispondente. Questi pesi vengono successivamente sommati in un accumulatore. La procedura viene ripetuta per tutti gli hit di un evento indirizzati a quello specifico engine. Dopo che tutti gli hit di un evento sono stati processati, ciascun engine invia il contenuto del proprio accumulatore agli engine vicini, e controlla se sia un massimo locale. In caso positivo, l'engine invia i valori degli accumulatori del cluster 3×3 all'ultimo blocco dell'architettura della TPU.

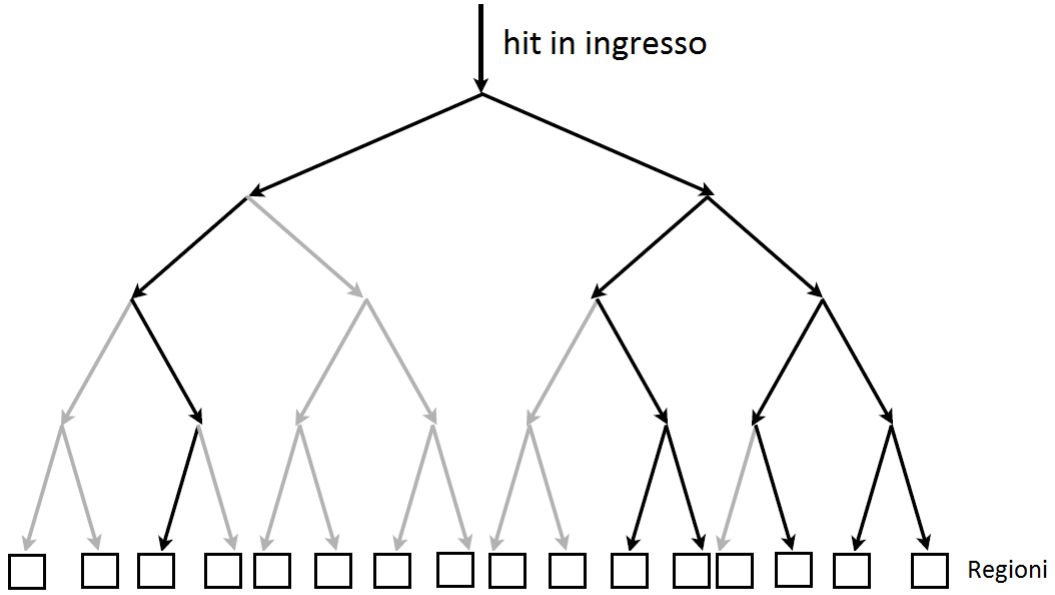


Figura 3.7: Esempio di funzionamento dello switch su chip. L'hit in ingresso ha la possibilità di essere inviato ad ogni engine, e viene indirizzato attraverso la rete in accordo il proprio group number. In questo esempio l'hit viene inviato a 6 engine.

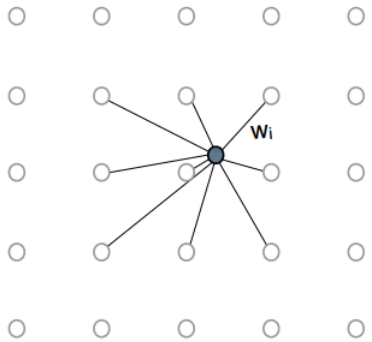


Figura 3.8: Calcolo del baricentro. La risoluzione ottenuta è maggiore del passo della retina.

3.3.3 Il calcolo dei parametri delle tracce

Il modulo per il calcolo dei parametri riceve in ingresso i valori degli accumulatori degli engine all'interno del cluster 3×3, e calcola il baricentro, come illustrato in Figura 3.8, con la seguente formula:

$$m_i = \frac{m_0}{d_k} + \frac{\sum_{i,j} m_i \cdot w_{i,j}}{\sum_{i,j} w_{i,j}} \tag{3.4}$$

dove m_0/d_k rappresenta una traslazione globale, che dipende solamente dalla posizione assoluta dell'engine massimo all'interno dello spazio dei parametri, e $w_{i,k}$ sono i pesi delle celle $(i,\ j)$ del cluster.

3.4 Studio di fattibilità di alto livello

Sono stati eseguiti studi sulla TPU ad alto livello [53, 54], usando una simulazione scritta in C++. Si è scelto di applicare l'algoritmo al caso di LHCb, esperimento in cui eseguire una buona tracciatura è fondamentale per acquisire eventi di quark pesanti quanto più puri possibile. L'applicazione della TPU ha riguardato il rivelatore Inner Tracker (IT), attualmente in uso, e i rivelatori di vertice (VELO) e l'Upstream Tracker (UT) nella configurazione prevista per il 2020. Di seguito illustreremo le tecniche che vengono utilizzate nella simulazione, come la mappatura del rivelatore, l'uniformazione dello spazio dei parametri. Tuttavia la scelta della parametrizzazione e' differente nei due casi e sara' descritta in dettaglio insieme ai risultati nel Capitolo 5 per l'IT e nel Capitolo 6 per il VELO e UT.

Spazio dei parametri uniforme

A causa della geometria dei rivelatori e le tipologie di eventi ricercati, le tracce non intersecano uniformemente i piani dei rivelatori ad LHCb. Usando un approccio basato sulla retina, questo diventa un problema, in quanto ci saranno regioni della retina con un'alta densità di tracce, ad esempio nei pressi della beam pipe, ed altre per lo più vuote. Dal punto di vista computazionale, questo significa sovraccaricare di lavoro solamente una parte di engine, lasciandone una certa porzione inattivi. Si rischia così di aumentare la latenza totale del dispositivo, fino quasi a bloccarlo, rendendolo inadatto allo scopo per cui è stato concepito. Per evitare questo effetto, è necessario rendere uniforme la distribuzione degli hit in ingresso agli engine.

Consideriamo un rivelatore posto intorno alla beam pipe, e assumiamo la distribuzione degli hit su di esso dipendente solo dalla distanza r dal centro. Possiamo quindi *fittare* la distribuzione normalizzata degli hit su ciascun piano, e usarla per ottenere una distribuzione uniforme $h(r)$:

$$h(r) = \frac{1}{N} \int_{r_{min}}^{r} f(r)dr \tag{3.5}$$

dove N e un fattore di normalizzazione, ed $f(r)$ è la funzione usata per il fit della distribuzione degli hit:

$$f(r) = \frac{1}{p_0 \cdot r + p_1} \tag{3.6}$$

dove p_0 e p_1 sono due parametri. Mostriamo in Figura 3.9 la distribuzione degli hit sui piani del VELO e la funzione $f(r)$.

Mappatura

Tramite la simulazione C++ si procede anche alla mappatura del rivelatore. per questa procedura generiamo tracce uniformemente distribuite nello spazio trasformato, dove gli hit sono distribuiti in modo uniforme. Questo significa che i recettori sono mappati nello spazio reale con maggiore densità nelle regioni maggiormente affollate, e viceversa.

Le tracce sono generate usando un generatore casuale di particelle, denominato *particles gun*[1], inizializzando i parametri delle tracce da un file esterno. In Figura 3.10 vediamo un esempio di tracce generate tramite il particle gun. Nella mappa-

[1]Il Particles Gun è uno strumento utile per simulare eventi di fisica. Difatti è possibile generare particelle seguendo la distribuzione di cui abbiamo bisogno, e propagarle all'interno dei rivelatori. Il particles gun risulta quindi uno strumento molto potente per studiare in dettaglio la TPU. Siccome l'implementazione fatta da LHCb per tale strumento mette a disposizione solamente le distribuzioni più semplici, è stato necessario personalizzarlo per avere più flessibilità, permettendo di definire i parametri delle particelle a partire da un file esterno di configurazione.

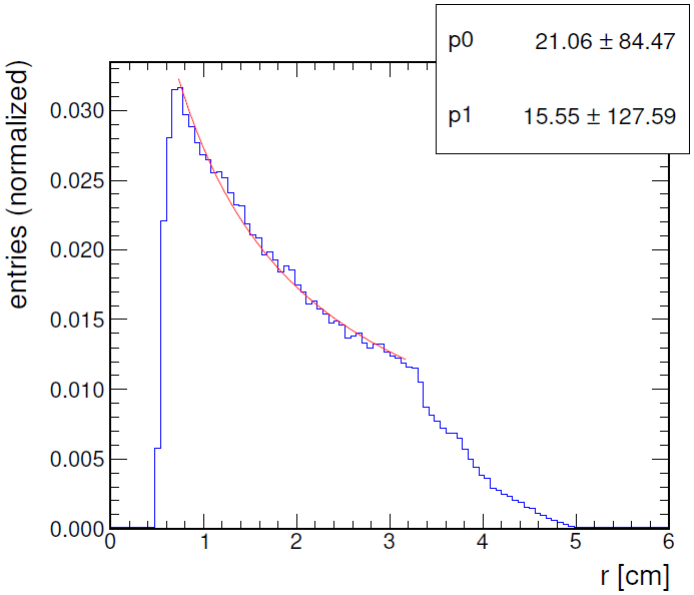


Figura 3.9: La distribuzione normalizzata degli hit sui piani del VELO, fittata dalla funzione $f(r)$.

tura consideriamo tracce ideali, cioè propagate all'interno del detector trascurando ogni effetto di interazione, come multiplo scattering, o ionizzazione.

Il passo della mappatura, come la varianza della funzione peso, sono parametri che vengono ottimizzati studiando l'efficienza e la risoluzione sulle tracce della TPU.

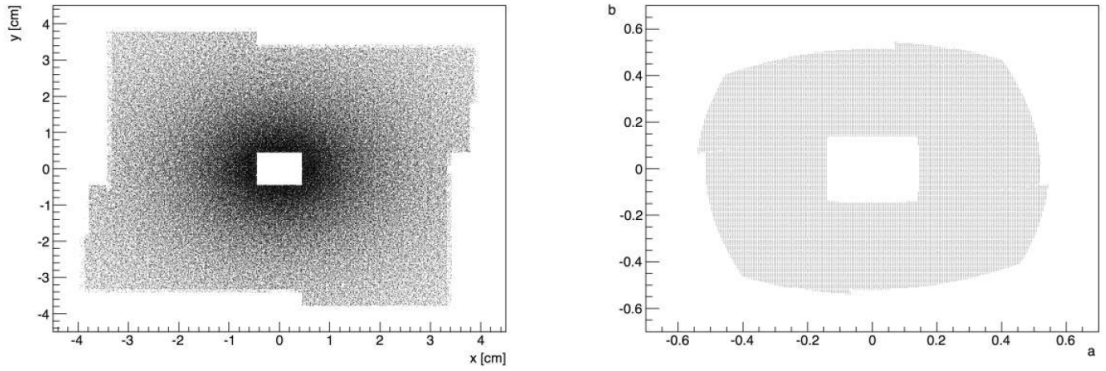


Figura 3.10: Distribuzione degli hit di tracce simulate su un piano del VELO, nello spazio reale (sinistra) e nello spazio trasformato (destra).

4 Implementazione su dispositivi a logica programmabile

Per implementare l'algoritmo della TPU si richiedono dispositivi ad alto grado di parallelizzazione ed elevata potenza di calcolo. Grazie ai continui progressi della microelettronica, da qualche anno sono disponibili dispositivi commerciali che soddisfano questi requisiti.

In questo capitolo descriveremo la caratteristiche dei dispositivi logici programmabili (*Programmable Logic Device*, PLD), su cui implementare la TPU.

4.1 Scelta della tecnologia per la TPU

Al giorno d'oggi la realizzazione di un dispositivo hardware con elevate prestazioni è effettuato su dispositivi logici programmabili, o su dispositivi realizzati in modo completamente personalizzato (approccio chiamato *full-custom*). La scelta è stata quella di basarci su dispositivi programmabili, utilizzando la tecnologia a logiche programmabili (*Field Programmable Gate Array*, FPGA; si veda la sezione 4.2.3). Infatti in una fase di studio e di sviluppo, l'elevata flessibilità, la velocità della progettazione e il basso costo iniziale sono elementi importanti, che favoriscono l'utilizzo di questa tecnologia, rispetto all'utilizzo di circuiti integrati specializzati progettati con un approccio full-custom, in cui i tempi di progettazione e il costo iniziale risultano molto più elevati.

L'utilizzo di FPGA è dettato anche dal fatto che il sistema di readout di LHCb è formato da schede che processano dati tramite FPGA. In previsione di eseguire un test della TPU direttamente sul rivelatore, non ci sarà quindi la necessità di progettare una nuova scheda, ma potremmo riutilizzarle dopo averne riprogrammato i chip, guadagnando sia in termini di tempo di sviluppo che di costo.

4.2 Introduzione ai dispositivi a logica programmabile

Fino agli anni '70, la progettazione di sistemi digitali era basata sull'utilizzo di circuiti logici standard, a bassa e a media scala di integrazione, combinati tra loro per generare la funzione logica desiderata. Questo approccio, al crescere della complessità del sistema da progettare, richiedeva un numero sempre maggiore di componenti, rendendo difficoltosa la progettazione dei circuiti stampati, aumentando il costo complessivo del progetto, il consumo di potenza e riducendone l'affidabilità, a causa dell'elevato numero di collegamenti da effettuare. Per sopperire a queste limitazioni furono introdotti nuovi metodi di progettazione.

Una prima soluzione prevedeva l'utilizzo di tecniche di sviluppo custom per la realizzazione di circuiti integrati specializzati per una determinata applicazione (chiamati *Application Specific Integrated Circuit*, ASIC), in cui l'utente realizza il progetto a partire da una libreria di componenti predefinita (approccio *standard-cell*) oppure partendo dalla realizzazione e il piazzamento di ogni singolo transistor (tecnologia *full-custom*). Tale scelta permette di raggiungere i livelli di massima ottimizzazione per quanto riguarda prestazioni, consumi di potenza ed area del chip, a scapito però di tempi di sviluppo molto dilatati dovuti alla difficoltà della progettazione, a cui vanno aggiunti costi di sviluppo e realizzazione sostenibili solo per produzioni su larga scala. Un ulteriore svantaggio è costituito dal fatto che il chip, una volta realizzato su silicio, non può essere più modificato, salvo tornare alle fasi di progetto, e ciò comporta un notevole incremento dei costi ogni qual volta si voglia rivedere, correggere o estendere il prodotto.

Un secondo approccio prevede l'utilizzo di PLD, cioè di un circuito ad elevata scala di integrazione, che può essere opportunamente programmato o personalizzato dall'utente finale, in modo da realizzare una specifica funzione. Questa soluzione offre enorme flessibilità in fase di progetto e permette di ridurre i costi ed i tempi di sviluppo e verifica, risultando in un'elevata versatilità, seppur con prestazioni finali inferiori rispetto ad un ASIC.

Le tecniche di programmazione sviluppate sono:

- tecnologia *antifuse*, in cui le connessioni non desiderati vengono eliminate facendo scorrere nelle connessioni grandi intensità di corrente;
- sistemi basati su RAM, in cui la configurazione delle connessioni è memorizzata su RAM. Il dispositivo va riprogrammato ogni volta che viene spento;
- sistemi basati su memorie di tipo *flash*, in cui viene memorizzata la configurazione delle connessioni. A differenza delle RAM, la configurazione non viene persa una volta spento il dispositivo.

La differenza sostanziale tra questi metodi sta nel fatto che la tecnologia antifuse è a programmazione singola, poiché non è possibile ricostruire le linee eliminate, mentre le altre due sono programmabili molteplici volte, andando a sovrascrivere il contenuto delle memorie.

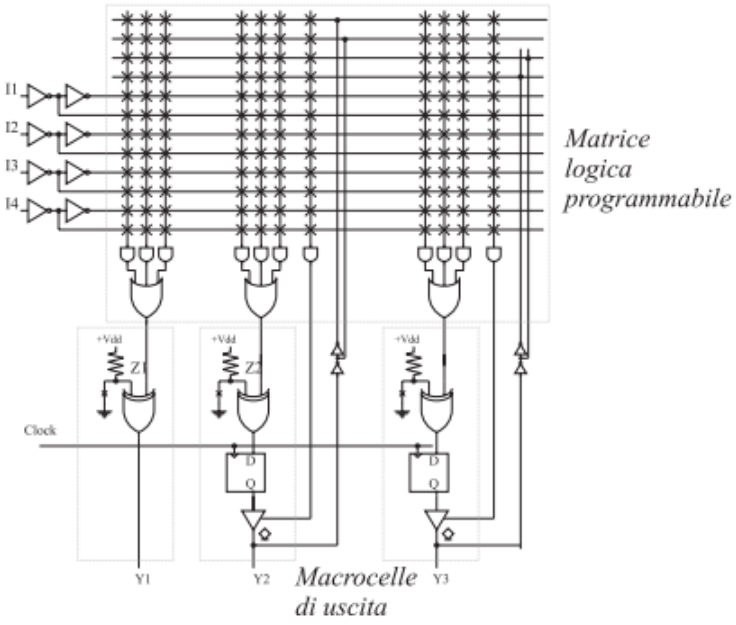


Figura 4.1: Schema a blocchi di una matrice logica programmabile. Le × identificano i nodi programmabili.

4.2.1 Matrici logiche programmabili

I primi PLD erano delle matrici logiche programmabili (*Programmable Array Logic*, PAL), e consentivano di realizzare semplici funzioni combinatorie utilizzando dei piani programmabili di porte logiche di AND od OR. In Figura 4.1 si mostra uno schema a blocchi di tali dispositivi. Per realizzare funzioni più complesse, erano presenti alcune macrocelle in uscita dalla matrice (come in Figura 4.2), usate per funzioni di feedback e programmabili in modalità sequenziale o combinatoriale. La modalità sequenziale permetteva di implementare semplici registri e macchine a stati finiti[1].

4.2.2 Dispositivi logici programmabili complessi

Per aumentare la capacità di calcolo delle PAL, la direzione presa fu quella di integrare più blocchi logici in un unico dispositivo. In questo caso si parla di dispositivi logici complessi, o CPLD. In questi dispositivi una rete di interconnessioni programmabili metteva in comunicazione diversi blocchi logici, come in Figura 4.3, permettendo lo scambio di dati per realizzare funzioni più complesse. Il blocco lo-

[1]Una macchina a stati finiti è un modello che permette di descrivere con precisione e in maniera formale il comportamento di molti sistemi, elettronici e non. In ogni stato della macchina si eseguono delle operazioni ed il passaggio allo stato successivo è determinato da condizioni interne od esterne alla macchina stessa.

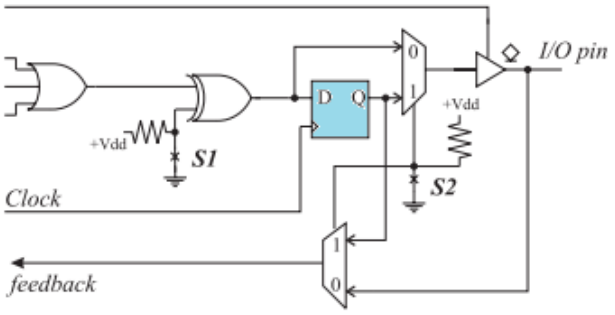


Figura 4.2: Schema a blocchi di una macrocella di una PAL. I due multiplexer programmabili consentono l'utilizzo della macrocella in modalità sequenziale o combinatoriale (escludendo il flip-flop, il blocco celeste in figura).

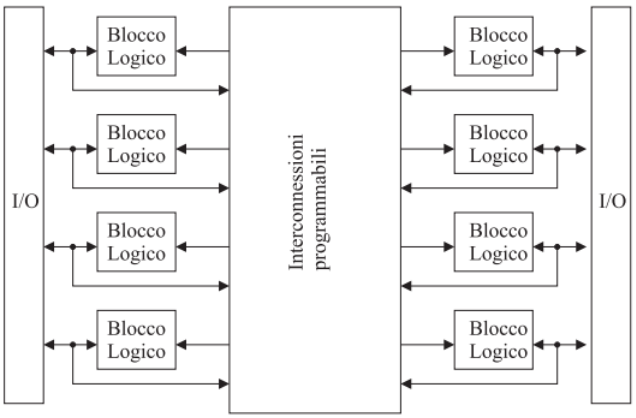


Figura 4.3: Schema di funzionamento di un CPLD.

gico principale è schematizzabile come una PAL di dimensioni maggiori, cioè una matrice di AND programmabile e varie macrocelle in uscita.

4.2.3 Matrici di porte programmabili

La famiglia di PLD più potenti risultano essere le matrici di porte programmabili sul campo (*Field Programmable Gate Array* o *FPGA*), che rappresentano uno sviluppo della tecnologia dei *gate-array* programmabili. In quest'ultima, viene fabbricata una matrice di celle logiche tutte uguali fra loro, che può essere personalizzata realizzando le sole linee metalliche di interconessione per realizzare la funzione desiderata. I costi e i tempi di sviluppo sono ridotti, ma non è possibile riprogrammare il dispositivo. Negli FPGA invece sia le celle che le interconessioni sono prefabbricate, permettendo all'utente di programmarle a seconda delle esigenze.

Rispetto ai CPLD, le celle logiche di un FPGA hanno generalmente una funzionalità ridotta. D'altro canto il numero di celle logiche di un FPGA è molto maggiore rispetto al numero di blocchi logici di un CLPD. In sintesi, un FPGA è formato da

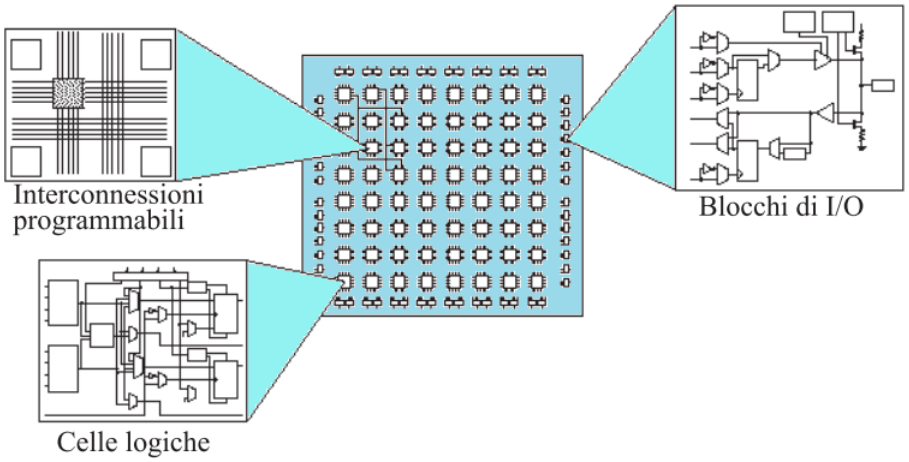


Figura 4.4: Schema interno di un FPGA.

blocchi logici di dimensione minore rispetto a quelle di un CPLD, che gli consente di avere un numero maggiore di flip-flop al suo interno e di realizzare funzioni estremamente complesse collegando opportunamente in cascata varie celle logiche.

Un FPGA è formato da tre blocchi principali (vedi Figura 4.4):

- i blocchi logici configurabili (*Configurable Logic Block* o CLB, oppure *Logic array Block*, LAB, a seconda del costruttore);
- i blocchi di ingresso/uscita (*Intput/Output Block*, IOB);
- le linee di interconnessione.

i blocchi logici configurabili

Blocchi logici Le LAB sono solitamente composte da 2 o 4 celle logiche (*Adaptive Logic Memory* o ALM) che eseguono operazioni booleane. Lo schema base di una ALM prevede l'uso di una o più *Look-Up Table* (LUT) programmabili, un sommatore completo (*Full Adder*, FA) ed un flip-flop di tipo D. La scelta di utilizzare LUT a soli quattro ingressi risiede nel fatto che la complessità di una LUT cresce esponenzialmente all'aumentare del numero di ingressi, e risulta dunque poco gestibile avere grandi LUT. Raggruppare insieme alcune LUT in una LAB e connetterle con una rete locale di interconnessioni consente infatti una maggiore velocità, dovuta al fatto che questo tipo di interconnessione è più veloce di quella generale tra blocchi logici distinti. In Figura 4.5 si mostra un esempio di ALM di un FPGA moderno.

Blocchi di ingresso/uscita Gli IOB si occupano della gestione dei segnali in ingresso e in uscita dall'FPGA, attraverso il controllo dei pin del chip. A seconda della funzione riservata, un pin che può essere configurato come ingresso, uscita, o bi-direzionale. Posizionati lungo il perimetro della matrice di LAB, gli IOB sono

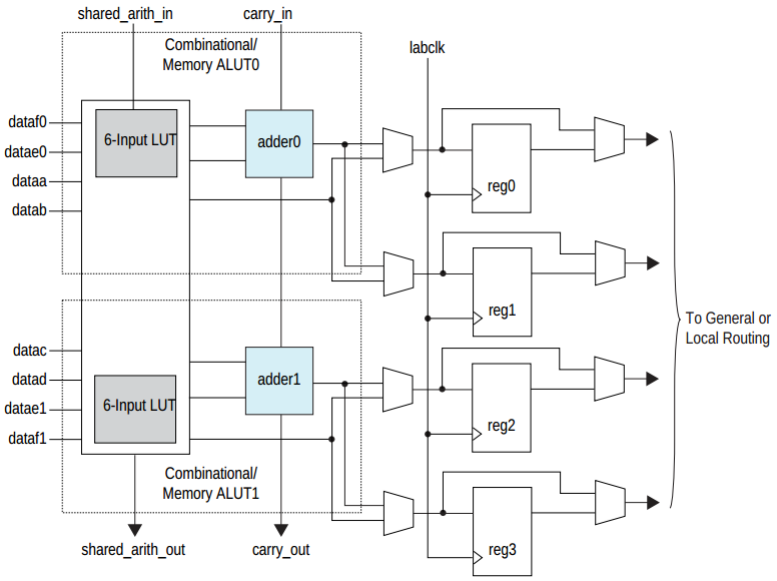


Figura 4.5: Schema a blocchi di una ALM di un FPGA di Altera [55].

generalmente composti da flip-flop dedicati alla sincronizzazione dei dati, da multiplexer che gestiscono i segnali in modalità DDR (Double Data Rate) e da buffer per la gestione dei diversi standard logici.

Linee di interconnessione Le linee di interconnessione hanno il compito di mettere in comunicazione le diverse risorse del dispositivo. Queste si dividono in linee locali e linee globali: le linee locali sono interconnessioni che connettono LAB adiacenti, minimizzando il ritardo del segnale; le linee globali consentono invece di mettere in comunicazione risorse distanti tramite percorsi che non attraversano le matrici di scambio e che sono pertanto caratterizzati dal non introdurre ritardi significativi. Ciò consente di utilizzare LAB distanti tra loro per implementare funzioni logiche complicate. La distribuzione dei segnali privilegiati, quali clock e reset, avviene invece su una rete di linee dedicate, in modo da garantire una distribuzione sincrona a tutto il chip col minimo ritardo possibile. Un esempio di linee di interconnessione è quello mostrato in Figura 4.6.

Tecniche di programmazione Le interconnessioni di un FPGA possiedono dei nodi programmabili, mostrati in Figura 4.7(a) su cui andiamo ad agire ogni volta che programmiamo il dispositivo. I moderni FPGA sono programmati con sistemi basati su SRAM o memorie flash, come mostrato in Figura 4.7(b). Ad ogni cella di memoria è collegato il gate di un transistor NMOS, che opera come porta di trasmissione. Utilizzando la configurazione mostrata in Figura 4.7, si possono effettuare tutti i possibili collegamenti fra i quattro terminale N, W, E e S. Il vantaggio di disporre di un sistema riprogrammabile comporta una maggiore occupazione di area e con maggiori ritardi di propagazione rispetto alla tecnologia basata sui fusibili.

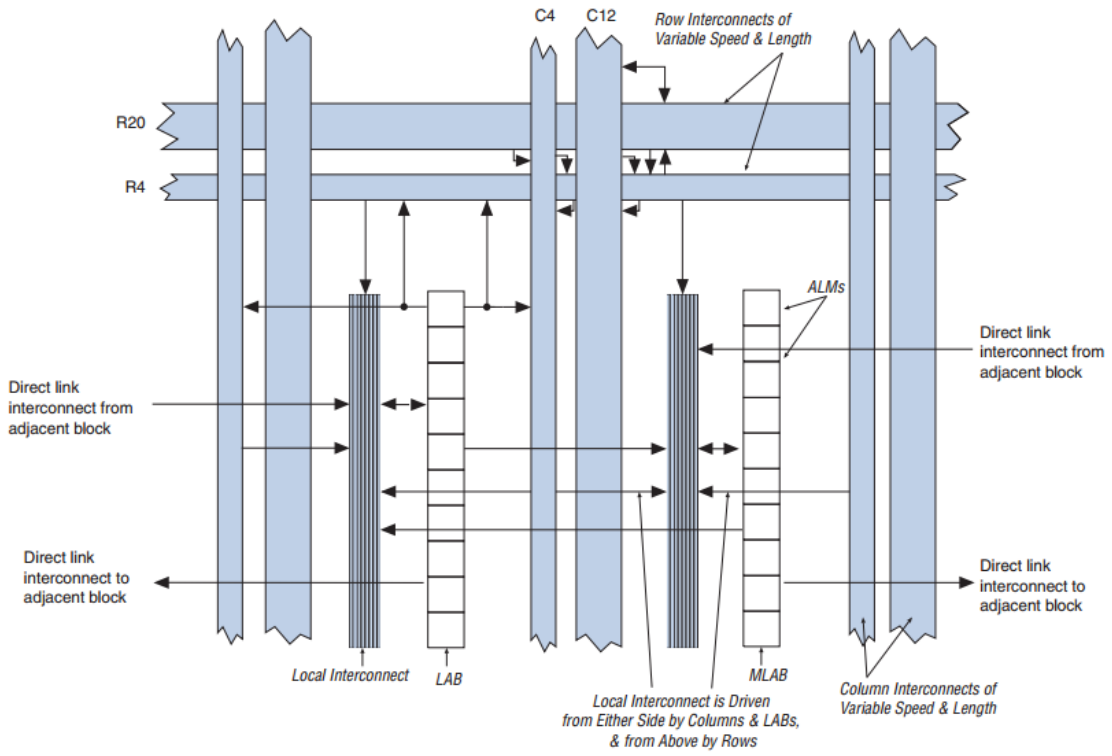
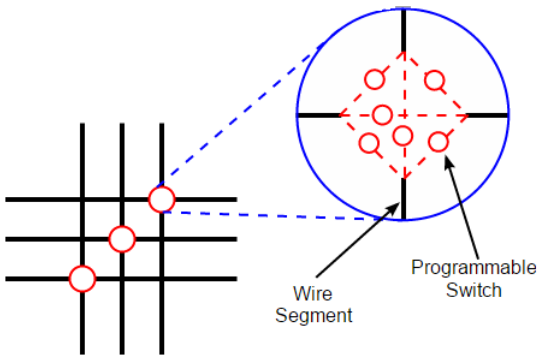
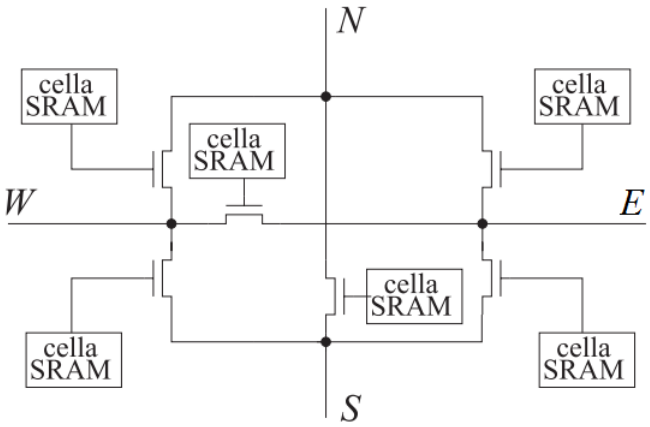


Figura 4.6: Schema a blocchi delle connessioni interne di un FPGA. Si nota come ogni blocco di LAB è connesso con se stesso, con quelli adiacenti e con i blocchi più distanti.



(a)



(b)

Figura 4.7: (a) un particolare dei nodi programmabili delle interconnessioni. (b) uno schema a blocchi della programmazione tramite SRAM.

La differenza nell'uso di SRAM o memorie flash sta nel fatto che le prime sono memorie volatili a differenza delle seconde, per cui è necessario avere una memoria non volatile esterna il cui contenuto viene trasferita nella SRAM una volta acceso l'FPGA.

4.3 I dispositivi a logica programmabile di Altera

Negli ultimi anni i settori di applicazione degli FPGA sono significamente aumentati. Alcune delle applicazioni più di successo riguardano il processamento di segnali digitali, l'*imaging* medico, la *computer vision*, la crittografia e le applicazioni militari. I maggiori produttori di FPGA sono le aziende Altera e Xilinx, presenti sul mercato con dispositivi di livello basso, intermedio ed alto. A parità di livello, i prodotti delle due aziende sono molto simili in termini di prestazioni.

Abbiamo scelto di realizzare il progetto della TPU su dispositivi Altera per due motivi:

- l'esperimento LHCb utilizza quasi esclusivamente FPGA Altera come suoi dispositivi a logica programmabile;

- la sezione di Pisa dell'Istituto Nazionale di Fisica Nucleare ha una lunga esperienza nell'utilizzo di dispositivi Altera, che include la progettazione di schede dedicate contenenti i suddetti dispositivi. Inoltre alcune di queste schede sono ottimi candidati per la futura realizzazione pratica del nostro progetto.

Le tre serie di FPGA Altera, dalla fascia più bassa a quella più alta sono: Cyclone, Arria, Stratix. In Tabella 4.1 sono riassunte alcune delle principali caratteristiche delle tre famiglie. La serie Stratix è stata scelta per l'applicazione della TPU, poiché

	Cyclone V	**Arria V**	**Stratix V**
Blocchi logici	110000	500000	952000
Celle (ALM)	41500	190240	359200
Memoria (kb)	13000	27000	62000
I/O pin	208÷560	416÷704	600÷840
Alimentazione (V)	1.1 V	0.85, 0.9 ,1.1	0.85, 0.9
PLL	8	24	28
DSP	36÷112	240÷1100	256÷2000
Clock massimo (MHz)	550	650	800
Costo ($)	30÷700	200÷3000	2000÷18000

Tabella 4.1: Caratteristiche di ogni famiglia di FPGA di Altera. Il confronto è stato fatto sulle ultime versioni, realizzate in tecnologia 28 nm, la più avanzata con cui siano stati sviluppati questi FPGA.

l'alto grado di parallelizzazione della TPU richiede un'elevata potenza di calcolo, che può essere offerta solo dagli FPGA della fascia più alta. L'algoritmo della TPU è stato sviluppato per gli FPGA Stratix III e gli Stratix V. Andremo nel dettaglio di questa scelta nei prossimi capitoli.

La cella logica principale ed il sistema di interconnessioni molto simili per tutti gli FPGA di Altera, e riportiamo nelle Figure 4.5 e 4.6 quelle inerenti al dispositivo Stratix V. Le ALM sono molto flessibili e possono essere configurate in varie modalità di funzionamento. Le memorie interne si dividono in blocchi interni (*embedded memories*) e blocchi logici, che combinati fra loro, formano le così dette MLAB. Queste memorie sono piccole (640 bit) rispetto alle altre (decine di kb) e risultano utili per implementare piccoli registri o LUT, senza intaccare le memorie interne, che possono essere utilizzate per altre applicazioni, ad esempio FIFO o grandi RAM.

Negli FPGA della fascia più alta diventa rilevante il consumo della potenza elettrica, per cui è necessario elaborare delle tecniche per ridurlo, diminuendo i consumi e le problematiche legate alla dissipazione del calore. Nel caso della serie Stratix, le innovazioni sono state sia a livello tecnologico, cioè nella progettazione del layout

dei transitori e nei materiali usati, in modo da diminuire gli effetti dovuti ad elementi parassiti, e nel minore voltaggio usato dal core (0.9V o 0.85V), sia a livello di programmazione. Altera ha infatti sviluppato un nuovo strumento in grado di regolare la potenza consumata da ogni blocco interno, in base ai requisiti del progetto. In questo modo possiamo limitare il maggior consumo di potenza ai soli percorsi che hanno vincoli temporali stringenti, riducendo il consumo rispetto ad avere tutti percorsi di questo tipo, come si vede in Figura 4.8.

Ogni tipo di FPGA è inoltre venduto in tre tipologie di velocità (*speed grade*), a seconda dei requisiti di timing dell'utente.

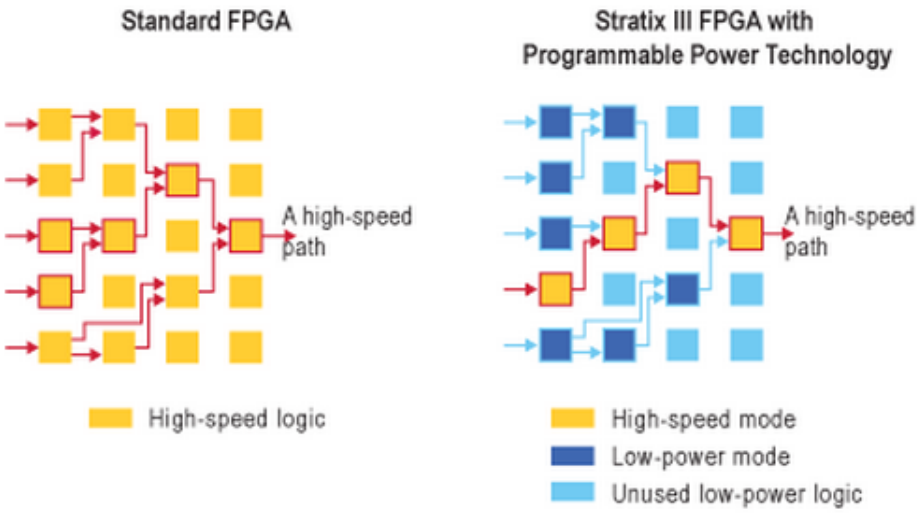


Figura 4.8: Confronto tra un FPGA standard ed uno Stratix III. Questa tecnologia, chiamata *Programmable Power Technology*, comporta una sostanziale riduzione nel consumo di potenza.

Le tre famiglie si differenziano molto oltre che per la densità di logica, anche per la presenza di interfacce I/O più sofisticate. In generale ogni FPGA è in grado di gestire più standard elettrici, come CMOS, TTL, segnali differenziali (LVDS, LVDT) e di gestire diversi protocolli di comunicazione per la presenza di blocchi specializzati, come serializzatori. Le prestazioni di questi serializzatori cambiano a seconda della famiglia: per esempio quelli della famiglia Cyclone sopportano fino a 6 Gbs in uscita, contro i 28.5 Gbs della Stratix V.

Per maggiori dettagli sulle logiche di Altera si consiglia di consultare la referenza [56].

4.3.1 Software per la progettazione del firmware

I dispositivi logici programmabili sono tali in quanto l'utente può, in modo semplice, programmare il dispositivo. Il processo non è molto diverso da quello utilizzato nella programmazione classica (C, java ecc.): si parte da una descrizione ad alto

livello che, attraverso un programma software specifico, viene convertito e codificato in un linguaggio "comprensibile" ai dispositivi. I linguaggi più utilizzati per la descrizione dell'hardware (*Hardware Description Language, HDL*) sono il VHDL [57] (*Very high speed integrated circuits Hardware Description Language*) ed il Verilog. Tali linguaggi differiscono dai linguaggi di programmazione poiché il codice non rappresenta una lista di istruzioni da eseguire sequenzialmente, ma una descrizione del funzionamento della logica, definendo i vari blocchi e le loro connessioni. In questo senso, il VHDL risulta essere un linguaggio parallelo e non sequenziale.

I due linguaggi VHDL e Verilog si differenziano per lo standard seguito: mentre il VHDL segue una propria sintassi diversa da altri linguaggi, il Verilog è sintatticamente più simile al linguaggio C. Il codice HDL è esportabile con facilità: i blocchi logici creati possono essere utilizzati in altri progetti, senza alcuna modifica, e uno stesso codice può essere sintetizzato per una vasta gamma di dispositivi. Ad esempio un codice scritto in solo VHDL può definire la logica da utilizzare sia in un FPGA sia in un ASIC.

Altera ha sviluppato il pacchetto di software Quartus II che permette di elaborare il progetto in tutte le sue fasi, dalla descrizione della logica nei diversi standard fino alla generazione del file che serve a programmare il dispositivo.

QuartusII [58] è utilizzato per lo sviluppo del codice VHDL, e la gestione delle risorse dell'FPGA, come le porte di I/O, la disposizione della logica, il controllo dei segnali interni e altro ancora. Il flusso di programmazione utilizzato in Quartus segue lo schema di Figura 4.9:

1. descriviamo il progetto attraverso un linguaggio HDL;

2. procediamo con una simulazione funzionale (*behavioural*) eseguita direttamente del codice scritto, in cui si tiene conto solamente delle operazioni descritte e non della disposizione dei blocchi sul componente;

3. se la simulazione risulta corretta procediamo con la sintesi del progetto, interpretando la descrizione a livello di porte logiche;

4. possiamo eseguire un'ulteriore simulazione funzionale (*register tranfer level*, RTL) per verificare la correttezza della sintesi, seppur anche questa non tenga conto dei ritardi introdotti dalla logica e dalle connessioni utilizzate;

5. se anche questa simulazione risulta corretta, eseguiamo il *place and route*, ovvero ogni elemento logico viene piazzato sul componente ed implementato attraverso le celle logiche dell'FPGA;

6. la simulazione eseguita a questo livello tiene conto della struttura interna dell'FPGA e di come sono state utilizzate le risorse interne. In questo caso, simulando il progetto, vengono tenuti in considerazione i ritardi di propagazione dei segnali attraverso le porte logiche e le connessioni;

7. se la simulazione è corretta, possiamo generare il file di programmazione dell'FPGA, contenente il Bitstream con cui vengono programmate le celle logiche e le linee di interconnessione, nei modi descritti nei paragrafi precedenti;

8. il file viene caricato sull'FPGA, che andrà ad eseguire le operazioni che sono state definite nel punto 1.

Nel caso in cui le simulazioni non risultano corrette, è necessario andare a modificare il codice che descrive la logica da implementare. Questa suddivisione è adottata dalla maggior parte dei software di programmazione. Nella compilazione è possibile aggiungere esternamente delle condizioni che vogliamo far rispettare, come ad esempio vincoli temporali tra ingresso ed uscita, che vengono interpretati da Quartus, modificando il piazzamento dei blocchi sull'FPGA.

Quartus fornisce vari strumenti per analizzare il modo in cui è stato sintetizzato il progetto nell'FPGA, ad esempio la visualizzazione della disposizione dei blocchi logici utilizzati; è inoltre possibile gestire completamente l'assegnazione dei pin di ingresso/uscita; si possono anche monitorare alcuni segnali interni all'FPGA, una volta che questo è stato programmato per facilitare la diagnostica.

Ultimata la compilazione, Quartus produce un report dettagliato sul consumo di logica dei singoli blocchi del progetto, oltre ad una stima precisa sulla temporizzazione dei segnali utilizzati, in particolare fornisce un limite superiore per la massima frequenza di clock che l'FPGA può garantire per quel determinato progetto.

Sviluppare un progetto con il supporto di questi software presenta evidenti vantaggi pratici ed economici. È infatti possibile prevedere la fattibilità e le prestazioni del nostro progetto con grande precisione, senza dover ogni volta programmare ed eseguire dei test sul dispositivo reale.

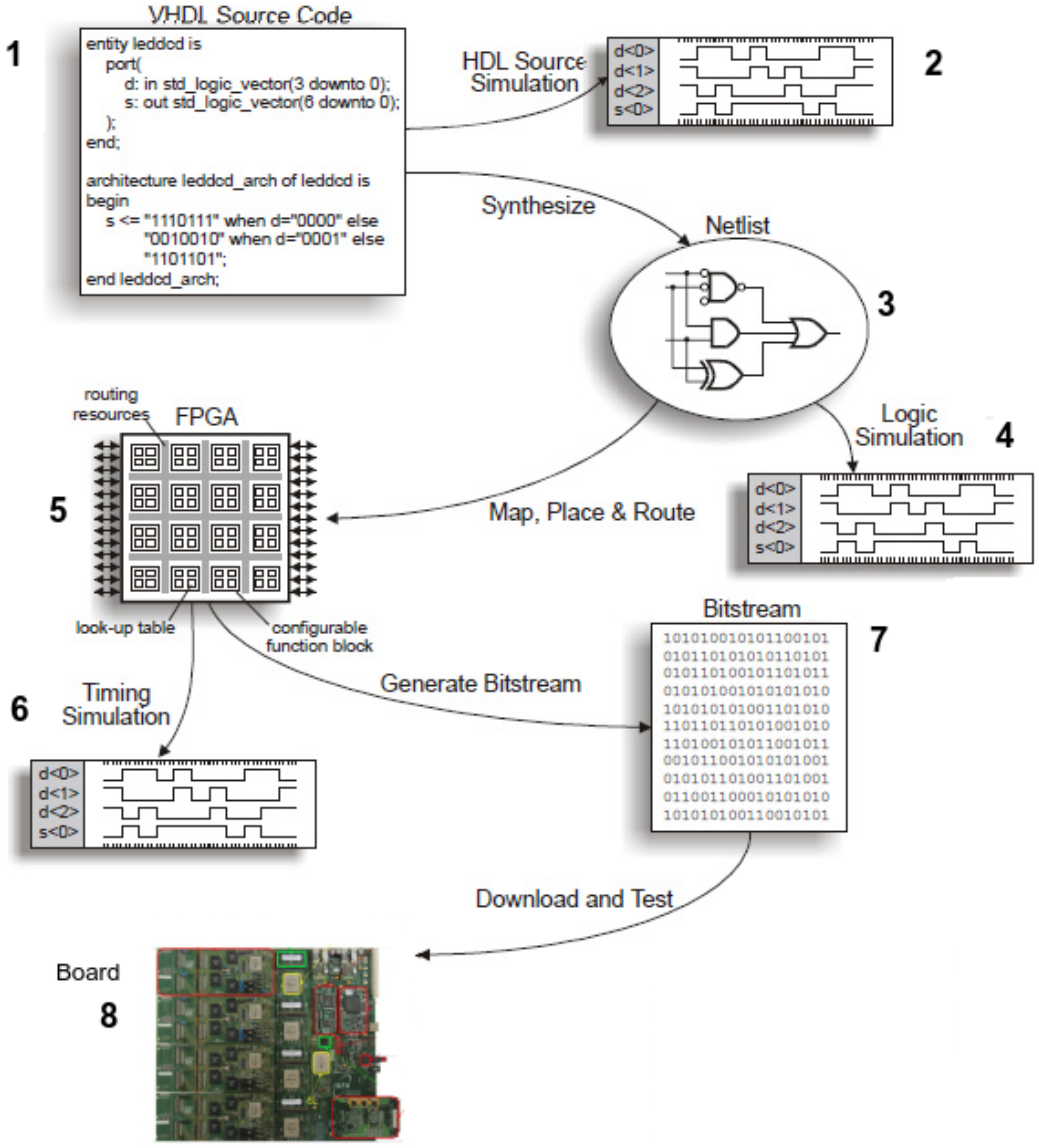


Figura 4.9: Flusso della programmazione di un FPGA tramite i software forniti da Altera. I dettagli delle varie fasi, contrassegnati dai numeri, sono riportate nel testo.

5 Applicazione all'Inner Tracker di LHCb

In questo capitolo verrà descritta un'implementazione della TPU da applicare all'Inner Tracker (IT) di LHCb. Di seguito andremo ad illustrare la parametrizzazione del rivelatore, la realizzazione del firmware, e i risultati ottenuti, confrontandoli con quelli della simulazione di alto livello.

5.1 Parametri della TPU applicata all'Inner Tracker

5.1.1 Caratteristiche dell'Inner Tracker

Il tracciatore a microstrip di silicio IT è situato nella regione esterna al campo magnetico (si veda il Capitolo 2.2.1 per maggiori dettagli). Il rivelatore è formato da 3 stazioni, ognuna contente 4 piani. I piani sono distribuiti in 4 sezioni disposte intorno alla beam pipe, due moduli laterali e due assiali (Figura 5.1). In ogni stazione il primo e l'ultimo piano hanno le strips di silicio orientate verticalmente, mentre i piani centrali hanno le strips orientate di +5° e -5° rispetto alla verticale. Questa configurazione identifica un set di coordinate che indichiamo con x-u-v-x. La frequenza di lettura degli eventi è di 1 MHz. I beetle chip (si veda la sezione 2.2.1) leggono i segnali da tutti i canali dei vari moduli. L'informazione in formato digitale viene inviata allo stadio successivo che effettua la soppressione dei canali compatibili con zero e crea i cluster calcolando anche il relativo baricentro. Come in tutto l'esperimento, il sistema di riferimento usato per descrivere le le tracce è destrorso con l'asse z coincidente con la linea di fascio, l'asse x punta verso il centro dell'acceleratore, l'asse y punta in alto lungo la verticale.

5.1.2 Parametrizzazione del rivelatore

La struttura descritta sopra permette un'implementazione modulare per la TPU, utilizzando due sistemi distinti basati su retina, uno dedicato ai moduli laterali ed

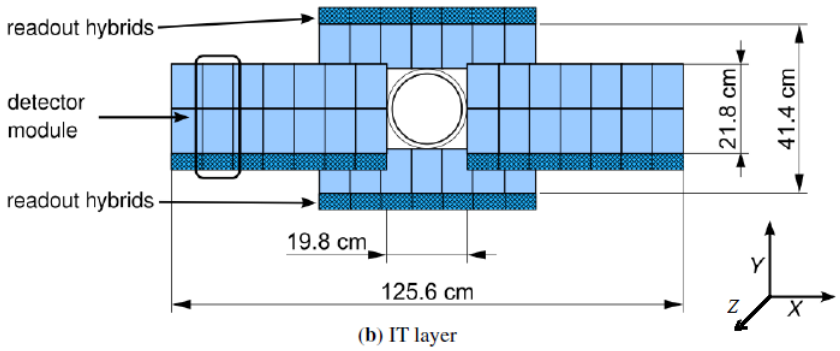


Figura 5.1: Schema di uno strato del rivelatore IT nel piano xy.

uno per i moduli in alto e in basso. L'applicazione descritta in seguito riguarda l'utilizzo dei soli moduli laterali.

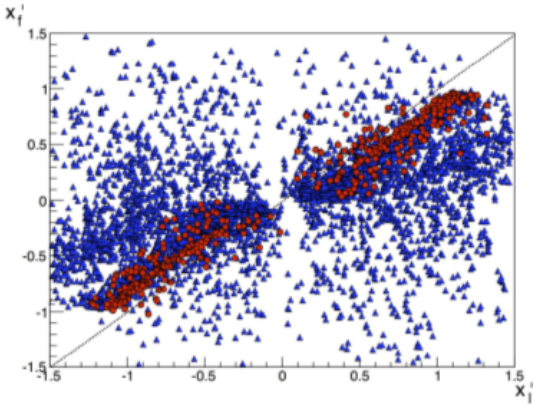
Abbiamo deciso di effettuare la ricostruzione delle tracce proiettandole sul piano xz, ed utilizzando come unica variabile la sola coordinata x di intersezione delle tracce sui piani del rivelatore. Assumendo che la curvatura sia trascurabile per i nostri scopi, per cui consideriamo tracce rettilinee, e che ogni traccia provenga dal vertice nominale dell'interazione, la sua proiezione sul piano xz di curvatura corrisponde univocamente al valore p_{xz} del momento iniziale della particella. Quindi, la ricostruzione delle tracce sul piano xz utilizzando la sola coordinata x di intersezione, fornisce una misura della componente p_{xz} del momento trasversa al campo mmagnetico.

Una semplice parametrizzazione per tracce rettilinee può essere effettuata andando a considerare le coordinate x dei punti di intersezione tra il primo e l'ultimo piano del rivelatore e la traccia, indicate rispettivamente con x_f e x_l. In questo modo abbiamo ottenuto uno spazio dei parametri bidimensionale.

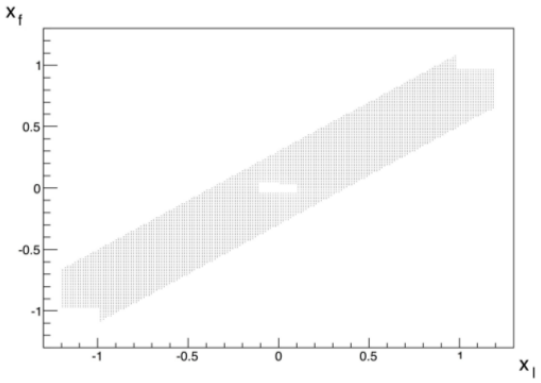
Le tracce ricostruite con alta precisione, chiamate *longable*, sono quelle che attraversano il VELO, il TT e le stazioni IT, ovvero hanno hit in ciascuno dei suddetti rivelatori. Queste tracce sono quelle preferite per la ricostruzione di eventi nelle analisi dati. Analizzando la distribuzione delle tracce in funzione dei parametri x_f e x_l, riportata in Figura 5.2(a), possiamo notare come le tracce longable, rappresentate in rosso, si dispongano lungo la bisettrice del piano, ovvero tendano a formare un angolo piccolo con l'asse z. Per questo motivo, abbiamo deciso di mappare solo una regione intorno alla bisettrice contenente il 95% delle tracce longable, mostrata in Figura 5.2(b). In accordo con gli studi effettuati dalla simulazione di alto livello, questa regione può essere efficientemente mappata utilizzando ~ 6000 engines [53].

5.1.3 Calcolo dei pesi

Tramite studi effettuati con la simulazione di alto livello, la parametrizzazione che abbiamo descritto può produrre una grande eccitazione di alcune aree della retina,



(a)



(b)

Figura 5.2: (a) parametri delle tracce che intersecano l'IT. I punti in rosso rappresentano le tracce longable, i triangoli blu le tracce non longable. (b) La mappatura dei parametri effettuata per l'IT.

in cui non è semplice distinguere il contributo dovuto alla traccia vera. Infatti una traccia che interseca un recettore con coordinate (x_r, z_r) sarà comune a tutte le tracce che intersecano il piano z_r in x_r con qualunque andamento lungo y. Un primo approccio studiato per ridurre questo effetto è stato effettuato andando a modificare l'algoritmo del calcolo dei pesi, introducendo l'algoritmo dei doppietti. Questo comporta di raggruppare i piani del rivelatore in gruppi di 2, applicare la soglia ad ogni doppietto e sommare i valori ottenuti in una retina finale. Nella configurazione scelta per l'IT, con 6 piani, avremo in totale 3 doppietti.

5.2 Scelta del dispositivo

La progettazione della TPU è stata portata avanti con lo scopo di implementarla su una scheda già esistente, in modo da ridurre i costi e il tempo di sviluppo della parte hardware e per avere più forza lavoro da dedicare allo sviluppo del firmware. La scheda è la TEL62, sviluppata da un gruppo della sezione pisana dell'INFN per l'esperimento NA62. La scheda è equipaggiata con Stratix III, FPGA prodotti da Altera. La scheda è stata sviluppata a partire dall'attuale scheda di readout di LHCb, la TELL1, mantenendo la completa compatibilità con il sistema di acquisizione dati di LHCb.

5.2.1 Stratix III

Gli FPGA con cui sono equipaggiate le schede TEL62 sono della famiglia Stratix, in particolare gli Stratix III EP3SL200F1152C4N. Il prodotto è stato lanciato sul mercato intorno al 2006, disegnato in tecnologia da 65 nm, e risulta, ad oggi, un FPGA con caratteristiche elevate disponibili con un prezzo contenuto. Riportiamo in Tabella 5.1 le capacità in termini di blocchi logici e di risorse di memoria, dell'FPGA montato sulle TEL62, lo Stratix III EP3SL200F1152C4N.

Stratix III EP3SL200F1152C4N			
Blocchi logici	200k	**Configurazione MLAB**	16×8
Celle (ALM)	80k	**640bit per le ROM**,	16×9
Memorie 9k	468	**320bit per altri usi**	16×10
Memorie 144k	36		16×16
Memorie MLAB	4000		16×18
Totale memoria (kbit)	10646		16×20
Alimentazione (V)	1.1, 0.9		

Tabella 5.1: Caratteristiche dello Stratix III EP3SL200F1152C4N.

In questo caso le memorie interne sono di tre tipi: 9k, 144k ed MLAB. Le MLAB sono utilizzabili nelle configurazioni mostrate in Tabella 5.1.

5.2.2 La scheda di readout TEL62

La scheda TEL62 [59] è un'evoluzione della scheda di readout TELL1, attualmente usata nell'esperimento LHCb. La TEL62 è equipaggiata con dispositivi più performanti, che le conferiscono:

- logica interna circa 8 volte superiore per ogni FPGA;
- memoria interna circa 5 volte superiore;
- frequenza massima di clock circa 4 volte superiore.

Il progetto ha mantenuto la compatibilità meccanica ed elettrica con la TELL1.

La scheda è dotata di 5 FPGA Altera della serie Startix III (EP3SL200F1152C4N): 4 FPGA di pre-processamento (PP), ed 1 di sincronizzazione, chiamato *sync link* (SL), disposti in una configurazione a stella. Ogni PP è collegata ad una mezzanino attraverso un connettore a 200 pin, dal quale la TEL62 riceve dati su 4 bus da 32 bit a 40 MHz, e si interfaccia con un memoria RAM DDR2 (*double data rate*) di 2 GByte. Le PP sono connesse tra loro attraverso 2 bus di comunicazione da 16 bit, ed ognuna è connessa all'FPGA SL attraverso due bus indipendenti da 32 bit, con una frequenza di 160 MHz, uno riservato per i dati ed uno per le informazioni di trigger nell'applicazione di NA62. L'SL è collegata ad un mezzanino di uscita con un altro bus ad alta velocità. Una memoria QDR da 1 Mb è usata come buffer intermedio tra la SL e l'uscita. Il mezzanino di uscita è lo stesso utilizzato nella TELL1, il "Quad-GbEthernet", dotato di 4 connessioni Ethernet ad 1 Gbit.

Lo *slow control* e i segnali di configurazione della scheda, vengono gestiti da altri 2 mezzanini: un mezzanino commerciale equipaggiato con PC (*credit card PC*, CCPC), che utilizza linux come sistema operativo, ed una scheda custom di input/output, chiamata *Glue Card*, connessa al CCPC con un bus PCI.

La Glue Card gestisce tre diversi protocolli di comunicazione, usati per tutti i dispositivi della TEL62: JTAG, I2C e ECS. Il JTAG è usato per configurare tutti i dispositivi della scheda da remoto; l'I2C è principalmente usato per configurare i registri dei mezzanini di ingresso, mentre l'ECS è un protocollo custom che utilizza bus a 32 bit usato per un accesso veloce ai registri interni degli FPGA della TEL62. Il segnale di clock arriva dall'esterno tramite fibra ottica, ma c'è la possibilità di generarlo direttamente sulla scheda per effettuare dei test. Il firmware degli FPGA, è memorizzato in due EEprom da 64 Mbit, una per l'SL ed una per le 4 PP; le 4 PP vengono quindi programmate con il medesimo firmware.

Tutte le linee del circuito stampatosono controllate con un impedenza caratteristica di 50 Ω. I voltaggi richiesti per generare le tensioni richieste dai componenti della scheda sono: 48V, 5V, 3V, +A5V, -A5V.

5.3 Progettazione logica della TPU

Per avere la possibilità di effettuare test parassitici su un esperimento reale, la prima implementazione della TPU è stata progettata per ricostruire le tracce usando i dati

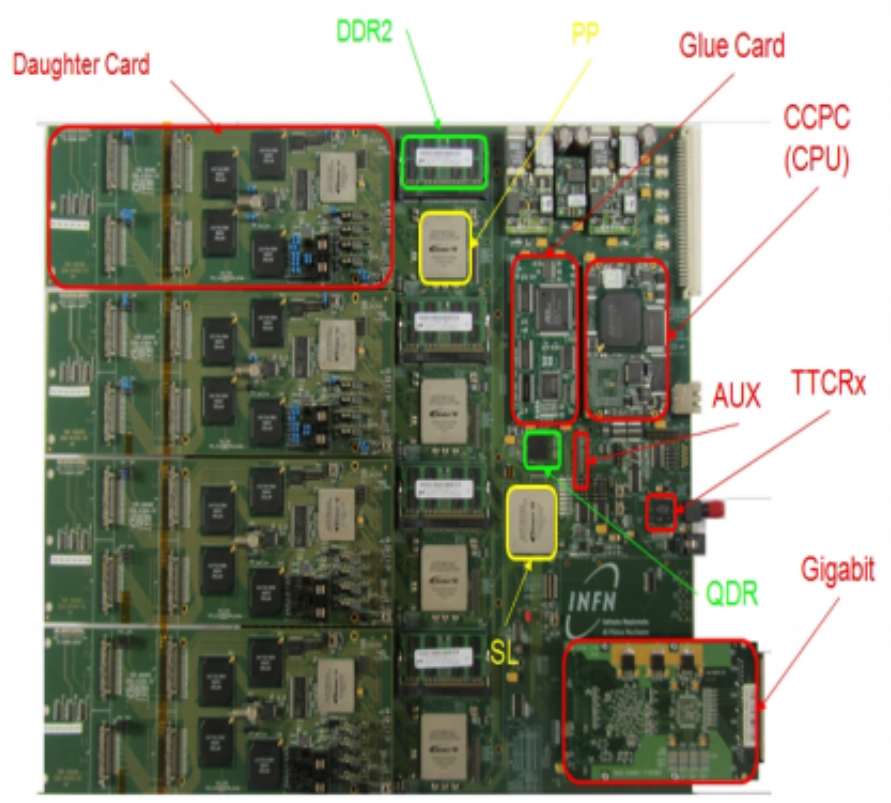


Figura 5.3: La scheda TEL62 equipaggiata con i mezzanini usati in NA62.

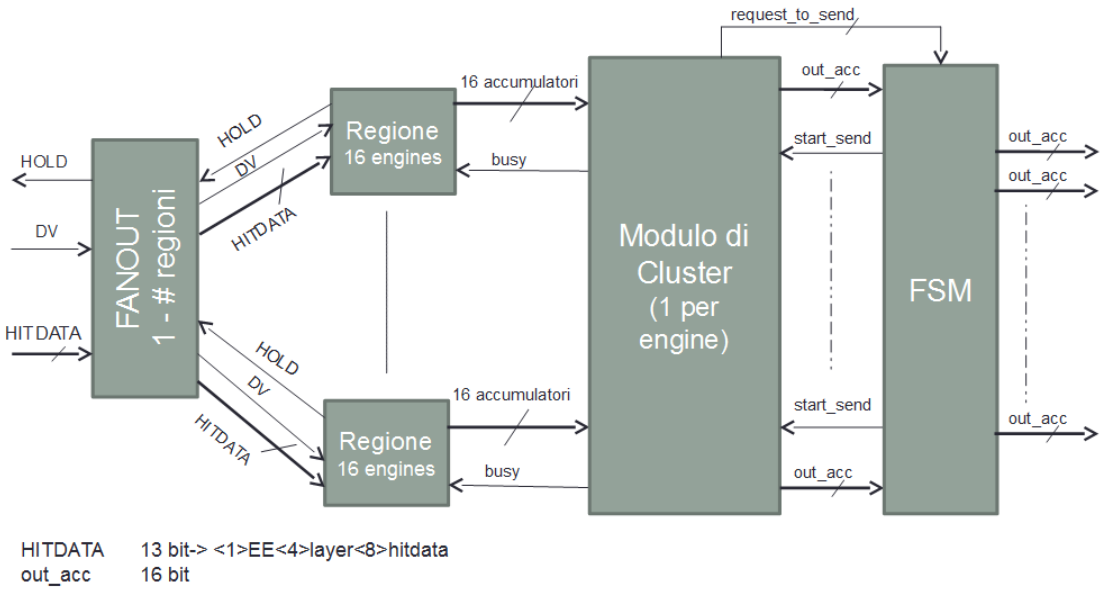


Figura 5.4: Diagramma a blocchi della TPU applicata all'IT.

del tracciatore IT dell'esperimento LHCb. La lettura dei dati digitalizzati è effettuata tramite fibre ottiche, i cui segnali possono essere divisi attraverso dispositivi ottici, ottenendo così una copia esatta senza interferire con la regolare acquisizione dei dati.

Il firmware della TPU per l'applicazione all'IT è stato realizzato sulla base della descrizione presentata nella sezione 3.3, con l'uso del software di sviluppo di Altera Quartus II versione 13.1. Al momento pensiamo che l'ultimo stadio dello switch e gli engine possano essere contenuti nelle PP, mentre il calcolo dei parametri della traccia e la formattazione dei dati possa avvenire all'interno della SL.

Come è mostrato in Figura 5.4, la TPU è formata da 4 blocchi principali:

- un fanout iniziale, che distribuisce gli hit a tutti gli engine;
- le regioni, al cui interno vengono gestiti il protocollo per i dati in ingresso e il calcolo dei pesi relativi all'hit in ingresso;
- i moduli per la ricerca dei cluster;
- un modulo finale per la gestione dei dati in uscita.

Fanout

In questa prima fase di progettazione, abbiamo sostituito lo switch con un semplice fanout, perché il nostro scopo è verificare la funzionalità dell'engine. Inoltre struttura dello switch per l'IT e i percorsi da assegnare ad ogni hit sono ancora in fase di

studio. Quindi il primo modulo è un semplice fanout che distribuisce i dati a tutti i moduli della TPU.

La divisione in regioni è stata mantenuta per avere un progetto compatibile con gli sviluppi futuri.

5.3.1 Formato dati in ingresso

I dati in ingresso alla TPU, contengono le informazioni relative a ciascun hit, necessarie per il corretto calcolo dei pesi:

- la coordinata x dell'hit;
- il numero del piano a cui l'hit appartiene;
- indicatore che l'evento è terminato.

Abbiamo deciso di usare 8 bit di risoluzione per la coordinata x, perché così possiamo memorizzare i valori dei recettori nelle MLAB utilizzando la configurazione standard di 16×8 bit, come mostrato in Tabella 5.1. Inoltre, 8 bit sono pienamente sufficienti per descrivere un hit. Infatti una regione di 2 recettori adiacenti, nella nostra mappatura, corrisponde ad una distanza di 0.179 cm lungo la coordinata x, e poiché la risoluzione spaziale dell'IT è di $\sim 50\,\mu$m, il minimo numero di bit per rappresentare un hit è:

$$\log_2\left(\frac{1790\,\mu\text{m}}{50\,\mu\text{m}}\right) = 5.2 \tag{5.1}$$

Possiamo definire quindi una dato formattato in una parola di 13 bit come segue:

`<12>EE<11..8>layer_identifier<7..0>coordX.`

Sono stati utilizzati 4 bit per l'identificatore dei piani, per avere la flessibilità nella scelta del numero di piani da utilizzare nella TPU. Nello studio fatto abbiamo scelto di utilizzare i 6 piani con le strip verticali.

La fine dell'evento viene identificata con una parola speciale, detta di *End Event* (EE) in cui il tredicesimo bit è 1 e tutti gli altri zero (in esadecimale: "0x1000"). Tale parola viene inviata una volta terminati gli hit relativi ad uno specifico evento.

5.3.2 Protocollo per la gestione dei dati in ingresso

All'interno delle regioni è stato implementato il protocollo per la gestione del flusso degli hit in ingresso agli engine. Il protocollo è stata realizzato con una macchina a stati finiti, il cui diagramma è mostrato in Figura 5.5. In ingresso alla TPU arrivano dall'esterno i dati corredati di un bit di validità (*Data_Valid*, DV) Analizziamo la struttura del protocollo:

- in S0 viene processato un hit, standard, senza EE, distribuendolo agli engine della regione;
- in S1 si processa la parola di EE, quando i moduli degli engine possono processarla, ovvero senza la presenza di una condizione di busy. Spiegheremo poi in dettaglio a cosa corrisponde tale condizione. In questo stato si alza un bit chiamato *EE_fsm*, utilizzato negli stadi successivi dell'engine. Inoltre si attiva un contatore che alza un bit di busy per 7 cicli. Descriveremo in seguito il motivo di questa condizione. La macchina sta un ciclo in S1, dopodiché torna in S0;
- in S2 si processa l'arrivo di un EE quando è presente una condizione di busy. La TPU non può processare ulteriori eventi, per cui blocchiamo il flusso di dati in ingresso, generando verso gli stadi precedenti un segnale *hold_fsm*. Quando la condizione di busy viene a mancare, la macchina a stati si porta in S1 per completare il processamento dell'EE.

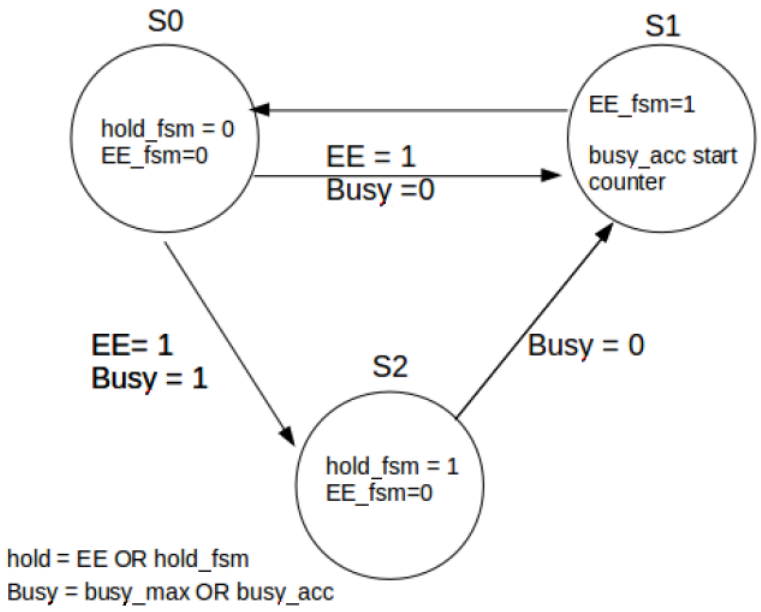


Figura 5.5: Schema a blocchi della macchina a stati finiti utilizzata per il controllo del protocollo per la gestione del flusso dati in ingresso.

La condizione di *Hold* fornita ai moduli precedenti è il risultato dell'OR logico tra i bit di EE e di hold_fsm. Questo è necessario perché l'hold_fsm viene generato un ciclo dopo l'arrivo dell'EE e l'HOLD sarebbe letto dai moduli esterni con un ciclo di ritardo. La condizione di *Busy* è invece l'OR di due bit di busy: uno generato dal contatore menzionato nello stato S1, relativo al modulo per il calcolo dei pesi, l'altro è generato invece dal modulo per il calcolo dei cluster. Per semplicità ci riferiremo

nelle prossime sezioni alla macchina a stati che controlla il protocollo dei dati in ingresso come *modulo DV-Hold*.

5.3.3 Modulo per il calcolo dei pesi

I dati in ingresso alle regioni vengono inviati in parallelo agli engine che la compongono. Il primo modulo di un engine esegue il calcolo del peso inerente a ciascun hit, ed è stato implementato come una *pipeline* sincrona, mostrata in Figura 5.6, in cui le varie operazioni sono eseguite in sequenza, sul fronte di salita del segnale di clock.

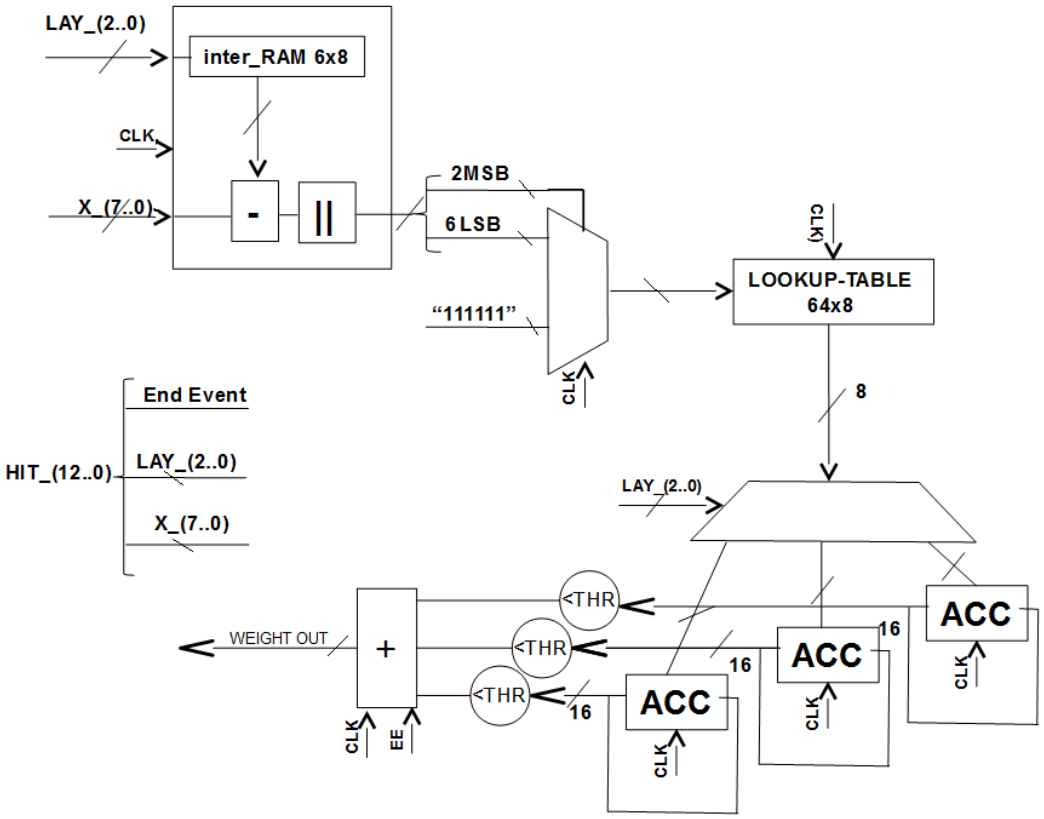


Figura 5.6: Il diagramma a blocchi della pipeline usata per il calcolo dei pesi.

L'hit in ingresso viene sottratto dal valore del recettore, memorizzato in una RAM indirizzata dall'identificatore del piano. I 6 bit meno significativi (LSB) del risultato sono usati come indirizzo per la look-up-table dei pesi, solamente se i 2 più significativi sono uguali a 0, altrimenti si forza l'indirizzo al massimo valore possibile, che equivale ad avere un peso nullo. Il peso viene quindi sommato in tre diversi accumulatori da 16 bit, a seconda del doppietto di appartenenza del piano. Una volta che l'evento è finito, si applica una soglia prefissata ad ogni accumulatori di doppietti per poi sommarli in un unico valore. La somma viene copiata un in buffer, il cui

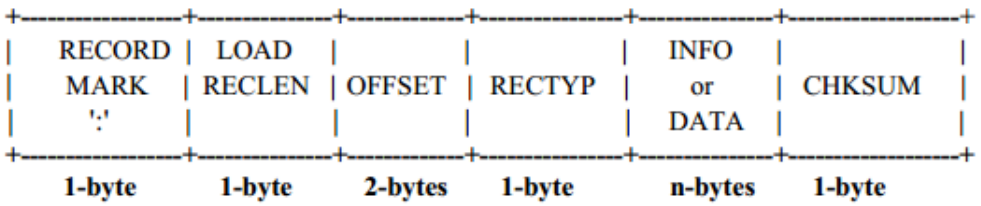


Figura 5.7: Codifica dello standard Intel Hex.

valore sarà usato nella ricerca dei cluster. In questo modo è possibile processare un evento di seguito all'altro, poiché il calcolo dei pesi e la ricerca dei cluster lavorano su due copie differenti degli accumulatori. Una volta copiato il valore della somma, si azzerano gli accumulatori per l'evento successivo. La struttura a pipeline non permette di processare altri EE fino a che la somma degli accumulatori non è stata copiata nei buffer; per questo motivo viene generato un segnale di busy all'arrivo di ogni EE, per un numero fissato di cicli di clock, come spiegato nella sottosezione precedente.

Le RAM sono inizializzate con un file esterno creato con uno script in C, contenente i valori delle intersezioni o dei pesi codificate nel formato ***Intel Hex***, mostrato in Figura 5.7, uno standard usato per la programmazione di microcontrollori e memorie. I valori dei recettori sono calcolati a partire dai valori calcolati dalla simulazione ad alto livello, codificati in una parola di 8 bit, mentre i pesi sono calcolati a partire da una gaussiana avente la stessa varianza usata nella simulazione ad alto livello. La funzione peso è discretizzata in un massimo di 64 valori da 8 bit. Le dimensioni delle RAM sono quindi di 6×8 bit per le intersezioni e di 64×8 bit per i pesi, utilizzando le MLAB dello Stratix III in una delle configurazioni riportate nella Tabella 5.1.

5.3.4 Modulo per la ricerca massimi locali

Al termine dell'evento i valori degli accumulatori copiati nei buffer vengono usati dai moduli di ricerca dei massimi. In questo progetto abbiamo un modulo di ricerca del massimo per ogni engine, per tenere bassa la latenza totale del dispositivo. La ricerca del massimo viene gestita attraverso l'uso di una macchina a stati, di cui mostriamo il diagramma in Figura 5.8. Analizziamo nel dettaglio il funzionamento della macchina:

- la macchina viene attivata tramite il bit EE_fsm (sezione 5.3.2);
- in S1 viene alzato un segnale di busy verso il modulo DV-Hold, e si ricevono in ingresso il peso dell'engine relativo al modulo di ricerca del massimo e quelli dei suoi vicini. Dobbiamo verificare se l'engine è il massimo di un quadrato 3×3, in quanto stiamo lavorando in uno spazio dei parametri bidimensionale. Si procede quindi ai confronti tra i pesi. Se l'engine risulta essere un massimo locale la macchina si porta in S2, altrimenti si torna allo stato di attesa S0;

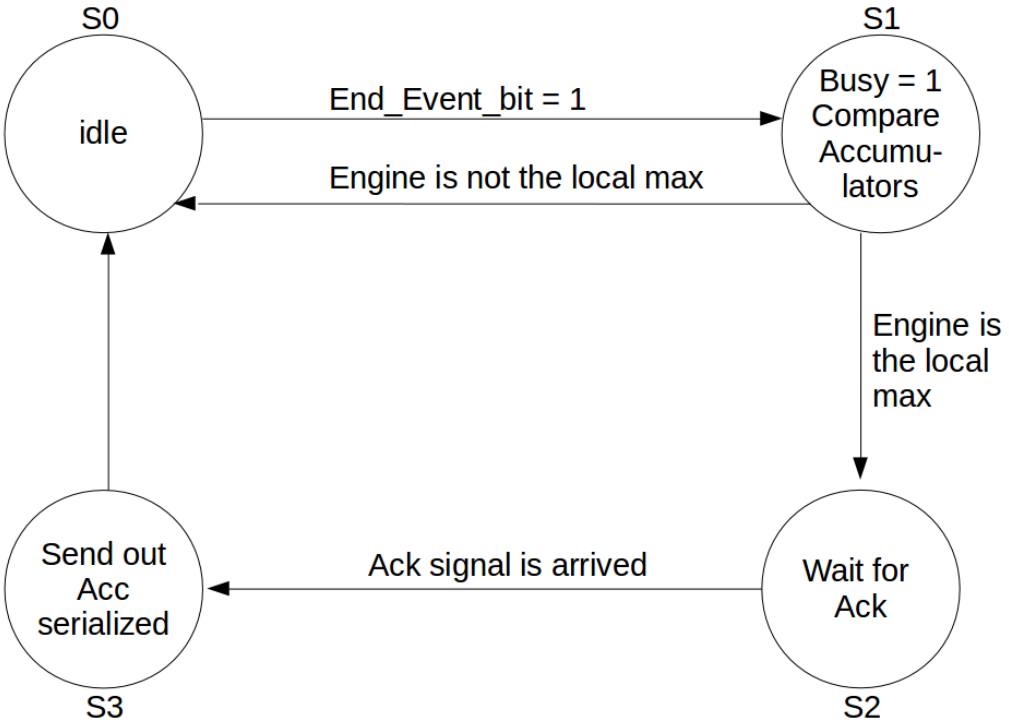


Figura 5.8: Il diagramma a blocchi della macchina a stati che implementa la ricerca dei cluster.

- in S2 si invia verso l'esterno un bit di *Request*[1] per richiedere l'invio dei valori del cluster. Quando si riceve dall'esterno il segnale di *Acknowledgment* (*Ack* in Figura) ci portiamo all'ultimo stato;

- in S3, si inviano serialmente i valori dei pesi degli engine appartenenti al cluster. Finita l'operazione di invio, la macchina si riporta in S0, in attesa dell'arrivo di un altro EE.

5.3.5 Protocollo per la gestione dei dati in uscita

Una macchina a stati, posta a valle di tutta la TPU, gestisce i segnali di Request e Acknowledgment verso i moduli per la ricerca dei massimi, emulando la presenza del modulo per il calcolo dei parametri. Il protocollo usato prevede che con l'arrivo dei Request venga generato un Acknowledgment comune a tutti i moduli, supponendo di poter inviare in uscita contemporaneamente tutti i bus dei moduli che hanno trovato un massimo locale.

[1]Il risultato dei confronti consiste in un bit, che vale 1 se il confronto è positivo. L'AND logico questi bit forma il segnale di Request.

Non è stato implementato in questa versione nessun modulo per il calcolo dei parametri, cioè del baricentro del cluster per non eccedere nel consumo di logica e per non avere una latenza troppo elevata. Difatti, come vediamo nell'equazione 3.4, il calcolo contiene una divisione, che in hardware richiede diversi cicli di clock e registri a scorrimento per essere eseguita. Questa operazione, al momento si pensa che possa essere eseguita all'interno dell'FPGA SL della TEL62. Un possibile modo per facilitare il calcolo del baricentro e la velocità di comunicazione, è di calcolare già all'interno dell'engine alcune somme, utili per il passo successivo. Se infatti andiamo a sostituire nell'equazione 3.4 il valore del parametro scritto come $m_i + k \cdot dm_i$, otteniamo, esplicitando il calcolo rispetto ad un engine laterale del cluster:

$$\tilde{m_i} = m_i^0 + (k-1) \cdot dm_i + dm_i \cdot \frac{\sum_{n,m} 2 \cdot W_{k+1,m} + W_{k,m}}{\sum_{n,m} W_{n,m}} \tag{5.2}$$

in cui:

- $\tilde{m_i}$ è un parametro della traccia del cluster trovato;
- m_i^0 è il parametro relativo ad un engine fissato;
- k rappresenta la coordinata dell'engine massimo all'interno della griglia nello spazio dei parametri;
- dm_i è il passo della retina nella dimensione del parametro considerato;
- $W_{n,m}$ sono i pesi degli engine del cluster,
- n, m assumono valori da 1 a 3.

In questa configurazione, possiamo ridurre l'informazione in uscita dalla TPU, calcolando $\sum_{n,m} 2 \cdot (W_{k+1,m} + W_{k,m})$ per ogni parametro e $\sum_{n,m} W_{n,m}$. Questi possono essere fatti uscire su bus a 20 bit e k su un bus ad 8 bit. I valori dm_i e m_i^0 sono delle costanti e si possono memorizzare allo stadio successivo. In questo modo potrebbe essere possibile ridurre i tempi di calcolo della TPU. Questa alternativa è ancora in fase di studio, e non implementata nella versione della TPU descritta in questa tesi.

5.3.6 Parametri di configurazione

La TPU è stata progettata configurando alcuni parametri. Come prima cosa abbiamo deciso di organizzare gli engine in una matrice quadrata, divisa in regioni da 4×4 engine ciascuna. Il numero totale di engine per FPGA è di 256, equivalente a 4×4 regioni.

I moduli per la ricerca dei cluster sono implementati fuori dalle regioni, per semplicità. Abbiamo un totale di 14×14 moduli, poiché gli engine che sono al bordo della matrice non appartengono ad un cluster completo.

Per quanto riguarda la scelta del numero di bit per le coordinate, dipendono dalla granularità della retina,nel modo mostrato ad inizio della sezione. Sia la granularità

che la σ della funzione peso, sono stati ottimizzati tramite la simulazione di alto livello.

Con questa configurazione occupiamo circa l'85% delle risorse logiche dello Stratix III. La maggior parte delle risorse sono utilizzate per effettuare i collegamenti tra i moduli per il calcolo dei pesi e per la ricerca dei cluster, in quanto si tratta di dover ridistribuire in modo corretto tutte le 256 uscite degli accumulatori.

Tutta l'architettura è stata concepita per essere modulare e flessibile; infatti si ha la possibilità di controllare le dimensioni dei vettori, il numero di engine e il numero di bit degli hit, attraverso alcuni parametri di configurazione. Al momento, la flessibilità non è totale, in quanto si possono avere, per semplicità nell'implementazione, solamente matrici quadrate di regioni quadrate, controllando la dimensione dei lati dei quadrati, come mostrato nella Tabella 5.2.

	$n = \sqrt{Engine\_in\_a\_region}$ $m = \sqrt{Number\_of\_regions}$	$n = 4$ $m = 2$	$n = 4$ $m = 4$
Engine in una Regione	n^2	16	16
Numero di Regioni	m^2	4	16
Numero di Engine	$n^2 \times m^2$	64	256
Numero di Cluster_Module	$(n \cdot m - 2)^2$	36	196

Tabella 5.2: Esempi di parametrizzazione del numero di engine implementabili in un modulo di TPU. n è la $\sqrt{\cdot}$ del numero di engine in una regione, m è la $\sqrt{\cdot}$ del numero di regioni.

5.4 Simulazione della logica della TPU

La simulazione logica della TPU è stata effettuata grazie all'utilizzo di Modelsim Altera, versione 10.1, con un livello di astrazione chiamato *Register-Transfer Level* (RTL), in cui esegue una simulazione funzionale del dispositivo in considerazione. Tramite la simulazione sono state studiate le prestazioni della TPU in termini di latenza e velocità, e in termini di risoluzione sulla ricerca dei cluster, in confronto con la simulazione di alto livello.

Gli eventi usati nei test seguenti sono stati generati in parte simulando singole tracce che attraversano una versione semplificata del rivelatore e in parte utilizzando la simulazione Montecarlo ufficiale di LHCb per il 2015.

5.4.1 Impostazione dei segnali in ingresso

I test su di un modulo scritto in VHDL prevedono "l'allestimento" di un banco di prova virtuale, detto *testbench*[2], con una struttura simile a quella mostrata in

[2]Nei firmware scritti in HDL, il testbench è un una struttura che fornisce al circuito in esame, i segnali di ingresso e raccoglie le uscite a tali stimoli. In questo senso la simulazione in VHDL è

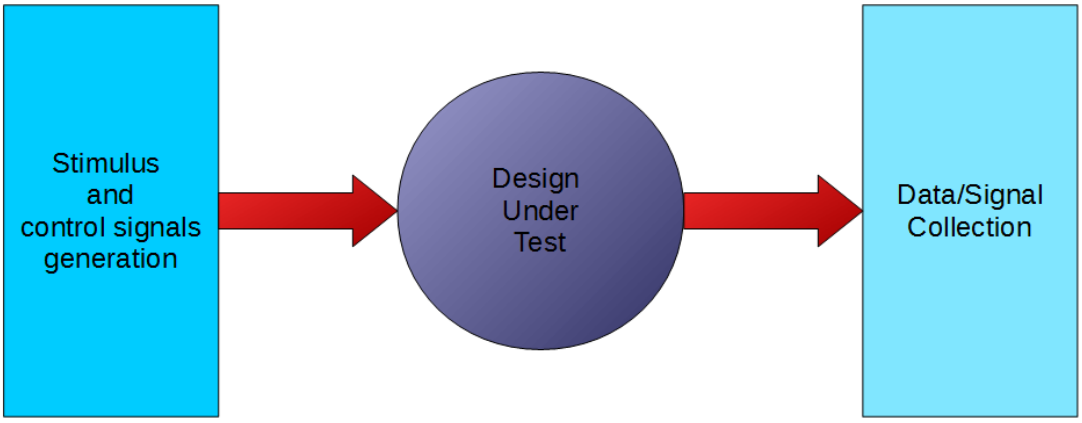


Figura 5.9: Schema a blocchi del funzionamento di un testbench. Il circuito sotto esame si trova inglobato all'interno di una struttura di test che fornisce gli stimoli esterni e raccoglie le risposte del circuito.

Figura 5.9.

Nel nostro caso il testbench è formato da 3 blocchi principali:

- un parte di generazione degli hit, che legge i dati da un file esterno e li invia alla TPU, corredati ognuno del segnale di DV. Questa parte è gestita da una macchina a stati sincrona, che rispetta il protocollo di gestione dei dati illustrato nella sottosezione 5.3.2;
- il firmware completo della TPU;
- una macchina stati sincrona che raccoglie i segnali di uscita della TPU e li scrive su un file, per le successive analisi.

All'interno del testbench si generano anche i segnali di clock e reset, distribuiti a tutti i moduli così da sincronizzare le operazioni tra i vari blocchi. Il segnale di reset viene inviato all'inizio della simulazione, per portare tutto alla configurazione iniziale. La frequenza di clock scelta è di 100 MHz, valore che può essere modificato in tutto il range di frequenze supportato dall'FPGA.

5.4.2 Prestazioni temporali del dispositivo

Le caratteristiche temporali studiate sono la latenza totale del dispositivo, andando a misurare quanti cicli di clock necessitano ad ogni modulo per completare le

come fare un esperimento virtuale, in cui il circuito fisico è rappresentato dal suo codice VHDL ed il testbench rappresenta un circuito stampato con una postazione vuota, su cui inserire il circuito sotto esame.

funzioni, e il *throughput*, ovvero la frequenza a cui si riescono a processare eventi consecutivi. La frequenza massima a cui possiamo lavorare è stata calcolata utilizzando lo strumento di analisi di Quartus II chiamato *Timing Analyzer*. Risulta che una TPU con 256 engine implementata nello Stratix III, può operare con una frequenza di clock massima di ~ 150 MHz, come riportato in Figura 5.10.

Slow 1100mV 85C Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name
1	154.44 MHz	154.44 MHz	CLOCK

(a)

Slow 1100mV 0C Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name
1	161.86 MHz	161.86 MHz	CLOCK

(b)

Figura 5.10: Frequenze massime permesse per la TPU con 256 engine su Stratix III applicata al rivelatore IT. I due valori sono stati calcolati tramite il Timing Analyzer di Quartus II, per per temperature di lavoro di 85°C (a) e 0°C (b).

In Figura 5.11 è mostrata una simulazione del modulo per la ricerca del massimo, per un evento formato da una singola traccia. Notiamo che la latenza tra l'arrivo dell'EE e la copia della somma degli accumulatori nei buffer è di 10 cicli di clock. Sempre dalla figura vediamo che la pipeline ha una lunghezza di 7 cicli di clock. L'inizio non corrisponde con l'ingresso dei dati nella TPU, poiché ci sono dei buffer intermedi nel fanout e nelle regioni per permettere la sincronizzazione dei segnali.

Consideriamo adesso il modulo di ricerca dei massimi locali: la latenza tra l'arrivo dell'EE e l'uscita dell'ultimo valore del cluster è di 21 cicli di clock, come vediamo in Figura 5.12. Nel nostro esempio questa è la massima latenza della TPU, supponendo la configurazione descritta nella sezione 5.3.5. Nel caso in cui un evento non presenti nessun engine sopra soglia, la latenza del sistema si riduce ad 11 cicli di clock.

L'ultima misura di tempo effettuata riguarda il throughput della TPU, ovvero la frequenza media a cui possiamo ricevere un hit in ingresso. In questo caso abbiamo utilizzato un campione di 1000 eventi presi dalla simulazione Montecarlo ufficiale di LHCb per il 2015, che prevede rispetto al Run del 2012 un aumento dell'energia nel centro di massa da 8 TeV a 14 TeV. Il tempo medio di processamento di un evento risulta essere di $\sim$ 36 cicli clock. Confrontando questo risultato con la latenza massima della TPU di 21 cicli, in questo caso il limite sul throughput dipende dal numero medio di hit in un evento e dalla lettura degli stadi successivi alla TPU, ma non dall'architettura attualmente implementata. Il massimo throughput che

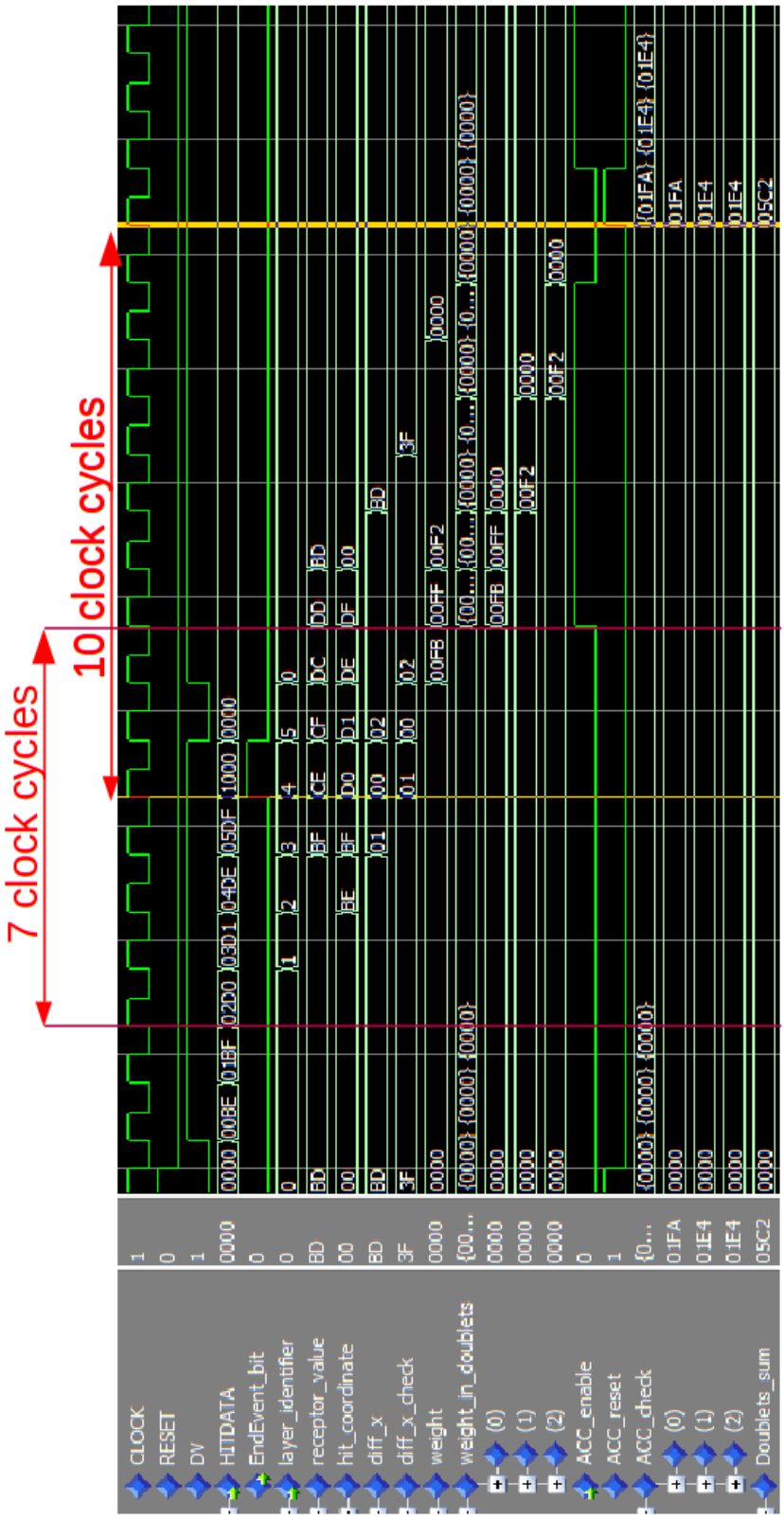


Figura 5.11: La simulazione Modelsim del modulo per il calcolo dei pesi, dall'ingresso dei dati nella TPU fino all'uscita degli accumulatori. La distanza tra le due linee gialle rappresenta la latenza tra l'arrivo dell'EE e la copia degli accumulatori nei buffer. La distanza tra le linee viola rappresenta la lunghezza della pipeline per il calcolo dei pesi di Figura 5.6.

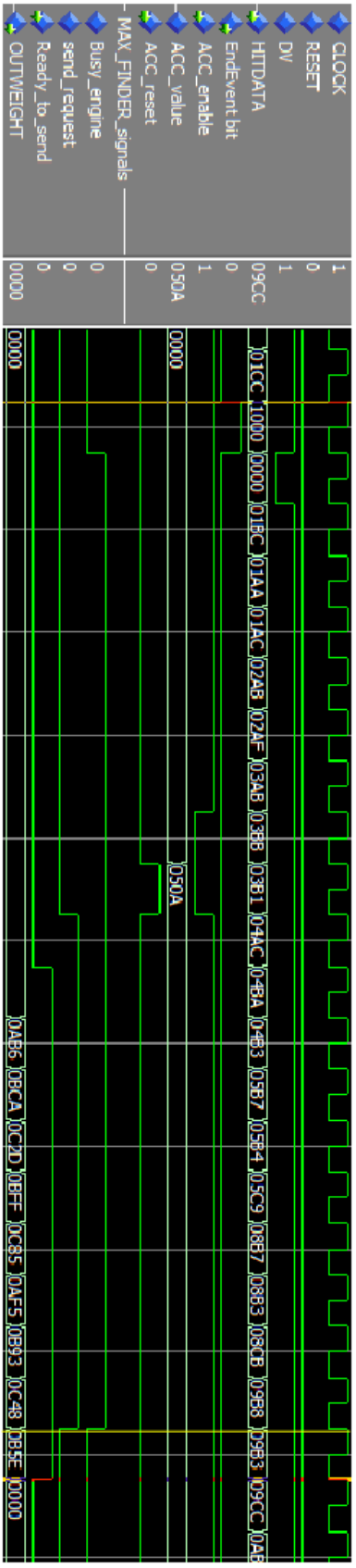


Figura 5.12: La simulazione ModelSim che include la ricerca dei massimi locali. La distanza tra le due linee gialle rappresenta la latenza totale del sistema, dall'arrivo dell'EE, fino all'uscita dell'ultimo valore del cluster.

	Cicli di Clock	t(ns) @150 MHz
Calcolo Pesi	10	67
Ricerca Cluster	21	140
Tempo per evento	36	240
Throughput = 4.1 MHz		

Tabella 5.3: Prestazioni temporali della TPU.

riusciamo a sostenere è quindi

$$f_{clk}/N_{cyc} = 150/36 \approx 4.1\,\text{MHz} \tag{5.3}$$

superiore a quella a cui lavora attualmente l'IT.

Riassumiamo quindi in Tabella 5.3 i risultati sulle prestazioni temporali della TPU.

5.4.3 Risposta della TPU ai segnali in ingresso

Per studiare la risposta della TPU ad un segnale di ingresso, abbiamo utilizzato due eventi, uno con una singola traccia ed uno con due tracce all'interno della porzione di spazio dei parametri mappato dai 256 engine. Nei grafici di Figura 5.13 sono riportati i valori della somma degli accumulatori dei doppietti per gli eventi di singola e doppia traccia, simulati ad alto livello ed a livello logico.

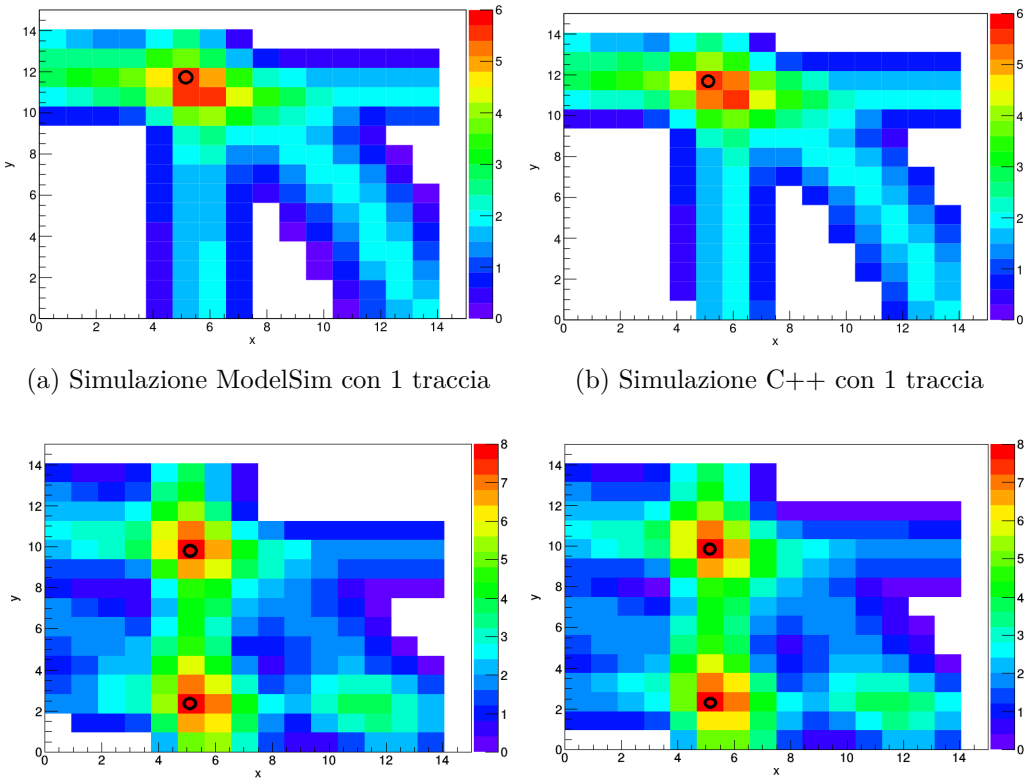
In questi grafici non è stata applicata nessuna soglia per i pesi. A questo livello vediamo che si trovano gli stessi cluster, ed i massimi sono nella stessa posizione in entrambe le simulazioni. Maggiori informazioni si possono ottenere dai grafici della differenza e del rapporto delle due simulazioni, mostrati in Figura 5.14.

Di seguito riportiamo i valori medi, con relativi errori, della differenza e del rapporto tra i valori degli accumulatori ottenuti dalle due simulazioni:

$$\begin{cases} modsim_{plot} - highlev_{plot}\big|_{1track} = 0.01 \pm 0.12; \\ \frac{modsim_{plot}}{highlev_{plot}}\Big|_{1track} = 0.99 \pm 0.17; \end{cases} \tag{5.4}$$

$$\begin{cases} modsim_{plot} - highlev_{plot}\big|_{2track} = 0.03 \pm 0.13; \\ \frac{modsim_{plot}}{highlev_{plot}}\Big|_{2tracks} = 0.93 \pm 0.22; \end{cases} \tag{5.5}$$

I due bin con valore prossimo a 2 in Figura 5.14(b) corrispondono ad accumulatori con valori simili al livello di approssimazione introdotto dall'aritmetica intera usata nei calcoli del dispositivo. Inoltre come si può vedere dalle Figure 5.13(a) e 5.13(b), il valore di tali bin è inferiore ad 1, quindi al di sotto della soglia che sarà applicata agli accumulatori.



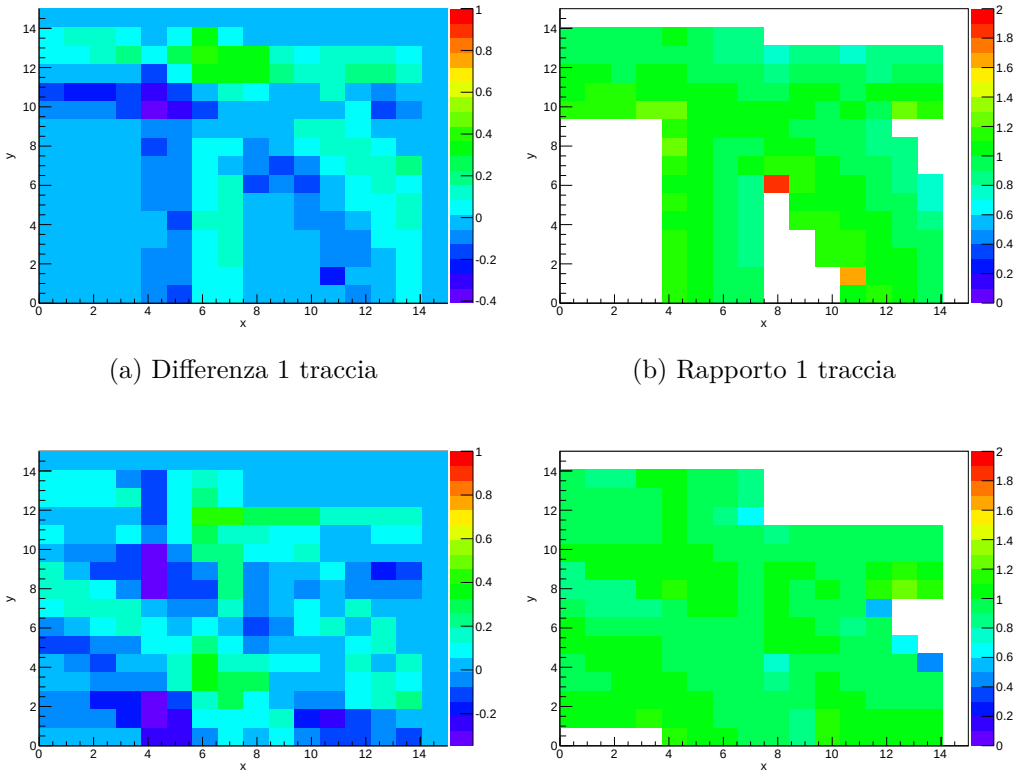
(a) Simulazione ModelSim con 1 traccia

(b) Simulazione C++ con 1 traccia

(c) Simulazione ModelSim con 2 traccia

(d) Simulazione C++ con 2 traccia

Figura 5.13: I grafici della risposta della retina, a livello di accumulatori, per eventi di particle gun a singola e doppia traccia, simulati con ModelSim e con la simulazione di alto livello in C++. I cerchi neri rappresentano la posizione dei massimi locali. Sugli assi sono riportate le coordinate degli engine all'interno della matrice considerata.



(a) Differenza 1 traccia

(b) Rapporto 1 traccia

(c) Differenza 2 traccia

(d) Rapporto 2 tracce

Figura 5.14: I grafici della differenza e del rapporto tra i valori degli accumulatori ottenuti con la simulazione ModelSim e con quella di alto livello, per gli eventi considerati. Sugli assi sono riportate le coordinate degli engine all'interno della matrice considerata.

I risultati mostrano che l'implementazione della TPU riproduce esattamente la simulazione di alto livello in termini di ricerca di massimi locali, in entrambi per entrambi gli eventi considerati. Non notiamo, in questa analisi, alcun degrado delle prestazioni dovuto alle approssimazioni numeriche effettuate nel dispositivo.

6 Applicazione ai rivelatori VELO e UT di LHCb

A partire dal 2020, LHC raggiungerà un'energia nel centro di massa di 14 TeV, una luminosità istantanea dell'ordine di $10^{34}\,\mathrm{cm}^{-2}\mathrm{s}{-}1$ ed una frequenza dei trigger di livello 0 di 40 MHz. Allo stesso tempo è previsto un miglioramento di tutti gli esperimenti situati presso LHC, per mantenere o aumentare le loro prestazioni in presenza delle nuove condizioni di luminosità. Anche LHCb, che avrà a disposizione una luminosità di $\sim 10^{-33}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ sostituirà alcuni rivelatori e schede di elettronica per la presa dati, come già descritto in sezione 2.3. Presentiamo in questo ultimo capitolo uno studio per applicare la TPU ai rivelatori VELO ed UT nella configurazione prevista per il 2020.

6.1 Parametri della TPU applicata a VELO e UT

I rivelatori VELO ed UT sono i rivelatori di traccia più vicini al punto di interazione (si veda il Capitolo 2 per maggiori dettagli). Il VELO, il rivelatore di vertice, è situato attorno al punto di interazione nominale dei fasci, in una zona priva di campo magnetico. L'UT si trova invece in una regione in prossimità del campo magnetico, di cui se ne percepiscono gli effetti.

La presenza del campo magnetico e la vicinanza al vertice delle tracce complica l'approccio della TPU rispetto al caso dell'IT, poiché la definizione delle tracce richiede l'uso di un numero maggiore di parametri. Infatti, oltre all'informazione degli hit sul detector, la posizione delle tracce cambia significativamente con la posizione del vertice primario l'impulso delle particelle. La parametrizzazione scelta per definire le tracce consta di 5 parametri, (u, v, z_0, d, k), dove:

- u, v sono le rispettive coordinate di intersezione x, y della traccia su un piano virtuale, che possiamo pensare come un piano aggiunto al rivelatore, ortogonale all'asse z, e posizionato tra i piani del rivelatore, come si vede dalla Figura 6.1;

- z_0 è la coordinata z del punto di massimo avvicinamento della traccia con l'asse z;

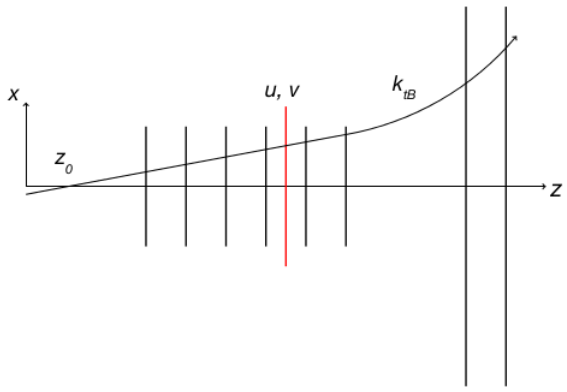


Figura 6.1: Illustrazione schematica dei parametri utilizzati per la configurazione VELO e UT. Il piano virtuale è rappresentato dalla linea rossa.

- d è il parametro di impatto trasverso con segno, definito come definito come $d = ((\vec{p_t} \times \vec{x_v}) \cdot \hat{z})/p_t$, dove p_t è la componente del momento trasversa rispetto alla direzione del fascio e x_v è la componente lungo l'asse x di d. In questo modo possiamo distinguere due tracce provenienti dallo stesso vertice secondario, aventi lo stesso modulo di d, come mostrato in Figura 6.2;

- $k = q/\sqrt{p_x^2 + p_z^2}$ è la curvatura della traccia con segno, ortogonale al campo magnetico, in cui q è la carica della particella e p_x e p_z sono le componenti del momento ortogonali alla direzione principale del campo magnetico ($\vec{B} = B\hat{y}$).

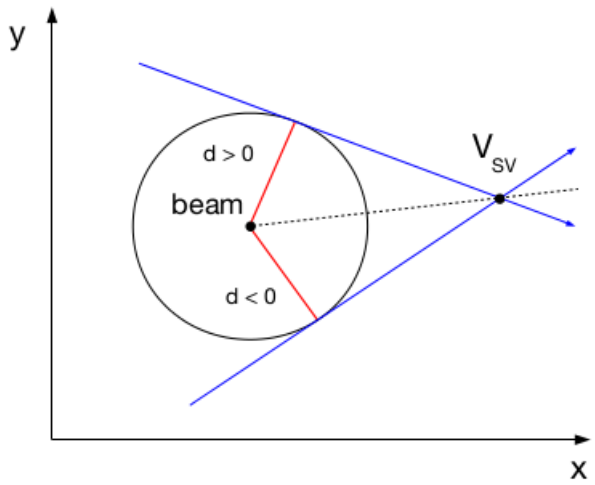


Figura 6.2: Rappresentazione del parametro di impatto trasverso con segno. Sono rappresentate in figura due tracce provenienti dallo stesso vertice secondario con lo stesso modulo di d e con segno opposto.

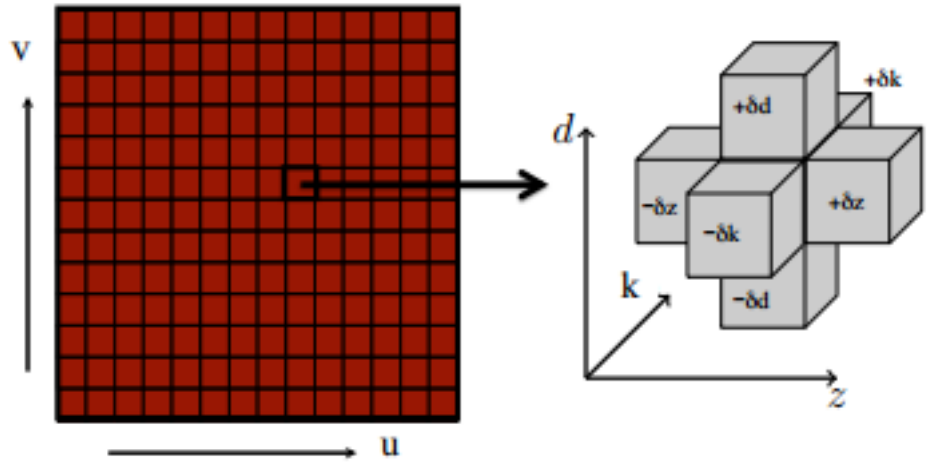


Figura 6.3: Rappresentazione dello spazio dei parametri considerato per l'applicazione VELO UT. La cella principale è affiancata da sei laterali, che contengono i valori perturbati.

Tuttavia una retina artificiale a 5 dimensioni è difficilmente realizzabile in pratica, visto l'alto numero di celle da implementare anche usandone un basso numero per ciascuna dimensione Se andiamo ad analizzare la topologia di una tipica traccia che intersechi i rivelatori scelti, vediamo che è possibile fattorizzare lo spazio delle fasi nel prodotto di due sottospazi di dimensione minore:

$$(u, v, d, z_0, k) \quad = \quad (u, v) \otimes (d, z_0, k), \tag{6.1}$$

poiché possiamo considerare la variazione dei parametri d, z_0, k come perturbazioni dei parametri u, v. Questo ci permette di lavorare nello spazio bidimensionale (u, v), in cui gli altri parametri sono nulli ($z_0 = d = k = 0$) e di trattare le perturbazioni come celle laterali. Come illustrato in Figura 6.3, abbiamo sei sottocelle per ogni cella principale nella spazio (u, v), corrispondenti alle variazioni di $(\pm\delta z_0, \pm\delta d, \pm\delta k)$.

Per ogni cella vengono calcolati i recettori sui piani dei rivelatori, per cui avremo un set di recettori per la cella centrale ed uno per ogni perturbazione, in accordo con le variazioni di d, z, k. In accordo con gli studi fatti nella simulazione di alto livello, lo spazio dei parametri può essere efficientemente mappato con ~ 22000 celle principali, all'interno della regione dello spazio dei parametri mostrata in Figura 6.4.

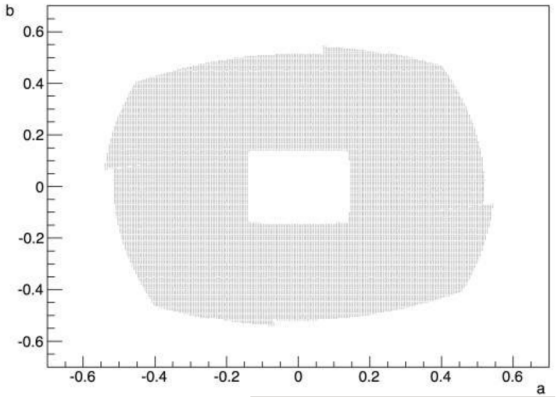


Figura 6.4: Rappresentazione della mappatura del piano (u v), nello spazio trasformato (si veda la sezione 3.4).

6.2 Scelta del dispositivo

Come illustrato nella sezione 4.3, abbiamo deciso di implementare il progetto della TPU applicata ai rivelatori VELO UT, sull'FPGA Stratix V di Altera [60]. La scelta è dovuta al fatto che il nuovo sistema di readout di LHCb prevede l'upgrade della scheda TELL1 alla scheda TELL40, la quale sarà equipaggiata con 5 FPGA Stratix V. Per un maggior dettaglio sulla TELL40 si rimanda alla sezione 2.3.

6.2.1 Lo Stratix V

Lo Stratix V è l'ultimo modello di FPGA della fascia più performante al momento prodotto da Altera. Questo FPGA è stato prodotto tecnologia a 24 nm, ed ottimizzato in particolare per applicazioni di ricetrasmittenza di segnali ad alta frequenza. Lo Stratix V montato sulle TELL40 sarà il 5SGXEA7N2F45C2ES. Riportiamo in Figura 6.1 le capacità in termina di blocchi logici e di risorse di memoria dell'FPGA considerato, confrontate con quelle dello Stratix III usate nella TEL62.

	Stratix III TEL62	**Stratix V TELL40**
Blocchi logici	200k	622k
Celle (ALM)	79k	234k
Memorie interne(Mb)	10	50
Memorie MLAB	4000	15k
Alimentazione (V)	1.1, 0.9	0.9, 0.85

Tabella 6.1: Caratteristiche dello Stratix V utilizzato sulla TELL40, confrontate con lo Stratix III montato sulla TEL62.

Lo Stratix V contiene memorie interne da 20 kb, che all'occorrenza possono essere divise in 2 blocchi indipendenti, con ingressi ed uscite separate. Questa nuova caratteristica verrà utilizzata nel nostro progetto delle TPU, come descriveremo nella prossima sezione.

6.3 Progettazione logica della TPU

L'implementazione della TPU nel caso dei rivelatori VELO UT è molto simile a quella per il rivelatore IT. Si rimanda quindi alla sezione 5.3 per le caratteristiche generali, mentre di seguito verrà posta particolare attenzione alle modifiche rese necessarie dalla diversa parametrizzazione della retina e dal differente dispositivo.

L'architettura proposta prevede la realizzazione di un'unica matrice di engine, che ricevono dati in ingresso tramite un semplice fanout, con uscite parallele per ciascun engine. In questa fase preliminare non abbiamo implementato né uno stadio di switch né un modulo per il calcolo dei parametri per i motivi illustrati nella sezione 5.3.

Come è mostrato in Figura 6.5, la matrice è formata da engine doppi, che chiameremo *double-engine*, al cui interno sono contenuti tre moduli principali:

- un modulo che implementa il protocollo di gestione dei dati in ingresso (DV_Hold in figura);
- il modulo per il calcolo dei pesi relativi ad ogni hit (pipeline in figura);
- il modulo per la ricerca dei cluster e dei massimi locali (max_module).

Notiamo alcune differenze rispetto al progetto per l'IT:

- non è stata effettuata nessuna suddivisione in regioni, per semplicità nella progettazione;
- gli engine sono implementati come moduli doppi, cioè ogni double-engine contiene al suo interno due engine;
- i moduli per la ricerca dei cluster sono implementati dentro i double-engine, per una gestione più semplice dei segnali;
- ciascun double-engine è inizializzato con un segnale che identifica le coordinate dei sui engine all'interno della matrice.

La scelta del double-engine è conseguenza della necessità di poter sfruttare al meglio le grandi memorie dello Stratix V, usando una singola memoria di tipo M20k per implementare le ROM in cui memorizzare i valori dei recettori e dei pesi di due engine.

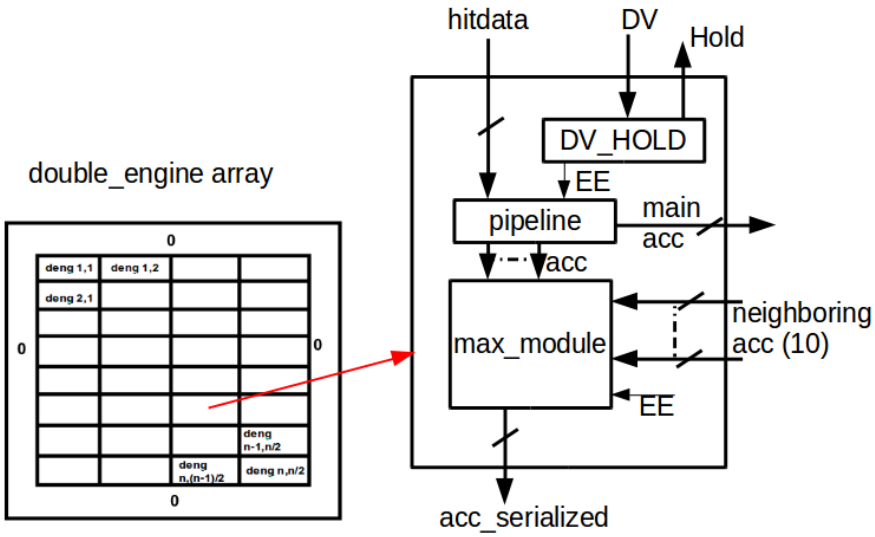


Figura 6.5: Schema a blocchi della TPU implementata.

6.3.1 Formato dati in ingresso

In ingresso alla TPU, i dati sono stati formattati in una parola contenente le informazioni necessarie per il corretto processamento degli hit:

- il valore delle coordinate (x, y) degli hit sui piani dei rivelatori;
- il numero del piano di appartenenza dell'hit;
- un valore di *timestamp*, che identifichi un evento dal punto di vista temporale;
- un indicatore della fine di un evento.

Abbiamo scelto di introdurre una timestamp per garantire la compatibilità con un futuro sistema di acquisizione dati che potrebbe fornire gli hit non ordinati temporalmente.

I dati in ingresso possono quindi essere codificati in una parola di 41 bit formattata come segue:

```
<40>EE<39..28>timestamp<27..24>layer_identifier<23..12>coordY<11..0>coordX.
```

Per non degradare la risoluzione del detector abbiamo usato 12 bit per identificare una coordinata. Il numero del piano di appartenenza dell'hit è un valore a 4 bit, che permette di utilizzare un numero di piani fino a 16, maggiore rispetto alla configurazione usata che prevede l'utilizzo di 6 piani del VELO e 2 piani appartenenti all'UT. Nella parola di EE, che identifica la fine di un evento, il bit di EE è fissato ad 1, il timestamp è quello che corrisponde a tale evento e il resto dei bit a 0, che

in esadecimale corrisponde ad esempio a 0x10TT0000000 (TT rappresenta il valore del timestamp).

6.3.2 Protocollo per la gestione dei dati in ingresso

Il flusso dei dati in ingresso viene gestito da un macchina a stati finiti. Come già illustrato in sezione 5.3.2, il controllo del flusso avviene attraverso la gestione dei segnali di *Data_Valid* e di *Hold*. Analizziamo gli stati della macchina, in riferimento al diagramma a blocchi mostrato in Figura 6.6:

- in S0, si aspetta l'arrivo degli hit;
- la macchina si porta in S1 quando si ha in ingresso un hit standard, ovvero senza EE. Viene quindi generata la condizione di hold alzando il segnale *hold_fsm* e tramite un contatore si incrementa il segnale *cell*, il numero della cella, che viene inviato agli engine. La macchina torna in S0 quando il numero di cella è uguale a 6.
- la macchina si porta in S2 quando arriva la parola di EE e non è presente nessuna condizione di Busy. Viene alzato il segnale di *hold_fsm* e si genera il bit di *start_max*, per attivare il modulo per la ricerca dei massimi locali. La macchina si mantiene in questo stato 7 cicli di clock, necessari a completare il calcolo dei pesi per tutte le celle laterali. Questo controllo si esegue incrementando la variabile *count* fino a 6;
- la macchina si porta in S3 se arriva un EE quando è presente la condizione di Busy. In questo caso, si alza il segnale di hold, e la macchina rimane in S3 finché persiste la condizione di Busy. Quando il segnale di *Busy* va a 0 ci portiamo in S2 per processare correttamente l'EE.

Il segnale di *Hold* che viene inviato all'esterno è l'OR logico tra i segnali *DV* e *hold_fsm*. Questa operazione è necessaria per bloccare il flusso di dati al ciclo successivo l'arrivo del DV. La condizione Busy viene generata dal modulo per la ricerca dei massimi quando il modulo si attiva.

6.3.3 Modulo per il calcolo dei pesi

Il dato che entra in ogni engine, viene memorizzato in un registro attivato dal DV e rimane all'interno dell'engine per 7 cicli di clock. Il modulo che calcola il valore dei pesi relativi agli hit è implementato come una pipeline sincrona, di cui mostriamo un diagramma a blocchi in Figura 6.7. In ciascun double-engine, sono implementate due pipeline, una per ciascun engine. Rispetto all'implementazione per l'IT, le parti aggiuntive sono necessarie in quanto le operazioni vengono effettuate per due coordinate su 7 celle.

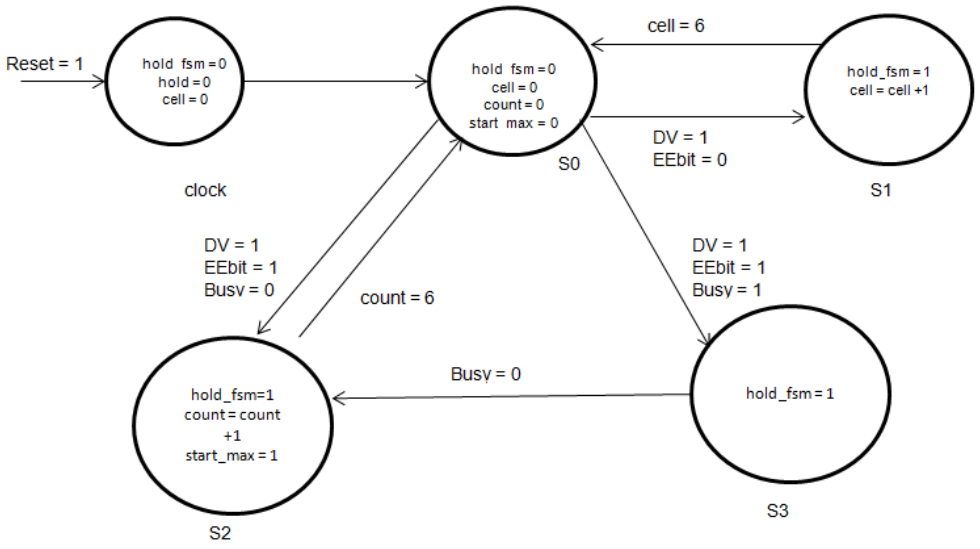


Figura 6.6: Diagramma a blocchi della macchina a stati per la gestione del flusso di dati in ingresso.

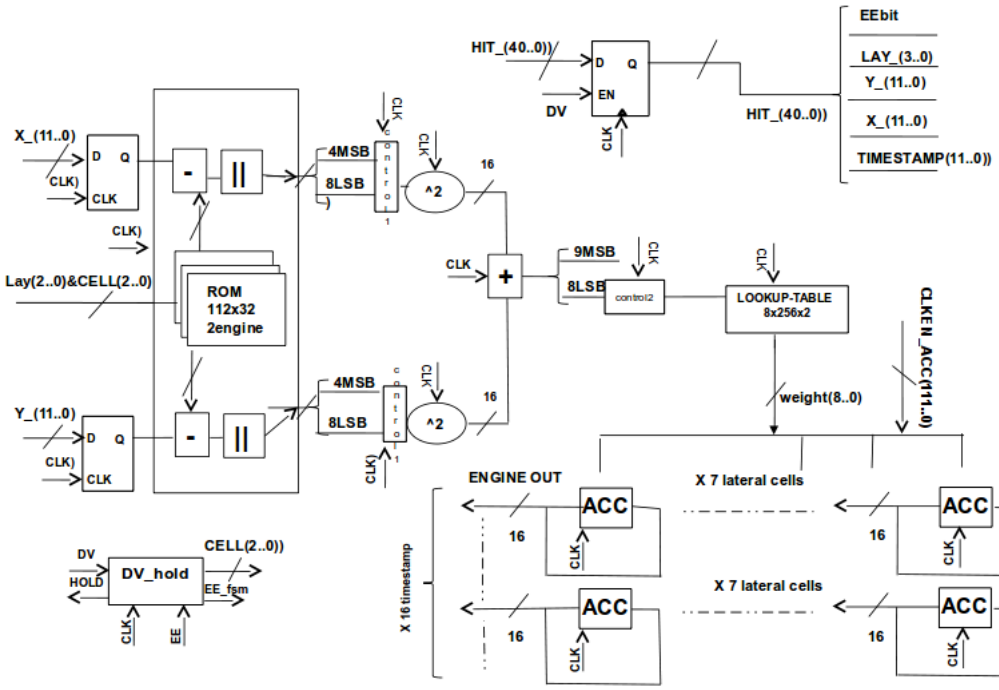


Figura 6.7: Schema a blocchi della pipeline usata in ogni engine per il calcolo dei pesi nella configurazione della TPU applicata ai rivelatori VELO e UT.

Le operazioni svolte sono le stesse descritte nella sezione 5.3.3, con la differenza che un hit viene processato per calcolare il peso relativo alle 7 celle considerate. Questo viene eseguito utilizzando come indirizzo della ROM, in cui sono stati memorizzati i valori dei recettori, una combinazione del numero di cella (si veda la sottosezione 6.3.2) e del numero del piano a cui l'hit appartiene. L'algoritmo prevede di calcolare prima il peso della cella centrale e poi quelli delle celle laterali, poiché la ricerca dei cluster viene effettuata sui valori delle celle centrali, cioè quelle con $d = z_0 = k = 0$. Si procede quindi al calcolo della distanza euclidea tra l'hit ed il recettore, secondo la formula $\delta^2 = (\Delta x)^2 + (\Delta y)^2$. Poiché non è necessario eseguire operazioni con un alto numero di bit, tronchiamo il numero di bit dei vettori usati nel calcolo della distanza prima di ogni operazione, e non una volta giunti al risultato finale. Questi troncamenti vengono fatti controllando se i bit più significativi scartati sono nulli. In tal caso si procede col calcolo normale, altrimenti il risultato viene forzato al valore massimo, corrispondente ad un valore nullo del peso. Il valore del quadrato della distanza è usata come indirizzo della look-up table che contiene i pesi, codificati in un vettore di 8 bit, i quali vengono memorizzati in accumulatori. Il numero totale di accumulatori usati risulta essere:

$$\#accumulatori = \#celle \cdot \#timestamp = 7 \cdot \#timestamp.$$

La scrittura su più accumulatori viene gestita generando un segnale di *enable* che ad ogni ciclo di clock abilita la scrittura sull'accumulatore giusto. Questo segnale viene generato come in Figura 6.8, utilizzando demultiplexer in cascata, controllati dal segnale di timestamp e dal segnale del numero di cella.

L'architettura è stata proposta in due versioni: una con 16 valori di timestamp ed una con 1 unico timestamp, per studiare quanto incida questo parametro nell'occupazione del chip. Il limite superiore di 16 valori di timestamp, equivalenti a 112 accumulatori, è solo una stima effettuata sulla base delle esperienze precedenti.

Le memorie utilizzate per i recettori e i pesi sono di tipo ROM. Queste vengono inizializzate con file nel formato Intel Hex (si veda la sezione 5.3.3). I recettori sono formattati in parole da 12 bit per ciascuna coordinata, a partire dai valori calcolati con la simulazione di alto livello. In totale abbiamo $7\,celle \cdot 8\,piani = 56\,recettori/engine$, per cui 112 parole in ogni double-engine. Il dato memorizzato è di 32 bit invece di 24, per facilitare la codifica del file intel hex; questo non crea problemi in termini di occupazione delle memorie, in quanto si sfrutta meno della metà della capacità dei blocchi di M20k dello Stratix V. I pesi sono stati mappati a partire da una funzione peso gaussiana con 256 valori da 8 bit. I file .hex sono stati creati col medesimo script, opportunamente modificato, usato per lo Stratix III.

La pipeline termina le sue operazioni con l'arrivo del segnale EE_fsm, con il quale si inviano sequenzialmente i valori degli accumulatori al modulo per il calcolo dei cluster, partendo da quello della cella laterale. Nel caso in cui si abbiano più timestamp, gli accumulatori vengono selezionati valori tramite un multiplexer controllato dal timestamp. Una volta trasmesse le informazioni, gli accumulatori relativi a quel timestamp vengono resettati.

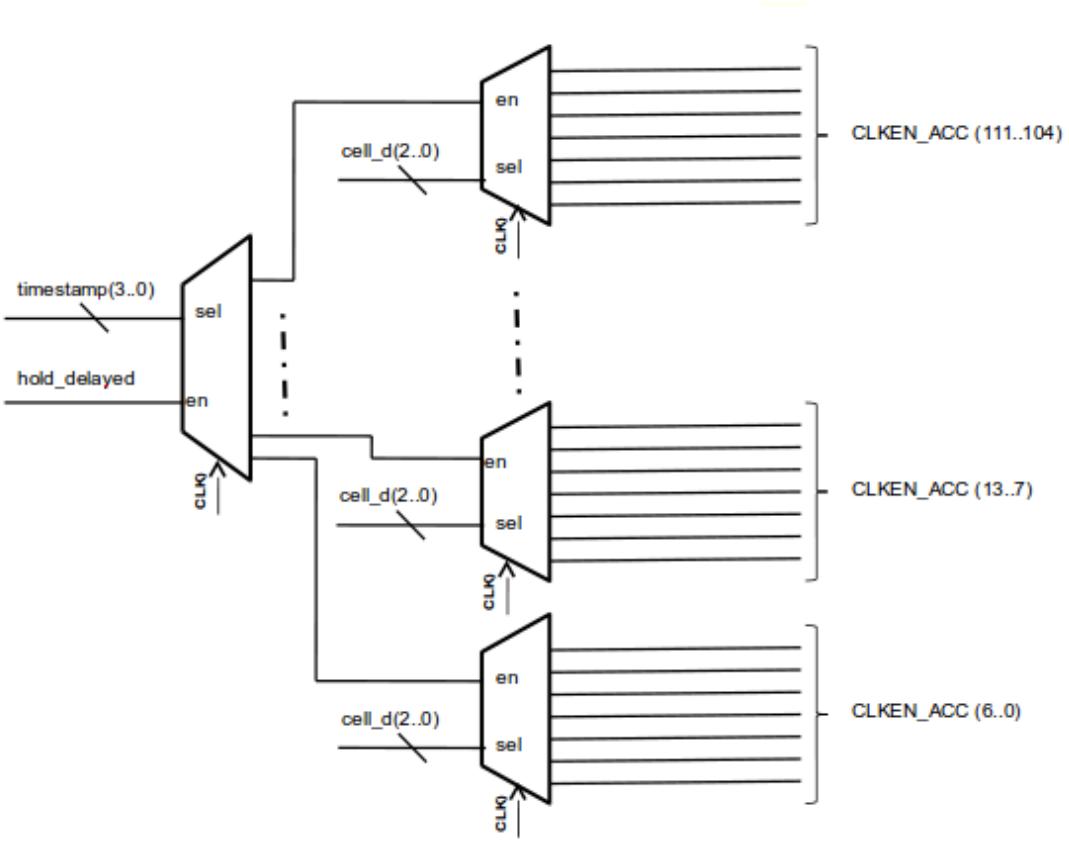


Figura 6.8: Diagramma a blocchi del modulo che genera il segnali di abilitazione di scrittura sugli accumulatori nella configurazione che utilizza 16 timestamp.

6.3.4 Modulo per la ricerca dei massimi locali

La ricerca dei cluster avviene in un modulo posto all'interno degli stessi double-engine. L'implementazione avviene tramite una macchina a stati. il cui diagramma è riportato in Figura 6.9:

- la macchina a stati si attiva passando dallo stato S0 ad S1 con l'arrivo dell'EE_fsm; in questa transizione viene attivata immediatamente la condizione di busy, così da inibire l'invio di altri EE prima che siano concluse tutte le operazioni di clustering;
- in S1 si procede alla copia dell'accumulatore centrale dell'engine per poi inviarlo agli engine vicini;
- i pesi degli engine vicini arrivano in S2 e vengono memorizzati all'interno della macchina stessa. In questo stato si effettuano anche i controlli sulla soglia dei pesi della cella centrale dell'engine e sulle sue coordinate, controllando se è un engine di bordo;
- si procede quindi in S3 per confronto tra i pesi degli engine del cluster e si alza una flag in caso di massimo relativo; la macchina si riporta invece allo stato iniziale se non risultano valide le condizioni richieste in S2;
- se non si riceve nessun segnale di attesa dall'esterno, la macchina si sposta in S4 per inviare in uscita i valori dei pesi nel cluster e quelli delle celle laterali dell'engine massimo, in modo seriale. In totale abbiamo quindi $9 + 6 = 15$ parole da inviare. Quando sono stati inviati tutti i valori, la macchina si riporta in S0, in attesa di un altro EE.

All'interno di ogni double engine abbiamo implementato una sola macchina a stati, in quanto, essendo contigui, i due engine non possono risultare massimi locali contemporaneamente. Quindi, una volta che l'EE è arrivato, si confrontano i pesi delle celle centrali del double engine e si opera sul cluster centrato sull'engine con il peso maggiore. Per mantenere la simmetria nelle connessioni, tutti gli engine vicini inviano egualmente il contenuto delle loro celle centrali anche se non appartengono al cluster su cui si ricerca il massimo. In Figura 6.10 vengono riportate le configurazioni dei due possibili cluster.

I valori degli accumulatori relativi alle celle laterali vengono copiate all'interno del modulo quando tutti i pesi sono stati calcolati, e solamente se l'engine risulta essere un massimo locale. La selezione del gruppo di celle laterali viene eseguita con un multiplexer pilotato dal segnale di timestamp.

Gli engine che si trovano ai bordi della struttura della TPU non devono eseguire la ricerca del massimo poiché non appartenono a cluster completi. Dette $\bar{\imath}$, $\bar{\jmath}$ le coordinate con cui dell'engine è stato istanziato (si veda l'inizio della sezione 6.3), controlliamo che queste siano $0 < \bar{\imath} < i_{max}$, stessa cosa per $\bar{\jmath}$. Solamente in questo caso viene eseguito il calcolo del massimo. Tuttavia, poiché ogni double-engine

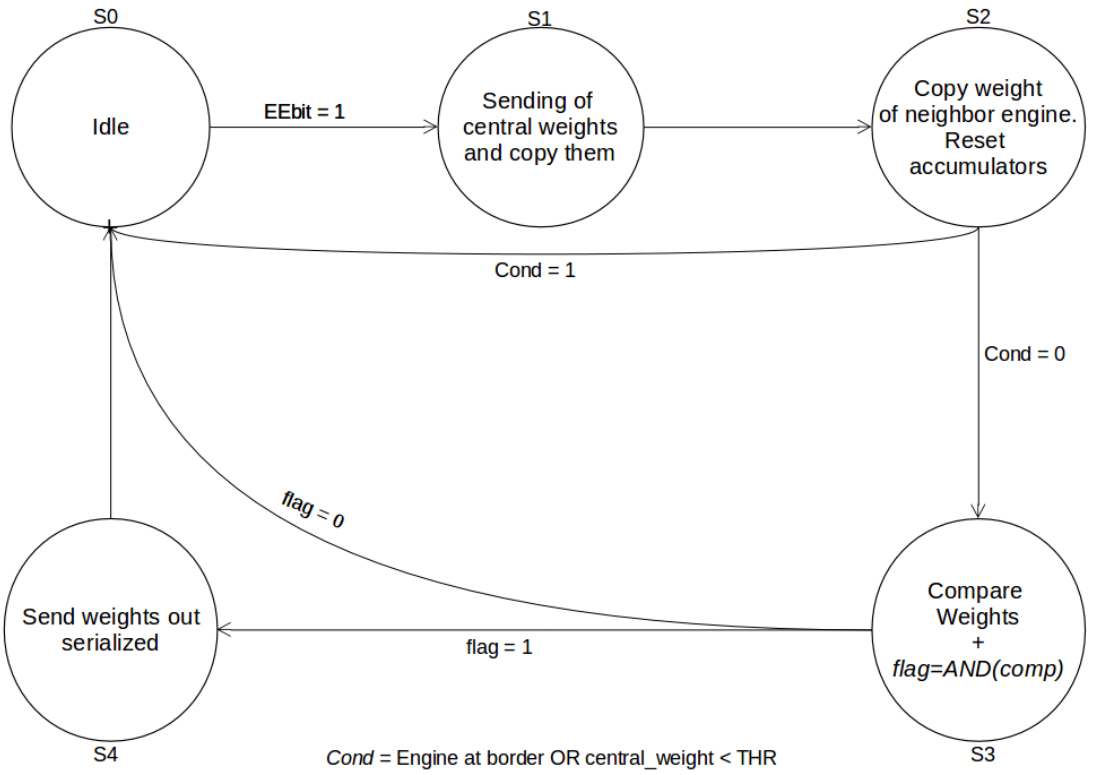


Figura 6.9: Un diagramma a blocchi della macchina a stati che ricerca i cluster.

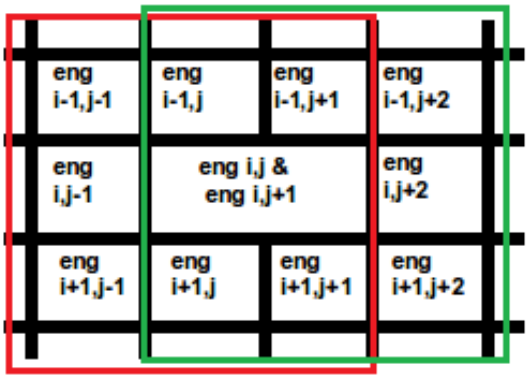


Figura 6.10: La ricerca dei massimi nel caso del double-engine. In rosso e in verde sono rappresentati i due possibili cluster.

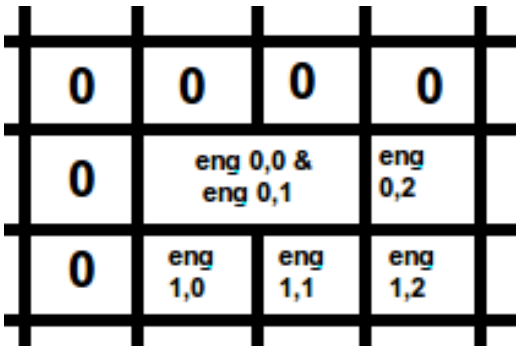


Figura 6.11: Esempio di configurazione del cluster per un double engine situato al bordo della matrice.

viene implementato con la stessa macchina a stati, per mantenere la simmetria nel numero di segnali in ingresso al modulo per il calcolo del massimo, è presente un ulteriore bordo di segnali inizializzati a "0" intorno alla matrice di double engine. In Figura 6.11 vediamo uno schema per la soluzione adottata nel caso di un angolo della matrice di engine.

6.3.5 Protocollo per la gestione dei dati in uscita

Ogni modulo che identifica un engine come massimo locale, invia in uscita i valori dei pesi delle celle centrali relative agli engine del cluster e i pesi delle celle laterali dell'engine massimo in modo seriale. Questi valori saranno utilizzati per il calcolo dei parametri (u, v, z_0, d, k), secondo la formula 3.4.

Attualmente non è stato implementato nella TPU nessun modulo per il calcolo dei parametri, per lo stesso motivo descritto nella sezione 5.3.5.

6.3.6 Parametri di configurazione

Come per la versione per l'IT, abbiamo cercato di implementare una TPU modulare, per mantenere la flessibilità del numero di engine da implementare sul dispositivo. Siccome è stato deciso di disporre i double-engine su una matrice quadrata, la definizione del numero di engine avviene tramite un parametro n che risulta uguale alla radice quadrata del numero totale di engine. Tuttavia, l'aver usato i double engine impone che n sia un numero pari.

6.4 Simulazione della logica della TPU

Le caratteristiche dell'architettura della TPU descritta sopra sono state studiate con i software forniti da Altera Quartus II, versione 13.1 e ModelSim Altera versione 10.1.

	Vers. 1 timestamp	Vers 16 timestamp
Num Engine	676	144
Logica utilizzata (%)	83	92
Freq. di clock max (MHz)	140	190

Tabella 6.2: Caratteristiche delle versioni della TPU con 1 e 16 valori del timestamp.

Abbiamo studiato due versioni dell'engine, una con un valore di timestamp, ed un'altra con 16 valori di timestamp, per capire quanto questo parametro influenzi il consumo di blocchi logici. L'allocazione della matrice di engine nel dispositivo è stata effettuata tramite il software Quartus II ed i risultati sono riportati in 6.2. Il basso numero di engine con 16 valori di timestamp allocabili in un singolo dispositivo rende questa soluzione poco probabile. Al contrario per implementare un sistema completo di tracciatura utilizzando la versione della TPU con un solo timestamp, sono necessari 35 FPGA. Mostriamo in Figura 6.12 i risultati di Quartus II per la frequenza massima permessa, nella versione con un unico valore di timestamp.

Slow 850mV 85C Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	141.56 MHz	141.56 MHz	CLOCK	

Figura 6.12: Frequenze massime permesse per la TPU con 676 double-engine su Stratix V per i rivelatori VELO UT. I valori sono stati calcolati tramite il Timing Analyzer di Quartus II.

Le caratteristiche temporali della TPU sono state simulate tramite il software ModelSim. Il flusso della simulazione segue lo stesso schema descritto nella sezione 5.4, impostando un opportuno testbench che invii gli hit in ingresso, rispettando il protocollo di gestione del flusso dati. Per verificare la funzionalità della TPU abbiamo selezionato i recettori in una regione generica dello spazio dei parametri ed abbiamo utilizzato un campione di hit appropriato. I recettori non sono stati selezionati per nessuna regione specifica dello spazio dei parametri, ed abbiamo utilizzato un campione di hit costruito specificatamente per verificare il corretto funzionamento di tutta la TPU. La simulazione mostra che l'intervallo temporale tra due DV consecutivi (DV-nextHit) è lungo 8 cicli, mentre quello tra un DV e la scrittura sull'accumulatore (DV-writing) è lungo 10 cicli, come mostrato in Figura 6.13. L'intervallo temporale tra l'arrivo di un EE e l'uscita dell'ultimo valore del cluster (che equivale alla latenza del sistema) è lungo 18 cicli di clock, come mostrato in Figura 6.14.

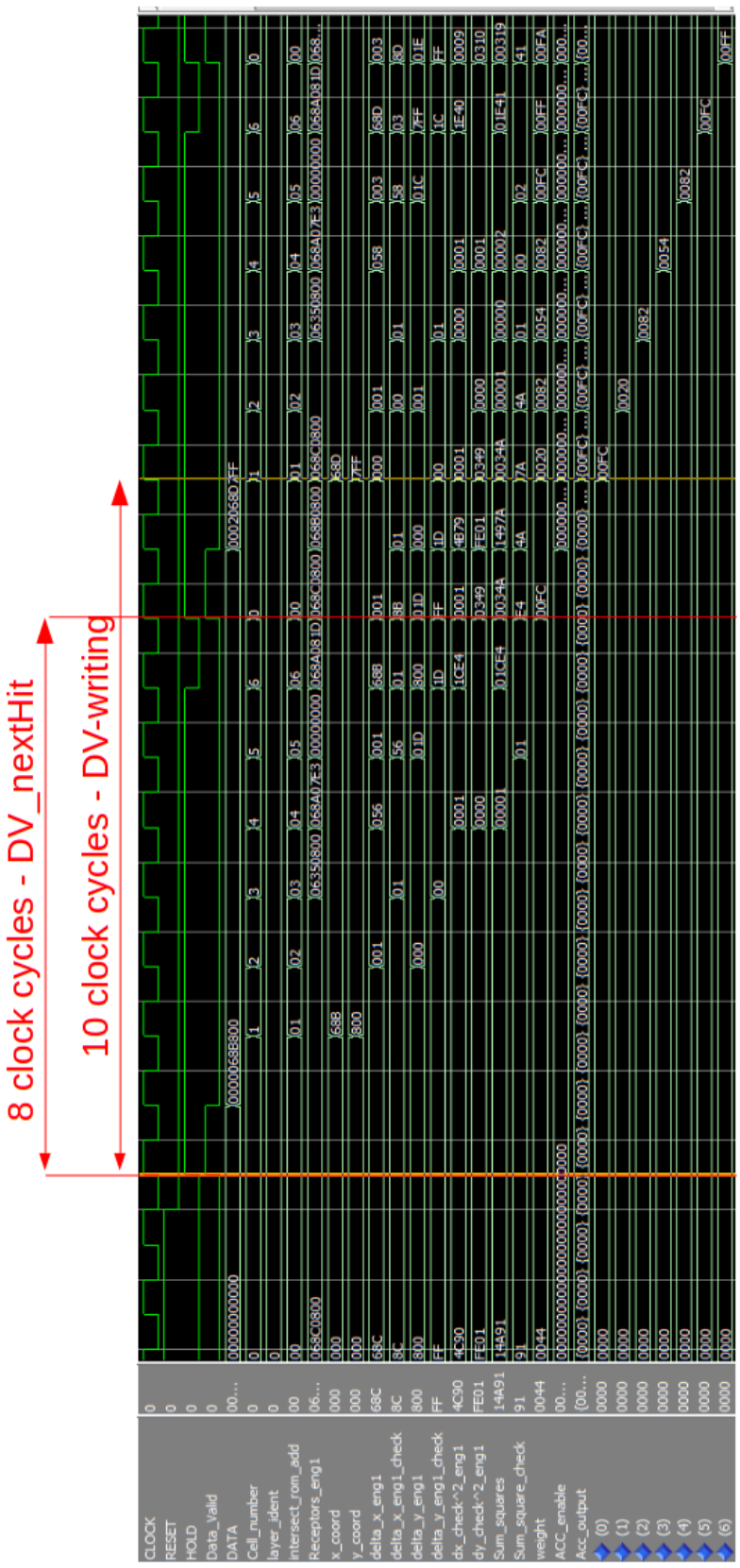


Figura 6.13: La simulazione ModelSim della pipeline che calcola i pesi per la TPU applicata al VELO e UT. L'intervallo temporale tra due DV consecutivi (DV-nextHit) è lungo 8 cicli, mentre quello tra un DV e la scrittura sull'accumulatore (DV-writing) è lungo 10 cicli.

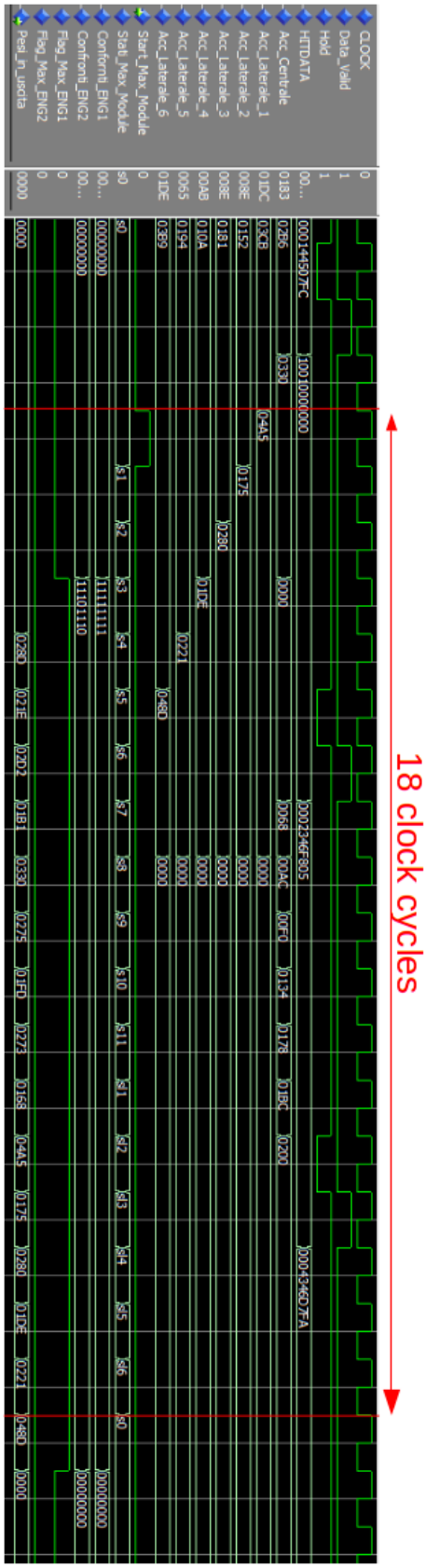


Figura 6.14: La simulazione ModelSim del modulo per la ricerca dei massimi locali. L'intervallo temporale tra l'arrivo di un EE e l'uscita dell'ultimo valore del cluster (che equivale alla latenza del sistema) è lungo 18 cicli di clock.

6.4.1 Prestazioni temporali del dispositivo

Dai risultati della simulazione logica si evince che il sistema è in grado di accettare un hit ogni 8 cicli di clock, quindi nel caso in cui arrivi un EE che genera un massimo locale, non possiamo ricevere altri EE per 18 cicli, equivalente al processamento di 2.25 hit. Considerando che un engine che risulta essere massimo locale riceverà un numero di hit prossimo 8, pari al numero dei piani utilizzati, il tempo per il calcolo del massimo non incide sulle prestazioni temporali della TPU.

La TPU deve essere in grado di identificare le tracce corrette entro alcuni μs ($4 \div 6\,\mu$s nel caso di LHCb), pari al tempo in cui i dati rimangono disponibili nei buffer in attesa della decisione di trigger. Il tempo che si impiega a processare un evento è dato dalla formula:

$$T_{evt} = \frac{1}{f_{clk}} \cdot n_{clk/evt} = \frac{1}{f_{clk}} \cdot (n_{clk/hit} \cdot n_{hit/evt} + n_{clk/max}) \tag{6.2}$$

dove $n_{clk/evt}$ è il numero di cicli per processare un evento, $n_{clk/hit}$ è il numero di cicli per processare un hit, $n_{clk/max}$ il numero di cicli per estrarre il massimo locale, $n_{hit/evt}$ è il numero di hit che l'engine massimo locale riceve per evento ed f_{clk} la massima frequenza di clock permessa. Dalla simulazione di alto livello possiamo estrarre la distribuzione di $n_{hit/evt}$ per gli engine che hanno l'accumulatore centrale sopra soglia, riportata in Figura 6.15. Integrando la distribuzione troviamo che il 90% degli eventi ha un valore di $n_{hit/evt}$ pari o inferiore a 16. Quindi, assumendo $n_{hit/evt} = 16$, $n_{clk/hit} = 8$, $n_{clk/max} = 18$ e $f_{clk} = 140\,\text{MHz}$, otteniamo un valore di T_{evt} pari a $\approx 1.04\,\mu$s.

Se aggiungiamo anche i cicli di clock necessari agli altri moduli che compongono la TPU, stimati tramite simulazioni logiche [52] e riportati in Tabella 6.3, vediamo che il numero totale di cicli di clock è di 192, equivalenti a $T_{evt} \approx 1.4\,\mu$s. Siamo quindi in grado di completare la lettura di un evento entro la latenza limite imposta dal sistema di readout dei rivelatori di LHCb dopo l'upgrade del 2020, pari a $4 \div 6\,\mu$s.

Inoltre questo valore del tempo impiegato a processare un evento potrà essere ridotto ottimizzando ulteriormente il piazzamento dei blocchi logici sul chip, in modo tale da ridurre i ritardi di comunicazione tra le diverse celle ed aumentare la massima frequenza di clock raggiungibile.

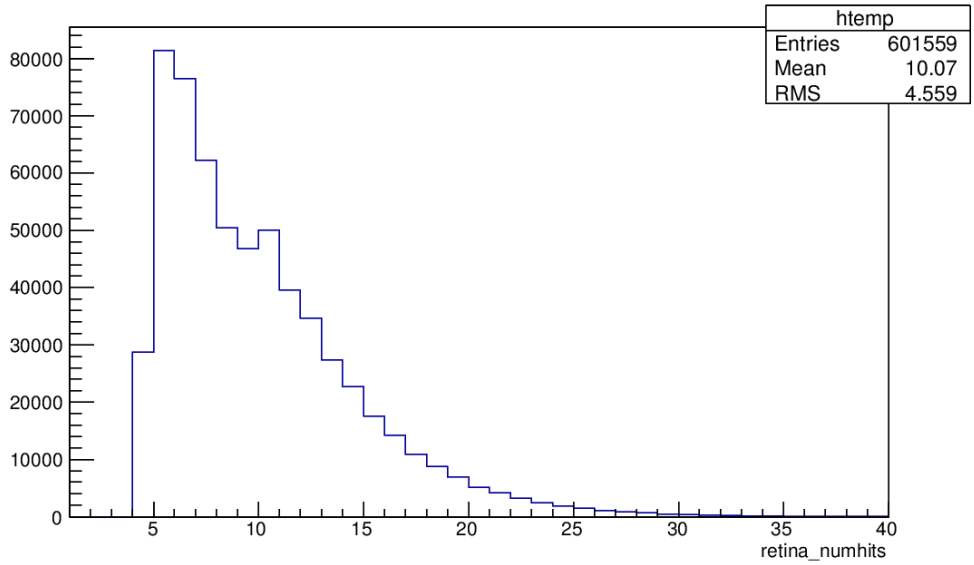


Figura 6.15: Distribuzione del numero di hit in ingresso agli engine il cui accumulatore centrale è sopra la soglia fissata.

	Cicli di clock
Pre-switch sulle schede di readout	15
Switch in TPU - *dispatcher*	15
Switch in TPU - *fanout*	6
Calcolo dei pesi	$8 \cdot 16$
Calcolo dei massimi	192
Uscita dei dati	10
Total	$192 \xrightarrow{@\ 140\ \mathrm{MHz}} 1.4\ \mu\mathrm{s}$

Tabella 6.3: Cicli di clock necessari ai vari moduli logici utilizzati nell'implementazione della TPU per il VELO e UT. I valori sono stati calcolati tramite simulazioni logiche dei moduli.

Conclusioni

In questa tesi abbiamo studiato per la prima volta l'implementazione in hardware di una unità di processamento di tracce (Track Processing Unit, TPU) per gli esperimenti di fisica delle alte energie, basata su un algoritmo nuovo, e in grado di operare alla frequenza di intersezione dei fasci di LHC (40 MHz). In particolare abbiamo curato la progettazione del modulo fondamentale della TPU, chiamato engine, che identifica le possibili tracce candidate a partire dagli hit sul rivelatore. Tale modulo processa i dati in parallelo ad altri moduli simili organizzati in una struttura a matrice ed un sistema intelligente di distribuzione dei dati minimizza il numero di copie necessarie. Uno studio precedente ha dimostrato tramite una simulazione ad alto livello che è possibile applicare questo algoritmo a un sistema di rivelatori di traccia attualmente presente nell'esperimento LHCb (sistema IT) e a un sistema di rivelatore di vertice la cui installazione è prevista nel 2020 nell'ambito di un upgrade dello stesso esperimento (sistema VELOUT). Per ottenere un'alta efficienza di ricostruzione delle tracce lo stesso studio ha dimostrato la necessità di un numero di engine dell'ordine di alcune migliaia.

Per l'implementazione di questo primo prototipo di TPU abbiamo scelto dispositivi a logica programmabile di tipo FPGA della serie Stratix progettata da Altera. Dato che il progetto è ancora in fase di sviluppo, la scelta del dispositivo è stata effettuata considerando la flessibiltà, rispetto ad altri dispositivi non riprogrammabili, ed il costo minore. Durante il nostro lavoro di tesi abbiamo sviluppato il firmware per la TPU in linguaggio di descrizione della logica di alto livello VHDL sia per il sistema IT che per quello VELOUT. Lo scopo è stato quello di verificare la fattibilità tecnica della TPU, valutando le dimensioni dell'apparato, i costi, e le prestazioni in termini di velocità di processamento degli eventi. Per la progettazione del firmware abbiamo utilizzato il software di sviluppo Quartus II, fornito dalla ditta produttrice delle FPGA, e per la simulazione il software ModelSim della ditta Mentor Graphics.

Alla fine della progettazione per il sistema IT abbiamo ottenuto che una FPGA di dimensione medie può contenere un massimo di 256 engine, occupando circa l'80% del chip e utilizzando una frequenza massima per il clock interno di circa 150 MHz.

Considerando che per questo sistema lo studio precedente ha stimato che circa 6000 engine sono sufficienti a ricostruire tracce in modo efficiente, la TPU risulta quindi realizzabile utilizzando circa 32 di queste FPGA. Inoltre utilizzando dati provenienti dalla simulazione MonteCarlo ufficiale dell'esperimento LHCb abbiamo misurato che la TPU riesce a processare eventi con una frequenza di 4 MHz, ampiamente superiore rispetto a quella con cui gli stessi dati sono letti dal rivelatore, che è pari a 1 MHz. Abbiamo inoltre verificato che, nonostante alcune approssimazioni numeriche necessarie per rispettare le requisiti di velocità e spazio del dispositivo, la capacità della TPU di identificare le tracce risulta perfettamente compatibile con quella ottenuta dalla simulazione di alto livello.

Poiché nel sistema VELO-UT è prevista la ricostruzione di tracce a curvatura non trascurabile, l'algoritmo di tracciatura presenta un configurazione più complessa rispetto a quella del sistema IT e un numero maggiore di parametri da calcolare. Per questo motivo abbiamo progettato il firmware per il dispositivo FPGA con la massima dimensione attualmente in commercio. Abbiamo quindi ottenuto che in queste FPGA possono essere implementati 676 engine, occupando circa l'85% del chip e con una frequenza di clock massima di circa 140 MHz. L'intero sistema necessita di circa 22000 engine per una ricostruzione efficiente delle tracce, quindi sono necessarie circa 35 FPGA per implementare la TPU corrispondente. Tramite la simulazione logica abbiamo stimato il tempo totale per processare un evento pari a 1.4 μs, ben al di sotto del tempo in cui i dati rimangono memorizzati in attesa della decisione di trigger (questo valore per LHCb dopo l'upgrade del 2020 Ã¨ pari a 4-6 μs). Inoltre sono possibili ulteriori ottimizzazioni nella disposizione delle celle logiche all'interno dell'FPGA per aumentare la frequenza massima disponibile. In alternativa nel prossimo futuro potrebbero essere utilizzati dispositivi che sono già stati annunciati da Altera con prestazioni ancora più elevate.

In conclusione, in questo lavoro di tesi abbiamo dimostrato la possibilità di realizzare su un dispositivo di tipo FPGA una unità di processamento di tracce in tempo reale ad alta frequenza. Al momento è in corso l'allestimento di una postazione di test con cui si possa valutare la funzionalità del firmware sottoposto ad un flusso di dati simile a quello dell'esperimento reale.

Bibliografia

[1] J. H. Christenson, J.W Cronin, et al. Evidence for the 2π decay of the K_2^0 meson. *Phys. Rev. Lett.*, 13(4):138–140, 1964.

[2] J. Beringer and others (Particle Data Group). Review of particle physics. *Phys. Rev. D*, 86(010001), 2012.

[3] The BABAR Collaboration. BaBar Techinical Design Report. Technical Report SLAC-457, SLAC-R-95-457, 1995.

[4] The Belle Collaboration. The Belle detector. *Nucl. Instrum. Meth.*, A479:117–232, 2002.

[5] G. Antchev, P. Aspell, et al. First measurement of the total proton-proton cross section at the LHC energy of $\sqrt{s} = 7$ TeV. *Europhys. Lett.*, (96), 2011.

[6] The LHCb Collaboration. Prompt charm production in pp collisions at $\sqrt{s} = 7$ TeV. (LHCb-CONF-2010-013), 2010.

[7] Y. Guz. Studies of open charm and charmonium production at LHCb. *Conf. Proc.*, 2010.

[8] A. Achilli, R. M. Godbole, et al. Total and inelastic cross-sections at LHC at $\sqrt{s} = 7$ TeV and beyond. *Phys. Rev. D*, (84:094009), 2011.

[9] The CDF Collaboration. The CDF-II detector: Technical Design Report. Technical Report FERMILAB-DESIGN-1996-01, FERMILAB-PUB-96-390-E, 1996.

[10] A. Bardi, S. Belforte, et al. The CDF Online Silicon Vertex Tracker. *Nucl. Instrum. Meth.*, A485:178–182, 2002.

[11] M. Dell'Orso and L. Ristori. VLSI structures for track finding. *Nucl. Instrum. Meth.*, A278:436–440, 1989.

[12] B. Ashmanskas, A. Barchiesi, et al. The CDF Silicon Vertex Trigger. *Nucl. Instrum. Meth.*, A518:532–536, 2004.

[13] S. R. Amendolia, S. Galeotti, et al. The AMchip: a full custom CMOS VLSI associative memory for pattern recognition. *IEEE Trans. Nucl. Sci.*, 39(4):795–797, 1992.

[14] The CDF Collaboration. Measurement of CP-violating asymmetries in $D^0 \to \pi^+\pi^-$ and $D^0 \to K^+K^-$ decays at CDF. *Phys. Rev. Lett.*, 109(111801), 2011.

[15] The CDF Collaboration. Measurements of Direct CP-Violating Asymmetries in Charmless Decays of Bottom Baryons. *arXiv*, (1403.5586), 2014.

[16] The CDF Collaboration. Measurements of Direct Violating Asymmetries in Charmless Decays of Strange Bottom Mesons and Bottom Baryons. *Phys. Rev. Lett.*, 106:181802, 2011.

[17] The CDF Collaboration. Evidence for the charmless annihilation decay mode $B_s^0 \to \pi^+\pi^-$. *Phys. Rev. Lett.*, 108:211803, 2012.

[18] The CDF Collaboration. Measurements of branching fraction ratios and CP asymmetries in $B^\pm \to D_{CP}K^\pm$ decays in hadron collisions. *Phys. Rev. D*, 81:031105, 2010.

[19] The CDF Collaboration. Observation of $B_s - \bar{B}_s$ Oscillations. *Phys. Rev. Lett.*, (97:242003), 2006.

[20] Evelyn J. Thomson, C. Ciobanu, J.Y. Chung, J. Gerstenslager, J. Hoftiezer, et al. Online track processor for the CDF upgrade. *IEEE Trans. Nucl. Sci.*, 49:1063–1070, 2002.

[21] The ATLAS Collaboration. Technical Design Report Fast TracKer (FTK). Technical Report CERN-LHCC-2013-007, 2013.

[22] G. Bagliesi. Tau tagging at ATLAS and CMS. *arXiv*, 0707.0928, 2007.

[23] Oliver Sim Bruning, Paul Collier, P Lebrun, Stephen Myers, Ranko Ostojic, John Poole, and Paul Proudlock. *LHC Design Report*. CERN, Geneva, 2004.

[24] R Bailey and Paul Collier. Standard Filling Schemes for Various LHC Operation Modes. Technical Report LHC-PROJECT-NOTE-323, CERN, Geneva, Sep 2003.

[25] The LHCb Collaboration. The LHCb detector at the LHC. *Journal of Instrumentation*, 3(08):S08005, 2008.

[26] M. L. Mangano. Two lectures on heavy quark prodcution in hadronic collisions. Technical Report CERN-TH-97-328, 1997.

[27] V. P. Andreev. B production at the LHC/QCD aspects. *arXiv*, 0706.1789, 207.

[28] LHCb Report: Level Best. Dernières nouvelles du LHC: à son meilleur niveau. 2012.

[29] *LHCb : Technical Proposal*. Tech. Proposal. CERN, Geneva, 1998.

[30] LHCb VELO TDR: Vertex locator. Technical design report. 2001.

[31] R.P. Bernhard, M. Agari, C. Bauer, J. Blouw, W. Hofmann, et al. The LHCb silicon tracker. *Nucl. Instrum. Meth.*, A596:17–20, 2008.

[32] LHCb outer tracker: technical design report. 2001.

[33] J. van Bakel, N. A.and van den Brand, H. Verkooijen, D. Baumeister, W. Hofmann, et al. Performance of the Beetle readout chip for LHCB. 2001.

[34] M. Adinolfi et al. Performance of the LHCb RICH detector at the LHC. *Eur. Phys. J.*, C73:2431, 2013.

[35] I. Machikhiliyan et al. The LHCb electromagnetic calorimeter. *J. Phys. Conf. Ser.*, 160:012047, 2009.

[36] R.I. Dzhelyadin et al. The LHCb hadron calorimeter. *Nucl. Instrum. Meth.*, A494:332–339, 2002.

[37] S. Filippov, Y. Gavrilov, et al. Experimental Performance of SPD/PS Detector Prototypes. Technical Report CERN-LHCb-PUB-2000-031, 2000.

[38] Jr. Alves, A.A., L. Anderlini, M. Anelli, R. Antunes Nobrega, G. Auriemma, et al. Performance of the LHCb muon system. *Journal of Instrumentation*, 8:P02022, 2013.

[39] R Aaij, J Albrecht, F Alessio, S Amato, E Aslanides, et al. The LHCb Trigger and its Performance in 2011. *Journal of Instrumentation*, 8:P04022, 2013.

[40] G. Haefeli, A. Bay, A. Gong, H. Gong, M. Muecke, et al. The LHCb DAQ interface board TELL1. *Nucl. Instrum. Meth.*, A560:494–502, 2006.

[41] The LHCb Collaboration. Framework TDR for the LHCb upgrade. Technical Report CERN/LHCC 2012-007, 2012.

[42] The LHCb Collaboration. LHCb VELO Upgrade Technical Design Report. Technical Report CERN/LHCC 2013-021, 2013.

[43] The LHCb Collaboration. LHCb Tracker Technical Design Report. Technical Report CERN/LHCC 2014-001, 2014.

[44] The LHCb Collaboration. LHCb Particle Identification Upgrade Technical Design Report. Technical Report CERN/LHCC 2013-022, 2013.

[45] Collaboration LHCb. LHCb Trigger and Online Upgrade Technical Design Report. Technical Report CERN-LHCC-2014-016. LHCB-TDR-016, CERN, Geneva, May 2014.

[46] H. Kirchner and S. J. Thorpe. Ultra-rapid object detection with saccadic eye movements: Visual processing speed revisited. *Vision Research*, 46(11):1762 – 1776, 2006.

[47] D. H. Hubel. The visual cortex of the brain. *Scientific American*, 17:54 – 62, 1963.

[48] W. N. Grimes, G. W. Schwartz, and F. Rieke. The synaptic and circuit mechanisms underlying a change in spatial encoding in the retina. *Neuron*, 82(2):460 – 473, 2014.

[49] M. Del Viva, G. Punzi, and D. Benedetti. Information and perception of meaningful patterns. *PLoS ONE*, 8(7):e69154, 07 2013.

[50] L. Ristori. An artificial retina for fast track finding. *Nucl. Instrum. Meth.*, A453:425–429, 2000.

[51] P.V.C. Hough. Machine Analysis Of Bubble Chamber Pictures. *Conf.Proc.*, C590914:554–558, 1959.

[52] A. Abba, F. Bedeschi, M. Citterio, F. Caponio, A. Cusimano, A. Geraci, F. Lionetto, P. Marino, M. J. Morello, N. Neri, D. Ninci, A. Piucci, M. Petruzzo, G. Punzi, F. Spinella, S. Stracka, D. Tonelli, and J. Walsh. A specialized track processor for the LHCb upgrade. Technical Report LHCb-PUB-2014-026. CERN-LHCb-PUB-2014-026, CERN, Geneva, Mar 2014.

[53] A. Piucci. Reconstruction of tracks in real time in the high luminosity environment at LHC. Tesi di laurea magistrale, Università di Pisa, 2014.

[54] A. Abba, F. Bedeschi, M. Citterio, F. Caponio, A. Cusimano, et al. Simulation and performance of an artificial retina for 40 MHz track reconstruction. *arXiv:1409.0898*, Set 2014.

[55] Altera Corporation. *Stratix III Handbook*, 2001. Available on line.

[56] Atera Corporation. http://www.altera.com/devices/fpga/fpga-index.html, 2014. FPGA overview.

[57] P. P. Chu. *RTL hardware Design Using VHDL, Coding for Efficiency, Portability and Scalabiliy.* Willy-IEEE Press, 2006.

[58] Altera Corporation. *Quartus II Handbook*, 2014. Available on line.

[59] B. Angelucci, E. Pedreschi, M. Sozzi, and F. Spinella. TEL62: an integrated trigger and data acquisition board. pages 823–826, 2011.

[60] Altera Corporation. *Stratix V Handbook*, 2013. Available on line.