HAL
open science

Le LHC vu des cuisines

Sebastien Viret

To cite this version:

Sebastien Viret. Le LHC vu des cuisines. Physique des Hautes Energies - Expérience [hep-ex]. Université Claude Bernard Lyon 1, 2021. tel-03170368

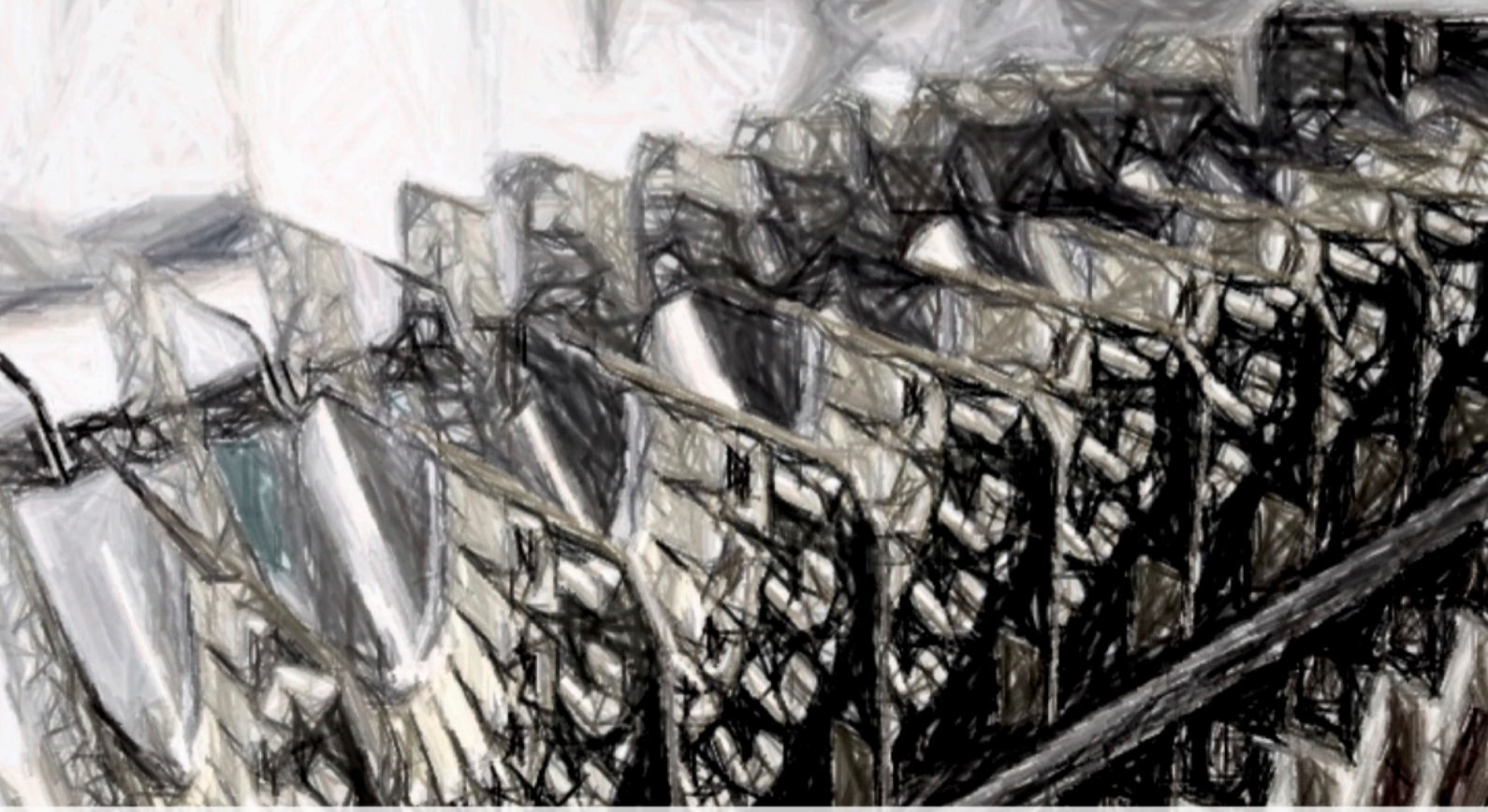
HAL Id: tel-03170368

https://theses.hal.science/tel-03170368v1

Submitted on 16 Mar 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Habilitation à diriger des recherches

Le LHC vu des cuisines

De LHCb à CMS, en passant par ATLAS

SEBASTIEN VIRET

HABILITATION À DIRIGER DES RECHERCHES

de l'Université Claude Bernard Lyon 1

préparée au sein de l'Institut de Physique des 2 Infinis de Lyon

Spécialité : **Physique des hautes énergies**

présentée par

SEBASTIEN VIRET

Le LHC vu des cuisines

De LHCb à CMS, en passant par ATLAS

Soutenue publiquement le 25 Janvier 2021 devant le Jury composé de :

PAULA COLLINS	Rapporteur	CERN
CHRISTOPHER PARKES	Rapporteur	University of Manchester
LAURENT SERIN	Rapporteur	CNRS/IN2P3
GIOVANNI CALDERINI	Examinateur	CNRS/IN2P3
DAVID CALVET	Examinateur	CNRS/IN2P3
JACQUES MARTEAU	Examinateur	Université Claude Bernard Lyon 1

À Laure, Lisa, Raphaël, et Lucie

SOMMAIRE

I Introduction **1**

II Du VELO aux mégots **5**

1 Contributions au projet LHC **7**

1.1 Le LHC, en quelques mots 7

1.2 Le rôle de la calibration 8

1.3 Alignement du détecteur de vertex de LHCb : 2005-2007 9

1.3.1 Le détecteur LHCb 9

1.3.2 Le problème de l'alignement 12

1.3.3 Contribution personnelle 13

1.4 Mise en route du LASER de calibration du calorimètre à tuiles dans ATLAS : 2008-2010 14

1.4.1 Le détecteur ATLAS 14

1.4.2 Le calorimètre à tuiles d'ATLAS et son étalonnage 14

1.4.3 Le système LASER en quelques mots 16

1.4.4 Contribution personnelle 18

2 Le HL-LHC et les mégots de CMS **19**

2.1 Le projet HL-LHC 19

2.2 Le détecteur de traces de CMS 20

III Produire et extraire les données **25**

3 Le frontend du trajectographe **27**

3.1 Description de la chaine d'acquisition 27

3.1.1 Principe général 27

3.1.2 Chaîne d'acquisition d'un module 2S 28

3.1.3 Chaîne d'acquisition d'un module PS 29

3.2 Le stub : objet fondamental du futur tracker 30

3.2.1 Présentation générale 30
3.2.2 Relation entre la largeur du stub et l'impulsion transverse 30
3.2.3 La correction de parallaxe 31
3.3 sélection des stubs 33
3.3.1 Typologie 33
3.3.2 sélection des stubs au niveau du frontend 34
3.3.3 Utilisation de la largeur des stubs dans la reconstruction des traces 37

4 Développement et réalisation du concentrateur 41
4.1 Le rôle du concentrateur 41
4.1.1 Pourquoi un ASIC ? 41
4.1.2 Principe de fonctionnement 42
4.2 Cahier des charges 43
4.2.1 Contexte 43
4.2.2 Fonctionnement du CIC 44
4.2.3 Definition des formats d'entrées/sortie et évaluation des pertes dynamiques 45
4.2.4 Contraintes techniques et cahier des charges 50
4.3 Modele du CIC 50
4.3.1 Présentation générale 50
4.3.2 Fonctionnement du coeur de l'ASIC 52
4.3.3 Vérification du modèle : simulation comportementale 54
4.4 Design et réalisation du CIC 54
4.4.1 Présentation du flot de conception 54
4.4.2 Le CIC : du modèle à la fonderie 55
4.5 Caractérisation du CIC1 58
4.5.1 Description du banc de test 58
4.5.2 Résultats 59
4.5.3 Le CIC1 dans la chaîne d'acquisition du module 2S 61
4.6 Développement du CIC2 61
4.6.1 Description et performances attendues 61
4.6.2 Et après 62
4.7 Bilan 62

IV Analyser les données 63

5 Construire des traces, vite... 65

5.1 Introduction . . . 65

5.1.1 Pourquoi ? . . . 65

5.1.2 Etat de l'art . . . 66

5.2 Un track trigger pour CMS . . . 67

5.2.1 Le problème . . . 67

5.2.2 Identification rapide de traces dans CMS . . . 69

6 Utiliser les mémoires associatives dans CMS 71

6.1 Mise en place de l'algorithme . . . 71

6.1.1 Préambule . . . 71

6.1.2 Banques, motifs, et autres superstrips . . . 72

6.1.3 Définition du superstrip . . . 73

6.1.4 Utilisation de bits ternaires . . . 75

6.2 Développement d'un émulateur de chip AM dans CMS . . . 77

6.2.1 Générateur de banque . . . 77

6.2.2 Émulateur . . . 79

6.3 Performance de l'algorithme dans CMS . . . 80

6.3.1 Optimisation des banques . . . 80

6.3.2 Définition de l'efficacité et du taux de faux . . . 81

6.3.3 Résultats . . . 84

6.3.4 Bilan . . . 86

7 Développement d'un démonstrateur : le projet FastTrack 89

7.1 Introduction et objectifs . . . 89

7.2 Préparation de l'ajustement, le filtrage post-AM . . . 90

7.2.1 Principe . . . 90

7.2.2 Principe de fonctionnement . . . 91

7.2.3 Performance et integration à l'algorithme complet . . . 94

7.3 Le démonstrateur . . . 98

7.3.1 Description générale . . . 98

7.3.2 Implémentation de l'algorithme dans le démonstrateur . . . 100

7.3.3 Performances et comparaison données/simulation . . . 101

7.4 Vers un système complet . . . 103

7.5 Bilan 103

V Conclusion 105

VI Annexes 121

A Notes LHCb 123

A.1 Alignment procedure 124

A.2 Alignment tests 125

B Notes ATLAS 126

B.1 LASER description 127

B.2 LASER commissioning 128

C Notes CMS 129

C.1 Stubs study 130

C.2 Front end dynamic inefficiencies 131

C.3 CIC1 specifications 132

C.4 Track trigger development 133

I

INTRODUCTION

INTRODUCTION

Quinze ans se sont écoulés depuis le passage de ma thèse en Septembre 2004. Il y a 15 ans, le grand collisionneur de hadrons (LHC) était toujours en construction et le boson de Higgs une hypothèse. Les choses ont évolué depuis....

A commencer par le LHC lui-même. Après un faux départ en 2008, le collisionneur fonctionne sans problème majeur depuis 2010. Il a fourni depuis cette date des quantités énormes de données dont l'analyse a permis d'effectuer de nombreuses découvertes, la plus célèbre étant celle du boson de Higgs en 2012.

Toutes ces découvertes n'auraient sans doute pas eu lieu, en tout cas pas aussi rapidement, si l'énorme et complexe machine qu'est le LHC n'avait pas aussi bien fonctionné. On peut en dire autant des détecteurs assemblés autour des points de collisions : CMS, ATLAS, LHCb, et ALICE. Avant les résultats, il y a les machines, et derrière les machines, il y a des centaines de chercheurs, d'ingénieurs, et de techniciens qui ont consacré plusieurs années à les développer et à les faire fonctionner. Les coulisses du LHC sont un monde à part et souvent passionnant.

Depuis 2005, l'essentiel de mes activités s'est déroulé dans ces coulisses, plus précisément dans trois des quatre expériences principales : LHCb (2005-2007), ATLAS (2008-2010), et finalement CMS (2011-2019). Ce document, s'il retrace l'ensemble de mes contributions, porte toutefois une attention plus particulière à CMS.

La première partie commence par un rappel de mes contributions à LHCb et ATLAS (Chapitre 1), et se termine par une description succincte de mes activités dans CMS (Chapitre 2), où depuis 2012 je consacre l'essentiel de mon attention au projet de remplacement du trajectographe. Cette activité est l'objet des deux parties suivantes.

La deuxième partie est consacrée à l'extraction des informations du futur trajectographe. C'est un élément essentiel de ce détecteur, car ces données devront être utilisables dans une latence très courte de quelques microsecondes. La problématique est décrite plus en détails dans le Chapitre 3. Le Chapitre 4, quant à lui, présente le développement d'un élément essentiel de la chaîne d'extraction : le concentrateur.

Enfin, la troisième et dernière partie se concentre sur l'exploitation des informations extraites afin de reconstruire des traces en temps réel. Un état des lieux des techniques existantes est dressé dans le Chapitre 5, puis le potentiel d'une de ces techniques dans le futur trajectographe est évalué dans le Chapitre 6. Enfin, le Chapitre 7 présente le démonstrateur qui a été développé pour confirmer ces résultats.

II

DU VELO AUX MÉGOTS

1

CONTRIBUTIONS AU PROJET LHC

Ce chapitre est un tour d'horizon rapide de mes contributions aux expériences LHCb et ATLAS entre 2005 et 2010. Ces travaux s'inscrivaient dans la préparation au démarrage du LHC, et furent essentiellement consacrés à la mise en place de systèmes de calibration.

1.1/ LE LHC, EN QUELQUES MOTS

Si vous avez ouvert ce document, c'est que vous avez sans doute déjà entendu parler du LHC. Je ne vais donc pas trop m'attarder à décrire le grand collisionneur de hadrons qui est en fonction depuis 2010 au CERN à Genève, de nombreux ouvrages le font déjà, de manière très détaillée et satisfaisante (voir par exemple [1]).

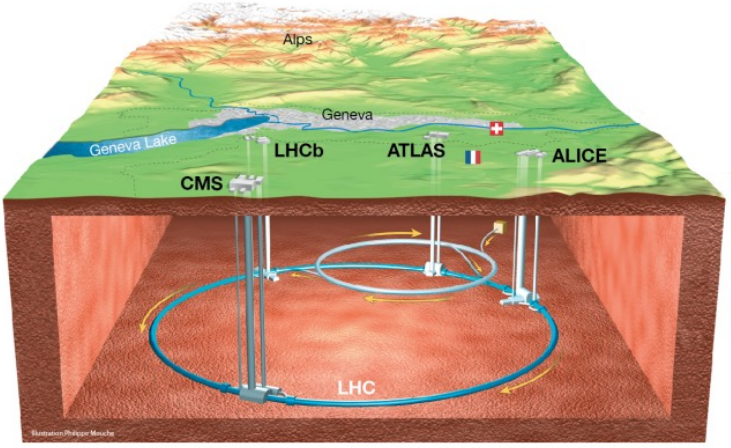


FIGURE 1.1 – Le LHC

Quelques mots tout de même... Le LHC permet d'effectuer des collisions de particules à un niveau d'énergie extrêmement élevé. Ces collisions ont lieu en quatre points de l'anneau souterrain de 27 km. Autour de chacun de ces points se situe un détecteur permettant d'analyser de manière précise ces collisions. Parmi ces quatre expériences,

on distingue 2 systèmes généralistes : ATLAS et CMS, et 2 détecteurs plus spécifiques : ALICE et LHCb.

Les deux objectifs principaux d'ATLAS et CMS sont :

— **Comprendre** l'origine de la masse en établissant l'existence du boson de Higgs. Ce premier objectif a été atteint en 2012 [2, 3].
— **Observer** la matière avec un grossissement jamais atteint jusqu'à présent pour rechercher des phénomènes nouveaux ou des objets inconnus. On suppose en effet que le modèle standard est une théorie effective, valable jusqu'à une énergie de l'ordre du TeV $(10^{12}eV)$. Au-delà, de nouveaux phénomènes sont attendus. Leur observation permettrait d'ouvrir un nouveau chapitre de notre discipline, et de préparer le terrain pour de futures expériences.

LHCb est une expérience plus spécialement dédiée à l'étude des quarks lourds (b et c), quant à ALICE, son programme de physique est consacré à l'étude précise du plasma de quarks et de gluons, un état particulier de la matière.

Le point commun de toutes ces expériences est leur complexité. Ces détecteurs sont des puzzles géants permettant de détecter et de mesurer les propriétés (charge électrique, quantité de mouvement, énergie) des particules créées lors de la collision des protons circulant dans le LHC. Chacun de ces puzzles contient environ 100 millions de cellules de détection indépendantes dont les informations doivent être lues et analysées dans un laps de temps très court.

L'assemblage de ces détecteurs, ainsi que leur mise en route et leur suivi, ont été et sont encore aujourd'hui des expériences à part entière. Des milliers de personnes venant du monde entier ont contribué à leur construction. Des centaines de personnes contribuent chaque jour à leur bon fonctionnement.

Sans ce travail essentiel, les nombreuses découvertes du LHC n'auraient sans doute pas eu lieu, en tout cas pas aussi rapidement. Le plus beau restaurant du monde n'est rien sans ses cuisines...

1.2/ Le rôle de la calibration

Lorsque j'ai soutenu ma thèse en 2004 [4], la plupart des détecteurs qui fonctionnent aujourd'hui au LHC entraient à peine dans leur phase de construction. Les étapes de recherche et développement qui avaient précédé cette phase avaient été franchies avec succès et il fallait réfléchir à leur mise en route et au suivi de leur fonctionnement.

En effet, les détecteurs installés autour du LHC sont non seulement des assemblages complexes, mais ils sont surtout très fragiles, et leur mode d'emploi n'est pas fourni à la livraison. Leur bon fonctionnement exige un suivi très strict.

Le but de ce suivi est simple : comment interpréter correctement le signal détecté et faire en sorte que celui-ci soit toujours disponible lorsque le collisionneur fonctionne ? Comment faire en sorte que le signal mesuré par un détecteur corresponde le plus précisément possible à la réalité ? Le rôle de la calibration est de fournir des réponses à ces questions.

La calibration est une discipline vaste qui dépend du signal que l'on étalonne. Pour chaque situation la méthode est différente, mais elle se base toujours sur le même constat

paradoxal : comment améliorer la qualité du signal mesuré en partant des seules informations disponibles avant étalonnage, c'est à dire des signaux non calibrés ?

Dans un détecteur, on mesure des temps, des dépôts d'énergie, et des positions. Les techniques de calibration couvrent logiquement chacune de ces 3 catégories.

Au LHC, l'étalonnage en temps se fait principalement lors de la mise en route du système (voir par exemple [5]). La synchronisation des sous-détecteurs est fondamentale ; les millions de signaux échantillonnés toutes les 25 ns doivent pouvoir être mis en commun de manière rigoureuse. Les dépôts d'énergie de la collision N doivent en effet être comparés aux traces de la même collision. Au niveau des détecteurs, le temps peut également être utilisé pour analyser les propriétés d'un dépôt d'énergie. Jusqu'à récemment, cependant, la précision des mesures de temps ne permettait pas d'en faire un élément central des systèmes de détection au LHC. Cette situation est aujourd'hui en train d'évoluer rapidement, d'autant plus qu'il va désormais devenir nécessaire de distinguer l'origine temporelle des particules au sein d'une même collision. De nouveaux systèmes de détection [6] dédiés à la mesure de temps sont en cours de développement, et il y a fort à parier que les méthodes de calibration dans ce secteur vont devoir s'adapter à ces nouvelles performances dans les années à venir.

L'étalonnage en position se fait également en plusieurs étapes, il concerne plus spécifiquement les détecteurs permettant de reconstruire la trace des particules et leur site de désintégration (vertex). Une trace mal reconstruite peut en effet fausser radicalement l'information mesurée. On doit connaitre précisément la position de chaque canal de détection. A titre d'exemple, dans les chambres à muons d'ATLAS, pour estimer la charge de muons de très grande impulsion transverse, on doit être capable de mesurer une déviation d'environ 30 microns sur l'ensemble du rayon du détecteur (plus d'une dizaine de mètres).

L'étalonnage en énergie est également une tâche aussi fondamentale que complexe, d'autant plus qu'il est parfois impossible, contrairement aux 2 cas précédents, de disposer d'une référence précise pour effectuer la correction. Difficile, en effet, de connaître avec précision l'énergie initiale d'un jet. Ce domaine de la calibration met donc en oeuvre des méthodes très élaborées et plusieurs niveaux de corrections sont généralement nécessaires avant de parvenir à une précision acceptable.

Ces deux dernières catégories ont constitué l'essentiel de mes activités entre 2005 et 2010. Ces années furent divisées en 2 périodes distinctes :

— **2005-2007** : travail sur l'étalonnage en position dans LHCb. Cette activité est décrite plus en détails dans la Section 1.3.
— **2008-2010** : travail sur l'étalonnage en énergie dans ATLAS. Cette activité est décrite plus en détails dans la Section 1.4.

1.3/ ALIGNEMENT DU DÉTECTEUR DE VERTEX DE LHCB : 2005-2007

1.3.1/ LE DÉTECTEUR LHCB

Si ATLAS et CMS ont une vocation généraliste, LHCb [7] est une expérience résolument tournée vers les saveurs lourdes. Son principal objectif est d'étudier avec précision la

physique des quarks b et c, domaine extrêmement riche et susceptible de fournir des informations précises sur les limites du modèle standard.

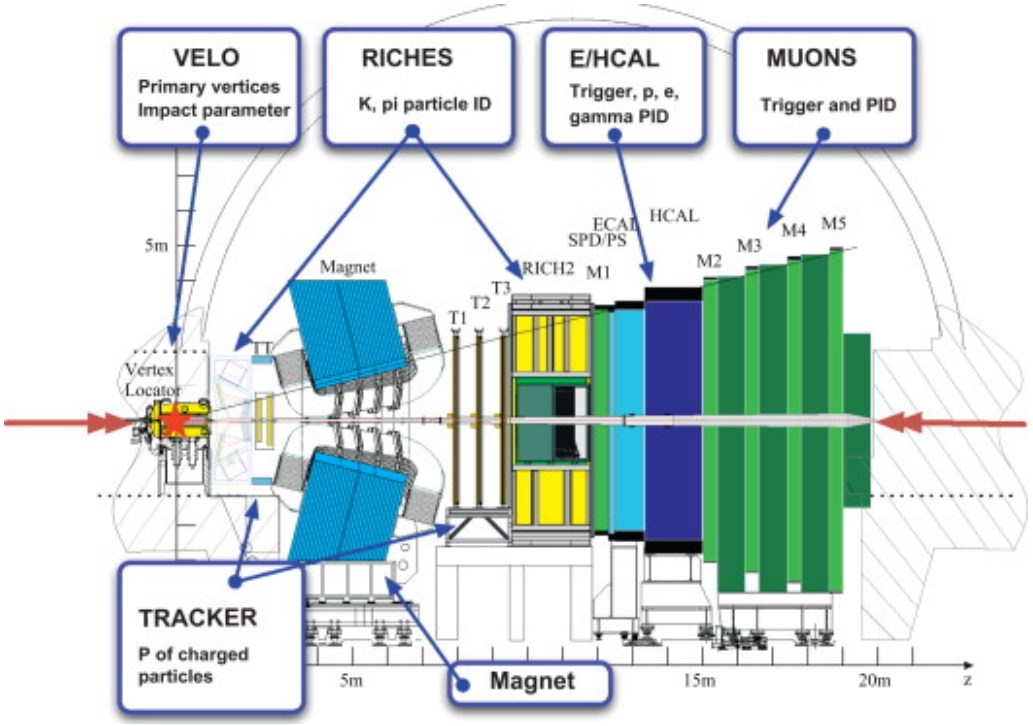


FIGURE 1.2 – Le détecteur LHCb

La structure de LHCb (cf. Fig. 1.2) est relativement différente de celle d'ATLAS ou CMS. Les paires de quarks lourds étant principalement produites vers des pseudo-rapidités élevées, le détecteur n'a pas besoin d'être hermétique et il couvre une acceptance beaucoup plus réduite située surtout vers l'avant. Par contre, LHCb est un assemblage de sous-détecteurs dont les principes de détection sont proches de ceux de CMS et ATLAS. La principale différence étant la présence dans LHCb d'un imageur Cherenkov (le RICH), permettant d'identifier précisément les mésons légers.

Dans une expérience de physique des saveurs lourdes, l'identification du vertex de désintégration du hadron b ou c est un point fondamental. En effet, compte-tenu du temps de vie relativement long des mésons beaux ou charmés, ce vertex secondaire est distant du vertex primaire de quelques millimètres. Le système de déclenchement de LHCb est largement basé sur l'identification de cette signature particulière, le détecteur réalisant cette identification est donc essentiel.

Ce détecteur de vertex de désintégration, c'est le VELO (contraction de VErtex LOcator) [8]. Ce trajectographe de poche (il mesure à peine 1m de long) doit être capable d'identifier un vertex avec une résolution spatiale de quelques microns, et ce à proximité du point d'interaction.

Ce détecteur (voir les figures 1.3 et 1.4) est constitué de deux boîtes de 21 stations, chaque station étant elle-même composée de deux détecteurs comportant chacun plusieurs milliers de strips de silicium. Ce sont ces deux détecteurs qui permettent de reconstruire le parcours de la particule : l'un donne la position radiale, l'autre la position azimutale. Il s'agit d'une topologie très originale, elle est présentée sur la Figure 1.5.

FIGURE 1.3 – Le VELO de LHCb

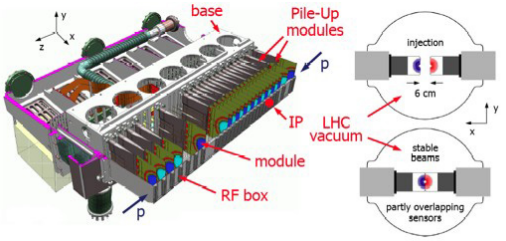


FIGURE 1.4 – Principe de fonctionnement du VELO

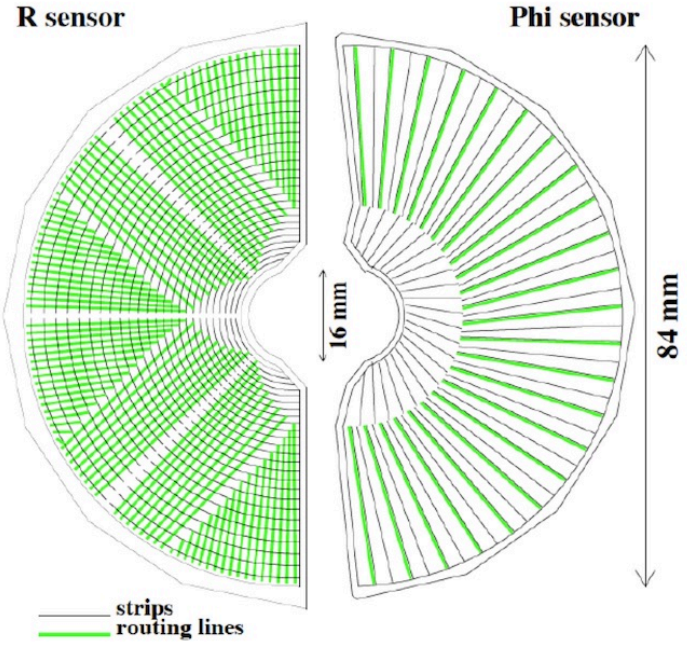


FIGURE 1.5 – Vue détaillée des strips d'un module du VELO

Outre ces modules, ce qui rend le VELO très singulier par rapport aux autres trajectographes, c'est sa proximité avec le faisceau du LHC (seulement 8 mm). Cette proximité impose une condition supplémentaire et très particulière. En effet, afin de protéger les détecteurs des instabilités du faisceau, les boîtes contenant les stations sont rétractées de 3 cm tant que celui-ci n'est pas stabilisé.

Or, pour assurer un fonctionnement optimal du système de déclenchement de LHCb, il est nécessaire de connaître avec précision la position de chaque détecteur, et ce à tout moment, durant la prise de données. Les mouvements fréquents du VELO, ainsi que la précision nécessaire au bon fonctionnement du détecteur, rendent cet étalonnage particulièrement complexe. Il doit en effet non seulement être très précis, mais également réalisable en quelques minutes au début de chaque session de prise de données.

1.3.2/ LE PROBLÈME DE L'ALIGNEMENT

La problématique de l'alignement est expliquée sur la Figure 1.6. On y voit représenté un détecteur très basique constitué de 5 modules, traversé par une particule chargée (trait bleu). Comme on peut le constater sur la figure de gauche, les 5 modules ne sont pas parfaitement alignés. Le trait est un peu forcé dans cet exemple, les déformations sont bien évidemment bien plus petites en pratique. Mais lorsque l'on veut obtenir une résolution de quelques micromètres, un déplacement de 50 microns (l'épaisseur d'un cheveu) doit être corrigé.

FIGURE 1.6 – Alignement et reconstruction de traces

Un problème survient lorsque l'on tente de reconstruire la trace sans connaître les déformations, c'est à dire en considérant que les modules sont à leur place nominale. C'est ce que l'on voit sur la figure de droite. Dans ce cas, les propriétés de la trajectoire, et en particulier son origine, sont erronées. Il peut même arriver, si les déformations sont trop importantes, que la trace ne soit pas détectée du tout. Il faut donc établir un moyen d'évaluer ces déformations, afin de réaligner le détecteur.

Il existe de nombreuses méthodes pour aligner un détecteur. Généralement on réalise cette opération en deux étapes. Dans un premier temps, on effectue une métrologie précise, qui permet de connaitre la position des éléments avec une précision de quelques dizaines de microns. Cette étape sert de point de départ aux méthodes numériques qui utilisent les signaux détectés pour abaisser la précision à quelques microns.

Plusieurs solutions sont alors possibles, mais le principe de base est toujours le même : on part d'un signal identifié et connu (muons cosmiques, vertex de désintégration d'une particule connue) et on compare le signal reconstruit avec le signal attendu. Dans le cas présent, le signal peut être la position de la trace, du vertex, ou la masse de la particule. L'algorithme d'alignement a pour but de minimiser l'écart entre la valeur mesurée et la valeur attendue. Cet écart est ce qu'on appelle le résidu.

Un bon alignement nécessite donc 2 ingrédients : un bon algorithme, et un bon échantillon de données.

Un bon algorithme est un processus qui permet de minimiser rapidement les résidus sans biaiser le résultat final (certains modes de déformation peuvent être insensibles à la minimisation). Un bon échantillon de données est un ensemble d'informations qui permettent à l'algorithme de converger le plus rapidement possible. Plus la topologie des traces utilisées est variée, moins le risque de biais sur le résultat final est élevé.

Pour une revue des différentes techniques d'alignement mises en place au LHC, on pourra se référer au document suivant [9].

1.3.3/ CONTRIBUTION PERSONNELLE

Mon travail entre 2005 et 2007 a consisté à développer et mettre en place l'alignement du VELO de LHCb. La procédure mise au point, qui est encore utilisée aujourd'hui, est décrite en détails dans [10] (note reproduite dans l'Annexe A.1).

A priori le VELO est un détecteur qui se prête relativement bien à l'alignement. Il n'y a pas de champ magnétique et peu de modules. On peut donc mettre en place des méthodes numériques non itératives (minimisation de chi-deux global) assez aisément. L'algorithme a donc été rapidement choisi.

La complexité du problème réside plutôt dans l'adaptation de l'algorithme au cas particulier du VELO. La symétrie polaire des canaux de détections, entre autres, a nécessité une attention particulière par la mise en place d'une étape d'alignement intermédiaire. Le choix des traces nécessaires à l'alignement optimal du système, ainsi que leur sélection, a également requis la mise en place d'une stratégie dédiée. On ne peut pas réaliser un alignement avec seulement des traces provenant de collisions, il faut varier les topologies et utiliser par exemple des traces provenant des muons du halo du faisceau, qui traversent l'ensemble du détecteur. Mais pour les utiliser, il faut les avoir sélectionnées en amont, et donc avoir élaboré au préalable les procédures de sélection adéquates.

La procedure mise en place a pu être vérifiée en conditions réelles dès 2006 lors de test en faisceaux [11] (note reproduite dans l'Annexe A.2).

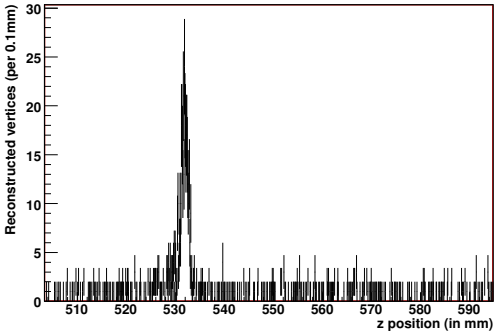


FIGURE 1.7 – Vertex reconstruits avant alignement

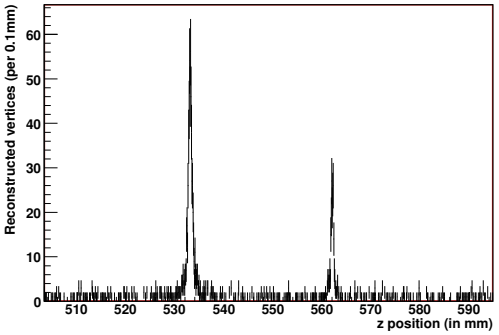


FIGURE 1.8 – Vertex reconstruits après alignement

Les Figures 1.7 et 1.8 illustrent très simplement l'impact que peut avoir l'alignement sur la qualité des informations reconstruites. Lors des tests en faisceaux, 2 cibles avaient été placées en amont des modules à tester. Les modules recevant les traces issues des interactions du faisceau primaire avec les 2 cibles, on pouvait s'attendre à observer une augmentation notable du nombre de vertex reconstruits autour de ces positions. La Figure 1.7 montre qu'avant correction de l'alignement, une seule des 2 cibles est identifiée. Les traces venant de la 2ème cible ne sont pas reconstruites, car la position des modules est trop éloignée de la réalité. Après application des corrections obtenues par la procédure d'alignement, la deuxième cible apparait et le nombre d'interactions provenant de la première cible augmente significativement (Figure 1.8). L'effet est visible et immédiat.

A l'heure actuelle, c'est toujours cette procédure qui est utilisée pour aligner le VELO de LHCb au début de chaque prise de données. Pour les données enregistrées, la rapidité

d'exécution n'est pas une contrainte et cet alignement est complété par une méthode plus complexe qui permet d'aligner l'ensemble des sous-détecteurs de traces entre-eux [12].

1.4/ Mise en route du LASER de calibration du calorimètre à tuiles dans ATLAS : 2008-2010

1.4.1/ Le détecteur ATLAS

Sans minimiser l'importance de LHCb et ALICE, on peut dire qu'ATLAS et CMS sont les 2 expériences principales du LHC. Le programme de recherche de ces dernières couvre en effet un spectre beaucoup plus étendu.

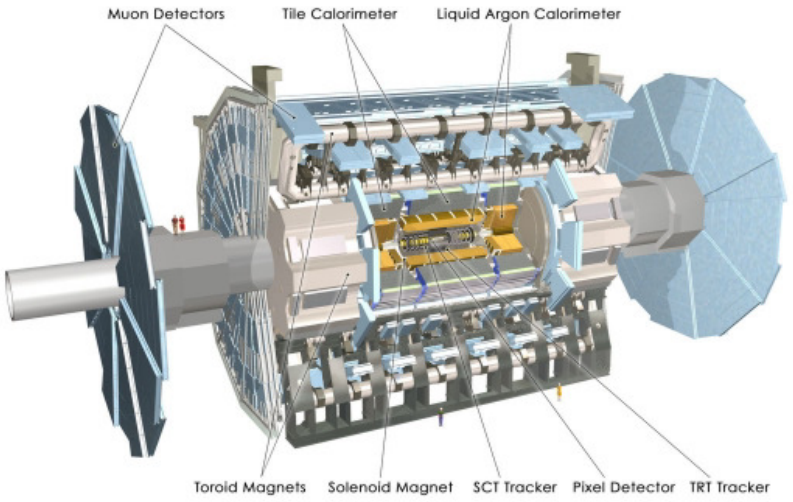


FIGURE 1.9 – Le détecteur ATLAS

Comme on peut le voir sur la figure 1.9, ATLAS [13] est un détecteur immense. Sa complexité est à l'avenant, l'ensemble du détecteur ATLAS contenant environ 100 millions de cellules de détection indépendantes. Il s'agit d'une des machines les plus complexes réalisées à ce jour.

La taille de la collaboration ATLAS est également impressionnante, plusieurs milliers de personnes contre quelques centaines pour LHCb (ce qui est déjà tout à fait respectable, au demeurant). L'organisation est forcément très différente. Dans ATLAS, chaque sous-détecteur est une expérience à part entière, il est beaucoup plus difficile d'avoir une perception précise de l'ensemble du détecteur. C'est clairement une autre manière de travailler, avec ses avantages et ses inconvénients.

1.4.2/ Le calorimètre à tuiles d'ATLAS et son étalonnage

Le calorimètre à tuiles (TileCal) [14] est un des sous-détecteurs d'ATLAS dont le but est de mesurer l'énergie des particules qui le traversent. Ce sous-détecteur est un sandwich de plaques de fer et de tuiles scintillantes. La particule arrivant dans le calorimètre se désintègre en créant une gerbe de particules d'énergies moindres. Les particules

chargées font scintiller les tuiles, et la lumière ainsi créée est transmise via des fibres optiques à des photo-multiplicateurs (PMT) (Figure 1.10). Les charges ainsi collectées sont numérisées puis transmises au système d'acquisition d'ATLAS. Les informations de chaque PMT du calorimètre (il y en a exactement 9852) sont alors mises en commun afin de remonter à l'énergie de la particule ayant initié la gerbe.

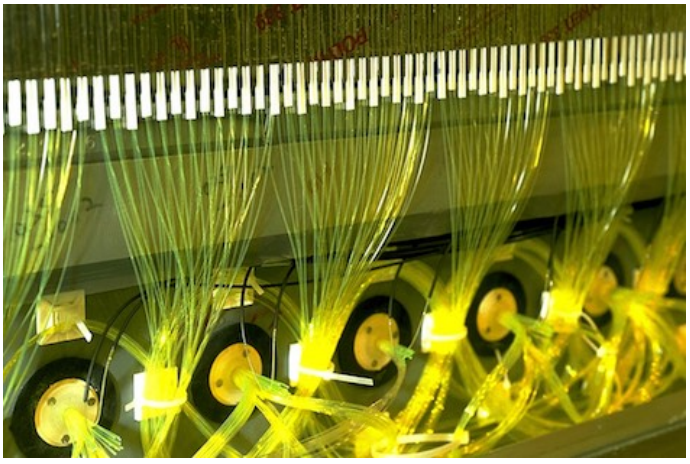


FIGURE 1.10 – Vue des fibres et des photomultiplicateurs d'un secteur du calorimètre à tuiles

Afin d'obtenir les mesures les plus précises possibles, la résolution en énergie du détecteur doit être optimale. Le but de l'étalonnage en énergie est que la différence entre l'énergie reconstruite et l'énergie réelle de la particule soit la plus petite possible...

Pour y parvenir on met en oeuvre 2 niveaux de calibration :

— **Calibration de bas niveau :** le TileCal est un système complexe. Cela signifie que son bon fonctionnement requiert le suivi et le contrôle d'un nombre important de paramètres (bruits, gains des PMT,...), ces paramètres pouvant dépendre du temps, de la température, etc... La connaissance de ces informations est critique pour ATLAS. Par exemple, une valeur ADC sera totalement inutile si l'on ne dispose pas du niveau de bruit correspondant (piédestal). En conséquence de quoi tous ces paramètres doivent être mesurés précisément et pris en compte lors de la reconstruction des données (via des bases de données).

— **Calibration de haut niveau :** une fois que la partie précédente est sous contrôle, l'énergie effectivement reçue par le détecteur est connue précisément. Il reste néanmoins à estimer correctement l'énergie de la particule initiale. En effet, le calorimètre ayant une granularité finie, l'énergie reconstruite par ce dernier sera forcément biaisée. Ceci est corrigé par des méthodes dédiées (algorithme de reconstruction des jets,...).

Le LASER est l'un des systèmes de calibration de bas niveau du calorimètre à tuiles. Il fournit, avec 2 autres systèmes (Césium et injection de charges), un étalonnage complet de la chaîne d'acquisition. Ces systèmes permettent de transformer le signal effectivement mesuré par le détecteur en une énergie utilisable par les logiciels de reconstruction de données. Leurs rôles respectifs sont illustrés sur la figure 1.11.

Le système Césium [15] fournit une calibration de toute la chaîne (hormis l'électronique).

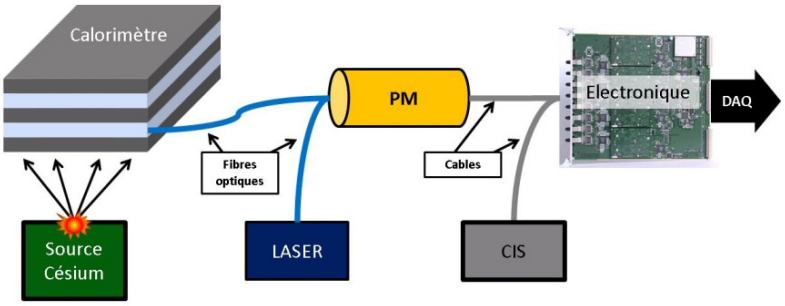


FIGURE 1.11 – Calibration bas niveau du calorimètre à tuiles

Une source de Césium 137 est déplacée dans le calorimètre. Le signal alors produit est collecté et mesuré pour chacune des cellules du TileCal. On peut ainsi obtenir une comparaison de la réponse de chaque cellule, et en déduire les corrections qui permettent d'uniformiser la réponse du calorimètre. Il est en effet très important, pour que la résolution du détecteur soit optimale, que la réponse d'une cellule à un dépôt d'énergie donné soit la même partout.

Le système Césium mesure un signal traversant à la fois la partie active du detecteur et le PMT. Néanmoins, si l'énergie mesurée dans une cellule est plus faible que la normale, cela peut tout aussi bien être dû à l'instabilité d'un PMT. Il faut donc pouvoir contrôler indépendamment ces-derniers. C'est là qu'intervient le système LASER[16]. La lumière d'un LASER est envoyée directement dans les PMTs via des fibres optiques. Comme l'intensité de la lumière envoyée est connue avec précision, il est possible d'en déduire la stabilité de la réponse du PMT, mais aussi de l'électronique de lecture (qui cette fois-ci est prise en compte) en reconstruisant le signal envoyé. Qui plus est, l'information ainsi obtenue peut-être utilisée pour égaliser la réponse du calorimètre à tuiles entre 2 analyses Cesium (celles-ci ne sont en effet effectuées que tous les 3 ou 4 mois).

Le système LASER permet donc de suivre également l'électronique de lecture du calorimètre à tuiles. Afin de caractériser les problèmes propres à cette partie du détecteur, un système d'injection de charges [17] complète la procédure de calibration de bas niveau.

Le groupe ATLAS de Clermont, que j'ai rejoint lors de mon recrutement au CNRS en 2007, est à l'origine du développement de ce système de calibration. Ses contributions concernent, outre le LASER lui-même, toute son électronique de contrôle et l'ensemble du système de calibration interne. J'ai eu la responsabilité de ce système entre 2008 et 2010.

1.4.3/ LE SYSTÈME LASER EN QUELQUES MOTS

1.4.3.1/ ASPECTS MECANIQUES

L'origine du système LASER dans ATLAS remonte à 1993 [18], mais c'est un procédé qui a déjà été utilisé avec succès par de nombreuses expériences de physique des hautes énergies (dans le calorimètre électromagnétique de CMS par exemple [19]). L'idée est d'envoyer de courtes impulsions lumineuses dans les PMTs du détecteur. Ceux-ci reproduisent en effet relativement bien les signaux créés par de véritables

particules. L'avantage, dans le cas du LASER, est que l'on connait précisément l'énergie de l'impulsion émise. Cela permet donc de suivre et de contrôler efficacement la stabilité et la linéarité des PMTs.

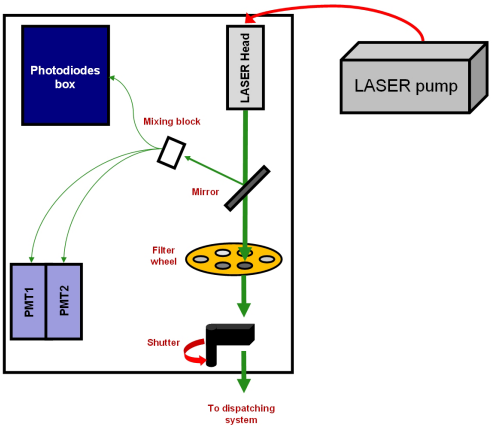


FIGURE 1.12 – Description de la boite LASER

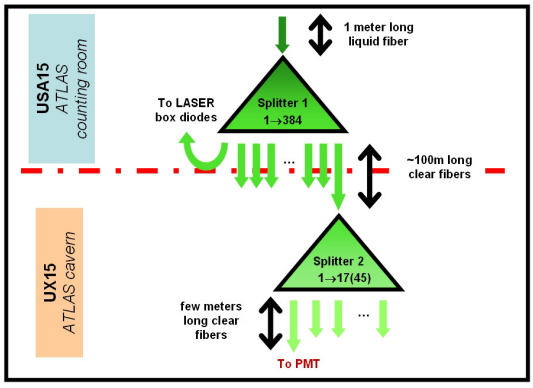


FIGURE 1.13 – Système de distribution de lumière du système LASER

Le système est divisé en deux parties distinctes, la boite LASER (figure 1.12) et le système de distribution de lumière (figure 1.13).

La boite LASER constitue le coeur du système. Elle contient en effet tous les éléments qui permettent de mesurer précisément l'amplitude de l'impulsion LASER émise, mais aussi de contrôler son déclenchement. Il est en effet possible d'envoyer des impulsions LASER pendant les prises de données afin de suivre en direct l'évolution du gain des PMTs.

La mesure de l'amplitude du signal émis est très importante. Cela permet en effet de normaliser les signaux mesurés dans les PMTs du TileCal et ainsi de gommer les instabilités éventuelles du LASER lui-même. Cette mesure est faite au moyen de 4 photodiodes qui collectent la lumière à différents niveaux. Ces diodes sont contenues dans une boite thermalisée, et leur stabilité absolue est régulièrement mesurée au moyen d'une source alpha d'$^{241}\mathrm{Am}$.

L'étude de la linéarité impose de couvrir précisément l'intégralité de la dynamique du détecteur (de 100 MeV à 1.5 TeV par PMT). Pour y parvenir, la boite LASER comporte une roue à filtres qui permet d'atténuer plus ou moins, selon sa position, la lumière émise.

La lumière sortant de la boite LASER est ensuite distribuée aux 9852 PMTs en deux étapes. Une première distribution est effectuée à la sortie de la boite, par un premier élément distribuant la lumière à environ 400 fibres optiques. La quantité de lumière distribuée par ces fibres peut alors être égalisée au moyen de connecteurs spécifiques, afin d'uniformiser la réponse du système. Elle est ensuite transportée dans chacun des modules du TileCal (100m plus loin), où elle est finalement distribuée à un certain nombre de fibres reliées directement aux PMTs.

1.4.3.2/ ASPECTS ELECTRONIQUES

L'électronique du système LASER, entièrement développée par les services techniques du LPC, permet de contrôler les divers élements de la boite LASER. Elle permet également de contrôler très précisément l'émission des impulsions. Comme dit précédemment, le système LASER permet d'envoyer des impulsions pendant que le LHC fonctionne. La procédure, décrite sur la figure 1.14, profite du temps mort de $3\ \mu s$ survenant à la fin de chaque cycle du LHC.

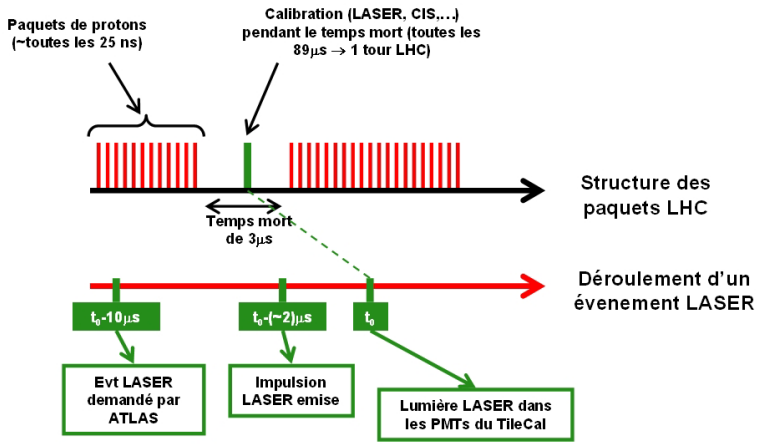


FIGURE 1.14 – Déroulement d'un évenement LASER

Le système a été dimensionné afin de permettre aux impulsions LASER d'arriver dans le TileCal exactement au milieu du temps mort. Il a été démontré qu'une précision d'environ 15ns sur le temps d'arrivée pouvait être obtenue, ce qui est largement suffisant.

1.4.4/ CONTRIBUTION PERSONNELLE

Responsable du système LASER du TileCal d'ATLAS entre 2008 et 2010, j'ai suivi et encadré les étapes qui ont permis de faire fonctionner et d'intégrer dans ATLAS ce système complexe [16]. J'ai ensuite pris une part importante dans l'analyse des résultats obtenus au cours de très nombreuses phases de tests. Finalement, j'ai développé et validé au CERN, les procédures qui permettent d'utiliser le système LASER pour le suivi des PMTs du TileCal in situ.

En parallèle, j'ai établi un cahier des charges en vue du remplacement du système initial par un nouveau LASER de calibration répondant mieux à nos attentes : le LASER II. Ce cahier des charges a servi de base au développement de cette nouvelle version, qui est aujourd'hui déployée dans ATLAS [20]. L'ensemble de ces travaux a été documenté en détails dans des notes internes. Ces notes sont reproduites dans les Annexes B.1 et B.2 de ce document.

2

LE HL-LHC ET LES MÉGOTS DE CMS

Le projet HL-LHC correspond à une montée en puissance significative de l'accélérateur. Les détecteurs vont subir des modifications profondes pour répondre à ces nouvelles conditions de fonctionnement. L'amélioration du trajectographe de CMS s'inscrit dans ce contexte. C'est un projet très ambitieux qui va bien au-delà d'un simple rafraichissement. L'innovation principale de ce nouveau détecteur, son introduction au premier niveau du système de déclenchement, est un défi technique inédit pour un système de cette taille.

2.1/ LE PROJET HL-LHC

Le LHC fournit depuis 2015 des données à une énergie record des collisions proton-proton de 13 TeV. D'ici 2025, plus de 300 fb^{-1} de données auront été enregistrés avec sa configuration actuelle. A ce stade, certains détecteurs et aimants soumis à des doses de radiations importantes seront en fin de vie. Un programme d'adaptation du collisionneur et des détecteurs sera alors nécessaire pour aller au-delà et envisager de collecter encore 10 fois plus de données. C'est le projet de Haute Luminosité du LHC (HL-LHC) [21], qui vise une collecte d'au moins 3000 fb^{-1} sur une période de 10 ans à partir de 2026. Ces différentes étapes sont résumées sur la Figure 2.1.

L'augmentation importante de la luminosité aura une incidence majeure sur le fonctionnement des futurs détecteurs. C'est particulièrement vrai concernant l'augmentation de l'empilement, c'est à dire du nombre moyen d'interactions survenant à chaque collision. Un événement contenant 5 à 10 fois plus d'interactions secondaires est naturellement plus complexe à analyser, mais surtout beaucoup plus complexe à sélectionner. Or la sélection des événements doit rester aussi efficace qu'au LHC, sans quoi les bénéfices d'une augmentation de la luminosité seront perdus. Il faut donc construire des détecteurs adaptés à ce nouvel environnement, à la fois plus précis et permettant une analyse plus rapide des informations reçues.

On parle souvent de mise à niveau pour qualifier ces nouveaux détecteurs. Ce terme est trompeur car il donne l'impression d'une simple amélioration. Comme nous allons le voir dans la suite, ce sont des systèmes complets qui sont développés, dans des conditions beaucoup moins favorables. En effet, ces développements doivent se faire

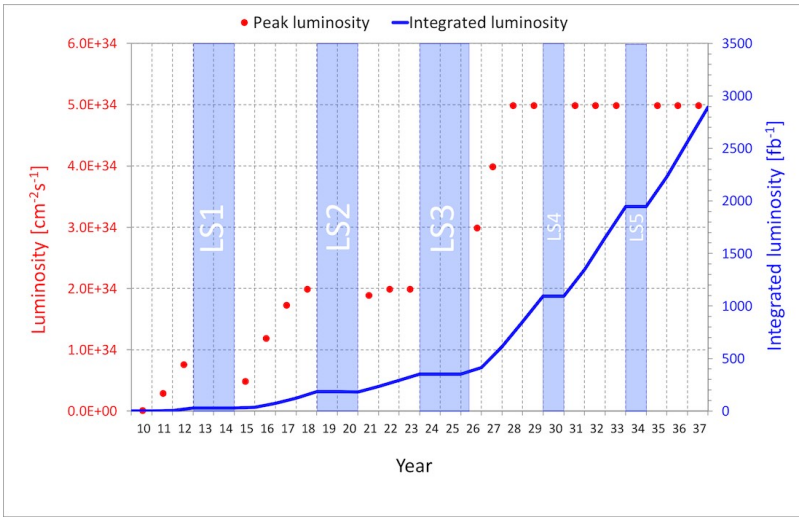


FIGURE 2.1 – Les différentes phases du projet LHC

alors que les expériences actuelles sont en fonctionnement, avec des moyens humains fortement réduits. Au début des années 2000, lors de la construction du LHC, la majeure partie des services techniques des laboratoires impliqués prenait part à cet effort. La situation du HL-LHC est bien différente, les responsables de projets doivent s'y adapter au quotidien.

2.2/ LE DÉTECTEUR DE TRACES DE CMS

Comme nous l'avons déjà dit plus haut, la montée en puissance du LHC s'accompagne d'une augmentation de la complexité des événements à reconstruire. Cette complexité accrue a pour effet une augmentation du taux d'occupation, augmentation qui pose plusieurs problèmes : reconstruction plus complexe et sélection en ligne moins efficace.

La conséquence directe de ces 2 facteurs est une baisse de la performance du détecteur, ce qui n'est pas vraiment le but recherché...

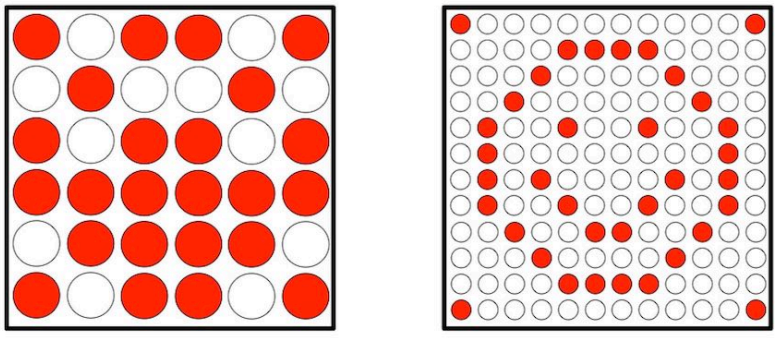


FIGURE 2.2 – Taux d'occupation et granularité : illustration

Il y a 2 façons de diminuer le taux d'occupation dans un détecteur : diminuer l'activité (moins de matière traversée), et augmenter sa granularité.

L'effet d'une augmentation de la granularité est illustré de manière simple sur la Figure 2.2. Avec des canaux 4 fois plus petits, on divise le taux d'occupation par 4 et on améliore la résolution. Par contre le nombre de canaux à lire est multiplié par 4, l'extraction des informations devient donc plus complexe.

Cela devient un problème lorsqu'il s'agit de sélectionner rapidement les événements. Si l'on veut disposer plus rapidement d'informations plus précises, il faut modifier en profondeur le système de déclenchement. C'est d'ailleurs de là, comme nous le verrons bientôt, que provient la contrainte la plus importante pour le futur trajectographe.

Le système de déclenchement de CMS (on parle souvent de trigger) sera en effet très sensible aux montées en puissance prévues. Le défi pour le nouveau système consistera à conserver un excellent niveau de performance au premier niveau (lecture des informations à 40MHz, latence de 12μs), voire à l'améliorer.

Jusqu'à présent, le niveau 1 du trigger de CMS est basé uniquement sur les détecteurs externes possédant un taux d'occupation moyen bien plus faible que le trajectographe : le spectromètre à muons et les calorimètres. Dans le contexte du HL-LHC, les taux d'empilements attendus seront tels qu'il deviendra impossible de conserver un niveau 1 aussi performant sans inclure le trajectographe dans celui-ci. La granularité des détecteurs externes ne sera en effet pas ou peu modifiée, et leur taux d'occupation sera beaucoup plus important. Il deviendra difficile d'interpréter leurs signaux sans utiliser les informations du trajectographe.

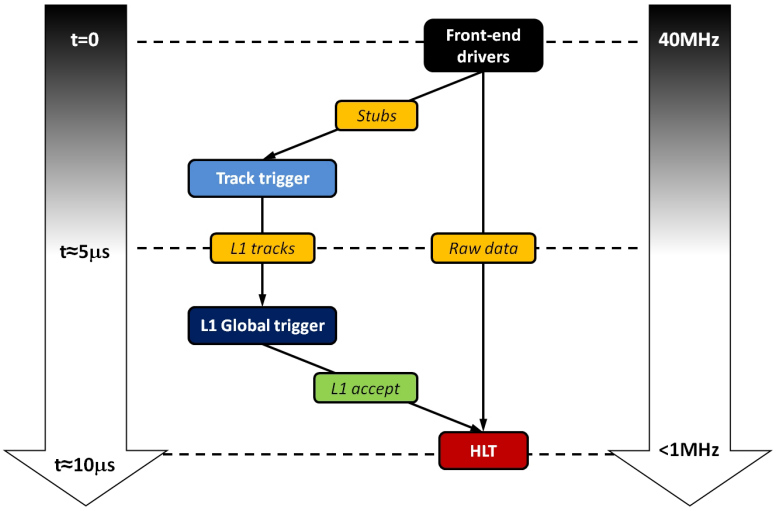


FIGURE 2.3 – Flux de données dans le trajectographe phase II

L'utilisation du trajectographe au niveau 1 est un problème ardu, dont le principe est résumé sur la Figure 2.3. Il faut être capable d'extraire les informations utiles du trajectographe à 40 MHz, de reconstruire électroniquement les traces associées à un événement (track trigger), et d'exploiter ces informations avec celles des autres sous-détecteurs (L1 global trigger), le tout dans la latence disponible qui sera, au HL-LHC de 12.5 microsecondes. En parallèle, la totalité des données du trajectographe devra pouvoir être lue pour chaque événement passant le niveau 1, à une fréquence pouvant aller jusqu'à 1MHz. Il faut donc être capable d'extraire 2 flux de données différents en parallèle, mais également d'analyser un de ces flux quasiment en temps réel (moins de 5 microsecondes). Dans le trajectographe actuel de CMS, un seul flux de données est extrait à la fréquence de

100kHz. Le nouveau détecteur est un changement complet de paradigme.

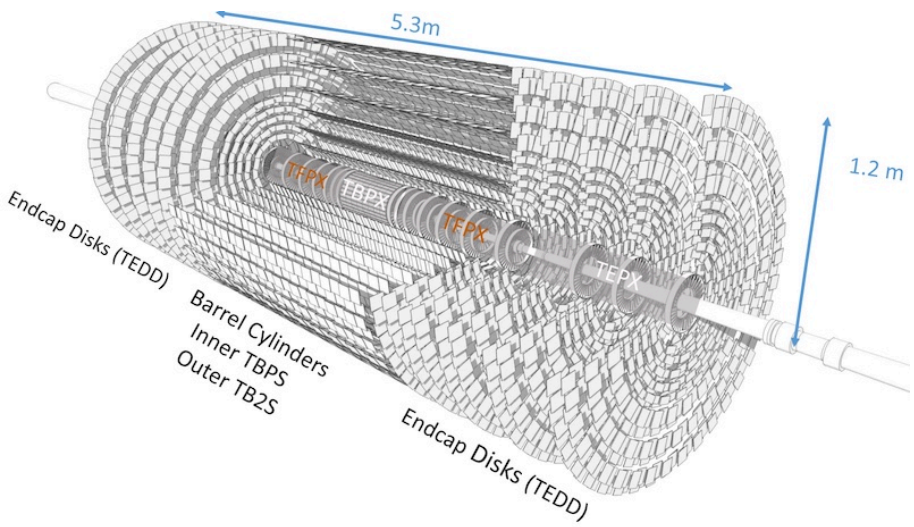


FIGURE 2.4 – Le futur trajectographe de CMS

La Figure 2.4 est une représentation en 3 dimensions du futur trajectographe. Il s'agit d'une version ancienne (connue sous le nom de géométrie plate), mais elle permet d'avoir une idée précise de la structure globale du système. On distingue les 2 régions qui permettent de couvrir l'ensemble de l'espace de phase : le tonneau central et les bouchons. Le tonneau est composé de 6 couches, et chaque bouchon de 5 disques. Les 15000 modules de détection sont répartis sur ces supports de manière à assurer un taux de couverture optimal. Afin de diminuer la quantité de matière à traverser, le nombre de couches est moindre que dans le détecteur actuel. Néanmoins, une particule chargée traversant le futur trajectographe laissera un nombre de points de mesures suffisant pour le track trigger (5 à 6 en moyenne). La structure complète du détecteur a été entièrement repensée : supports, services, modules,... Une cure d'amaigrissement généralisée a été effectuée. En moyenne, la quantité de matière traversée a été divisée par 2 par rapport au détecteur actuel.

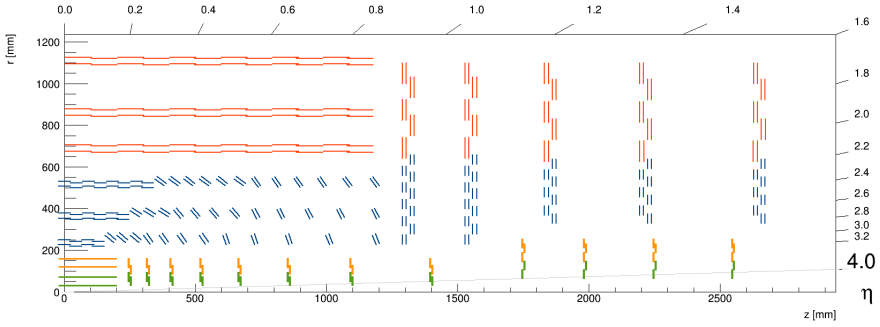


FIGURE 2.5 – Vue de la répartition des modules de détection dans le plan R-Z

Les modules de détection sont les éléments fondamentaux de la nouvelle stratégie de lecture des données. La Figure 2.5 fournit une vue plus détaillée de la répartition de ces modules, dans le plan R-Z. Les lignes rouges et bleues indiquent les positions des 2 types de modules qui équiperont le détecteur à terme. Afin de conserver un taux d'occupation raisonnable dans ces zones internes, les modules bleus, dénommés PS (Figure 2.7),

auront une granularité plus importante que les rouges, nommés 2S (Figure 2.6). Les modules jaunes et verts, quant à eux, correspondent à un détecteur plus interne constitué de pixels de silicium. Pour le premier niveau du futur système de déclenchement, il n'est prévu d'utiliser que les modules 2S et PS.

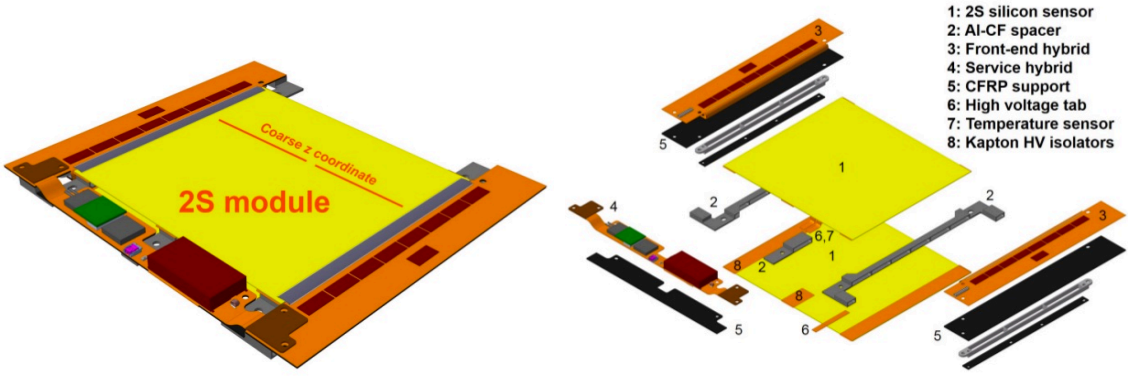


FIGURE 2.6 – Vue d'un module 2S (ensemble et détail).

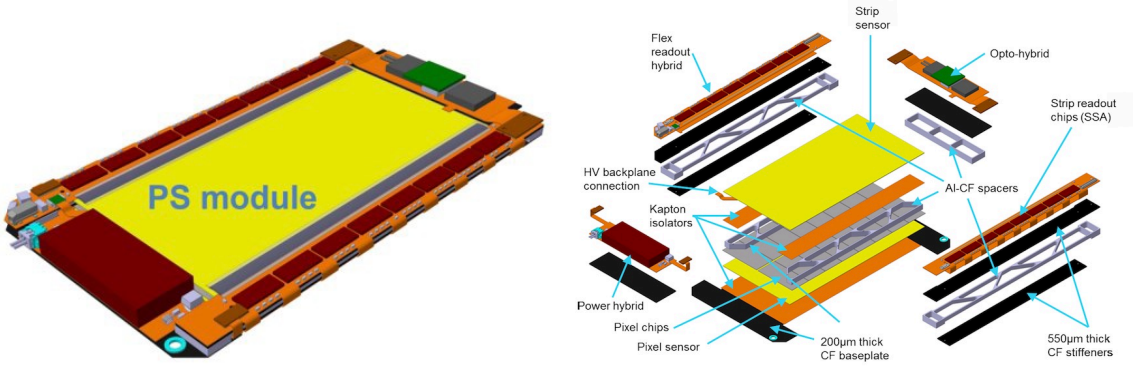


FIGURE 2.7 – Vue d'un module PS (ensemble et détail).

Mécaniquement, ces modules sont constitués de 2 couches de silicium rectangulaires et adjacentes (la distance entre les 2 couches est de quelques mm), représentées en jaune sur les Figures 2.7 et 2.6.

Pour les modules 2S, les 2 couches sont identiques et chacune d'entre elles contient environ 1000 strips de 90μm de large par 10 cm de long. Pour les modules PS, une couche contient des strips de 1.5mm de long au lieu de 5cm, d'où la granularité plus importante de ces modules. Cette nouvelle configuration explique le fait qu'au final, le futur trajectographe contiendra environ 10 fois plus de canaux de détection que le précédent.

Le flux de données sortant du trajectographe sera donc beaucoup plus important que dans le détecteur actuel. A 40MHz, l'ensemble des signaux représentera une masse de données de 10^4Tbps !!! Il serait totalement irréaliste, et inutile, de chercher à extraire une telle quantité d'informations. Un filtrage est donc effectué dans chaque module. Les signaux des 2 couches de silicium sont mis en coïncidence afin de former des superclusters nommés stubs, ou mégots si l'on tient à tout prix à chasser les anglicismes. Le principe de base d'un stub est illustré sur la Figure 2.8.

L'utilisation des stubs doit permettre de diviser la quantité totale de données à extraire par environ 200, soit un taux de données en sortie de 50 Tbps. C'est toujours relativement

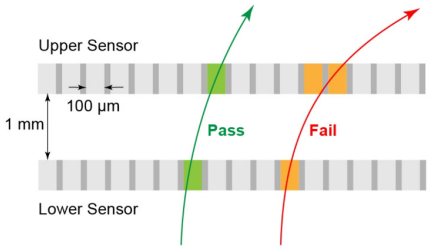


FIGURE 2.8 – Stub : principe de base

important, mais techniquement cela devient acceptable. Par contre, ce concept simple requiert une électronique de lecture très élaborée, mais c'est le prix à payer pour permettre au système de reconstruction de trace de disposer de toute l'information nécessaire au niveau 1. Nous reviendrons plus en détails sur l'extraction des informations dans la deuxième partie de ce document.

Une fois le problème de l'extraction des données résolu, reste celui de leur traitement. C'est le second défi de ce futur système, auquel sera consacrée la troisième partie : comment traiter un flux de 50 Tbps en quelques microsecondes ?

III

Produire et extraire les données

3

LE FRONTEND DU TRAJECTOGRAPHE

Ce chapitre s'ouvre sur une description détaillée de l'électronique frontale des modules de détection. C'est grâce à ce système complexe que sont reconstruits les stubs, objets fondamentaux pour le système de reconstruction de traces rapides. La deuxième partie est consacrée à ces objets, dont il est primordial de comprendre les atouts, mais aussi les limites.

3.1/ DESCRIPTION DE LA CHAINE D'ACQUISITION

3.1.1/ PRINCIPE GÉNÉRAL

Les modules ont été décrits dans la partie précédente ; nous allons désormais nous concentrer sur l'élément qui permet de reconstruire et de mettre en forme le signal : l'hybride. Les hybrides sont des circuits imprimés qui connectent les zones de détection (strips de silicium) aux systèmes de reconstruction (ASICs). Leur principe de fonctionnement, commun aux 2 types de modules, est schématisé sur la Figure 3.1.

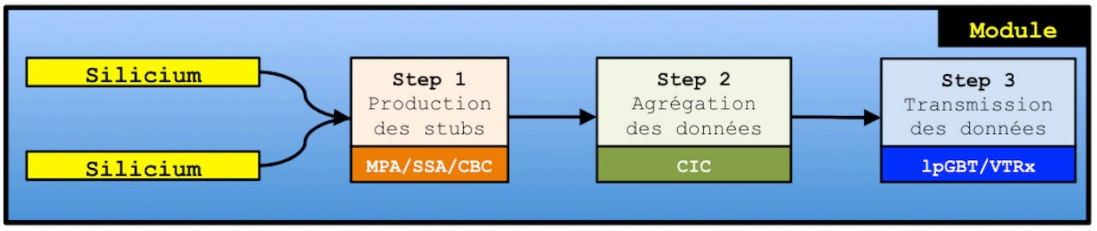


FIGURE 3.1 – Readout : principe de base

Le traitement et l'extraction du signal se font en trois étapes distinctes. Tout d'abord, le signal collecté sur les 2 couches de silicium est transmis simultanément à une série de circuits intégrés dédiés (ASICs) qui l'amplifient, le numérisent, et forment les stubs. Ces ASICs sont les CBC [22] dans les modules 2S, et les MPA [23] et SSA [24] dans les modules PS. Les signaux numérisés sont ensuite transmis à un autre ASIC qui collecte les informations et les met en forme : le concentrateur (CIC) [25]. Le CIC, qui constitue le

deuxième étage de la chaîne, est également localisé sur l'hybride de frontend.

La troisième étape, quant à elle, se déroule sur d'autres hybrides, eux-aussi rattachés aux modules : les hybrides de service. Il y en a un par module 2S et 2 par module PS. Le rôle de ces hybrides est d'assurer la communication entre le module et le monde extérieur. C'est par eux qu'arrive l'alimentation électrique du module, et c'est de là que le signal est extrait du détecteur après avoir été empaqueté dans une trame permettant de s'assurer que les données ne seront pas corrompues (les données sont précieuses et doivent voyager vers la salle de comptage sans encombre, enveloppées dans beaucoup de papier-bulle...). Ce paquetage est pris en charge par le lpGBT [26], qui est lui aussi un ASIC développé pour répondre aux contraintes du projet HL-LHC. Enfin, le signal électrique soigneusement emballé est transmis vers l'extérieur via une fibre optique. La conversion optique du signal est là aussi effectuée par un connecteur spécial développé par le CERN : le VTRx [27].

3.1.2/ Chaîne d'acquisition d'un module 2S

Le fonctionnement du module 2S est représenté sur la Figure 3.2. Les 2 couches de 2032 strips de silicium sont séparées de 1.8 ou 4mm. L'écart entre les couches dépend de la distance du module par rapport au point d'interaction. Plus le module est proche, plus son épaisseur devra être importante afin de conserver une qualité de stub acceptable.

Les strips ont une largeur de 90 microns, et une longueur de 10 cm divisée en 2 sections indépendantes de 5 cm. Ces deux segments définissent les 2 zones d'un module, zones lues par 2 hybrides distincts (gauche et droit).

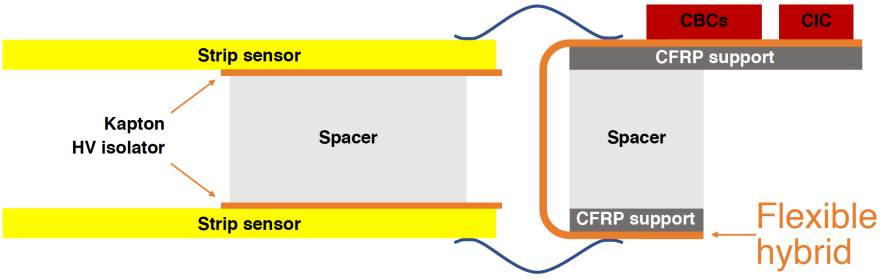


FIGURE 3.2 – Principe de lecture détaillé d'une moitié de module 2S

La Figure 3.2 montre par exemple l'hybride du côté droit. Les strips sont reliés à celui-ci par 2032 micro-câbles (wirebonding). L'hybride étant flexible, le signal des 2 faces peut-être mis en commun et transmis à un seul composant. En pratique, ce sont 8 CBCs qui collectent chacun le signal de 127 strips consécutifs de chaque face (soit 254 voies au total). Les CBCs numérisent ce signal et génèrent les stubs. En parallèle, ils transmettent l'intégralité des données de lecture pour chaque événement passant le niveau 1 du système de déclenchement.

Les signaux des 8 CBCs sont routés vers un CIC qui met en forme et transmet ensuite les informations à l'hybride de service.

Au total, chaque module 2S contient une vingtaine d'ASICs situés sur 3 hybrides distincts (les hybrides de frontend gauche et droit sont différents). La majeure partie des difficultés réside néanmoins dans la mise au point des hybrides de frontend. La réalisation d'un

hybride flexible comportant un routage aussi complexe et embarquant autant d'ASICs est un véritable défi technologique.

3.1.3/ CHAÎNE D'ACQUISITION D'UN MODULE PS

Le principe de fonctionnement du frontend du PS est schématisé sur la Figure 3.3. Là aussi le système est divisé en 2 sections (gauche et droite) et le signal est collecté et transmis par une chaîne CIC/lpGBT/VTRx. La principale différence est dans la granularité du détecteur. La couche de silicium du bas est composée de macro-pixels de 1.5 mm de longueur sur 90 microns de large. La longueur totale du senseur étant de 5 cm, il est donc divisé en 2 côtés de 16 segments chacun.

Comme pour le module 2S, le signal est collecté par un hybride flexible, mais son fonctionnement est radicalement différent. En effet, les macro-pixels sont directement reliés à chaque MPA via un réseau de 1920 billes (une pour chaque voie). L'ensemble des 16 MPAs et du senseur silicium reliés entre eux par 30720 points de contact forment ainsi un objet unique : le MAPSA.

Le signal des strips de la face opposée est quant à lui reconstruit par un autre ASIC, le SSA. Les informations du SSA sont transmises via l'hybride flexible au MPA, qui met alors en commun les signaux des deux côtés du module, et construit les stubs, au même titre que le CBC. Comme dans le cas des modules 2S, chaque paire MPA/SSA couvre seulement une partie d'un module : 120 strips dans le cas des modules PS.

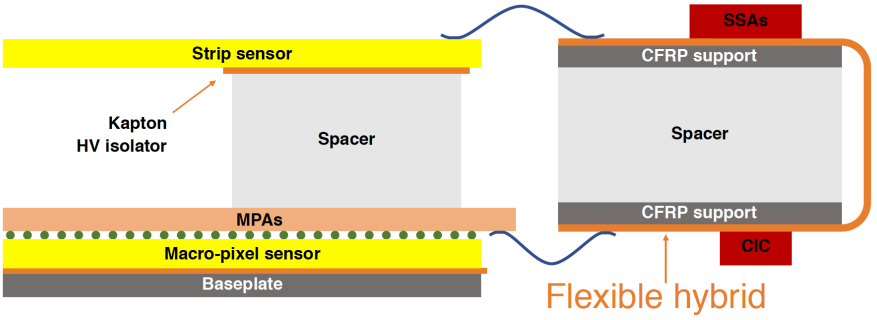


FIGURE 3.3 – Principe de lecture détaillé du module PS

Comme pour les modules 2S, l'épaisseur des modules varie entre 1.6 et 4 mm, avec cette fois une valeur intermédiaire de 2.6 mm, selon leur position dans le détecteur.

L'extraction des informations se fait via un hybride de service relativement différent de celui du module 2S. En effet, celui-ci ne contient que les éléments liés à la transmission des informations de et vers l'extérieur. Contrairement aux modules 2S, l'alimentation du module PS est gérée par un hybride dédié situé de l'autre côté du module.

D'une manière générale, les modules PS sont encore plus complexes que les modules 2S. Le MAPSA et le routage de l'hybride de frontend sont les principales difficultés à surmonter. Les multiples ASICs présents sur le module sont également des projets difficiles à mener à bien, nous le verrons dans la partie consacrée au concentrateur.

Pour le module PS comme pour le 2S, réaliser un premier prototype fonctionnel sera une étape majeure, tant ces systèmes sont ambitieux sur le plan technique.

3.2/ LE STUB : OBJET FONDAMENTAL DU FUTUR TRACKER

3.2.1/ PRÉSENTATION GÉNÉRALE

Si les modules PS et 2S sont aussi complexes, c'est avant tout pour pouvoir fournir des stubs au système de reconstruction de traces rapide. Il est temps de s'intéresser de plus près à cet objet fondamental.

Comme on peut le voir de manière détaillée sur la Fig. 3.4, un stub est reconstruit chaque fois que deux clusters corrélés provenant d'une trace intéressante sont détectés.

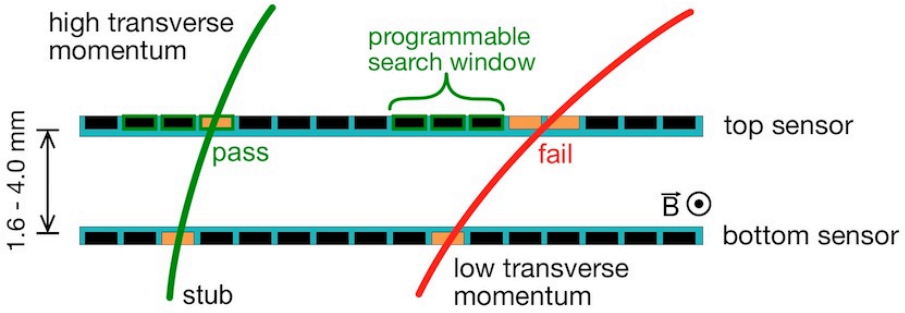


FIGURE 3.4 – Stub : principe de base

On comprend alors aisément le rôle des stubs dans la réduction du flux de données. Le bruit électronique, qui peut induire un faux cluster, est éliminé par construction par ce mécanisme. D'autre part, la largeur du stub, qui est la distance entre les centres des 2 clusters, est directement reliée à l'impulsion transverse (p_T) de la particule incidente (si celle-ci provient du point d'interaction). Il est donc possible, en modulant la largeur de la fenêtre de détection utilisée lors de leur reconstruction, de sélectionner uniquement les stubs provenants de traces de p_T élevé. Une telle coupure permet de réduire significativement l'impact de l'empilement, qui induit principalement des traces de bas p_T, sur la quantité de données à extraire.

Nous allons voir dans la suite de ce document que les stubs sont des objets dont la maitrise peut fournir des informations extrêmement utiles. J'ai été personnellement en charge de leur implémentation dans la simulation officielle de CMS entre 2014 et 2017. Durant cette période, j'ai pu approfondir ma connaissance de ces objets, et fournir à la collaboration des méthodes pour exploiter au mieux leur potentiel. Les conclusions de ce travail sont résumées dans une note de synthèse que j'ai rédigée sur le sujet en 2018 [28] (note reproduite dans l'Annexe C.1). Je résume ci-dessous quelques éléments décrits dans cette note qui me paraissent important pour la bonne compréhension des chapitres suivants.

3.2.2/ RELATION ENTRE LA LARGEUR DU STUB ET L'IMPULSION TRANSVERSE

La largeur d'un stub est fondamentale à plus d'un titre : pour la réduction du flux de données à extraire, bien sûr, mais aussi, nous le verrons, pour la reconstruction de traces. C'est en effet un paramètre qui peut être utilisé pour vérifier la cohérence entre plusieurs stubs d'un candidat trace. Si l'on ne considère que des particules primaires, la définition géométrique et simplifiée des stubs est décrite par la Figure 3.5. La géométrie exacte,

quant à elle, est présentée sur la Figure 3.6.

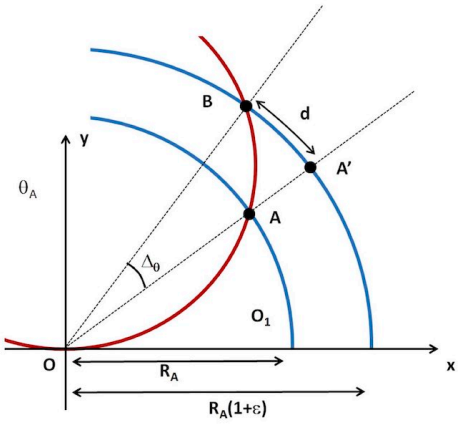


FIGURE 3.5 – Définition de base d'un stub provenant d'une particule primaire.

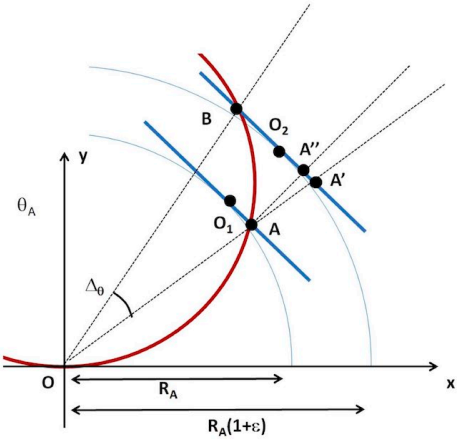


FIGURE 3.6 – Corrections appliquées pour prendre en compte la planéité des modules.

En partant de la Figure 3.5, on peut établir la relation suivante :

$$p_T = 0.57 \cdot q \cdot R_A \sqrt{1 + \left(\frac{\epsilon'}{pitch \cdot bend}\right)^2} \tag{3.1}$$

où R_A, rayon du cluster le plus proche du point d'interaction, est donné en mètres, ϵ' et $pitch$ en mm, et $bend$ en nombre de strips. ϵ' est la distance radiale entre les 2 clusters formant le stub ($\epsilon' = \epsilon \cdot R_A$), $pitch$ la largeur d'un strip et $bend$ la largeur du stub.

Cette formule permet d'établir des relations simples entre les différents paramètres. La largeur minimale reconstruite par les chips de frontend étant de 0.5 strip, on peut par exemple facilement estimer la sensibilité maximale d'un module, c'est à dire la valeur maximale d'impulsion à laquelle il est techniquement sensible. Si l'on prend par exemple un module 2S classique, pour lequel on a ϵ'= 1.8 mm et pitch = 0.09 mm, on voit que la sensibilité sera de 5 GeV/c à 20 cm contre 23 GeV/c à 1 m. Le message à retenir ici est que la largeur des stubs fournira toujours une information beaucoup plus précise dans les couches externes du détecteur. Il sera toujours plus facile de définir des fenêtres de sélection dans ces zones que dans les couches internes où la résolution sera moins bonne et les fenêtres moins précises. L'utilisation de modules plus épais dans les zones internes va tendre à atténuer ce phénomène, mais la résolution restera toujours meilleure dans les zones externes.

3.2.3/ LA CORRECTION DE PARALLAXE

La planéité des modules doit être prise en compte, plus particulièrement dans les couches internes où l'effet peut être relativement important (jusqu'à plusieurs strips). Si l'on se réfère à la Fig. 3.6, la distance effectivement mesurée par les chips de frontend est celle entre les points B et A'', alors que la largeur réelle du stub correspond à la distance entre B et A'. La mesure doit donc être corrigée de l'écart entre A' et A'', qui peut

se définir comme :

$$A'A'' = \epsilon' \tan \widehat{AOO_1}, \tag{3.2}$$

On peut facilement exprimer cette distance en unités de strips. On a en effet $N_{A'A''} = \frac{\epsilon'}{r_{O_1}} N_{AO_1}$. Une rapide application numérique permet de voir que cette valeur peut aller jusqu'à 5.5 strips pour un signal sur le bord des modules des couches les plus internes du détecteur. C'est loin d'être négligeable quand on sait que la taille des fenêtres de sélection dans cette zone sera au plus de 2 strips ! Cette correction est donc implémentée directement dans les chips de frontend (CBC/MPA). 8 corrections différentes par chip peuvent être pré-enregistrées (soit 1 valeur pour environ 15 strips), ce qui est largement suffisant pour conserver une résolution acceptable. Il est toutefois intéressant de noter qu'une valeur plus précise de la correction peut toujours être calculée avec précision au niveau du backend, afin d'améliorer la résolution sur la largeur du stub.

En pratique, il n'est possible de déterminer la valeur exacte de la largeur d'un stub et de sa correction de parallaxe que dans le cas où le module est parallèle à l'axe du faisceau (tonneau classique). Dans tous les autres cas, on peut montrer que le paramètre ϵ' dépend de l'angle entre la trace et l'axe du faisceau θ_0, comme représenté sur la Figure 3.7. Les autres paramètres sont δ, la distance entre les 2 couches de silicium, et α, l'angle entre le module et l'axe du faisceau. Dans la convention présentée ici, α vaut respectivement $\pi/2$ et 0 pour des modules du tonneau classique et des bouchons.

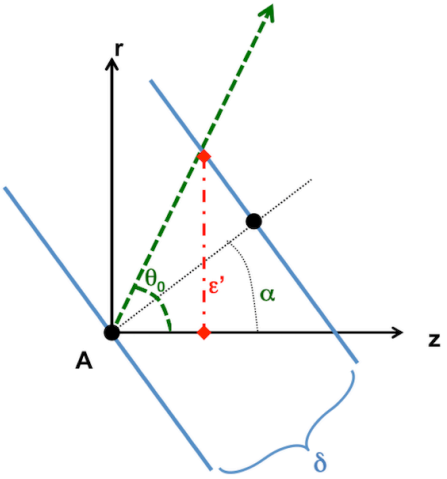


FIGURE 3.7 – Paramètres définissant le stub dans le cas d'un module incliné par rapport à l'axe du faisceau.

Avec cette nouvelle définition, on peut généraliser la relation entre p_T et largeur du stub :

$$p_T = 0.57 \cdot q \cdot R_A \sqrt{1 + \left(\frac{\sin\theta_0}{\cos(\theta_0 - \alpha)}\right)^2 \cdot \left(\frac{\delta}{pitch \cdot bend}\right)^2} \tag{3.3}$$

On peut également généraliser la correction de parallaxe :

$$N_{A'A''} = \frac{\delta}{r_{O_1}} \cdot \frac{\sin\theta_0}{\cos(\theta_0 - \alpha)} \cdot N_{AO_1} \tag{3.4}$$

Dans les 2 cas, on note l'apparition du terme θ_0 qui est bien sûr inconnu au moment de la création du stub. On vérifie bien que cette dépendance disparaît lorsque $\theta_0 = \pi/2$, c'est-à-dire pour un stub formé dans le tonneau classique. Pour les autres types de modules, la résolution sera donc forcément moins bonne. C'est bien ce que l'on observe sur les Figures 3.8 et 3.9, où sont comparées les résolutions des stubs issus de muons primaires dans le tonneau et les bouchons respectivement. On observe bien une résolution plus mauvaise dans les bouchons, due en partie au point évoqué plus haut.

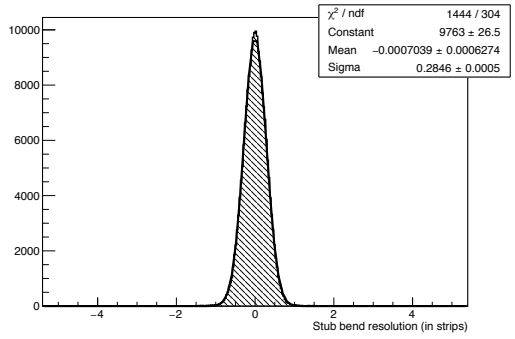


FIGURE 3.8 – Résolution des stubs provenant de muons primaires dans le tonneau

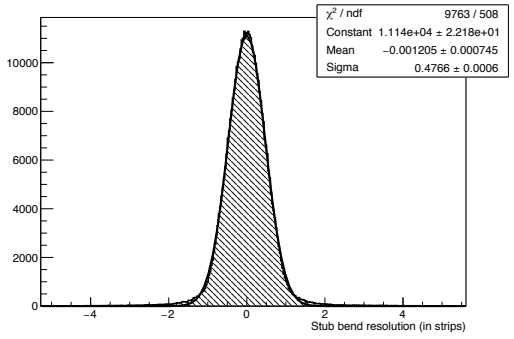


FIGURE 3.9 – Résolution des stubs provenant de muons primaires dans les bouchons

3.3/ SÉLECTION DES STUBS

3.3.1/ TYPOLOGIE

Qu'ils proviennent d'une particule secondaire ou d'une particule primaire, les stubs associés à une particule peuvent être classés en plusieurs catégories qui sont résumées sur la Figure 3.10.

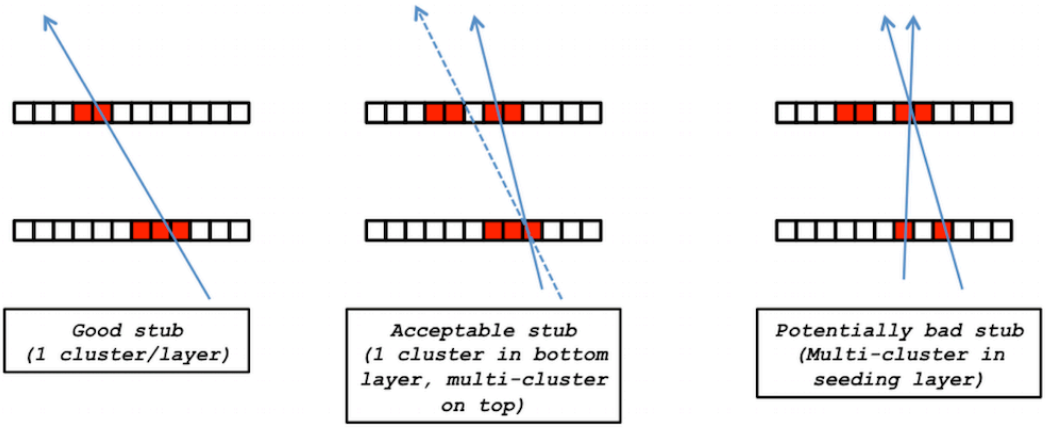


FIGURE 3.10 – Les différents types de stubs associés à une particule primaire

Pour bien comprendre les situations représentées sur cette figure, il faut se pencher sur la façon dont les stubs sont construits dans les CBC et les MPA. Dans les 2 cas les

ASICs se basent sur un senseur de référence situé au bas du module. Pour un module PS, par exemple, il s'agit toujours des macro-pixels. Les modules pouvant être placés de différentes façons dans le détecteur, le senseur de référence n'est donc pas toujours celui qui est le plus proche du point d'interaction.

Pour chaque cluster sur le senseur de référence, il n'est pas possible de produire plus d'un stub. Par contre, rien n'empêche un cluster de la couche supérieure de servir à former plusieurs stubs. D'autre part, dans le cas où plusieurs clusters sont trouvés dans la fenêtre de sélection de la couche supérieure, on construit le stub ayant la plus petite largeur (après correction de la parallaxe).

En partant de ce constat on peut définir trois types de stubs. Il est important de préciser que l'on parle ici de stubs associés à une particule primaire (au moins un des clusters est associé). Ces trois catégories sont :

— **Stub de bonne qualité** : exactement 1 cluster par senseur. La largeur de ce stub est bien corrélée avec l'impulsion de la particule (si celle-ci est primaire et en l'absence de brehmstrahlung). Les Figures 3.8 et 3.9 ont été obtenues avec des stubs rentrant dans cette catégorie.
— **Stub de qualité moyenne** : un seul cluster est induit dans le senseur de référence, mais il y a plusieurs clusters au dessus. Cela peut se produire si un strip est mort ou si le signal est au dessous du seuil d'un strip. Dans ce cas, seul un stub sera produit. Sa position sera correcte, mais la largeur sera potentiellement biaisée. Ce stub est de bonne qualité pour un système de reconstruction de traces utilisant seulement la position.
— **Stub de mauvaise qualité** : il y a plus d'un cluster dans le senseur de référence. Dans ce cas la position et la largeur sont potentiellement biaisées. Un tel stub, même si il est associé à une bonne particule, peut potentiellement biaiser n'importe quel algorithme de reconstruction.

.

Dans la pratique, tous ces stubs vont passer l'étape de sélection au niveau du frontend. La distinction présentée ici devient importante seulement dans la perspective de la reconstruction de traces. Il est en effet tentant d'utiliser la largeur des stubs pour sélectionner les traces candidates et réduire la proportion de fausses traces. Avant de mettre en place ce type de stratégie, il est important de comprendre l'impact du type de stub sur la résolution en largeur.

3.3.2/ SÉLECTION DES STUBS AU NIVEAU DU FRONTEND

Avant de s'intéresser plus précisément à l'utilisation de la largeur des stubs pour la reconstruction de traces, revenons à la raison d'être du stub : la réduction du flux de données à extraire du détecteur. Pour ce faire, la taille de la fenêtre de sélection va être d'abord contrainte par le taux de réduction à obtenir, l'efficacité venant au second plan. Pour définir ces fenêtres, on doit donc commencer par déterminer précisément la quantité de données maximale qui peut être extraite du détecteur.

Les nombres moyens de stubs, par module et par collision, qui peuvent être extraits au maximum du futur trajectographe sont résumés dans le tableau 3.1.

Ces valeurs sont calculées à partir des capacités d'extraction de données des futurs

Type de module	2S	PS (320MHz)	PS (640MHz)
Occupation moyenne maximale (stubs/module/BX)	4.8	4.4	8.8

TABLE 3.1 – Limites d'extraction du frontend, en nombre moyen de stubs par module et par collision.

modules. Nous reviendrons plus précisément sur ces chiffres dans le chapitre suivant, où nous verrons que ces taux permettent d'assurer des pertes relativement faibles dans l'ensemble du trajectographe, et ce jusqu'à des empilements moyens de plus de 250 collisions par événements.

A partir de là, il y a deux possibilités : soit le taux d'occupation estimé est supérieur à cette valeur, soit il est en dessous. Dans l'immense majorité des modules du futur trajectographe, pour un empilement moyen de 200, le taux d'occupation moyen sera largement inférieur. Dans ce cas, la fenêtre de sélection pourra être déterminée de manière à maximiser l'efficacité de reconstruction jusqu'au seuil de sélection minimal fixé à 2 GeV/c. Dans le cas contraire, on choisira une fenêtre qui permet de passer sous le seuil d'occupation maximal, au détriment de l'efficacité.

Evidemment, la taille des fenêtres va dépendre fortement du type de particules utilisé pour les calculer. Des fenêtres déterminées avec des électrons seront forcément plus larges que celles obtenues pour des muons. Mais des fenêtres plus larges signifient également plus de bruit de fond dû à des stubs secondaires, donc des taux d'occupation supérieurs. Enfin, les taux d'occupation dépendants de l'empilement moyen, les fenêtres dépendront également de ce paramètre.

Afin de prendre en compte tous ces facteurs de manière déterministe, une procédure semi-automatique de calcul des fenêtres de sélection a été développée [29]. Le trajectographe est découpé en 108 régions dans lesquelles les fenêtres sont identiques pour chacun des modules (Fig. 3.11).

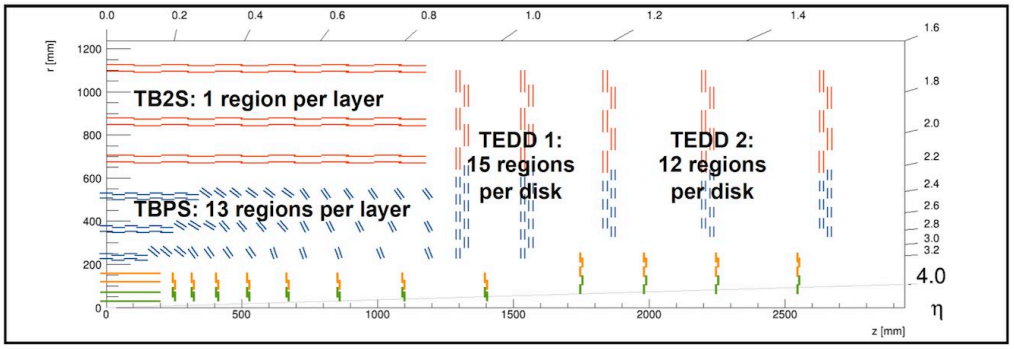


FIGURE 3.11 – Régions du trajectographe nécessitant une fenêtre de sélection distincte

Pour chacune de ces régions, et pour un empilement moyen donné, 2 largeurs de fenêtres sont calculées : une fenêtre stricte, déterminée avec des muons, et une fenêtre large, déterminée avec des électrons de haut p_T et des muons. A titre d'exemple, les fenêtres larges obtenues pour les différentes régions pour des empilements moyens de 140 et 200 événements sont données dans les Figures. 3.12 et 3.13. On y voit clairement que les valeurs restent inchangées dans les régions externes où l'occupation est faible, et où l'efficacité prime. Par contre, dans les zones internes, c'est le taux d'occupation moyen qui

dicte la taille de la fenêtre, et l'impact peut être important. On voit par exemple que dans le cercle le plus interne du disque 2 (TEDD2/R1), la fenêtre passe de 1.5 à 0.5 strips ! ! ! L'impact sur l'efficacité de cette zone, heureusement réduite, est loin d'être négligeable.

TB2S3	7,0												
TB2S2	6,5												
TB2S1	6,0												
TBPS3	4,5	5,0	5,0	5,0	5,0	5,0	5,0	5,5	5,0	5,0	5,5	5,5	5,5
TBPS2	3,0	4,0	4,0	4,0	4,0	4,0	4,0	4,5	5,0	5,0	5,0	5,0	5,0
TBPS1	2,5	3,0	3,0	2,5	3,0	3,0	2,5	2,5	2,5	2,0	2,0	2,0	2,0
	FLAT	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12

7,0	7,0	7,0	7,0	7,0	R15
7,0	7,0	7,0	7,0	7,0	R14
6,5	6,5	7,0	7,0	7,0	R13
6,5	6,5	7,0	7,0	7,0	R12
6,5	6,5	7,0	6,5	7,0	R11
6,5	6,5	6,5	6,5	7,0	R10
6,5	6,5	6,5	6,5	6,5	R9
6,5	6,5	6,5	6,5	6,5	R8
6,0	6,0	6,0	6,0	6,5	R7
5,5	6,0	6,0	6,0	5,0	R6
5,5	5,5	4,5	4,0	3,5	R5
5,5	5,0	3,0	2,5	2,0	R4
2,5	2,5				R3
2,5	2,5				R2
2,0	1,5				R1
TEDD1_1	TEDD1_2	TEDD2_1	TEDD2_2	TEDD2_3	

FIGURE 3.12 – Fenêtres larges pour un empilement moyen de 140 interactions par collision

TB2S3	7,0												
TB2S2	6,5												
TB2S1	6,0												
TBPS3	4,5	5,0	5,0	5,0	5,0	5,0	5,0	5,5	5,0	5,0	5,5	5,5	5,5
TBPS2	3,0	4,0	4,0	4,0	4,0	4,0	4,0	4,5	5,0	4,0	3,5	3,5	3,0
TBPS1	2,0	3,0	3,0	2,5	3,0	3,0	2,5	2,5	2,0	1,5	1,5	1,0	1,0
	FLAT	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12

7,0	7,0	7,0	7,0	7,0	R15
7,0	7,0	7,0	7,0	7,0	R14
6,5	6,5	7,0	7,0	7,0	R13
6,5	6,5	7,0	7,0	7,0	R12
6,5	6,5	7,0	6,5	7,0	R11
6,5	6,5	6,5	6,5	7,0	R10
6,5	6,5	6,5	6,5	6,5	R9
6,5	6,5	6,5	6,5	6,5	R8
6,0	6,0	6,0	6,0	4,5	R7
5,5	6,0	4,5	3,5	3,0	R6
5,5	5,0	3,0	2,5	1,5	R5
3,5	3,0	1,0	1,0	0,5	R4
2,5	2,5				R3
2,5	2,5				R2
1,0	0,5				R1
TEDD1_1	TEDD1_2	TEDD2_1	TEDD2_2	TEDD2_3	

FIGURE 3.13 – Fenêtres larges pour un empilement moyen de 200 interactions par collision

Pour évaluer cet impact, une estimation complète de la proportion de traces reconstructibles a été effectuée à partir de différents jeux de fenêtres. Une trace reconstructible est une trace qui génère suffisamment de stubs dans le trajectographe pour être identifiée par le track trigger, c'est à dire au moins 1 stub par couche sur 4 couches distinctes du trajectographe, et si possible 5. Au dénominateur on considère toutes les particules primaires produites dans l'acceptance du trajectographe ($|\eta| < 2.4$), induisant au moins 1 stub dans ce dernier. Les résultats obtenus sont résumés dans la Table 3.2.

De manière assez prévisible, on observe une baisse de l'efficacité pour un empilement de 200. Cette baisse reste néanmoins relativement faible, car le nombre de modules pour lesquels le taux d'occupation est important est relativement restreint.

Pour un empilement donné, on observe comme prévu une différence d'efficacité importante entre les jeux larges et stricts pour les électrons. La Table 3.3 compare la proportion d'électrons reconstructibles pour des fenêtres larges et strictes optimisées pour un empilement moyen de 140.

Cette différence est particulièrement importante à bas p_T où les pertes d'énergies sont les plus importantes.

	Fenêtre pour un empilement moyen de	200		140	
Type d'événement	Domaine en p_T (GeV)	$N_{hits} \geq 4$	$N_{hits} \geq 5$	$N_{hits} \geq 4$	$N_{hits} \geq 5$
Muon sans PU	[2,8]	99.85 ± 0.01	98.45 ± 0.01	99.87 ± 0.01	98.52 ± 0.01
	[8,50]	99.87 ± 0.01	98.77 ± 0.01	99.89 ± 0.01	98.78 ± 0.01
Electron sans PU	[2,8]	93.70 ± 0.01	89.09 ± 0.01	94.06 ± 0.01	89.66 ± 0.01
	[8,50]	98.52 ± 0.01	96.35 ± 0.01	98.58 ± 0.01	96.46 ± 0.01
Particules primaires, PU200	[2,8]	92.40 ± 0.01	86.72 ± 0.01		
	[8,50]	95.2 ± 0.1	92.5 ± 0.1		
Particules primaires, PU140	[2,8]			92.31 ± 0.01	88.04 ± 0.01
	[8,50]			97.0 ± 0.1	94.3 ± 0.1

TABLE 3.2 – Proportions de traces reconstructibles mesurées avec différents types de fenêtres de sélection.

	Fenêtres de sélection (en GeV)	Larges	Strictes
Electron seul	[2,8]	94.06 ± 0.01	89.82 ± 0.01
	[8,50]	98.58 ± 0.01	97.77 ± 0.01

TABLE 3.3 – Proportion d'électrons reconstruits, avec des fenêtres larges et strictes optimisées pour un empilement de 140.

Au final, le choix du type de fenêtre dépendra des particules requises par le système de reconstruction de traces. La méthode présentée ici permet de choisir entre différents cas de figure. Si l'on est surtout intéressé par les muons, on s'orientera vers les fenêtres strictes. Par contre, si il devient important de récupérer également les électrons, on pourra utiliser les fenêtres larges, le prix à payer sera une augmentation non négligeable du nombre de stubs en sortie (30%). Les fenêtres de sélection pourront de toute façon être modifiées à tout moment dans les CBC et les MPA.

3.3.3/ UTILISATION DE LA LARGEUR DES STUBS DANS LA RECONSTRUCTION DES TRACES

3.3.3.1/ RÉSOLUTION DE LA LARGEUR MESURÉE

En partant du principe que la correction de parallaxe peut être calculée de manière précise (ce qui est vrai pour tous les modules parallèles à l'axe du faisceau ($\alpha = \pi/2$)), la source principale d'erreur systématique sur la largeur est la position des 2 clusters du stub. Comme l'information fournie par les ASICs de frontend est purement binaire (un strip est on ou off), l'erreur digitale sur la position d'un strip est donnée par la relation classique $pitch/\sqrt{12}$[30]. On en déduit l'erreur attendue sur la largeur des stubs dans le cas où celui-ci est formé de deux clusters d'un seul strip : $pitch/\sqrt{6} \approx 0.41 \cdot pitch$. En pratique on observe un meilleur résultat (voir Figure 3.8 montrant la résolution obtenue pour des bons stubs provenant de muons, dans le cas de modules du tonneau ($\alpha = \pi/2$)), en particulier dans le tonneau. Ceci est dû à la plus forte proportion de clusters à 2 strips dans cette zone.

A l'inverse, dans les bouchons, les clusters d'un seul strip sont majoritaires, et la correction de parallaxe ne peut être calculée de manière exacte. On retrouve alors une valeur plus proche de la résolution digitale. La résolution observée est même un peu moins bonne, comme le montre la Figure 3.9, montrant la résolution obtenue pour des bons stubs provenant de muons des bouchons ($\alpha = 0$).

L'analyse de ces figures montre que, pour des stubs de bonne qualité, on peut estimer qu'une résolution de 0.5 strips est une approximation raisonnable, ce qui signifie que dans 99% des cas, pour cette catégorie de stubs, la différence entre valeur attendue et mesurée sera inférieure à 1.5 strips. Nous verrons dans une section ultérieure comment cette information peut être utilisée.

Cependant, ce constat ne vaut que pour les stubs de bonne qualité issus de particules dont la trajectoire est bien circulaire. Pour les stubs imparfaits ou provenant d'électrons, la résolution peut-être beaucoup moins bonne, comme on peut le voir sur la Fig. 3.14, obtenue avec des stubs de mauvaise qualité induits par des muons, ou sur la Fig. 3.15, obtenue avec des stubs de bonne qualité provenant d'électrons. Il faudra donc utiliser la résolution des stubs avec précaution.

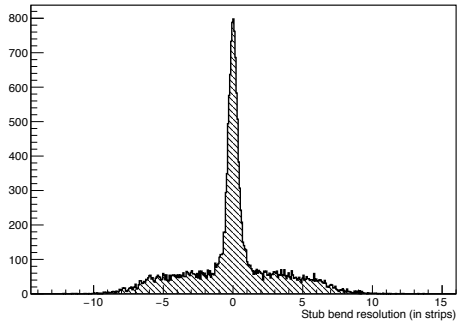


FIGURE 3.14 – Résolution des stubs de mauvaise qualité provenant de muons primaires dans le tonneau

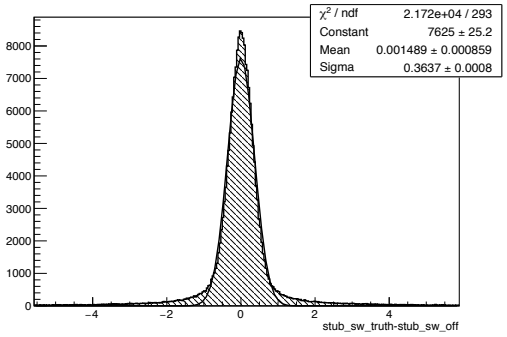


FIGURE 3.15 – Résolution des stubs de bonne qualité provenant d'électrons primaires dans le tonneau

3.3.3.2/ SÉLECTION DES STUBS PAR LA LARGEUR

Les figures 3.16 and 3.17 illustrent de manière très claire l'influence du type de stub sur la résolution. Les 2 figures ont été réalisées avec des stubs associés à des muons primaires, dans la couche extérieure du tonneau, là où le bras de levier est le plus important. On y voit clairement que la corrélation en impulsion est bien meilleure dans le cas des stubs de bonne qualité.

Dans l'étude que nous avons réalisée, la proportion de stubs de mauvaise qualité induits par une particule primaire normale peut aller jusqu'à 5% dans certaines zones. C'est loin d'être négligeable, surtout pour des stubs qui sont destinés au système de reconstruction de traces. Encore une fois, tous ces stubs ne sont pas totalement inutilisables, en particulier pour un algorithme n'utilisant que la position.

Pour illustrer plus précisément les risques que comporte l'utilisation de la largeur des stubs, la proportion de traces reconstructibles avec ou sans coupure sur la largeur a été évaluée.

La coupure sur la largeur utilisée est la suivante : le stub est rejeté si la différence entre la valeur attendue et la valeur mesurée est supérieure à 1.5 strips. Ce type de coupure très simple est exactement le genre de sélection qui pourrait être appliqué de manière efficace au niveau de la reconnaissance de traces, le p_T attendu de la graine pouvant être rapidement déterminé.

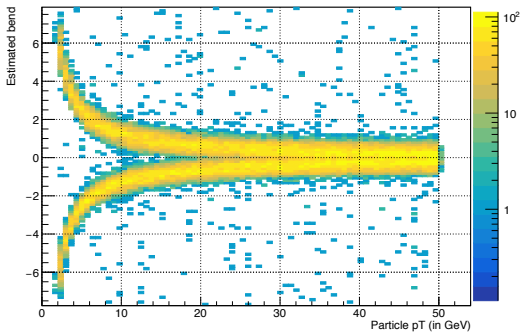


FIGURE 3.16 – Largeur de stubs de bonne qualité induits par des muons primaires en fonction de l'impulsion transverse du muon.

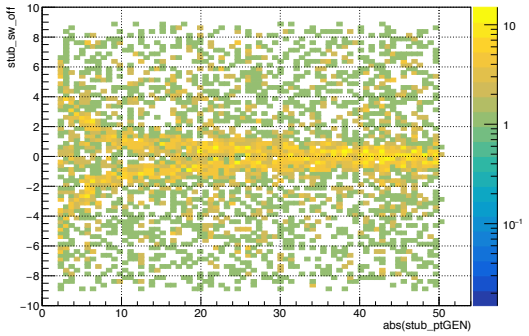


FIGURE 3.17 – Largeur de stubs de mauvaise qualité induits par des muons primaires en fonction de l'impulsion transverse du muon.

Dans la table 3.4 l'impact d'une sélection des stubs sur la proportion des traces reconstructibles est mesuré pour des muons et des électrons, sans aucun empilement (l'effet de l'empilement est de toute façon négligeable ici).

	sélection sur la largeur	Non		Oui	
Type de particule	Domaine en p_T (GeV)	$N_{hits} \geq 4$	$N_{hits} \geq 5$	$N_{hits} \geq 4$	$N_{hits} \geq 5$
Muon	[2,8]	99.87 ± 0.01	98.52 ± 0.01	99.19 ± 0.01	95.09 ± 0.01
	[8,50]	99.89 ± 0.01	98.78 ± 0.01	99.88 ± 0.01	98.62 ± 0.01
Electron	[2,8]	94.06 ± 0.01	89.66 ± 0.01	81.50 ± 0.01	72.70 ± 0.01
	[8,50]	98.58 ± 0.01	96.46 ± 0.01	94.38 ± 0.01	90.64 ± 0.01

TABLE 3.4 – Proportion de traces reconstructibles avec ou sans coupures sur la largeur des stubs.

Comme on pouvait s'y attendre, c'est dans les cas où le stub est le plus affecté (électron, bas p_T) que l'impact d'une coupure est le plus important. Par contre, pour des muons de haute impulsion transverse, la perte est faible, la coupure peut donc être très efficace dans ce cas.

Il est cependant important de bien comprendre que ce type de sélection peut avoir un impact notable sur la performance de l'algorithme, et surtout un impact variable dans le temps car la proportion de stubs de mauvaise qualité est directement liée à la dégradation du détecteur. Les strips inactifs, en particulier, pourraient jouer un rôle important à long terme.

En conclusion, l'utilisation de la largeur des stubs pour effectuer des sélections fines doit être faite avec discernement. Il ne faut pas oublier qu'initialement les stubs n'ont pas été introduits pour çà. Pour la petite histoire, lorsque nous avons commencé à définir les formats de données en sortie du concentrateur (voir le chapitre suivant), il n'était même pas prévu de transmettre la largeur des stubs au système de reconstruction de traces !

4

DÉVELOPPEMENT ET RÉALISATION DU CONCENTRATEUR

Ce chapitre est consacré au concentrateur, projet dont j'ai assuré la responsabilité scientifique à partir de 2013. Le rôle de cet élément de la chaîne d'acquisition du futur trajectographe est présenté de manière approfondie, ainsi que le long chemin menant du cahier des charges au composant physique, dont une première version a été testée avec succès en 2019.

4.1/ LE RÔLE DU CONCENTRATEUR

4.1.1/ POURQUOI UN ASIC ?

Le rôle du concentrateur a été évoqué rapidement dans le chapitre précédent. C'est en partie grâce à ce composant, qui regroupe et met en forme les données produites par les MPA et les CBC, que les stubs peuvent être distribués au système de reconstruction de traces rapides.

Le concentrateur est un ASIC (Application-Specific Integrated Circuit). Contrairement aux processeurs qui équipent nos ordinateurs et autres smartphones, les ASICs sont spécialement mis au point pour effectuer une tâche particulière.

Réaliser ces composants requiert une expertise technique importante, et on peut a priori se demander la raison pour laquelle nous en avons besoin, alors que les outils disponibles sur le marché, et en particulier les FPGAs, permettent d'effectuer énormément d'opérations très élaborées. En fait, si l'on se penche sur les contraintes que devront respecter les composants installés sur les modules, on s'aperçoit rapidement qu'un ASIC est la seule alternative.

Le premier argument en faveur d'un ASIC est la consommation. Le budget disponible pour le concentrateur de données est de quelques centaines de milliwatts seulement. On est très loin des quelques dizaines de watts d'un FPGA haut de gamme classique.

L'autre contrainte majeure est la résistance aux radiations. Le trajectographe étant proche du faisceau, il est soumis à des doses très importantes, et est traversé en permanence

par une multitude de particules chargées qui peuvent venir fausser le signal parcourant les modules. En inversant la valeur d'un bit, ce type d'événement (on les nomme SEU pour Single Event Upset) peut avoir un effet dévastateur sur la prise de données. Le composant doit donc être capable de résister à ce type de phénomène. Cette contrainte seule suffit à éliminer une grande majorité des FPGA.

Quand bien même ces deux arguments seraient insuffisants, on pourrait également citer les contraintes liées à la latence du traitement de l'information, à l'encombrement du composant, etc. Autant de raisons pour lesquelles il est nécessaire de développer un ASIC.

4.1.2/ Principe de fonctionnement

Tout au long de ce chapitre nous utiliserons plusieurs appellations pour le CIC : CIC1, CIC2, ou simplement CIC. Le CIC1 et le CIC2 sont les 2 versions physiques du CIC réalisées à ce jour. Elles effectuent le même traitement de données, et ne diffèrent que par certains aspects techniques sur lesquels nous reviendrons par la suite. En ce qui concerne le comportement général de l'ASIC, CIC1 et CIC2 sont équivalents.

Le principe de fonctionnement très simplifié du CIC est schématisé sur la Figure 4.1. Comme évoqué lors du chapitre précédent, le circuit reçoit les stubs, mais aussi des données brutes, fournies par 8 CBCs ou MPAs. Il reformate ces données, les traite, et les groupe en paquets avant de les envoyer vers le lpGBT.

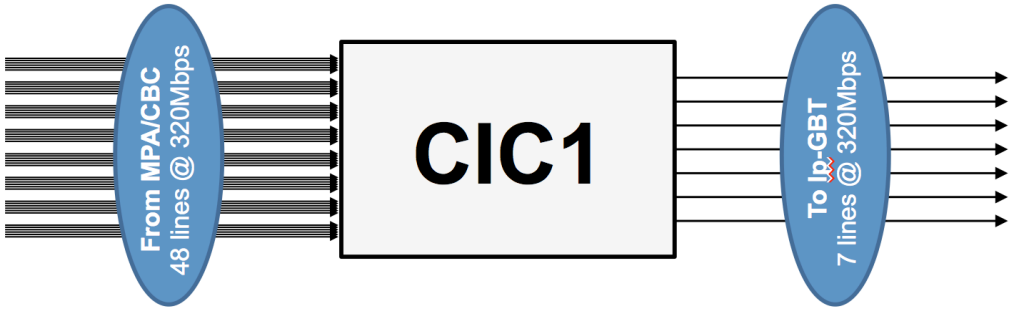


Figure 4.1 – Principe de fonctionnement du CIC

Le CIC doit donc être compatible avec 2 types de modules différents. C'est ce qui fait sa complexité. Un ASIC est par définition spécifique, lui demander d'être flexible est forcément contre-nature...

Dans le cas présent, la raison pour laquelle il n'y a pas un concentrateur différent sur chacun des modules est historique. A l'origine, il ne devait pas y avoir de CIC sur les modules PS.

Une telle modification du cahier des charges n'aurait pas dû être possible si le projet avait été clairement défini dès le départ. En effet, le processus normal de développement d'un ASIC se résume aux étapes suivantes :

— 1. Etablissement d'un cahier des charges détaillé et précis
— 2. Développement d'un modèle numérique et simulation comportementale
— 3. Design physique, validation, et simulation du design routé

Le respect de cette démarche est excessivement important. En effet, une fois que l'ASIC est produit, il n'y a aucun moyen de le modifier. Il est donc primordial de simuler dans le détail son fonctionnement avant le départ en fonderie. De la même manière, le cahier des charges doit être défini de la manière la plus précise possible dès le début du projet. Une modification simple sur le papier peut s'avérer impossible à mettre en place sur un design trop avancé.

Si l'on ne respecte pas les règles du jeu, on s'expose forcément à des retards et à d'autres situations complexes à gérer, le CIC en est un bon exemple. Les parties suivantes, qui décrivent le projet en détails, ne suivent donc pas forcément un ordre chronologique, mais elles reflètent l'ordre dans lequel je mènerais à bien ce projet si c'était à refaire.

4.2/ CAHIER DES CHARGES

4.2.1/ CONTEXTE

La définition du cahier des charges est une étape fondamentale. Dès que celui-ci est complet, une équipe de microélectroniciens expérimentés peut réaliser une première version d'un ASIC en 18 mois environ. A l'inverse, si le développement de l'ASIC débute avant la finalisation de cette étape, la situation peut rapidement s'enliser.

Le cahier des charges est défini d'abord et avant tout par les utilisateurs du composant, c'est-à-dire les physiciens. A minima, ceux-ci doivent fournir aux ingénieurs le format des données en entrée et en sortie du chip, mais aussi la vitesse à laquelle ces données vont arriver et repartir. Avec ces informations, il est possible d'évaluer la faisabilité technique du projet et de converger, moyennant quelques itérations, vers une solution réalisable. A partir de là, le rôle du physicien se cantonne normalement au suivi de l'avancement du projet.

Le cahier des charges est également défini par les spécifications techniques du module hôte. Son facteur de forme et ses connexions vont être fortement contraints par l'hybride auquel il sera connecté. Sa tension d'alimentation et sa consommation devront également suivre des règles très précises. La technologie utilisée, et en particulier la finesse de gravure, sont également une contrainte très importante pour le projet. Ces éléments, si il n'empêchent pas le développement du modèle du CIC, doivent être connus le plus tôt possible afin de pouvoir aborder le design physique du chip dans des bonnes conditions.

Tout ces points sont des évidences si vous avez déjà réalisé des ASICs. Dans le cas contraire, ce n'est pas aussi simple. C'est sans aucun doute ce manque d'expérience qui a handicapé ce projet lorsque celui-ci a démarré à la fin de l'année 2011. Le CIC a été introduit lorsqu'il est devenu clair que les données provenant des CBCs devaient être compressées pour pouvoir sortir des modules 2S. Le cahier des charges se résumait à peu prêt à ce constat, autant dire qu'il était flou...

Petit à petit, il s'est précisé. L'inclusion du CIC dans le module PS, par exemple, remonte à la fin de l'année 2013, soit presque 2 ans après le début du projet. La définition des formats de données et des vitesses de sortie entre 2014 et 2015 a permis de vraiment démarrer le développement du modèle, mais il a fallu attendre 2017 pour véritablement débuter le design physique de l'ASIC. Le cahier des charges tel qu'il est présenté dans ce document n'a été finalisé qu'au début de l'année 2018. Rien d'étonnant, dans ce contexte

à ce que la première version du CIC n'ait été disponible qu'en 2019, alors qu'elle était initialement prévue en 2015.

Le message à retenir est donc le suivant : sans format de données, point de modèle, et sans cahier des charges complet, point d'ASIC. Avec le recul, je suis convaincu que la connaissance des formats de données est la condition minimale pour accepter de prendre en charge ce type de projet.

4.2.2/ Fonctionnement du CIC

Les entrées/sorties d'un ASIC de traitement de données sont les premières informations à définir. Ces informations, dans le cas du CIC, sont résumées sur la Figure 4.2. Elles sont de 2 types : les informations liées aux données en transit dans le chip, et celles liées au contrôle de ce-dernier.

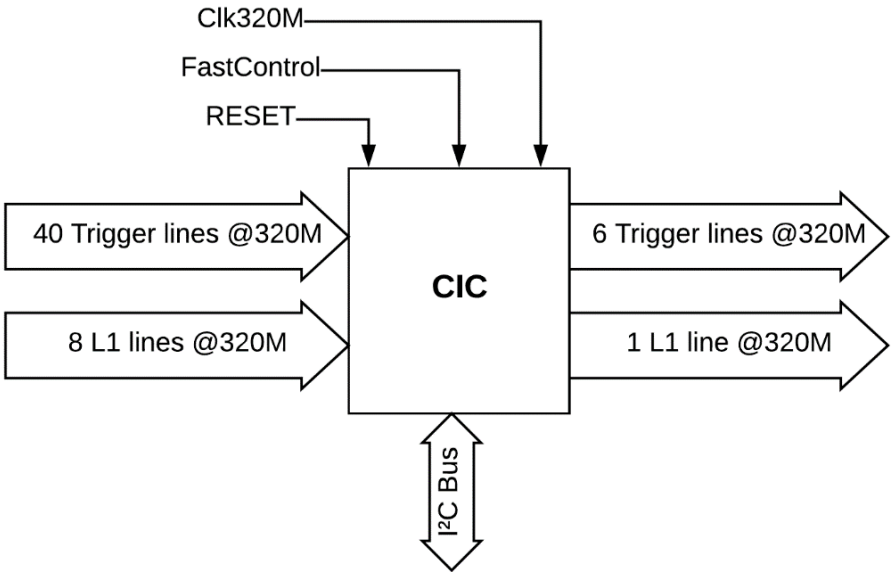


FIGURE 4.2 – Flux de données entrant et sortant du CIC

Comme nous l'avons déjà dit plusieurs fois, CBCs et MPAs fournissent deux flux de données indépendants :

— **Voie trigger** : données produites de manière synchrone à la fréquence de 40MHz. Il s'agit des stubs. Ceux-ci doivent être transmis le plus rapidement possible au système de reconstruction de traces situé hors du détecteur. Sur cette voie, on peut perdre des données (nous verrons lesquelles), mais on ne peut pas perdre de temps.
— **Voie données** : données brutes du détecteur, qui sont extraites pour chaque événement passant le premier niveau du système de déclenchement. Ces informations, par définition asynchrone, sont transmises à une fréquence moyenne pouvant aller jusqu'à 750kHz. Sur cette voie, on peut perdre du temps, mais on ne doit pas perdre de données.

Toutes ces informations arrivent au CIC via des lignes différentielles cadencées à 320MHz. Chaque CBC/MPA possède 6 lignes en sortie : 5 pour la voie trigger et 1 pour la voie données. En tout, chaque CIC comporte donc 48 entrées : 40 pour la voie trigger,

et 8 pour la voie données. Cette répartition est fixe et définitive.

En sortie, le cahier des charges initial prévoyait 6 lignes à 320MHz (5 trigger et 1 données), mais une étude détaillée des pertes dynamiques du CIC (voir section suivante) a poussé à modifier ce paramètre pour les modules PS. Pour ces derniers, le CIC comportera 7 lignes en sortie (6 trigger et 1 données) pouvant fonctionner non seulement à 320MHz, mais aussi à 640MHz pour certains modules PS très proches du point d'interaction.

Le pilotage de l'ASIC est effectué par les systèmes de contrôle suivants :

- **Clock** : l'horloge est ce qui permet au CIC de se synchroniser avec le LHC, et de générer l'ensemble des horloges internes de l'ASIC. Dans les 2 cas, cette horloge est fournie par le lpGBT. Elle est cadencée à la fréquence maximale nécessaire au CIC, soit 320MHz dans les modules 2S et 640MHz dans les modules PS.
- **Fast control** : les signaux de contrôle rapide correspondent à une trame de 8 bits envoyée de manière synchrone au CIC à une fréquence de 40MHz. La combinaison de 8 bits reçue fournit au CIC des informations sur les actions à mener immédiatement : resynchronisation, reception d'un L1A,... Ces signaux de contrôle sont envoyés au cours de la prise de données.
- **Slow control** : le contrôle lent, au contraire, correspond à l'ensemble des signaux permettant d'initialiser l'ASIC ou de modifier son mode de fonctionnement entre 2 prises de données. Celui-ci n'est en effet pas complètement contraint par construction. Un ensemble de paramètres peut être modifié via le bloc de slow control. On peut par exemple utiliser le slow control pour choisir si le CIC doit fonctionner en mode PS ou en mode 2S. Chaque paramètre est contenu dans un registre accessible en lecture et en écriture. Certains registres ne sont accessible qu'en lecture (comme par exemple les compteurs d'erreurs).
- **Reset** : cette ligne permet de lancer un redémarrage complet de l'ASIC, à n'importe quel moment. On parle alors de reset dur, par opposition à la resynchronisation qui est pilotée par le fast control et au cours de laquelle seul certains paramètres sont effacés (les paramètres de configuration, par exemple, sont conservés).

4.2.3/ DEFINITION DES FORMATS D'ENTRÉES/SORTIE ET ÉVALUATION DES PERTES DYNAMIQUES

Le dimensionnement des entrées et des sorties est étroitement lié aux flux de données qui traversent le concentrateur. Ces flux dépendent du taux d'occupation du détecteur, mais aussi de la taille des trames de données, et donc de leur format. Comme expliqué précédemment, ce travail est effectué en amont par le physicien. Ce fut donc logiquement ma première contribution lorsque j'ai repris la responsabilité du projet fin 2013.

Les informations décrites dans cette partie sont extraites du document résumant les spécifications de CIC1 [31] (Annexe C.1) , ainsi que d'une série de notes consacrées à l'étude des pertes dynamiques dans le concentrateur [32] dont la version la plus récente est reproduite dans l'Annexe C.2.

4.2.3.1/ FORMATS D'ENTRÉE

Ce qui rentre dans le CIC correspond à ce qui sort du MPA ou du CBC. Comme le CIC a été développé plus tardivement que ces ASICs, le format des données entrantes était déjà largement figé en 2013. C'est à la fois une bonne et une mauvaise chose. Bonne car on dispose tout de suite des informations dont on a besoin, mauvaise car on sait qu'il sera très difficile de faire modifier ces formats (on ne change pas le cahier des charges). Dans une chaîne de frontend contenant plusieurs ASICs, il est important que ces composants soient développés en étroite collaboration. Si un ASIC est trop en avance, comme le CBC, il va imposer son fonctionnement aux autres. Si un ASIC est trop en retard, comme le CIC, il va servir de poubelle aux autres. La chaîne doit être pensée dans sa globalité dès le départ.

Concernant les modules 2S, et donc le CBC, les données de la voie trigger (stubs) sont formatées sur un bloc de 40 bits qui contient l'ensemble des stubs reconstruits par le CBC lors d'une collision (BX). Chaque bloc peut contenir au maximum 3 stubs, le nombre maximum de stubs pouvant parvenir à 1 CIC pour un BX donné est donc 24. Pour la voie données, les informations brutes sont transmises. Cela signifie que le CIC reçoit de chaque CBC une trame de taille fixe contenant le signal de l'ensemble des 254 strips. Même les canaux sans signal sont transmis. On parle alors de données non-sparsifiées, ou sans suppression de zéros.

Pour les modules PS, le MPA formate les stubs sur un paquet de 80 bits contenant les informations de 2 collisions. Ce regroupement permet de transmettre jusqu'à 5 stubs par MPA pour 2 BXs, et permet ainsi d'absorber un événement pour lequel le nombre de stubs serait plus important. On peut dire que le MPA joue le rôle de pré-concentrateur, sur 2BX au lieu de 8. Quant aux données brutes, elles sont transmises au CIC sparsifiées. Cela signifie que seuls les strips activés sont transmis, sous forme d'amas (clusters). Il peut en effet y avoir plusieurs strips consécutifs activés. Le fait de transmettre des données sparsifiées permet de réduire de manière significative, dans l'immense majorité des cas, la taille de la trame à transmettre. Le seul inconvénient de ce procédé est que la taille de la trame envoyée par le MPA n'est pas fixe et dépend du nombre de clusters. Cela nécessite un traitement particulier au niveau du CIC qui doit regrouper des informations qui lui parviennent de manière totalement asynchrone.

4.2.3.2/ FORMATS DE SORTIE

Pour la voie trigger, le CIC regroupe les informations en temps et en espace. Un paquet de sortie contenant les stubs de 8CBC/MPA enregistrés pendant 8 collisions est préparé et envoyé toutes les 200 ns vers le lpGBT. La longueur de ce bloc est fixe et dépend de la fréquence de sortie des données : 384 bits à 320MHz et 768 bits à 640MHz. Si le nombre de stubs à transmettre est inférieur à la capacité de stockage, la trame est complétée par des 0 (padding bits). Dans le cas contraire, on sélectionne en priorité les stubs de largeur faible (nous reviendrons plus tard sur la manière dont cette sélection est opérée).

Pour la voie données, dans le cas 2S, le CIC forme les clusters à partir des trames issues de chacun des 8 CBCs, et les regroupe dans une trame unique contenant tous les clusters du demi module pour l'événement correspondant. Pour les modules PS les clusters étant déjà formés, le CIC effectue seulement le regroupement, mais la récupération des trames d'entrée, qui sont de tailles variables, est plus complexe.

Dans les 2 cas, le format des trames de sortie est le même pour les 2 types de modules (modulo l'encodage des clusters et des stubs, la granularité des modules étant différente). Cette uniformisation est importante, car elle simplifie le traitement des données à l'échelon suivant. Pour la voie trigger, où la latence est critique, c'est essentiel.

4.2.3.3/ ETUDE DES PERTES DYNAMIQUES

La voie trigger comporte 40 entrées à 320MHz pour 6 sorties à 320/640MHz. Le taux de compression à atteindre est donc au minimum 6.7 ou 3.35. Pour la voie données, le taux de compression est de 8 ou 4. Il faut donc s'assurer que l'occupation moyenne des détecteurs sera suffisamment faible pour atteindre ce facteur de réduction partout.

C'est particulièrement vrai pour la voie trigger où aucune sparsification n'est effectuée. Le facteur de réduction va donc ici uniquement dépendre du taux d'occupation moyen dans les modules. Si celui-ci est trop élevé, le CIC recevra un nombre trop important de stubs et il ne pourra pas tous les transmettre. Il est donc important, afin de dimensionner les sorties du CIC, d'évaluer les pertes potentielles, et de mettre en oeuvre, le cas échéant, les contre-mesures nécessaires. Le dimensionnement des fenêtres de sélection présenté au chapitre précédent est alors une des possibilités.

Cette évaluation a été effectuée avec des échantillons d'événements simulés dans les conditions attendues au HL-LHC, c'est-à-dire avec un empilement moyen pouvant aller jusqu'à 200, voire 250 si l'on prend en compte un facteur de sécurité. Pour évaluer au mieux les pertes attendues dans le CIC d'un module donné, des séquences de plusieurs milliers d'événements doivent être générées. Le framework de CMS n'étant pas adapté à ce type d'étude, un programme indépendant a été développé.

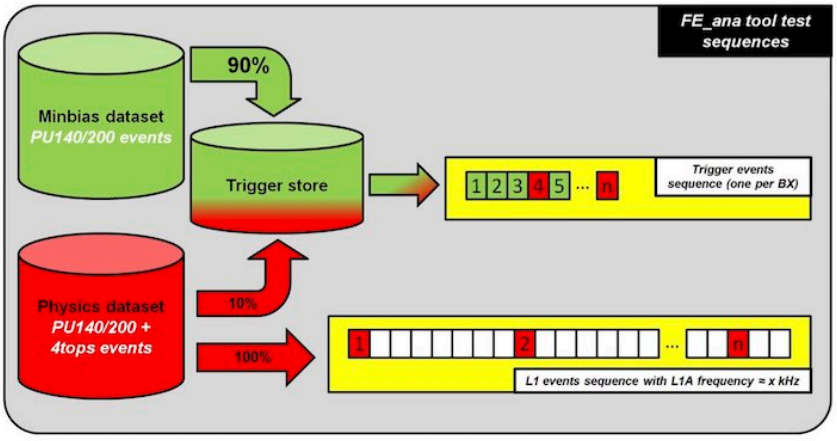


FIGURE 4.3 – Principe de fonctionnement de l'outil d'analyse des pertes dynamiques

Le principe de fonctionnement du programme mis en place pour étudier les pertes (nommé FE_ana) est décrit sur la Figure 4.3. Deux paquets de données sont créés à partir d'événements simulés dans le trajectographe : un échantillon contenant seulement des événement de biais minimum (le tout venant), et un autre avec des événements susceptibles de passer le niveau 1 du trigger (qui ont une occupation supérieure de 10/20% en moyenne au tout venant). Pour chacun de ces paquets, on stocke l'ensemble des informations relative aux ASICs de frontend : stubs, strips activés,... Ces paquets contiennent

quelques milliers d'événements seulement, générer de grands échantillons de données avec un empilement important est en effet très long, et peu utile pour ce type d'étude.

A partir de ces paquets, on crée des séquences de données type pour tester les deux voies du CIC. Pour la voie trigger, on mélange des informations des 2 paquets dans une proportion prédéfinie, alors que pour la voie données on ne sélectionne que des événements du paquet approprié. Concernant la taille de la séquence, on génère environ 10000 BX pour la voie trigger (1 événement pour chaque BX), et 100000 BXs pour la voie données (événements L1 répartis aléatoirement de manière à obtenir une fréquence finale d'environ 750kHz). Une fois la séquence créée, on peut calculer les pertes de manière relativement simple.

C'est particulièrement vrai pour la voie trigger. En effet, selon le format de sortie des données, la taille des blocs de données est prédéterminée. Par exemple pour 5 sorties à 320MHz, le bloc de sortie a une taille de 320 bits, ce qui permet de transmettre au maximum 16 stubs pour un module 2S, et 13 pour un module PS. Il suffit donc d'évaluer si cette taille est suffisante pour le groupe de 8 collisions dans la zone étudiée.

On peut ensuite établir, à partir des résultats obtenus dans chacune des zones définies dans le chapitre précédent, une cartographie complète des pertes dans le trajectographe. La Figure 4.4 présente la carte des pertes mesurées pour un empilement moyen de 250, en utilisant une vitesse de sortie à 320MHz pour l'ensemble du trajectographe et 5 lignes en sortie.

R15	0,00	0,00	0,00	0,00	0,00				0,00	0,00	0,00	0,00	0,00	R15
R14	0,00	0,00	0,00	0,00	0,00	L6	0,00	L6	0,00	0,00	0,00	0,00	0,00	R14
R13	0,00	0,00	0,00	0,00	0,00				0,00	0,00	0,00	0,00	0,00	R13
R12	0,00	0,00	0,00	0,00	0,00	L5	0,01	L5	0,00	0,00	0,00	0,00	0,00	R12
R11	0,00	0,00	0,00	0,00	0,00				0,00	0,00	0,00	0,00	0,00	R11
R10	0,02	0,03	0,02	0,01	0,01	L4	0,08	L4	0,01	0,01	0,02	0,03	0,02	R10
R9	0,06	0,02	0,02	0,02	0,02				0,02	0,02	0,02	0,02	0,06	R9
R8	0,09	0,09	0,05	0,05	0,03	L3	0,13	L3	0,03	0,05	0,05	0,09	0,09	R8
R7	0,37	0,09	0,10	0,04	0,03				0,03	0,04	0,10	0,09	0,37	R7
R6	0,34	0,59	0,17	0,18	0,07	L2	1,59	L2	0,07	0,18	0,17	0,59	0,34	R6
R5		1,12	1,88	0,81	0,83				0,83	0,81	1,88	1,12		R5
R4		6,15	8,40	4,85	2,33	L1	22,43	L1	2,33	4,85	8,40	6,15		R4
R3			12,59	15,21	9,44				9,44	15,21	12,59			R3
R2				22,23	16,13				16,13	22,23				R2
R1				37,93	35,23				35,23	37,93				R1
	D5-	D4-	D3-	D2-	D1-	BARREL			D1+	D2+	D3+	D4+	D5+	

FIGURE 4.4 – Proportion de stubs de bonne qualité perdus au niveau du concentrateur (en %), pour un empilement moyen de 250 interactions par événement, avec une vitesse de sortie de 320MHz.

On évalue les pertes à partir des stubs effectivement issus d'une particule primaire d'impulsion transverse supérieure à $2\ GeV/c$. On peut en effet tout à fait tolérer des pertes dans les autres catégories de stubs. Pour évaluer ces chiffres on utilise des fenêtres de sélection de stubs strictes pour un empilement moyen de 200, en cherchant à conserver une efficacité maximale partout.

On fixe à 1% le seuil en dessous duquel on considère que les pertes restent acceptables. On voit clairement que dans cette configuration, qui était à l'origine la configuration de base pour l'ensemble du détecteur, ce seuil est franchi dans les zones les plus internes. Ces résultats ont démontré qu'il était nécessaire de passer à 640MHz en sortie dans les 4 premiers anneaux des disques et dans la première couche du tonneau. Ce sont des changements importants pour le CIC, mais également pour les autres éléments du

détecteur comme par exemple les hybrides. On voit bien qu'il est capital de réaliser ce type d'étude le plus rapidement possible.

Dans les zones où le seuil est franchi mais les pertes en deçà de 5%, le taux de transmission reste à 320MHz, mais une ligne de sortie supplémentaire a été ajoutée, ainsi qu'un protocole de transmission plus simple au niveau du lpGBT (correction d'erreurs plus légère), libérant ainsi de l'espace pour la trame du concentrateur. En cas de problème persistant, la taille de la fenêtre de sélection des stubs fournit un dernier bras de levier. Mais, comme nous l'avons déjà dit auparavant, on utilise cette coupure en dernier recours, car elle a un impact direct sur l'efficacité de reconstruction.

Concernant la voie L1, compte-tenu de la taille variable des trames et de l'asynchronicité intrinsèque des paquets de données, les pertes sont un peu plus complexes à évaluer. Pour les mesurer, on évalue le temps d'extraction d'une trame L1. Ce temps est défini par la différence entre le BX où le dernier bit sort du concentrateur et celui où le premier bit entre dans le concentrateur. En conditions normales de fonctionnement, la valeur moyenne du temps d'extraction doit être constante au cours du temps. Celle-ci peut toutefois présenter des variations importantes (une trame vide prendra bien moins de temps qu'une trame contenant un nombre important de clusters). Par contre, si l'occupation du détecteur est trop importante, la taille moyenne des trames augmente et celles-ci prennent de plus en plus de temps à extraire. Dans ce cas de figure, le risque d'un remplissage du concentrateur augmente fortement. Lorsque la capacité de stockage est épuisée, on est en situation d'overflow et commence alors à perdre des données. Si cet overflow est lié à un taux d'occupation trop important, la situation est irréversible et le concentrateur doit être resynchronisé, jusqu'à l'overflow suivant. L'étude du temps moyen d'extraction est donc très importante car elle permet de prévenir ce type de problème, et fournit des indications précieuses sur le dimensionnement des vitesses des données en sortie et sur la taille de l'espace de stockage nécessaire dans le concentrateur.

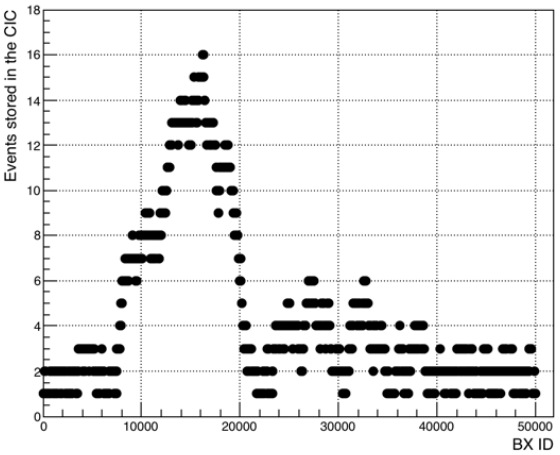


FIGURE 4.5 – Exemple d'evolution de la profondeur de la FIFO du CIC dans une zone à haute occupation.

La Figure 4.5 montre l'évolution du nombre d'événements stockés dans un CIC, obtenue avec cette simulation, dans le cas d'un module interne où la taille des événements est importante. On voit qu'il peut y avoir jusqu'à 16 événements en parallèle, mais que la vitesse de sortie est suffisamment importante pour que la FIFO se vide. Dans d'autres cas, on voit le nombre d'événements augmenter indéfiniment. C'est pour traiter ces cas

que le doublement de la vitesse en sortie dans cette voie a également été nécessaire pour certains modules.

Au final, une profondeur de 16 a été retenue pour la FIFO du CIC. Le cas observé sur la Figure 4.5 est très rare et ne concerne qu'un nombre très restreint de modules pour lesquels la vitesse de sortie n'a pas été doublée. On s'attend donc à observer quelques situations de ce type, mais dans une proportion très faible.

En conclusion, l'étude des pertes dynamiques liées au CIC a conduit à des modifications en profondeur du CIC, mais également du module de détection PS. Cela confirme que la définition du format des données et l'étude des pertes potentielles sont des éléments essentiels pour le cahier des charges du concentrateur, et qu'ils doivent être réalisés dès le début du projet.

4.2.4/ Contraintes techniques et cahier des charges

Les autres éléments du cahier des charges sont liés au dimensionnement physique du composant. Certaines de ces contraintes sont liées à l'hybride (pinout, tension d'alimentation, facteur de forme), d'autres sont liées au refroidissement du détecteur (finesse de gravure). Les contraintes principales sont résumées dans le tableau 4.1. Les autres contraintes techniques, comme le facteur de forme, découlent de ces conditions initiales.

Technologie	CMOS 65nm à 7 niveaux de métal
Entrées	48 voies différentielles à 320Mbps
Sorties	7 voies différentielles à 320/640Mbps
Alimentation I/Os	1.2V
Alimentation coeur	1V (PS) ou 1.2V (2S)
Consommation nominale (en mW)	250 (PS) ou 310 (2S)
Connectivité	Puce nue avec billes (Flip chip)

Table 4.1 – Caractéristiques techniques du CIC.

Là aussi une définition tardive de ces paramètres peut conduire à des situations complexes. Le fait que l'alimentation du coeur du CIC soit différente pour les 2 modules est une bonne illustration de ce problème. Un ASIC n'a pas vocation à fonctionner à 2 tensions différentes, c'est une source de problèmes. Néanmoins, le CIC arrivant en dernier, nous avons dû composer avec les alimentations restantes sur chacun des modules. Le 1.2V n'étant plus disponible en quantité suffisante sur le modules PS, il a fallu passer à 1V. Pourquoi dans ce cas ne pas passer à 1V également sur le module 2S ? Tout simplement car cette tension n'est pas disponible sur les modules 2S....

4.3/ Modele du CIC

4.3.1/ Présentation générale

Une fois les formats de données définis, ainsi que les vitesses des entrées/sorties, il est possible de développer le modèle. Le concentrateur étant un ASIC purement digital (à l'exception de quelques blocs périphériques qui n'ont pas été développés à l'IP2I), son

modèle est central. Celui-ci conditionne la faisabilité du composant physique, et doit être développé dès le départ avec cette contrainte en ligne de mire.

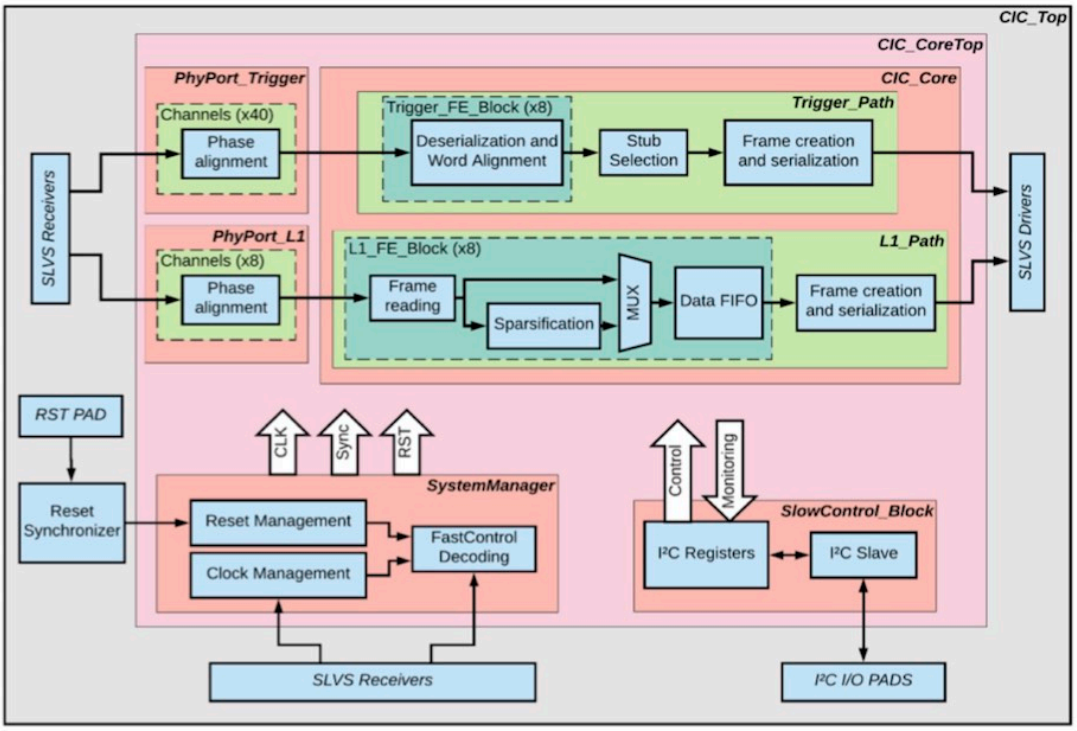


FIGURE 4.6 – Diagramme bloc du CIC1

Le modèle simplifié du concentrateur est présenté dans le diagramme bloc de la Figure 4.6. Les éléments de base sont représentés dans les cadres bleu clairs, organisés en quatre grandes structures (cadres rouges). Les blocs situés à la périphérie sont dédiés à la réception et à l'émission des données. Ils ne sont pas implémentés dans le modèle car on utilise des éléments précompilés que l'on peut introduire directement dans le design physique (on parle de bloc IP (Intellectual Property bloc)). Dans le cas des blocs d'entrée/sortie du CIC, ceux-ci sont fournis par l'INFN [33].

Le modèle proprement dit est le bloc nommé **CIC_CoreTop** sur la Figure 4.6. Les 4 grandes structures qu'il englobe sont les suivantes :

— **PhyPort** : ce bloc est une version simplifiée du phase aligner du lpGBT, dont le but est de remettre en phase l'ensemble des entrées du concentrateur par rapport à l'horloge d'entrée à 320MHz. La différence de parcours entre les différentes lignes peut en effet induire des déphasages mineurs. On ne peut pas parler réellement de bloc IP, car des petites modifications ont dû être apportées au modèle, et celui-ci a donc été recompilé, mais l'essentiel du travail était déjà fait en amont.
— **SystemManager** : ce bloc gère l'ensemble des horloges et des procédures de réinitialisation du concentrateur. A ce titre, il reçoit et décode le signal de FastControl. Il a été entièrement développé par le groupe de Lyon
— **SlowControl** : gère les opérations de contrôle du composant. Ce bloc a été fourni par le groupe de microélectronique du CERN.
— **CIC_Core** : c'est le bloc qui contient le coeur digital du concentrateur, dans lequel les données sont traitées. On y retrouve les 2 voies indépendantes : trigger et L1. Chacune de ces voies, est elle même divisée en 2 étapes : 8 sous-blocs de

reception de données (**FE_blocks**, en vert foncé), et un bloc de traitement et de formatage des informations en sortie. Il a été entièrement développé à l'IP2I.

Cette structuration modulaire présente plusieurs avantages. Le système est beaucoup plus simple à comprendre et les blocs peuvent être développés indépendamment les uns des autres. En terme de design physique, il est également beaucoup plus simple de partir d'un modèle très hiérarchisé. Cela facilite grandement le placement et le routage des différents éléments.

4.3.2/ FONCTIONNEMENT DU COEUR DE L'ASIC

4.3.2.1/ VOIE TRIGGER

Afin d'interpréter correctement les blocs de données provenant des 5 lignes dédiées d'un MPA ou d'un CBC, la mise en phase avec l'horloge à 320MHz effectuée par le bloc PhyPort ne suffit pas. En effet, le bit 0 de la ligne 1 doit correspondre au bit 0 des 4 autres lignes, sinon l'information ne peut pas être reconstruite. Il faut donc que les 5 lignes soient en phase entre elles. D'autre part, les blocs étant émis à 40MHz, il faut aligner l'ensemble des lignes avec l'horloge correspondante.

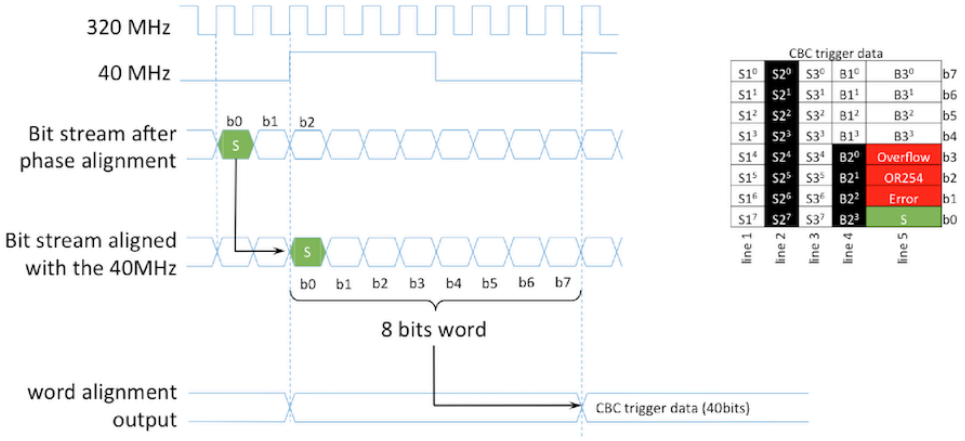


FIGURE 4.7 – Principe du réalignement des données dans le cas du module 2S

La première étape de la voie trigger est donc cette séquence d'alignement, dont le principe est présenté sur la Figure 4.7, dans le cas du module 2S. Une fois cette procédure appliquée, les 5 lignes de chaque CBC/MPA sont alignées avec une horloge à 40/20MHz respectivement.

La dernière étape d'alignement consiste à mettre en phase les données des 8 CBC/MPA. Le bloc de la collision N du CBC1 doit être interprété au même moment que le bloc correspondant des autres CBC. Une procédure similaire à la précédente est alors appliquée, mais cette fois-ci pour 40 lignes.

L'ensemble des voies étant aligné, l'interprétation des données peut s'effectuer. Durant 8 collisions, les stubs de chaque CBC/MPA sont collectés et regroupés dans un registre temporaire. Dans le cas où le nombre de stubs excèderait la taille du registre, priorité est donnée aux stubs de faible largeur. Les stubs sont donc triés au moment de leur stockage. Afin d'optimiser la latence et de minimiser la complexité du routage, un tri bitonique a été

implémenté [34].

Une fois les stubs de la dernière collision récupérés et stockés, le paquet final est prêt à être transmis au lpGBT. Les caractéristiques du paquet (nombre de stubs, erreurs potentielles, position dans le cycle), sont renseignées dans l'en-tête de celui-ci, qui est commune à tous les types de modules. L'architecture de cette voie est entièrement pipelinée, et la latence entre la réception des premiers stubs et l'émission du bloc de sortie est constante et inférieure à 400ns (cette valeur a été confirmée lors des tests du CIC1). Une latence de transmission faible est un point très important pour le système de reconstruction de traces.

4.3.2.2/ VOIE L1

Le principe de fonctionnement de la voie L1 est beaucoup plus complexe que pour la voie trigger. Le type des données reçues par ce bloc est en effet très hétérogène : données non-sparsifiées pour le CBC, sparsifiées pour le MPA. Les données des 8 MPA sont totalement asynchrones, alors que celles des 8 CBCs arrivent a priori en même temps pour les 8 chips. Par contre le temps entre la réception d'un signal L1A et l'arrivée de cet événement dans le CIC n'est pas rigoureusement fixe pour le CBC. La gestion de ces différences notables impose une flexibilité importante de la voie L1, synonyme de complexité. En raison de contraintes de planning strictes, une version simplifiée de la voie L1 a été développée pour le CIC1. Cette version traitait les données normales correctement, mais sa consommation n'était pas optimisée et le traitement des erreurs était incomplet. Nous décrivons ici la version finale de la voie L1, qui a été mise en place dans le CIC2.

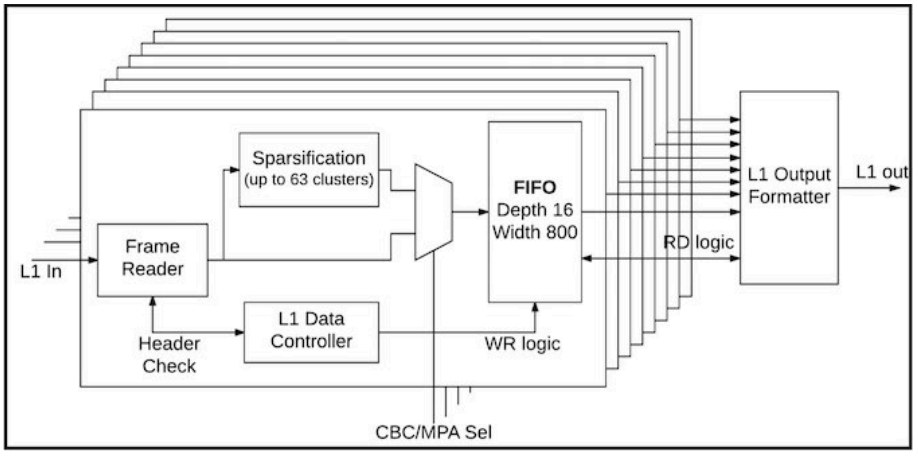


FIGURE 4.8 – Traitement des données de la voie L1

Son principe de fonctionnement est schématisé sur la Figure 4.8. Comme pour la voie trigger, la réception des données se fait séparément pour les 8 ASICs. Dans chacun des 8 blocs de réception, le contrôle des informations est effectué par un bloc L1 Data Controller. Ce bloc fondamental contrôle en particulier la validité du L1 ID reçu. Cet identifiant permet en effet d'étiqueter le bloc, et c'est grâce à lui que les 8 blocs d'un même événement pourront être réunis par le L1 output formatter. Une attention toute particulière est donc apportée à la reception de ce paramètre, et surtout à sa non-réception. En effet, les données étant reçues de manière asynchrone, le CIC doit obligatoirement signaler à la DAQ lorsqu'une trame est absente.

Une fois traitées, les données reçues par un bloc sont stockées dans une FIFO de profondeur 16. Cela signifie que l'on peut stocker jusqu'à 16 événements L1 en parallèle dans un CIC. Pour chacun de ces événements, la taille de la trame ne doit pas dépasser 800 bits. Ces paramètres découlent des études présentées précédemment. On constate au passage que le CIC doit contenir un espace mémoire de taille respectable, en effet : 8x16x800=102400 bits.

En plus de ces 8 FIFOs, les L1ID reçus sont stockés dans une FIFO référente (master FIFO), qui est utilisée ensuite par le L1 output formatter pour récupérer et valider les informations. Ce dernier bloc récupère, pour chaque L1ID contenu dans la master FIFO, les 8 trames correspondantes dans les blocs FE, afin de former la trame finale. Si un ou plusieurs blocs sont absents, des bits d'erreurs sont activés dans le bloc d'identification de la trame (header). Ce principe de fonctionnement permet d'assurer la continuité de la prise de données : le CIC ne se désynchronise pas en l'absence d'une ou plusieurs trames.

Voilà pour les grandes lignes, la voie L1 recèle en effet de nombreuses subtilités qui la rendent extrêmement complexe à élaborer, le lecteur pourra trouver une description détaillée de ces éléments dans la note décrivant les spécifications du CIC C.3.

4.3.3/ Vérification du modèle : simulation comportementale

Avant même de commencer le design physique de l'ASIC, il est important de vérifier que le modèle est fonctionnel. Cette condition n'est bien évidemment pas suffisante pour dire que l'ASIC fonctionne, mais elle est hautement nécessaire.

Le système de vérification développé pour le CIC s'inspire du banc de test qui avait été développé pour le système de reconstruction de traces rapides, qui est décrit dans la troisième partie de ce document. Le principe de base est la modularité, l'idée étant d'utiliser le même programme pour toutes les étapes du développement. Ainsi, le modèle de base pourra être testé avec le même programme que le chip final. L'autre avantage est que l'on est sûr qu'on injecte à chaque étape les mêmes trames, à la même fréquence,...

4.4/ Design et réalisation du CIC

4.4.1/ Présentation du flot de conception

Les différentes étapes de la conception d'un ASIC sont schématisées sur la Figure 4.9. C'est un processus itératif, qui peut se découper en 2 grandes parties consécutives : le design logique et le design physique.

Le design logique correspond au modèle, et au découpage du système en blocs distincts. C'est ce que nous avons décrit dans la partie précédente, et qui est résumé dans la Figure 4.6. Le produit final de cette partie est ce que l'on nomme la netlist synthétisée. Cette netlist correspond au schéma de câblage final du CIC. Elle contient l'ensemble des composants ainsi que leurs interconnexions, mais aucune information quant à leur placement dans le chip.

Cette netlist constitue le point de départ du design physique, dont le but va être de placer les composants dans un espace restreint de façon à aboutir à un ASIC fonctionnel. Il

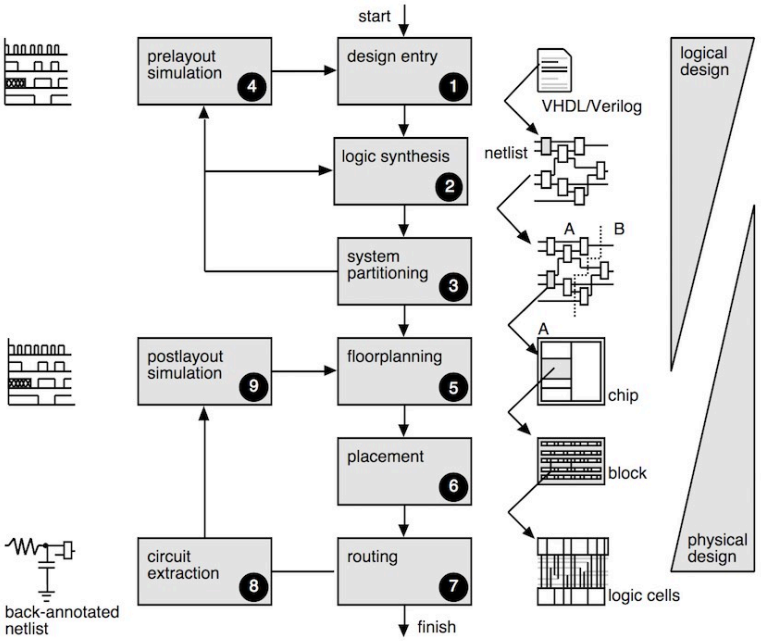


FIGURE 4.9 – Les étapes de la conception d'un ASIC [35]

s'agit d'une étape très complexe car elle doit respecter un nombre très important de contraintes. Dans le cas du CIC1, on parle d'environ 500000 portes logiques à placer et à interconnecter dans environ 15mm^2.

Bien évidemment, tout cela repose sur des logiciels de modélisation. Mais cela ne veut évidemment pas dire qu'il n'y a rien à faire. Un programme peut faire énormément de choses, à condition de lui expliquer en amont ce que l'on veut exactement. La définition des contraintes et des conditions initiales permettant une bonne convergence de ces outils est l'étape fondamentale du design physique d'un ASIC. Elle requiert une attention particulière et une très bonne compréhension de la technologie utilisée.

4.4.2/ LE CIC : DU MODÈLE À LA FONDERIE

La première étape est ce que l'on nomme floorplan : le plan d'implantation de l'ASIC. Celui-ci est contraint par les dimensions du CIC, qui sont elles-mêmes définies par la répartition de ses connexions avec l'hybride de frontend. Dans le cas du CIC, une contrainte supplémentaire vient du fait que le même ASIC doit être compatible avec 2 hybrides totalement différents (PS et 2S). Ces derniers ont été heureusement développés par la même équipe, cela a permis de minimiser les risques liés à ce point.

Une fois les dimensions et la connectique fixées, on peut commencer à placer les blocs IP dans l'espace disponible (Figure 4.10). On définit également les circuits qui vont permettre d'alimenter les cellules du CIC. Les rails d'alimentation sont contenus dans des couches dédiées de l'ASIC [1] et doivent être définis avant les autres car ils traversent l'ensemble du plan de l'ASIC (l'alimentation doit être la plus uniforme possible) et contraignent donc le routage dans toutes les autres couches. Si l'on compare l'ASIC à une maison, les rails d'alimentations sont les murs porteurs. La structure très régulière de

1. Le CIC comporte au total 8 couches superposées.

la grille d'alimentation du CIC2 est représentée sur la Figure 4.11. Nous verrons ensuite que des tests dédiés (IR drop) permettent de vérifier que ce maillage est suffisamment fin une fois le design physique terminé.

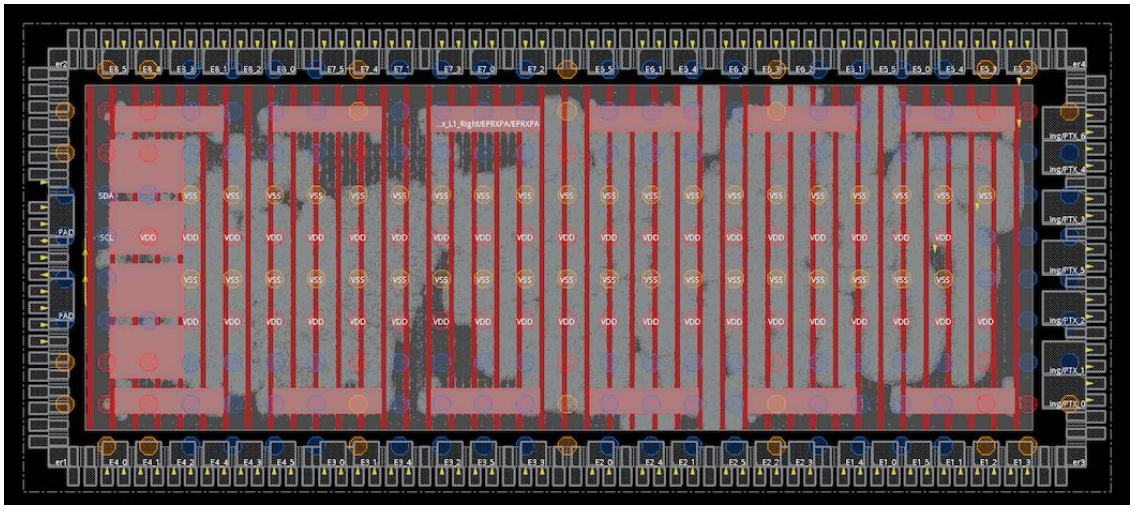


FIGURE 4.10 – Floorplan du CIC2

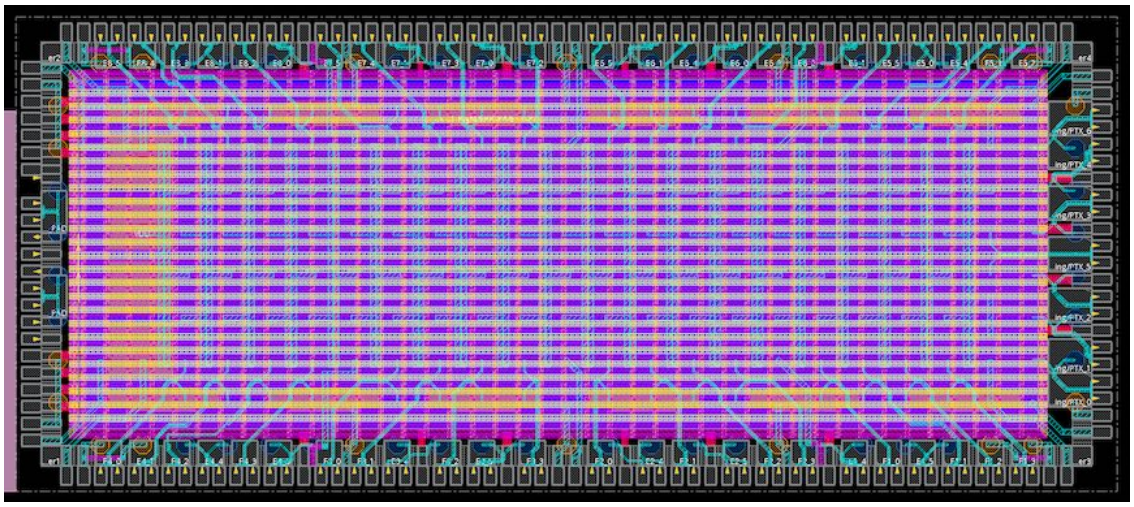


FIGURE 4.11 – Grille d'alimentation du CIC2

Une fois ces étapes réalisées, le programme effectue un premier placement rapide de l'ensemble de la netlist synthétisée. C'est ce premier placement qui est nommé floorplan. On est alors encore loin du design final, en particulier car l'ensemble des cellules n'est encore synchronisé à aucune horloge.

La propagation de l'horloge (ou plutôt des horloges) aux différentes cellules de l'ASIC est l'étape suivante. C'est une opération complexe car il faut que toutes les cellules reçoivent une information qui leur permette de traiter correctement les signaux en continu. Pour un système comme le CIC qui contient plusieurs centaines de milliers de portes logiques et qui reçoit des informations à une fréquence relativement élevée, c'est une opération délicate qui prend généralement plusieurs heures sur une machine puissante. Mais c'est cette étape qui fait rentrer l'ASIC dans le monde réel. En effet, une fois celle-ci franchie avec succès, on dispose presque d'une version réaliste. Il reste pour cela à effectuer la

dernière étape : le routage.

L'étape de routage consiste à optimiser l'ensemble des connexions de l'ASIC, désormais cadencé à la bonne fréquence, de manière à ce que tous les signaux puissent se propager correctement dans celui-ci. Un signal entrant dans le CIC à un instant donné dispose en effet d'une latence limitée pour passer d'une cellule à la suivante. S'il arrive en retard, le logiciel effectuant le routage signale une violation temporelle. Après la première étape de routage, il subsiste encore des centaines voire des milliers de violations de ce type. On procède alors à des optimisations, qui sont des étapes supplémentaires au cours desquelles le logiciel va redéfinir les connexions afin de supprimer les violations. C'est un processus itératif car de nouvelles violations vont apparaitre, mais si les conditions initiales du projet on été correctement définies, 4 ou 5 itérations suffisent généralement à obtenir un circuit sans violation. On parle alors de timing-closure.

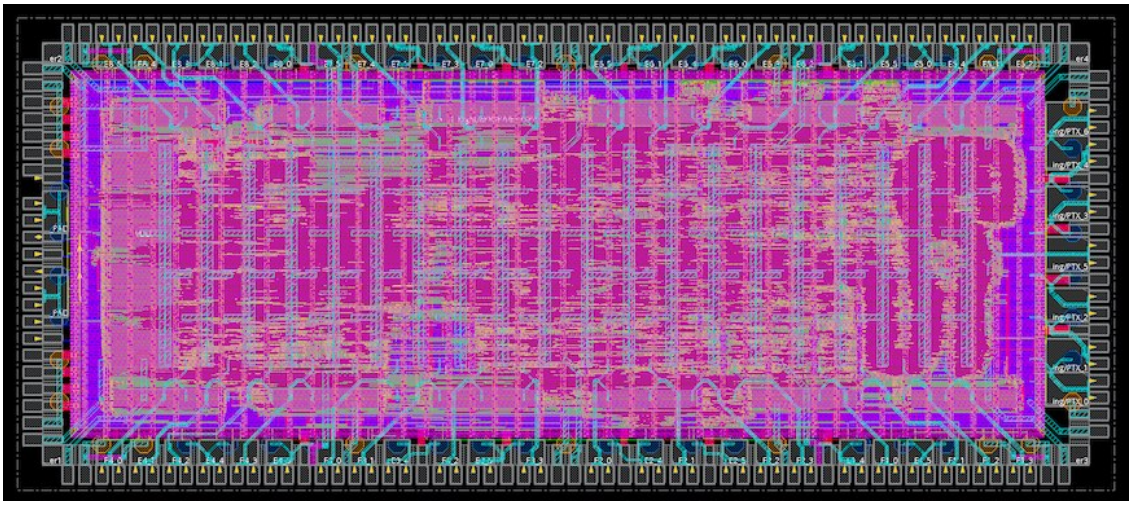


FIGURE 4.12 – CIC2 après routage complet

Dans le cas contraire, les contraintes du programme sont modifiées, voire le modèle si les violations sont irréductibles ou trop importantes. L'obtention de la timing closure est donc logiquement l'étape la plus longue du design physique d'un ASIC mais aussi la plus importante. Une fois celle-ci obtenue, on dispose d'un système physiquement réaliste, le design routé (Figure 4.12).

Mais avant d'envoyer ce circuit en fonderie, il doit encore franchir toute une série de vérifications. Comme dit plus haut, ce qui est gravé est gravé, il est donc important de s'assurer que ce que l'on va fabriquer correspond au cahier des charges.

Pour commencer, on vérifie avec la simulation développée pour le modèle que le comportement du système correspond à celui attendu : c'est la vérification comportementale, mais cette fois-ci avec un système réaliste (en particulier avec des parasites).

Ensuite, on estime la puissance consommée par l'ASIC, ainsi que les baisses de tension sur l'ensemble du système (IR drop). La puissance doit rester raisonnablement en dessous des valeurs nominales, et l'IR drop (Figure 4.13) doit être inférieur à 5% de la tension d'alimentation. En effet, si une cellule n'est pas alimentée correctement, celle-ci risque de mal fonctionner et d'engendrer un dysfonctionnement global. C'est pour cette raison que la grille d'alimentation est si importante.

Une fois toutes ces vérifications effectuées, on réalise une dernière série de contrôles

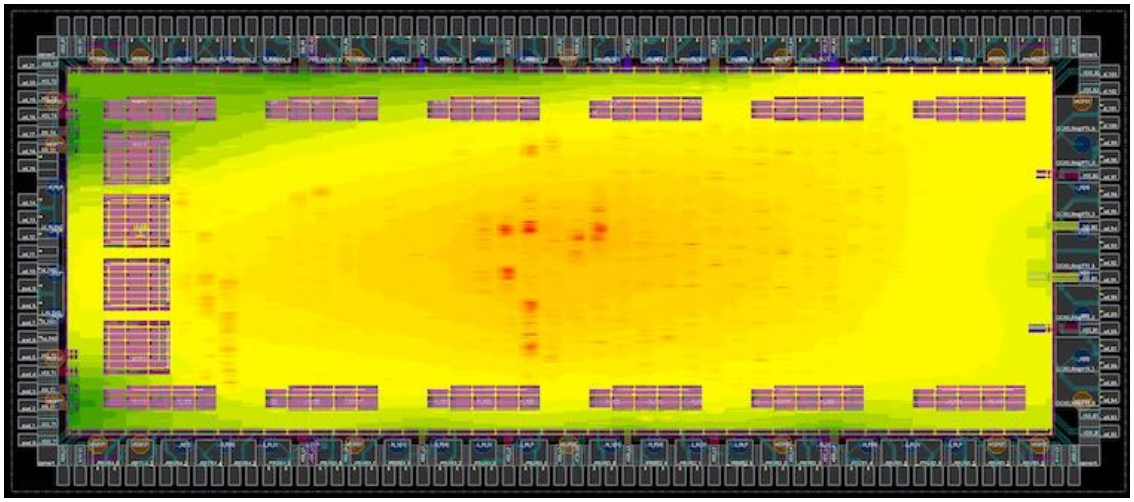


FIGURE 4.13 – Analyse IR drop pour le CIC2. L'IR drop maximal (cellules rouges) correspond ici à environ 10mV, soit moins de 1% de la tension nominale.

très importants : DRC et LVS.

Le DRC (pour Design Rule Check) est une vérification de la conformité du design produit avec les règles de fabrication du graveur (dans notre cas TSMC). Certaines règles doivent être obligatoirement respectées, mais pour d'autre il est possible de passer outre (le graveur se déchargeant évidemment de toute responsabilité en cas de problème).

Le LVS (pour Layout Versus Schematic) est une vérification de la compatibilité entre la netlist finale et la netlist synthétisée. La simulation comportementale ne permet effectivement pas de vérifier que l'ensemble des connexions du design final est rigoureusement identique aux connexions définies initialement. Celles-ci ont pu être modifiées au cours du processus de développement. Il faut s'assurer que ces modifications n'ont pas eu d'impact sur le comportement global de l'ASIC.

Une fois ces 2 dernières étapes validées, une image complète de l'ASIC est produite et transmise au graveur via un intermédiaire qui vérifie une dernière fois le projet. Le composant gravé sur le silicium revient environ 2 mois plus tard. On peut alors entamer la phase de tests.

4.5/ CARACTÉRISATION DU CIC1

4.5.1/ DESCRIPTION DU BANC DE TEST

En tant que responsable du développement du CIC, notre groupe est en charge des tests fonctionnels de l'ASIC seul. Le but de ces tests est de vérifier que le CIC répond au cahier des charges fonctionnel. Les points à vérifier sont la consommation, le traitement des données, et aussi la résistance aux radiations, si l'ASIC est durci.

Une fois ces vérifications effectuées, le CIC est testé dans son environnement final. Ce test, que nous évoquerons rapidement à la fin de ce chapitre, n'est pas sous la responsabilité de notre groupe.

Le CIC1 contient l'ensemble des fonctionnalités du chip final et possède la même em-

preinte pour le câblage sur carte. La principale différence est que les techniques de durcissement aux radiations n'ont pas été utilisées pour son design, alors que ce sera le cas pour la version finale (le CIC2). Le système de test des deux versions sera donc identique. Celui-ci est représenté sur la Figure 4.14, se compose de trois cartes : la mezzanine CIC, la carte d'interface, et la carte de pilotage. Les 2 premières cartes ont été développées à l'IP2I, la troisième est une carte d'évaluation Xilinx KCU105.

La mezzanine CIC est une petite carte sur laquelle l'ASIC est connecté. On peut voir une de ces cartes, avec l'ASIC, sur la Figure 4.15. Il s'agit d'un système de test, l'ASIC est connecté de manière définitive et ne peut pas être réutilisé ensuite. Une trentaine de cartes de ce type ont été réalisées et testées pour le CIC1.

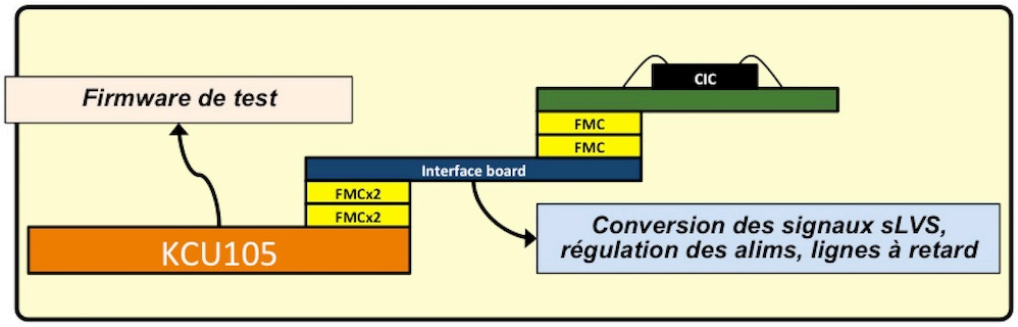


FIGURE 4.14 – Système de test du CIC

La carte d'interface permet de piloter de manière précise l'alimentation de l'ASIC, et de convertir les signaux envoyés par la carte d'évaluation dans un format compréhensible par le CIC. C'est une zone de transit.

Enfin, la carte d'évaluation contient l'ensemble du firmware de test et les outils permettant de mettre en oeuvre ce firmware. Celui-ci est très largement inspiré du système développé pour notre banc de test de reconstruction de traces rapides, qui est décrit dans le dernier chapitre de ce document.

Le pilotage final de tout ce système est réalisé à partir d'un environnement software codé en langage Python, interfacé depuis un PC avec la carte d'évaluation. Là encore, le banc de test du projet FastTrack (cf. Chapitre 7) a fourni une base de départ solide qui nous a permis d'être opérationnels dès la réception des CIC1 en Janvier 2019.

4.5.2/ RÉSULTATS

Lorsque l'on reçoit un ASIC, la première chose à vérifier est qu'il est capable de recevoir et de renvoyer des informations. Le premier bloc que l'on teste est donc celui de slow control, car sans lui, il est impossible de communiquer avec le composant. Une fois ce premier test validé, on peut passer aux choses plus sérieuses. Dans le cas contraire, on teste un autre chip, et si le problème persiste, on a un vrai souci... Heureusement, ce ne fut pas le cas pour le CIC1, et nous avons pu passer tout de suite à l'étape suivante.

Le CIC est un ASIC digital qui traite des données, on doit donc vérifier qu'il traite effectivement les informations dans une situation normale. Grâce aux outils de vérification fonctionnelle développés pour tester le modèle et la netlist, ce test a pu être effectué très rapidement. On a pu ainsi établir le bon fonctionnement du CIC pour toutes les étapes de

FIGURE 4.15 – Vue d'un CIC1 micro-câblé sur la mezzanine de test.

fonctionnement (configuration incluse), et pour tous les types de données.

Après cette validation, la consommation du chip a été mesurée. Le résultat de cette étude est présenté sur la Figure 4.16. On y voit la consommation mesurée d'un CIC1 en mode 2S avec transmission d'informations sur la voie données, pour toutes les étapes de fonctionnement de l'ASIC, du démarrage à la transmission. On a pu ainsi vérifier que la consommation était légèrement inférieure, tout en restant compatible, aux prévisions, et dans tous les cas inférieure à la consommation nominale.

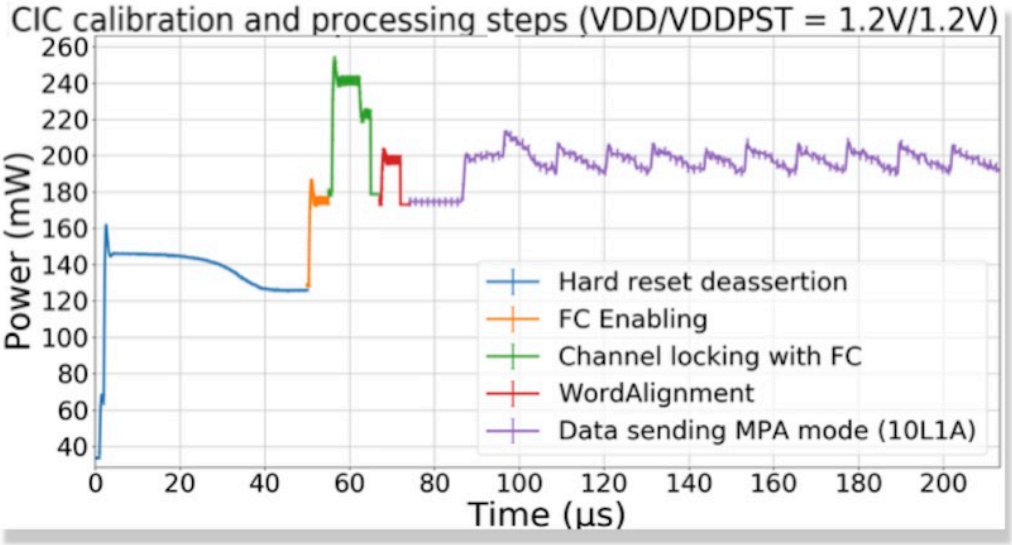


FIGURE 4.16 – Profil de consommation du CIC1

D'autre part le CIC1 a été exposé à une dose de 2.1 MGray de rayons X (la dose totale attendue sur l'ensemble du fonctionnement du HL-LHC étant de 1MGray). Ce test n'a pas révélé une dégradation prématurée du composant due au radiations. Cela ne veut bien évidemment pas dire qu'il ne faut pas durcir le CIC1. Le durcissement prévient les SEUs (Single Event Upset), qui n'ont pas de rapport avec le vieillissement dû à la dose reçue.

4.5.3/ Le CIC1 dans la chaîne d'acquisition du module 2S

Le CIC1 a également pu être testé dans une chaîne d'acquisition plus complète. Ces tests ont été effectués au CERN au printemps 2019. Le CIC1 a été connecté avec 8 CBC, émulant ainsi un hybride de frontend complet du module 2S. Ces tests ont permis de valider totalement la communication entre ces 2 ASICs, ainsi que le fonctionnement du CIC1 en conditions réalistes. C'est une étape majeure en vue de la réalisation d'un premier prototype de module 2S complet.

Au final, mis à part quelques petites corrections mineures, le CIC1 a parfaitement fonctionné. Cela nous a permis d'entamer le développement du CIC2 avec une relative sérénité.

4.6/ Développement du CIC2

4.6.1/ Description et performances attendues

Le CIC1 a dû être développé rapidement afin de respecter des délais de soumission très stricts. De ce fait, certains éléments ont été volontairement mis en suspens. Ces points sont le durcissement aux radiations, le fonctionnement à 640MHz, et la consommation de la voie L1.

Ils ont été pris en compte au cours du développement de la seconde version de l'ASIC : le CIC2, qui a eu lieu de Novembre 2018 à Juillet 2019. Le comportement de l'ASIC n'a pas été modifié, mais son architecture interne à été revue en profondeur.

4.6.1.1/ Durcissement de l'architecture

De part leur caractère imprévisible, les SEUs seront les principaux ennemis des ASICs du HL-LHC. Leur quantité est relativement difficile à prévoir. On peut au mieux faire en sorte que le composant puisse réagir correctement au cas où un SEU survient.

Pour rendre un ASIC résistant aux SEUs, on applique une procédure de triplication. Les registres les plus importants de l'ASIC sont tous triplés, et on mesure la valeur du registre en effectuant un vote majoritaire. De cette manière, même si un SEU traverse un des registres, les 2 autres ne sont pas affectés et le signal n'est pas corrompu.

Cette technique simple est très efficace, mais par construction elle triple la complexité du circuit. Néanmoins, les registres n'ont pas tous la même importance. En pratique on triplique seulement les registres de contrôle, et on ne protège pas les données. Les conséquences d'un SEU sur un signal de resynchronisation peuvent en effet être beaucoup plus graves que celles d'un SEU sur l'adresse d'un stub.

C'est la procédure qui a été appliquée pour le CIC2. Certaines données importantes, comme par exemple les en-têtes des trames de sortie de la voie données, ont également été protégées. Au final la triplication a augmenté la consommation totale d'environ 15%, ce qui est tout à fait acceptable.

4.6.1.2/ Optimisation de la consommation

La consommation du CIC1 n'a fait l'objet d'aucune optimisation. En particulier, nous savions que la voie données devait être améliorée de manière significative.

Ce travail a nécessité la réécriture de l'ensemble des blocs liés à la voie L1, mais l'effet de ces optimisations est, comme on peut le voir sur la Table 4.2, significatif.

Version	Consommation max (en mw)
CIC1 (1V)	241
CIC2 (1V)	207
CIC1 (1.2V)	322
CIC2 (1.2V)	235

Table 4.2 – Comparaison des consommation maximales estimées pour le CIC1 et le CIC2.

Bien que l'on ait ajouté la triplication, la consommation du CIC2 a sensiblement diminué par rapport à celle du CIC1. Ce résultat reste à confirmer avec la version physique du CIC2, mais il est très encourageant.

4.6.1.3/ Statut du projet

Le design du CIC2 a été complété en Juillet 2019, date à laquelle le projet est parti en fonderie. Les ASICs sont revenus en Octobre et sont actuellement en cours de test à l'IP2I. Les tests, qui vont cette fois inclure une campagne de tests en faisceaux, vont s'étendre jusqu'au premier trimestre de 2020.

4.6.2/ Et après

Le CIC2 est sensé être un ASIC de pré-production. Cela signifie que s'il fonctionne comme prévu, il devrait pouvoir être installé sur les modules finaux du trajectographe. Si les tests permettent de valider son fonctionnement, la fonderie des 30000 chips du futur détecteur est prévue pour l'été 2020.

4.7/ Bilan

A l'heure où j'écris ces lignes je ne sais pas encore si le CIC2 va répondre à nos attentes ou pas. Mais le CIC1 est totalement fonctionnel, et il s'agit du premier ASIC réalisé par l'IN2P3 dans cette finesse de gravure. A ce titre, on peut donc considérer que le projet est un vrai succès.

Pour en arriver là, il a fallu traverser un certain nombre de péripéties. La liste est longue, et je ne m'y attarderai pas ici, mais si il y a une leçon à retenir, c'est qu'un projet comme le CIC ne s'improvise pas. Les règles énoncées dans la partie 4.1.2 sont les conditions nécessaires au démarrage de ce type d'entreprise.

IV

Analyser les données

5

CONSTRUIRE DES TRACES, VITE...

Comment extraire et traiter les données du trajectographe le plus rapidement possible ? Nous avons consacré la partie précédente à résoudre le premier problème, il faut désormais s'attaquer au second. Ce chapitre dresse un état des lieux de la problématique et présente brièvement les solutions envisagées pour le trajectographe de CMS.

5.1/ INTRODUCTION

5.1.1/ POURQUOI ?

Comme nous l'avons déjà évoqué dans un chapitre précédent, la sélection en ligne des événements sera un problème majeur au HL-LHC. Comment savoir, lorsque chaque collision engendre un nombre important de signaux, si celle-ci doit être conservée ou pas ?

Le trajectographe est un outil de sélection très puissant, mais jusqu'à présent il n'était pas envisageable d'utiliser ses données dans un délai de quelques microsecondes. Nous avons vu dans la partie précédente que le futur trajectographe de CMS fournira une solution à ce problème via les stubs.

Mais que faire avec ces signaux ? Au HL-LHC, avec un empilement moyen de 200 interactions par collision, ce sont environ 20000 stubs qui seront collectés à chaque croisement des faisceaux. On disposera alors d'un laps de temps très court pour former des traces. Comment procéder ?

Généralement, la meilleure solution, pour résoudre un problème, c'est de le réduire, autant que possible, en sous-problèmes indépendants. Ce constat s'applique parfaitement à la reconstruction de traces rapides, où l'on peut immédiatement identifier 2 étapes distinctes : la distribution de données et la reconstruction des traces. Si nous aborderons rapidement le premier point dans le contexte de CMS, nous nous concentrerons essentiellement sur la deuxième étape dans la suite de ce document

Concernant la reconstruction de traces, il est là encore possible de réduire le travail à effectuer en plusieurs tâches précises. En effet, ce processus se fait systématiquement en deux temps : l'identification des candidats (pattern recognition), et l'ajustement de leurs paramètres (fit). En terme de latence, l'identification est le point critique. Pour être

rapide, un algorithme de reconstruction de traces doit avant tout s'atteler à débloquer ce premier verrou.

5.1.2/ ETAT DE L'ART

Jusqu'à une période très récente, la quasi-totalité des systèmes de reconstruction de traces en temps réel déployés dans des trajectographes se sont appuyés sur les mémoires associatives.

Une mémoire associative est basée sur un ensemble de mémoires adressables par le contenu, les CAM (Content-Addressable Memory). Il s'agit d'un type de mémoire particulier utilisé principalement dans les applications de recherche à très grande vitesse. Contrairement à la mémoire standard d'un ordinateur (RAM), pour laquelle l'utilisateur fournit une adresse et récupère le mot de données qui y est stocké, une CAM effectue l'opération inverse : l'utilisateur fournit un mot de données et récupère, le cas échéant, toutes les adresses dans lesquelles le mot est stocké.

Ce type de fonctionnement, qui peut paraître étrange à première vue, présente un intérêt énorme pour un problème d'identification en temps réel comme le notre. En effet, la demande de données étant diffusée simultanément sur tous les éléments de la mémoire, le temps d'identification dépend linéairement de la quantité de stubs en entrée, là où la plupart des méthodes classiques ont une complexité quadratique. Le potentiel de ce procédé est donc très clair, reste à l'adapter à l'identification de traces.

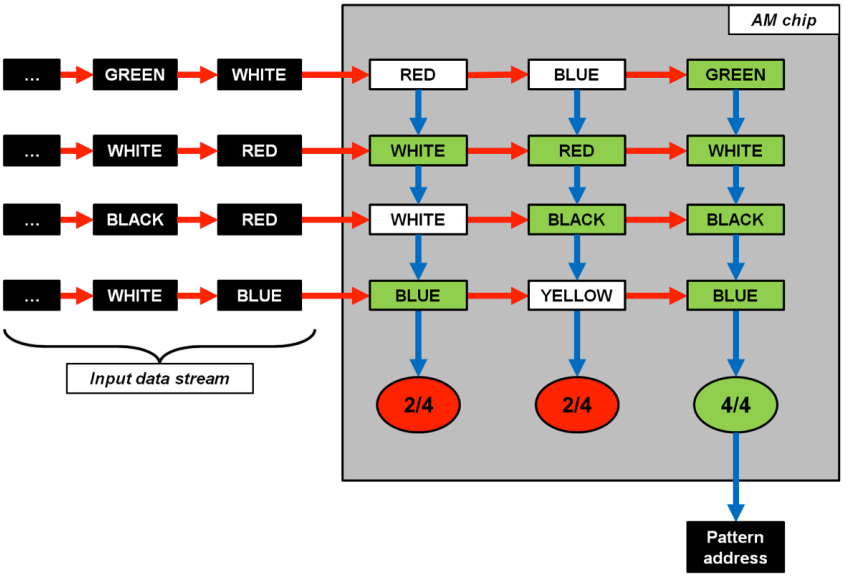


FIGURE 5.1 – Principe de fonctionnement d'un chip AM.

Des CAM commerciaux ont été utilisés dans l'expérience H1 du collisionneur positon-proton HERA dès le début des années 1990 [36]. Mais le principe de la mémoire associative tel que nous l'avons utilisé dans CMS, et tel qu'il est présenté sur la Figure 5.1 a été proposé pour la première fois à la fin des années 80 [37] et mis en application à la fin des années 90 dans l'expérience CDF [38, 39].

Dans ce système, chaque motif de trace est stocké dans une seule zone de mémoire, mais celle-ci est constituée de N mots indépendants : les adresses. Chaque adresse peut

être identifiée à la position d'un signal potentiel sur une couche de détecteur (par exemple un stub). L'association entre différentes couches de détecteurs est réalisée de manière interne pour chacun des motifs, sur le principe d'un vote majoritaire du même type que celui utilisé pour la triplication du CIC.

La Figure 5.1 permet d'expliciter simplement le principe de l'identification. Dans cet exemple, le motif est composé de quatre mots. Comme on peut le constater, les quatre lignes d'entrée sont indépendantes les unes des autres. Les mots d'une ligne peuvent être transmis dans un ordre aléatoire ; toutes les correspondances sont connues dès que le dernier mot est passé dans le système. Le nombre de mots pouvant différer d'une ligne à l'autre, le temps nécessaire pour faire la correspondance sur une ligne dépend linéairement du nombre de mots.

L'identification de motifs par mémoires associatives se prête particulièrement bien à un portage sur ASIC. Les premiers chips AM ont été développés au milieu des années 90 [40]. C'est avec la première génération de cet ASIC, qui contenait quelques centaines de motifs seulement, que cette technique a pu être utilisée avec succès dans CDF. Cela a permis au détecteur d'améliorer très sensiblement son système de déclenchement et de faire d'importantes découvertes, notamment dans le domaine des saveurs lourdes. Le système développé est devenu une référence, si bien qu'il est actuellement prévu de le mettre en place dans le détecteur ATLAS [41]. A cet effet, la septième génération de chip AM est en cours de finalisation. Les chips installés dans ATLAS contiendront plusieurs centaines de milliers de motifs.

Ce système a logiquement été à la base du projet développé pour CMS dès 2007 [42]. Cependant, nous verrons également que d'autres méthodes ont été mises au point depuis.

5.2/ UN TRACK TRIGGER POUR CMS

5.2.1/ LE PROBLÈME

Implémenter un système de reconstruction de traces rapide dans CMS ou ATLAS implique de franchir un ordre de grandeur en terme de taille et de complexité par rapport aux systèmes précédents. Dans le cas de CMS, le défi à relever est le suivant : comment traiter un flux de 50 Tbps de données en quelques microsecondes ? Ces 2 contraintes sont habituelles en physique des hautes énergies, mais il est beaucoup moins courant de les trouver rassemblées dans un seul et même problème.

Comme mentionné plus haut, il est important de considérer les deux étapes suivantes, distribution de données et reconstruction des traces, de la manière la plus indépendante possible. Dans le cas de CMS, on peut considérer que la latence disponible pour ces deux étapes est d'environ 2 et 4 microsecondes respectivement. Avant de nous intéresser plus précisément à la reconstruction des traces dans les chapitres suivants, nous allons décrire les grandes lignes d'un système complet dans le cas de CMS.

5.2.1.1/ DISTRIBUTION DES DONNÉES

La reconstruction des traces s'effectue sur des cartes électroniques localisées dans les salles de comptage situées hors du détecteur, à l'abri des radiations. A la fin de la partie précédente nous avions laissé les stubs dans les fibres optiques quittant les modules. Comment les acheminer à la carte qui va reconstruire les traces ? Avec quelle latence ?

Là aussi il faut diviser le problème. En effet, il est tout à fait inenvisageable d'acheminer une telle quantité de données chaque seconde dans une seule carte de traitement. On divise donc l'information en espace, mais aussi en temps. Le système final est ainsi composé d'un certain nombre de cartes qui traitent chacune les informations d'une partie du détecteur (diminution de la quantité de données à traiter) pour une partie des collisions (augmentation du temps disponible pour le traitement).

Ces secteurs indépendants sont nommés tours de trigger. Une projection dans le plan R/Z des 48 tours de CMS (le découpage en 8 sections égales en ϕ amène à 48 secteurs), correspondant à l'approche AM, sont représentées sur la Figure 5.2.

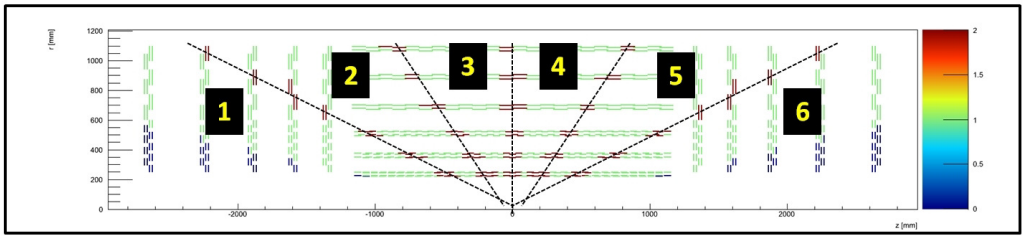


FIGURE 5.2 – Distribution de données : multiplexage en espace

Les données de chaque tour sont gérées par un ensemble de 11 cartes ATCA contenues dans un chassis adapté à ce standard de production très largement utilisé à l'heure actuelle dans les télécommunications. Le principe du multiplexage en temps mis en place pour l'approche AM de CMS est décrit sur la Figure 5.3.

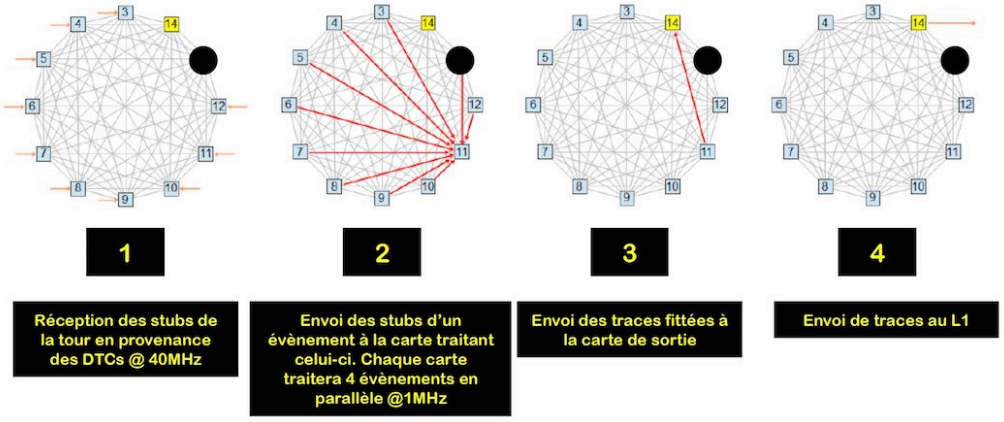


FIGURE 5.3 – Distribution de données : multiplexage en temps.

Toutes les 25 ns, les stubs de l'événement N contenus dans la tour considérée sont envoyés dans le chassis correspondant où ils sont distribués en parallèle à 10 cartes de traitement (étape 1). Chaque carte redistribue ensuite les fragments de l'événement N à la carte traitant cet événement (étape 2). Les cartes disposant chacune de 4 unités

de reconstruction, on peut ainsi traiter 40 événements en parallèle, et donc disposer de 1 microseconde au lieu de 25 ns pour effectuer la première étape de reconstruction (identification). Le multiplexage en temps permet ainsi de disposer de la latence nécessaire pour effectuer les différentes étapes de la reconstruction, à condition que la latence de chaque étape ne dépasse pas 1 microseconde.

5.2.1.2/ Reconstruction des traces

Une fois les données correctement transmises, nous disposons donc de l'intégralité des stubs de la collision N répartis dans chacun des 48 secteurs. Pour reconstruire les traces dans un secteur, nous avons 3 à 4 microsecondes au total, sachant que compte-tenu du multiplexage en temps, la reconstruction doit s'effectuer en étapes successives d'environ une microseconde.

Nous l'avons vu plus haut, la reconstruction de traces se base justement sur des étapes consécutives, identification et ajustement. L'identification est bien évidemment la plus complexe à réaliser dans une latence très courte. C'est donc logiquement surtout sur ce point que diffèrent les trois approches qui ont été évaluées dans CMS au cours des années précédentes.

En effet, même si ces méthodes peuvent paraitre très différentes sur le papier, une étude attentive montre que la principale différence réside surtout dans l'étape d'identification des candidats. Les autres étapes (distribution de données et ajustement des traces) sont très facilement adaptables d'une méthode à l'autre.

5.2.2/ Identification rapide de traces dans CMS

5.2.2.1/ Mémoires associatives

Cette approche a longtemps été, pour des raisons évoquées précédemment, la seule étudiée dans CMS. Le principal obstacle de cette technique étant le développement d'un chip AM dédié à CMS. Nous avons vu que la réalisation d'un ASIC est tout sauf un long fleuve tranquille, la crainte de retards de développement a été un des arguments principaux à l'encontre de cette approche. Nous reviendrons en détails sur ce point dans les chapitres suivants mais, vous l'aurez compris, ce n'est pas l'approche qui a été choisie au final par CMS.

5.2.2.2/ FPGA

Les FPGA (Field-Programmable-Gate-Array) sont des composants re-programmables in situ. Les capacités de ces systèmes se sont accrues de manière exponentielle au cours des 20 dernières années, et aujourd'hui les composants les plus puissants peuvent effectuer des opérations extrêmement complexes tel qu'un algorithme d'identification de traces. De nos jours, les FPGAs sont omniprésents dans les salles de comptage, il n'y a donc rien d'étonnant à les voir jouer un rôle majeur dans le domaine de la reconstruction en temps réel.

Le fait qu'ils soient reconfigurables peut être vu comme un avantage notable par rapport

à un ASIC tel que le chip AM [1] car il est possible à tout moment d'améliorer les performances du système en reprogrammant un algorithme plus puissant dans le FPGA. Par contre, la consommation d'un FPGA est plus importante que celle d'un ASIC, et l'identification par mémoires associatives, qui n'est pas adaptable aux FPGA actuels, est la seule méthode totalement linéaire et indépendante de l'empilement. Mais des algorithmes tels que la transformée de Hough, ou un filtre de Kalman simplifié [43, 44], ont fourni des résultats suffisamment prometteurs pour convaincre les responsables du trajectographe de CMS de privilégier cette approche à la fin de l'année 2016.

D'un point de vue strictement pragmatique, le développement d'un algorithme présente moins de risques que le développement d'un ASIC. Si les performances estimées paraissent comparables, le FPGA aura toujours un sérieux avantage.

1. Nous verrons cependant par la suite qu'un chip AM dispose également d'une certaine flexibilité

6

UTILISER LES MÉMOIRES ASSOCIATIVES DANS CMS

La clé d'un système de reconstruction de traces en temps réel est l'identification des traces. Avant de développer un démonstrateur complet autour des mémoires associatives, il faut s'assurer que celles-ci sont utilisables dans CMS. Notre première contribution, entre 2012 et 2014, a donc consisté à développer une émulation complète de chip AM afin d'évaluer le potentiel d'une telle approche.

6.1/ MISE EN PLACE DE L'ALGORITHME

6.1.1/ PRÉAMBULE

Le principe d'une identification de traces basée sur des mémoires associatives est schématisé sur la Fig. 6.1.

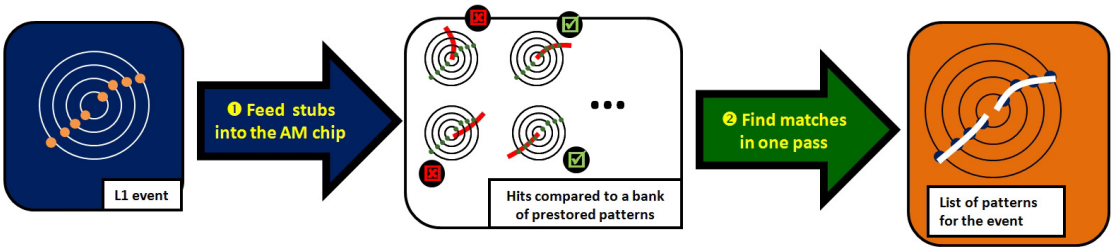


FIGURE 6.1 – Identification de traces avec des mémoires associatives.

Les stubs reconstruits pour chaque événement sont injectés dans une puce AM où ils sont comparés à une banque de motifs stockés au préalable. Les motifs activés sont ensuite transmis à un système permettant d'évaluer les paramètres des traces.

Dans le contexte du trajectographe ces motifs peuvent être vus comme des pistes de faible granularité. Ils sont définis une fois pour toutes à l'aide d'événements simulés et ils sont supposés représenter la majeure partie des traces reconstructibles dans le

détecteur.

Chacune des 48 tours du système aura sa propre banque de motifs indépendante. Le but du jeu est de créer les 48 banques les plus performantes pour le futur trajectographe.

6.1.2/ BANQUES, MOTIFS, ET AUTRES SUPERSTRIPS

6.1.2.1/ QU'EST QU'UNE BONNE BANQUE ?

Avant d'aller plus loin dans les aspects techniques de la reconnaissance de formes AM, il n'est pas inutile de préciser ce que nous recherchons exactement. Au sortir du trajectographe environ 95% des stubs ne proviennent pas des traces que l'on cherche à reconstruire. Le rôle des mémoires associatives est de réduire au maximum cette proportion. Il s'agit avant tout d'une étape de filtrage ; on ne reconstruit pas les traces, on crée un environnement propice à leur reconstruction. Nous devons donc essayer d'augmenter au maximum la pureté des stubs sans diminuer l'efficacité de reconstruction. On voit bien que si l'on veut respecter cette contrainte le nombre de motifs nécéssaire risque d'augmenter de manière démesurée, alors que nous devons prendre en compte les limites techniques des puces AM et tenter de réduire au maximum la taille des banques.

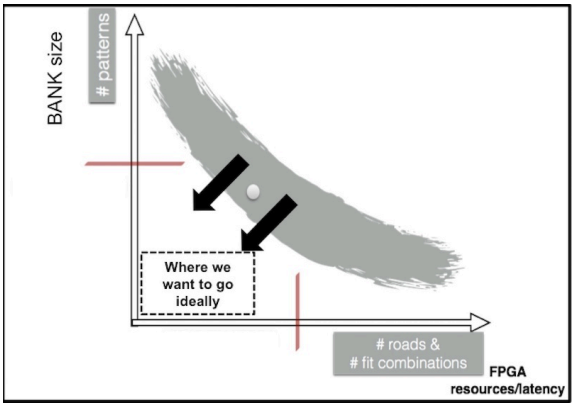


FIGURE 6.2 – Optimisation d'une banque de motifs

Comme toujours dans ce type de problème la solution va consister à trouver un point d'équilibre. La Figure 6.2 résume le problème et définit le point de fonctionnement que l'on va essayer d'atteindre. En ordonnée est représentée la taille de la banque et en abscisse un indicateur de la pureté des données en sortie (la pureté décroit lorsque cet indicateur augmente). La zone grisée représente les points de fonctionnement des banques de motifs. Une banque de petite taille aura des motifs de grandes dimensions qui seront activés plus souvent et seront moins purs. Le cas extrême serait une banque à 1 seul motif (le secteur entier), qui serait très efficace mais très bruité ! Dans le cas inverse, on peut imaginer une banque de taille très importante contenant des motifs de largeur minimale (1 strip). Une telle banque aurait évidemment une pureté optimale, mais nécessiterait un nombre de chips AM beaucoup trop important.

Si l'on s'appuie sur la Figure 6.2, on va donc chercher à atteindre le point le plus proche de l'origine. La banque idéale sera de petite taille, efficace et engendrera un nombre réduit de routes avec la pureté la plus grande possible. Pour déterminer cette banque, nous allons devoir nous pencher sur son élément de base : le motif.

6.1.2.2/ DÉFINITION DU MOTIF

Si le stub est l'élément clé de l'étape d'extraction, le motif est l'objet fondamental de l'étape AM. On peut le comparer à une route traversant l'ensemble du détecteur. Chaque route peut contenir une ou plusieurs trajectoires de particules, comme indiqué sur la Fig. 6.3.

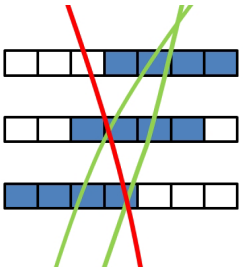


FIGURE 6.3 – Exemple de motif dans un détecteur simple. Les traces vertes seront associées au motif, pas les rouges.

Cet exemple donne une idée claire de la structure du motif. Chaque motif est constitué de plusieurs éléments, un pour chaque couche du détecteur traversée. Cet élément est appelé superstrip. Il s'agit d'un objet qui dépend fortement du détecteur lui-même. Sa dimension a un impact direct sur le pouvoir de filtrage et sur la taille de la banque. Un motif composé de superstrips étroits assurera un bon pouvoir filtrant, mais conduira à une banque de très grande taille, alors qu'un motif composé de superstrips larges conduira à une petite banque avec un faible pouvoir de filtrage.

6.1.3/ DÉFINITION DU SUPERSTRIP

Le superstrip est le constituant élémentaire du motif. En pratique il s'agit d'une adresse permettant de définir une zone de la tour de trigger couverte par la banque que l'on cherche à construire. Chaque stub appartenant à la tour de trigger peut être associé à un superstrip. Lors de l'étape d'identification, on commencera donc par convertir tous les stubs reçus en superstrips, qui sont les seules informations comprises par le chip AM. Du point de vue du chip AM, le superstrip doit donc avant tout être vu comme une adresse avec pour seule contrainte une taille fixe de 16 bits.

Une fois les propriétés du superstrip définies, les informations de la tour correspondante sont encodées dans des mots de 16 bits qui vont fournir l'ensemble des adresses du secteur. Toutes les informations relatives au détecteur sont contenues dans ces adresses qui vont être utilisées par le chip AM.

La Fig. 6.4 montre comment les superstrips sont définis géométriquement dans le futur trajectographe de CMS, dans le cas d'un module 2S. Le superstrip est simplement un groupe de strips et de segments. Son adresse est donc une coordonnée locale dans le secteur, il s'agit d'encoder deux informations : la position du module dans le secteur et la position du superstrip dans le module.

Les chips AM disposent de 8 lignes d'entrées indépendantes. Pour la plupart des secteurs chaque ligne est assignée à une couche de détecteur précise. Il n'est ainsi pas nécessaire d'encoder le numéro de la couche dans le superstrip. Les secteurs hybrides contenant

9 couches distinctes, il est nécessaire de partager une ligne entre 2 couches. Mais là aussi le numéro de la couche n'est pas nécéssaire pour le superstrip, car il s'agit de 2 zones géométriquement indépendantes. On encode donc dans le superstrip seulement 2 coordonnées dans chacun des cas : Z/ϕ dans le tonneau, et R/ϕ dans les bouchons.

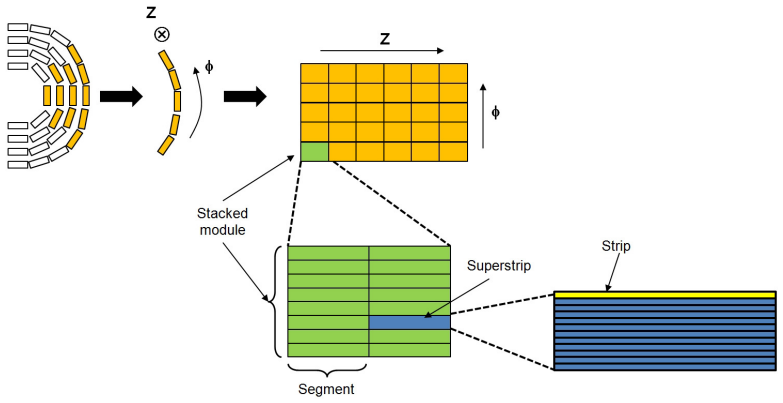


FIGURE 6.4 – Définition géométrique d'un superstrip.

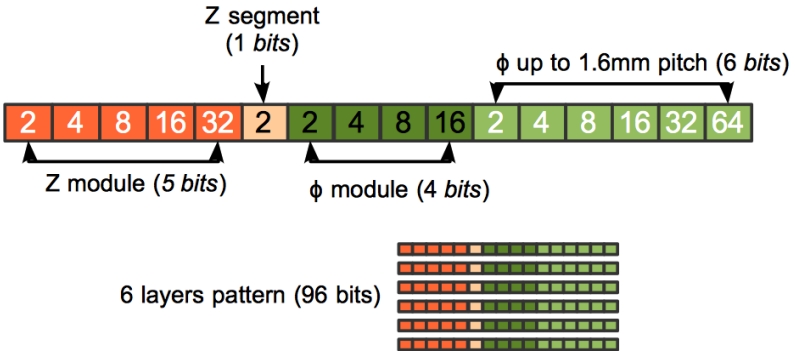


FIGURE 6.5 – Encodage d'un superstrip et représentation d'un motif dans le tonneau (6 couches).

Un exemple d'encodage est donné sur la Fig. 6.5 pour un module du tonneau. Chaque superstrip sur un layer/disque donné est défini par une adresse unique.Pour chacune des coordonnées à encoder, le nombre de bits nécessaire dépend simplement de la granularité du superstrip. Un premier ensemble de bits fournit le numéro de module le long de la coordonnée correspondante : 5 bits pour Z (on peut avoir jusqu'à 24 modules en Z dans une tour du tonneau) et 4 pour ϕ (jusqu'à 15 modules par tour en la couche la plus externe). Ensuite, une deuxième série de bits fournit la position du superstrip dans le module. La Fig. 6.4 montre un module 2S (en vert), divisé en 2 segments en Z et 1024 strips en ϕ. Par conséquent, pour décrire toutes les positions, il suffit d'un bit pour Z et de 10 bits pour ϕ. Cependant, la taille minimale pour un superstrip en ϕ étant de 16 strips, seuls 6 bits sont nécessaires. La même stratégie permet facilement de définir un superstrip dans les bouchons ou dans les modules tiltés.

L'encodage du superstrip peut tout à fait différer d'une couche à l'autre d'un secteur. Nous verrons par la suite qu'il peut être par exemple très intéressant d'augmenter la taille des superstrips lorsque le module est plus éloigné de l'axe du faisceau.

6.1.4/ UTILISATION DE BITS TERNAIRES

Comme mentionné précédemment, la taille d'une banque est étroitement liée à la granularité du superstrip. En pratique, si l'on divise par 2 la largeur des motifs, on multiplie par 4 la taille de la banque. Or plus les motifs seront fins, meilleure sera la pureté de la selection. Mais d'un autre côté, on veut conserver une taille de banque raisonnable. C'est afin de pallier ce problème que le concept de motif de résolution variable a été introduit [45].

FIGURE 6.6 – Définition d'un motif de résolution variable à partir de motifs de haute résolution.

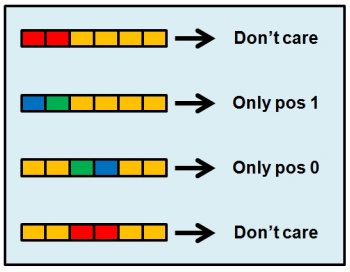


FIGURE 6.7 – Conversion du motif en bits ternaires (cas ou 1 seul bit ternaire est utilisé).

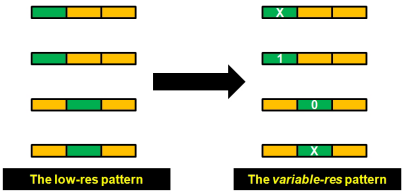


FIGURE 6.8 – Comparaison du motif de résolution variable avec celui de basse résolution.

Pour générer une banque de résolution variable, on commence par créer 2 banques : une banque haute résolution, et une banque basse résolution. Un motif basse résolution a une largeur supérieure d'un facteur 2^N où N est compris entre 1 et 3, par rapport au motif de haute résolution. Ensuite, comme indiqué sur la figure 6.6, les motifs haute résolution appartenant au même motif basse résolution sont fusionnés. Le motif de résolution variable est créé via la procédure illustrée à la Fig. 6.7. Si les deux superstrips haute résolution sont touchés, nous pouvons négliger le bit le moins significatif du superstrip, nous lui donnons alors la valeur DC (pour Don't Care) dans le motif, sinon nous utilisons la valeur du bit haute résolution. Le bit du superstrip peut donc prendre trois valeurs différentes, on parle de bit ternaire, ou DC bit. En pratique, un bit ternaire est formé de deux bits classiques. Cela signifie que chaque fois que l'on ajoute un bit DC sur un superstrip, on perd un bit sur la taille de celui-ci.

Le principal avantage de cette technique est pointé sur la Fig. 6.8 : au lieu d'avoir deux motifs haute résolution dans la banque, on a un seul motif utilisant partiellement la haute résolution. Pour le prix d'une banque basse résolution, on a donc partiellement accès à la haute résolution. Bien entendu, tout cela a un coût, et dès que c'est la valeur DC du bit

ternaire qui est utilisée, on retombe sur la basse résolution. Il faut donc s'attendre à un taux de faux plus élevé avec la banque à résolution variable qu'avec une banque haute résolution. Mais on vérifie que ce taux reste bien meilleur qu'avec une banque à basse résolution.

Cette procédure peut bien sûr être généralisée en ajoutant d'autres bits ternaires. Il n'y a en effet aucune limitation du nombre de bits ternaires par superstrip dans les puces AM (si ce n'est la taille du superstrip). Cependant, à partir de deux bits ternaires, il devient impossible de couvrir toutes les configurations géométriques possibles. Il faut donc choisir celles qui sont les plus probables. La figure 6.9 montre par exemple le codage utilisé dans le cas de deux bits ternaires. Dans ces cas, on dispose de seulement 9 codes pour 15 configurations possibles. On voit qu'il n'y a aucun gain par rapport à la résolution la plus basse pour 7 configurations, en particulier pour deux configurations ne comportant que deux superstrips atteints (dans la colonne du milieu), qui sont justement les plus probables en pratique.

Un encodage plus physique, présenté sur la Fig. 6.10, est utilisé. Il est obtenu en utilisant un codage en miroir de la position du superstrip (codage de Gray). Les configurations aboutissant à deux bits DC (basse résolution) sont les mêmes, à l'exception de deux configurations à deux superstrips pouvant survenir assez souvent dans le détecteur, qui sont échangées avec deux configurations beaucoup moins fréquentes. On utilise donc souvent le bit DC synonyme de basse résolution, mais pour des configurations qui sont en réalité peu fréquentes.

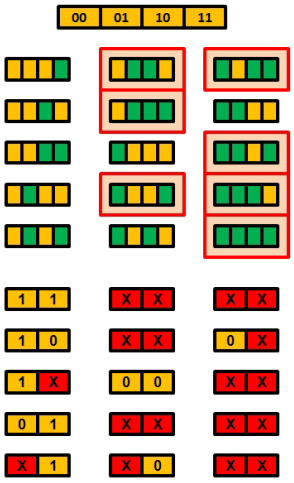


FIGURE 6.9 – Codage de 15 positions avec 2 bits ternaires (encodage classique).

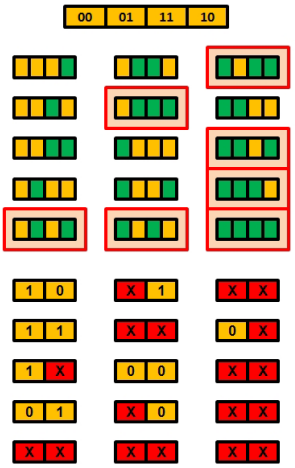


FIGURE 6.10 – Codage de 15 positions avec 2 bits ternaires (encodage Gray).

Avec trois bits DC, le phénomène s'accentue : 27 encodages disponibles pour 255 combinaisons. Cela signifie donc que nous ne pourrons utiliser des informations à plus haute résolution que pour 10% des superstrips. Cependant, une banque à 3DC bits restera intéressante en terme de réduction de la taille de la banque. Au-delà de 3 DC bits la situation commence à devenir moins avantageuse car on retombe dans une configuration 0 DC dans la majeure partie des cas.

Les chips AM étant configurables, il est possible d'utiliser un nombre différent de bits DC selon la couche considérée, cela devient particulièrement important quand la taille de

base du superstrip dépend du layer. Nous y reviendrons dans le chapitre suivant.

6.2/ DÉVELOPPEMENT D'UN ÉMULATEUR DE CHIP AM DANS CMS

La génération du motif optimal d'une banque met en jeu un grand nombre de paramètres. Afin d'étudier l'effet de chacun de ces paramètres, il est primordial de disposer d'outils de simulation adaptés et simples d'utilisation. Ces outils n'étant pas disponibles dans CMS lorsque nous avons rejoint cette activité, notre première contribution a consisté à mettre en œuvre une émulation logicielle complète de la reconstruction AM [1]. Elle est constituée de deux composants :

— **Un générateur de banque**, qui construit des motifs à partir de données simulées dans le trajectographe (généralement des muons, plus rarement des électrons). Les propriétés des motifs (superstrips, bits ternaires, nombre de couches,...) sont entièrement configurables à l'initialisation du générateur.

— **Un émulateur de reconnaissance de motifs**, permettant d'obtenir la liste des motifs activés par un événement quelconque. Ce composant a été développé tout d'abord sous la forme d'un code C++ indépendant, puis il a été porté dans l'environnement de simulation de CMS : CMSSW. Tout comme les chips AM classiques, ce code est indépendant de CMS et peut potentiellement être utilisé pour d'autres applications de reconnaissances rapides ou pour développer de nouvelles générations de chip AM (voir chapitre suivant).

6.2.1/ GÉNÉRATEUR DE BANQUE

6.2.1.1/ PRINCIPE

La génération de la banque de motifs est effectuée à l'aide de très grands échantillons (plusieurs milliards) de muons dont le parcours dans le trajectographe est préalablement simulé. On génère uniquement des particules primaires d'impulsion transverse supérieure à $2\ GeV/c$, ce qui est légèrement inférieur à la valeur du cahier des charges. Ceci permet d'évaluer plus précisément l'efficacité de la méthode dans la zone de transition en impulsion.

Pour chaque secteur de trigger nous créons un échantillon de bonnes particules susceptibles d'être utilisables pour la banque de ce secteur. Un bonne particule induit au moins un stub dans le secteur dans chaque couche qu'il traverse.

Le générateur se base sur cet échantillon de quelques dizaines de millions de particules plutôt que sur l'échantillon global, afin d'accélérer le processus. L'élaboration de la banque suit alors la procédure itérative suivante :

— On définit les motifs pour N bonnes particules, puis on les ajoute à la banque. Par construction, celle-ci ne peut pas contenir 2 fois le même motif, si bien que si celui obtenu est déjà dans la banque, il n'est pas ajouté de nouveau (on conserve néanmoins le nombre de fois où il a été obtenu, ce qui permet de définir sa popularité).

1. Développer cette émulation est, qui plus est, le meilleur moyen de comprendre en profondeur le fonctionnement du chip AM

— A la fin d'une itération, on dispose du nombre de nouveaux motifs effectivement ajouté dans la banque, dénoté N_p
— On peut alors calculer le taux de couverture de la banque pour cette itération :

$$coverage = 1 - \frac{N_p}{N} \tag{6.1}$$

— On réitère la procédure jusqu'à obtenir un taux de couverture satisfaisant, généralement 90%.

Une fois le seuil atteint, la procédure est stoppée et le programme prépare les motifs avec bits ternaires à partir des modèles haute résolution. On sauvegarde ensuite les motifs de résolution variable obtenus et un nouveau taux de couverture est calculé. Tout comme le chip AM réel, le générateur permet de prendre en charge jusqu'à 3 DC bits par superstrips pour une couche donnée, et un nombre variable de DC bits selon les couches.

L'utilisation des DC bits augmente mécaniquement le taux de couverture final. En stoppant la croissance de la banque haute résolution à 90% on obtient une banque de résolution variable avec un taux largement supérieur à 95%. L'inefficacité restante est compensée lors de l'identification en autorisant l'activation de motifs incomplets (on tolère un stub manquant sur les 6 ou 7 que contient le motif).

6.2.1.2/ DÉFINITION DE L'ÉCHANTILLON DE DONNÉES D'ENTRAINEMENT

Pour la génération de l'échantillon d'entrainement, il est important d'utiliser une densité spectrale en impulsion correcte afin d'assurer un taux de couverture homogène sur l'ensemble de l'espace de phase. En effet, les particules de faible impulsion ayant une courbure plus importante, elles vont par construction engendrer beaucoup plus de motifs que les particules de haut p_T. Si nous utilisons un spectre plat en p_T pour nos échantillons, la couverture à haut p_T augmentera donc rapidement et faussera la valeur moyenne du taux de couverture (Fig. 6.11 a).

On contourne ce problème en utilisant un spectre plat en $\frac{1}{p_T}$. Comme nous avons plus de pistes à bas p_T, les motifs correspondants sont créés beaucoup plus rapidement. Cependant, cette stratégie nous fait tomber dans l'excès inverse, avec une zone mal couverte à haut p_T (Fig. 6.11 b). La solution que nous avons finalement retenue pour régler ce dernier problème fut de créer 3 banques indépendantes sur des plages d'impulsion distinctes ([2,5], [5,20], [20,100]). Nous les fusionnons ensuite en une seule banque : cela garantit une bonne couverture de l'ensemble du spectre en impulsion (Fig. 6.11 c). Bien entendu, cette fusion est exclusive et un motif présent dans deux sous-banques ne sera compté qu'une fois dans la banque globale.

Une fois la banque terminée, il est possible de la tronquer à une taille donnée, afin d'émuler de manière rigoureuse la capacité d'un ou plusieurs chips AM. A l'intérieur de l'ASIC il est possible d'ordonner les motifs. Au cours de notre étude nous avons évalué 2 types de classements : par impulsion ou par popularité. L'impulsion d'un motif est l'impulsion moyenne des traces qui ont été utilisées pour le produire, sa popularité est le nombre de traces qui l'ont activé. Ces 2 classements ont chacun leurs avantages et leurs inconvénients, mais dans la suite, et sauf mention contraire, nous utilisons un classement par popularité, car il permet de conserver naturellement un taux de couverture élevé même avec une banque tronquée.

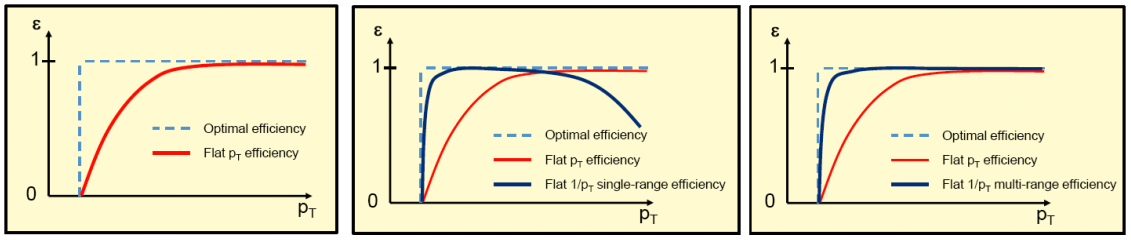


FIGURE 6.11 – Efficacité d'une banque produite avec (a) un spectre plat en p_T, (b) un spectre plat en $1/p_T$, (c) 3 zones à spectre plat en $1/p_T$

6.2.2/ ÉMULATEUR

Le but de l'émulateur est d'obtenir la liste des motifs activés par un événement donné. Les ingrédients de départ sont donc une banque et une liste de stubs.

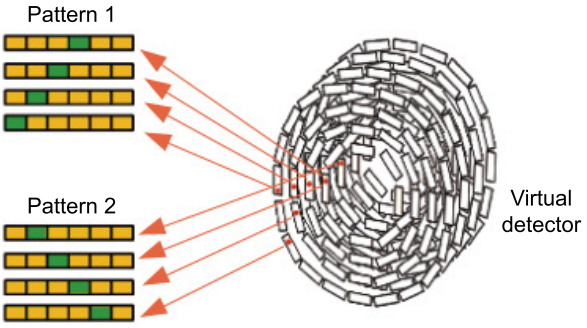


FIGURE 6.12 – Chaque superstrip de la banque pointe vers un superstrip du détecteur virtuel.

L'algorithme le plus simple à implémenter consisterait à vérifier pour chaque stub s'il est contenu ou non dans chaque motif. Mais cela conduirait à un nombre total d'itérations de $N_{stubs} \cdot N_{motifs}$, ce qui est clairement un problème pour les événements lourds du HL-LHC. De plus, cette solution très inefficace ne refléterait pas le comportement de la puce AM.

La solution que nous avons implémentée utilise une représentation virtuelle du détecteur contenant tous les superstrips. Nous avons vu précédemment qu'un motif est composé d'un superstrip par couche. Pour chacun de ces superstrips, on créé un pointeur sur le superstrip virtuel correspondant (voir Fig. 6.12).

Si plusieurs motifs utilisent le même superstrip, leurs pointeurs seront liés au même superstrip virtuel. En conséquence, l'activation d'un seul superstrip virtuel est immédiatement visible par tous les motifs qui le référencent. La procédure de correspondance est ensuite effectuée comme suit :

— Pour chaque stub, nous activons le superstrip correspondant dans le détecteur virtuel.
— Pour chaque motif, nous comptons le nombre de superstrips activés pour celui-ci dans le détecteur virtuel. Si nous sommes au dessus d'un seuil prédéfini (généralement on tolère l'absence d'un superstrip) : le motif est activé.

Le nombre d'étapes devient alors $N_{stubs} + N_{patterns}$. $N_{patterns}$ étant beaucoup plus grand que N_{stubs} (des millions de patterns par rapport à des milliers de stubs), le nombre de stubs de l'événement devient négligeable pour le temps de calcul qui devient linéaire avec le nombre de patterns. Comme dans la puce AM, le temps de traitement de l'émulateur est alors identique pour tous les types d'événements.

Afin de tester le potentiel des mémoires associatives dans CMS, cet émulateur a été porté dans CMSSW. Il a également été très largement documenté afin de permettre à l'ensemble des groupes impliqués dans le projet de réaliser leur études (voir par exemple le tutoriel suivant [46]). Comme nous le verrons par la suite, nous avons pu vérifier que cet émulateur reproduisait au bit près le comportement des chips AM actuel.

6.3/ Performance de l'algorithme dans CMS

6.3.1/ Optimisation des banques

Le but de l'étape d'identification est de fournir l'environnement le plus propre possible pour l'étape suivante d'ajustement des traces. Nous avons vu précédemment que seule une petite fraction des stubs provient des particules que nous recherchons. Il s'agit d'augmenter sensiblement cette fraction.

La procédure d'optimisation des banques se déroule en deux étapes : optimisation par secteur, puis évaluation de la performance pour le détecteur entier.

Les secteurs de trigger utilisés sont divisés en 3 catégories selon le domaine couvert en pseudo-rapidité : tonneau, bouchon, et hybride. Pour chacune de ces catégories il s'agit de définir la meilleure banque possible. Les paramètres sont bien évidemment différents pour les 3 types de secteurs, et l'optimisation va nous amener à des banques très différentes. Les motifs dépendant de plusieurs paramètres (largeur, nombre de DC bits,....), l'espace des paramètres à tester est très vaste. Les résultats qui sont présentés ici sont le fruit d'une longue campagne de travail menée par plusieurs groupes, mais de nombreuses améliorations sont encore possibles. Le lecteur intéressé par ces développements pourra par exemple consulter le document de synthèse suivant [47].

Nous présentons dans cette section les performances obtenues avec l'ancienne géométrie du tracker (sans modules tiltés). La géométrie n'a en effet été introduite que fin 2016, juste avant l'évaluation officielle des différentes approches proposées pour CMS. C'est donc avec la configuration plate que la majeure partie des optimisations a été effectuée jusqu'en 2017, date à laquelle nous nous sommes retirés du projet. Néanmoins, nous avons adapté le filtrage AM à la géométrie finale et les résultats obtenus sont en accord avec ceux obtenus pour la géométrie plate [48].

Les premières études ont été effectuées avec des banques contenant des motifs de largeur constante (Figure 6.13 gauche). Mais ce type de motif engendrait des banques de taille très importante, et ne prenait pas en compte l'augmentation de dispersion des traces avec le rayon du stub. Plus la quantité de matière traversée est importante, plus la trace a en effet des chances de s'éloigner de la trajectoire idéale à cause de phénomènes de diffusion. Ce phénomène est mieux pris en compte via l'utilisation de motifs projectifs (Figure 6.13 droite), pour lesquels la largeur du superstrip augmente avec la distance par rapport à l'axe du faisceau.

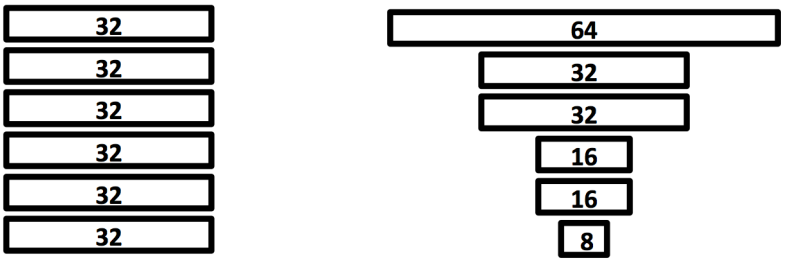


FIGURE 6.13 – Motif à largeur constante (gauche) et motif projectif (droite)

De la même manière, nous avons constaté que les performances des banques était meilleures lorsque le nombre de bits ternaires utilisables dans chacune des couches était variable.

Les banques finales dont sont issus les résultats présentés dans la suite sont caractérisables par la largeur des superstrips et le nombre de DC bits utilisés dans chaque couche. Ces informations sont résumées dans les Figs. 6.14, 6.15, and 6.16, pour les trois types de tours.

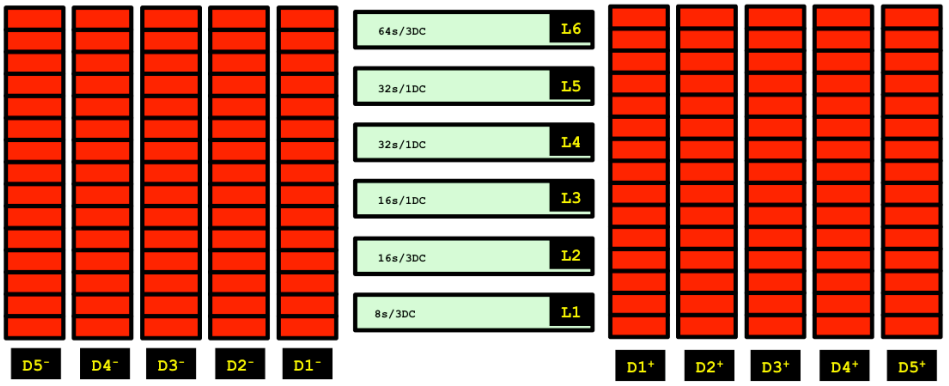


FIGURE 6.14 – Propriétés de la banque tonneau. Les parties du détecteur impliquées sont en vert. La taille correspondante des superstrips (en nombre de strips) et les nombres de bits ternaires maximum sont indiqués.

La taille des banques est limitée à 1M de motifs (2 chips AM) pour les secteurs tonneau et bouchon, et 1.5M de motifs (3 chips AM) pour les secteurs hybrides.

Au total, dans cette configuration, l'approche AM du trajectographe utiliserait 56M de motifs, soit environ 112 chips AM. Avec le taux de multiplexage en temps de 40 présenté auparavant, cela représente au total 4480 chips pour l'ensemble du trajectographe. C'est un chiffre tout à fait normal pour un système de cette ampleur.

6.3.2/ DÉFINITION DE L'EFFICACITÉ ET DU TAUX DE FAUX

Afin d'évaluer la performance d'une banque de motifs, plusieurs paramètres doivent être étudiés : l'efficacité, le taux de faux, la pureté des stubs en sortie, le nombre moyen de motifs activés ainsi que la limite haute (valeur englobant 95% des événements). Le

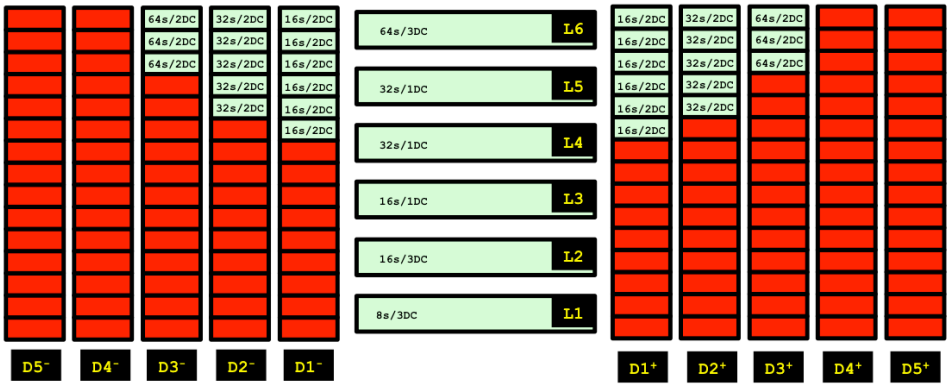


FIGURE 6.15 – Propriétés de la banque hybride. Les parties du détecteur impliquées sont en vert. La taille correspondante des superstrips (en nombre de strips) et les nombres de bits ternaires maximum sont indiqués.

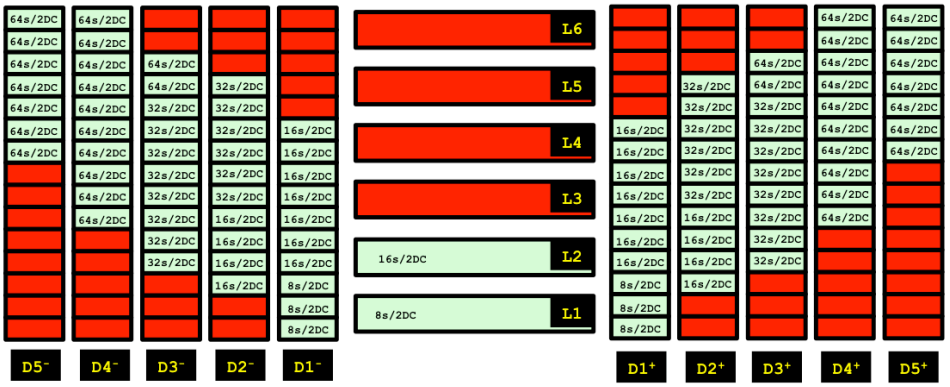


FIGURE 6.16 – Propriétés de la banque bouchon. Les parties du détecteur impliquées sont en vert. La taille correspondante des superstrips (en nombre de strips) et les nombres de bits ternaires maximum sont indiqués.

dernier paramètre est lié au caractère purement électronique de la procédure que l'on veut mettre en place. Il est important de connaitre la taille maximale des événements attendus, afin d'estimer dans quelle mesure le système sera capable de les assimiler et quelles seront les pertes dynamiques.

L'efficacité de la reconstruction de traces après mémoires associatives est définie de la manière suivante :

$$\epsilon^{SR \to AM} = \epsilon^{SR} \cdot \epsilon^{ST} \cdot \epsilon^{TW} \cdot \epsilon^{AM} \tag{6.2}$$

où :

— ϵ^{SR} : efficacité de reconstruction des stubs. Proportion de bonnes particules induisant au moins 4 stubs dans 4 couches distinctes du trajectographe. Cette efficacité correspond à l'acceptance que l'on a définie dans la Section 3.3.2.

— ϵ^{ST} : efficacité de transmission des stubs. Proportion des traces passant l'étape précédente pour lesquelles au moins 4 stubs dans 4 couches distinctes subsistent après prise en compte des pertes dues au frontend (pertes dans le concentrateur par exemple).

— ϵ^{TW} : efficacité des tours de trigger. Proportion des traces restantes contenues dans au moins une des tours de trigger.
— ϵ^{AM} : efficacité de reconstruction AM. Proportion de traces restantes contenues (4 stubs sur 4 couches) dans au moins 1 des motifs activés.

Ces définitions se basent sur la notion de bonne particule. Une bonne particule est définie par les caractéristiques suivantes :

— $|d_0| < 1cm$
— $|z_0| < 15cm$
— $|\eta| < 2.4$
— $p_T > 3GeV/c$
— Au moins 1 stub induit.

Comme nous l'avons déjà évoqué dans la partie 3.3.2, ϵ^{SR} et le dénominateur dépendent fortement du paramètrage des fenêtres de sélection des stubs utilisées. Avec une coupure à 2.5 strips dans la couche la plus interne du tonneau, plus de particules induiront un stub par rapport à une coupure à 1.5. D'autre part, comme la plupart de ces particules vont se désintégrer plus tôt, cela va contribuer à abaisser ϵ^{SR}. Dans l'absolu, chaque estimation de ce paramètre devrait donc être en principe accompagné de l'étalonnage utilisé pour construire les stubs.

Les inefficacités de transmission ne sont actuellement pas prises totalement en compte dans la simulation officielle de CMS, en particulier les pertes dues au concentrateur. Pour modéliser ces pertes de manière rigoureuse il faut en effet conserver l'information de 8 événements consécutifs, ce qui va à l'encontre des règles d'implémentation de CMSSW. Dans le même temps, nous avons vu dans la section 4.2.3 qu'en optimisant les capacités de transmission du concentrateur, ces inefficacités pouvaient être fortement réduites. Nous considérerons donc que $\epsilon^{ST} = 1$ dans la suite de ce document, sans oublier qu'une méthode d'estimation de ce paramètre devra être développée à terme. Tout comme ϵ^{SR}, ce paramètre va dépendre des fenêtres de sélection.

On remarquera que ces 2 paramètres sont totalement indépendants de la reconstruction de traces proprement dite, on parle donc plutôt d'efficacité intrinsèque au détecteur pour les caractériser.

La méthode de reconstruction va influer sur les deux derniers termes, et plus particulièrement ϵ^{AM}. Il est en effet relativement simple de définir les tours de trigger de façon à obtenir $\epsilon^{TW} \approx 1$ [2].

Pour le taux de faux, on se base sur le contenu du motif activé. Celui-ci est associé à une trace si plus de 4 stubs de la trace (sur 4 couches distinctes) sont contenus dans le motif. En se basant sur cette définition, un motif activé est considéré comme faux si :

— Il n'est associé à aucune trace
— Il est associé à une trace qui ne provient pas d'une bonne particule (en se référant aux critères définis plus haut).

On voit que cette définition ne prend pas en compte la pureté des stubs dans le motif. Les motifs étant relativement larges, il est en effet possible que celui-ci contienne des stubs n'appartenant pas à la trace considérée. Un motif vrai ne sera donc pas forcément un motif pur. Nous verrons dans le chapitre suivant comment ce paramètre influe sur la suite de l'algorithme de reconstruction.

2. Définir le câblage entre les modules de ces tours et le backend correspondant est par contre assez complexe et sort du cadre de ce document.

6.3.3/ RÉSULTATS

6.3.3.1/ EFFICACITÉ

L'efficacité a été mesurée avec des échantillons simples (muons et électrons) sans empilement et des échantillons de paires $t\bar{t}$ incluant les effets d'empilements avec des valeurs moyennes de 140 et 200.

La Table 6.1 présente les résultats obtenus pour la configuration de base décrite dans 2 domaines d'impulsion transverse : basse (3 à 8 GeV/c) et haute (8 à 50 GeV/c).

	Muons		Electrons		$t\bar{t}$+PU140		$t\bar{t}$+PU200	
	$3 \leq p_T \leq 8$	$8 \leq p_T \leq 50$	$3 \leq p_T \leq 8$	$8 \leq p_T \leq 50$	$3 \leq p_T \leq 8$	$8 \leq p_T$	$3 \leq p_T \leq 8$	$8 \leq p_T$
ϵ^{SR}	99.9 ± 0.05	$100.0_{-0.05}$	94.9 ± 0.1	98.3 ± 0.05	97.2 ± 0.05	97.8 ± 0.1	97.2 ± 0.05	97.5 ± 0.2
ϵ^{TW}	$100.0_{-0.05}$	$100.0_{-0.05}$	99.9 ± 0.05	99.9 ± 0.05	99.9 ± 0.05	99.9 ± 0.05	99.9 ± 0.05	99.9 ± 0.05
ϵ^{AM}	97.0 ± 0.05	98.4 ± 0.05	76.5 ± 0.1	92.5 ± 0.1	95.5 ± 0.05	98.4 ± 0.1	95.7 ± 0.05	98.5 ± 0.1

TABLE 6.1 – Efficacités d'identification des traces (en %), pour des particules primaires d'impulsion transverse $p_T \geq 3$GeV/c

Les Figures 6.17 et 6.18 présentent l'évolution de l'efficacité dans la zone de bas p_T pour des muons et des électrons respectivement. Les points noirs représentent ϵ^{SR} et les cercles bleus l'efficacité totale $\epsilon^{SR \to AM}$, qui comme on a $\epsilon^{ST} = \epsilon^{TW} \approx 1$ correspond à $\epsilon^{SR} \cdot \epsilon^{AM}$.

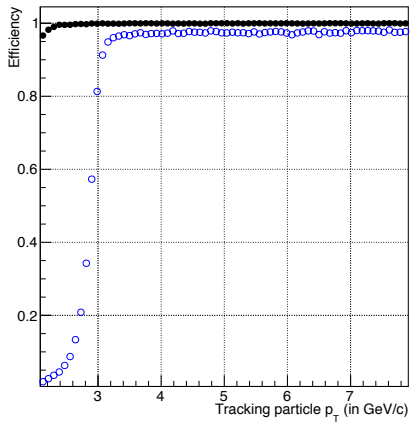


FIGURE 6.17 – Efficacité de reconstruction des traces pour des muons primaires sans empilement

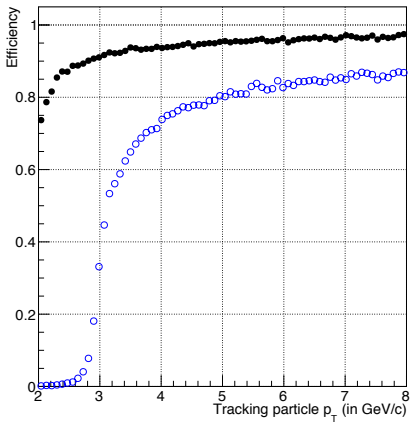


FIGURE 6.18 – Efficacité de reconstruction des traces pour des électrons primaires sans empilement

Comme prévu l'efficacité des muons est bien meilleure, en particulier à bas p_T. D'autre part l'augmentation de l'efficacité totale est très nette autour de 3 GeV/c, avec une efficacité presque nulle à 2 GeV/c, ce qui correspond au comportement attendu des banques AM. C'est un point important car dans le cas contraire un nombre important de traces de bas p_T risquerait de franchir l'étape d'identification, augmentant ainsi dangereusement le taux d'occupation pour les étapes suivantes.

La performance pour les muons seuls, dans la mesure où ils ont été utilisés pour construire les banques, correspond au meilleur résultat possible. On voit que pour les

événements contenant de l'empilement l'efficacité à haut p_T est comparable à celle des muons. Cela confirme le fait que l'empilement n'a pas d'impact sur ce paramètre. Un motif activé par une trace seule sera également activé si en plus de cette trace des signaux dus à l'empilement apparaissent.

Le fait que le plateau d'efficacité pour les muons soit en dessous de 99% est dû à la définition des banques dans les secteurs bouchons. Historiquement les banques de la géométrie plate étaient basées sur des tours de déclenchement étendues jusqu'à $|\eta| = 2.3$, au lieu de 2.4. Ce problème a été corrigé par la suite dans la géométrie inclinée, mais dans les résultats présentés ici on observe une baisse importante de l'efficacité au-delà de $|\eta| = 2.3$ qui impacte donc le niveau général du plateau.

L'efficacité est par contre moins bonne pour les électrons, en particulier à bas p_T. Ce n'est pas vraiment surprenant, les trajectoires des électrons étant beaucoup plus sensibles à la matière des couches traversées. On voit que même l'acceptance est affectée. Pour pallier ce problème nous avons vu dans la section 3.3.2 que l'on peut intervenir au niveau de ϵ^{SR}, en élargissant par exemple les fenêtres de sélection des stubs (mais on augmente les nombre de stubs). On peut aussi envisager d'améliorer ϵ^{AM} en utilisant des banques de motifs générées à partir d'électrons.

6.3.3.2/ TAUX DE FAUX ET NOMBRE DE MOTIFS ACTIVÉS

Les taux de motifs moyens pour les événements $PU + t\bar{t}$ sont présentés dans le Tableau 6.2, ainsi que les taux satisfaits par 95% des événements. Le taux de faux est également fourni. Les résultats sont présentés pour les 3 types de tour.

	Nombre moyen de motifs activés		Taux de faux (en %)	
Empilement moyen	140	200	140	200
Tour tonneau	11.0 (41)	19.6 (62)	24.4	52.4
Tour hybride	18.4 (63)	39.2 (109)	54.6	76.4
Tour bouchon	16.0 (54)	29.2 (84)	46.6	67.8

TABLE 6.2 – Nombre moyens de motifs activés et taux de faux pour des événements $t\bar{t}$. Pour le taux de sortie la limite à 95% est donnée entre parenthèses.

Comme nous utilisons un seuil minimal de 4 stubs dans certaines zones des tours hybrides et, dans une moindre mesure, dans les tours des bouchons, le taux de faux dans ces secteurs est plus important que dans les bouchons où le seuil minimal d'activation est à 5.

La dépendance à l'empilement est également plus importante dans les secteurs contenant des modules des disques. Ceci vient du fait que la résolution des stubs à bas p_T est beaucoup moins bonne dans ces zones (cf Chapitre 3). Il est donc beaucoup plus probable d'activer un motif pour une trace de $p_T < 3\ GeV$. Nous verrons plus loin que ces traces primaires de bas p_T constituent effectivement une proportion importante des fausses routes dans ces secteurs.

Concernant le taux en sortie, on voit qu'il est inférieur à 110 routes par tour et par événement pour 95 % des événements, quelle que soit la valeur d'empilement. C'est compatible avec l'objectif de 100 qui avait été fixé au début de ce travail d'optimisation. Les pertes dynamiques dues à un nombre trop important de motifs activés seraient sous

contrôle avec une telle configuration de banques.

6.3.3.3/ Pureté des stubs après l'étape AM

L'impact du filtrage AM sur la pureté des stubs est significatif, comme indiqué dans le tableau 6.3. Une fois encore, le pouvoir de filtrage est meilleur dans les tours de type tonneau. Cependant, dans tous les cas, la pureté augmente sensiblement par rapport à la situation avant AM, où la pureté est d'environ 5% (au mieux) dans l'ensemble du détecteur.

	Nombre moyen de stubs par tour		Pureté des stubs (en %)	
Empilement moyen	140	200	140	200
Tonneau	27.7	65.0	43.6	23.3
Hybride	33.6	87.0	32.0	15.6
Bouchon	28.2	63.8	32.7	18.8

TABLE 6.3 – Nombre moyen de stubs et pureté des stubs par tour de déclenchement après le stade AM. Un bon stub est un stub induit par une particule avec $|d_0| < 1cm$ et $p_T > 3GeV/c$

Le nombre de stubs à traiter par la suite est réduit quasiment d'un ordre de grandeur pour un empilement moyen de 140. La réduction est moins importante dans le cas d'un empilement de 200. Si l'efficacité de la méthode AM est indépendante de l'empilement, il n'est par contre pas surprenant de constater que les multiplicités en sortie sont notablement affectées par ce paramètre.

6.3.4/ Bilan

Le potentiel des mémoires associatives dans CMS a pu être évalué de manière complète grace à l'émulation présentée dans ce chapitre. Cette étude nous a permis de confirmer l'intérêt d'une telle approche, et a servi de base au développement d'une procédure globale.

Toutefois, ces résultats ne reflètent qu'une petite partie de l'ensemble des études effectuées, et seule une fraction de l'énorme potentiel des mémoires associatives a pu être exploitée. L'espace de phase est, vous l'aurez compris, très vaste, et la marge d'amélioration significative.

Parmi les pistes qui ont été explorées et que nous n'avons pas évoquées ici, on peut citer les points suivants :

— **Banques triées par** p_T : l'un des gros avantages des chips AM est la possibilité d'extraire les motifs activés dans un ordre prédéfini. Dans l'étude précédente, les routes étaient classées par popularité. Avec un tel ordre, il est possible de réduire considérablement la taille de la banque en supprimant les motifs moins fréquents. L'impact sur l'efficacité est relativement faible. On pourrait aussi imaginer ordonner les motifs par p_T afin d'extraire en priorité les motifs compatibles avec une haute impulsion transverse, ou définir un scénario hybride optimisant l'efficacité avec une taille de banque réduite. Les premières études dans ce sens ont été présentées [49, 50].

— **Banques pour électrons** : l'idée ici est de construire des banques dédiées utilisant des électrons à haut p_T afin de réduire l'inefficacité due à la diffusion multiple. Les motifs activés via ces banques peuvent être marquées en tant que candidats électrons et peuvent donc être dirigés vers un module d'ajustement dédié, avec des critères de sélection plus souples que les autres particules. Cette solution n'a pas encore été testée de manière approfondie.
— **Utilisation de la largeur des stubs** : l'utilisation de cette information pour la reconstruction a été l'objet de nombreux débats. Dans l'approche AM il avait été décidé de ne pas compter sur ce paramètre, dont la robustesse à moyen terme n'est pas démontrée (comment par exemple gérer un strip mort). Les autres approches firent un choix différent (et compréhensible dans leur cas..). Afin de comparer de manière équitable les résultats de l'approche AM avec les autres, il a été décidé de tester l'impact de cette information sur le filtrage AM. Ces études ont démontré que s'il n'était pas forcément très intéressant d'utiliser la largeur dans le chip AM [51, 52], son utilisation pour filtrer les stubs dans les motifs activés pouvait s'avérer très efficace [53, 54]).

Encore une fois, ces exemples ne sont que la partie émergée de l'iceberg. Ils démontrent toutefois clairement la richesse et la souplesse de cette approche.

7

Développement d'un démonstrateur : le projet FastTrack

Nous venons de démontrer la faisabilité d'une approche basée sur les mémoires associatives dans CMS. La suite logique de ces travaux est la mise en oeuvre d'un algorithme complet, et son déploiement dans un démonstrateur. Ce sont les étapes qui sont présentées dans ce dernier chapitre.

7.1/ Introduction et objectifs

Les résultats du chapitre précédent nous ont permis d'initier dès 2013 une collaboration avec le groupe ATLAS du LPNHE, groupe impliqué dans le projet FTK. Cette association s'est concrétisée en 2014 avec le démarrage du projet FastTrack, qui a bénéficié entre 2014 et 2018 d'un financement de l'agence nationale pour la recherche (Projet ANR-13-BS05-0011 [55]).

Ce projet se divisait en 2 volets complémentaires : développement d'une nouvelle génération de puces AM dans une finesse de gravure de 28 nm (LPNHE), et création d'un démonstrateur utilisant ces puces dans le cadre de CMS (IPNL).

Dans CMS, FastTrack s'est inscrit dans une collaboration plus large, avec en particulier l'INFN et Fermilab, dont l'objectif était le développement d'un démonstrateur complet de l'ensemble de la procédure. Nous aborderons ce point en fin de chapitre.

En parallèle au développement de ces démonstrateurs, un algorithme de reconstruction de traces complet a été mis au point afin de completer le plus efficacement possible l'étape de filtrage par les mémoires associatives. Nous allons nous concentrer tout d'abord sur ces travaux, et plus particulièrement sur le module de filtrage intermédiaire développé à l'IP2I.

7.2/ PRÉPARATION DE L'AJUSTEMENT, LE FILTRAGE POST-AM

7.2.1/ PRINCIPE

Une des méthodes les plus prometteuses pour l'ajustement des paramètres des traces est l'analyse en composantes principales (PCA) [56]. Cette technique permet d'établir une relation linéaire entre les coordonnées de la trace et ses caractéristiques intrinsèques. Outre le fait qu'elle offre une résolution optimale, cette méthode a l'immense avantage d'être très simple à mettre en œuvre sur un FPGA. En effet, les paramètres de la trace s'obtiennent à partir des coordonnées des stubs via un seul produit scalaire. C'est extrêmement rapide et efficace. Avantage supplémentaire, il s'agit d'une technique éprouvée, qui a été mise en place avec succès dans CDF [57]

Néanmoins, il s'agit d'une technique très sensible aux impuretés. Il est important de lui fournir en entrée le contenu le plus propre possible, et en particulier exactement 1 stub par couche. Cette condition était satisfaite par construction dans le cas de CDF, en particulier car les motifs étaient très fins. La situation de CMS est sensiblement différente.

Les motifs étant relativement larges, il peuvent contenir plusieurs stubs par couche. Pour un processeur PCA, cela signifie tester toutes les combinaisons possibles. Pour des motifs activés dans des jets de haut p_T, contenant des nombres importants de stubs dans chaque couche, le nombre de combinaisons à tester peut rapidement exploser.

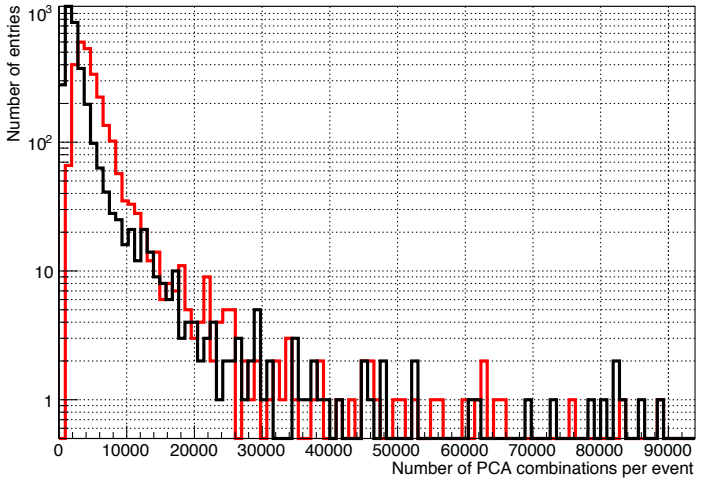


FIGURE 7.1 – Nombre de combinaisons par événement après le filtrage AM, pour des événements $PU140 + t\bar{t}$ (noir) et $PU200 + t\bar{t}$ (rouge)

On voit par exemple sur la Figure 7.1 le nombre total de combinaisons après le filtrage AM, pour des événements standards au HL-LHC. Ce nombre est très important, et si toutes les combinaisons doivent passer par un processeur PCA, même avec plusieurs processeurs fonctionnant en parallèle, cela peut devenir problématique pour la latence totale.

Il est donc important de limiter le nombre de combinaisons à tester en sortie de mémoires associatives. C'est l'objectif du track candidate (TC) builder.

7.2.2/ PRINCIPE DE FONCTIONNEMENT

Le TC builder est un algorithme qui doit être implémenté sur un FPGA. Il doit donc être le plus simple possible. Il est composé d'un nombre prédéfini de modules de filtrage fonctionnant en parallèle : les TC modules.

Le principe de fonctionnement d'un de ces modules est explicité sur la Figure 7.2.

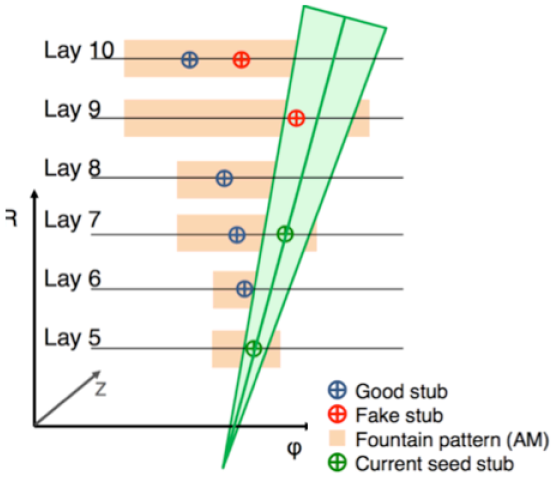


FIGURE 7.2 – Principe de fonctionnement du TC builder

Chaque TC module traite les stubs d'un motif activé. L'idée est de partir d'une graine formée de 2 stubs, et de projeter cette graine vers la couche suivante. Si on trouve un stub dans la zone projetée on poursuit la projection, et ainsi de suite. Afin de simplifier l'algorithme au maximum, la projection est toujours effectuée à partir de la graine de départ. Si l'algorithme se termine avec succès, on obtient un candidat trace contenant exactement un stub par couche, qui peut être envoyé au module effectuant la PCA.

Nous allons décrire brièvement chacune des étapes de ce second filtrage.

7.2.2.1/ GÉNÉRATION DES GRAINES

On définit la graine comme une paire de stubs appartenant à deux couches distinctes. Toutes les graines sont traitées séquentiellement par le TC module. Afin d'optimiser la latence et de libérer le TC module le plus rapidement possible, on stoppe le processus dès que l'on a obtenu une graine menant à un candidat complet (un stub sur chaque couche).

Pour former une graine, on utilise uniquement des stubs issus de modules PS afin d'obtenir la meilleure précision possible pour la projection. Dans les disques, cela signifie que seuls les anneaux les plus internes sont utilisés. Ce choix permet accessoirement de réduire naturellement le nombre de graines, et donc de conserver un temps d'exécution raisonnable.

Les couches choisies dépendent du type de tour de déclenchement. La répartition est la suivante :

— **Tonneau :** les 3 couches internes.

— **Hybride :** les 3 couches internes du tonneau.
— **Bouchon :** les 2 couches les plus internes du tonneau et les 9 premiers anneaux des deux disques les plus proches du point d'interaction.

Cette répartition garantit la présence d'au moins une graine pour chaque motif activé traité. Elle est facilement généralisable à la géométrie tiltée.

7.2.2.2/ PROJECTION ET FILTRAGE

C'est l'étape-clé de l'algorithme. Les projections sont effectuées dans les plans (r, ϕ) et (r, z), où la trajectoire des particules peut être linéarisée. À partir de la trajectoire circulaire de la particule primaire, il est facile de dériver la relation suivante :

$$r = 2R_c \sin(\phi - \phi_0), \tag{7.1}$$

menant à

$$\phi = \phi_0 + \arcsin \frac{r}{2R_c}, \tag{7.2}$$

où R_c est le rayon de courbure, égal à 1,75 m pour la piste 2 GeV/c et à 2,63 m pour la piste 3 GeV/c. Si $\frac{r}{2R_c}$ est suffisamment petit, on peut utiliser l'approximation suivante : menant à

$$\phi = \phi_0 + \frac{r}{2R_c}, \tag{7.3}$$

L'erreur ne dépasse jamais 9 mrad à 2 GeV/c et 2 mrad à 3 Gev/c, ce qui est acceptable pour les fenêtres d'extrapolation utilisées pour la recherche de stubs. De plus, la distance radiale entre la graine et le point d'extrapolation étant toujours inférieure à 1 m, l'erreur commise sera forcément inférieure.

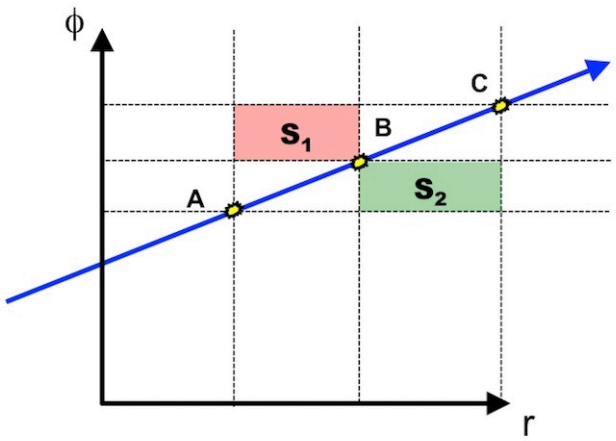


FIGURE 7.3 – Principe de sélection des stubs dans le module TC

Une fois la projection effectuée, la méthode utilisée pour sélectionner les bons stubs est explicitée sur la Fig. 7.3, pour le plan (r, ϕ) (le principe est le même dans le plan (r, z)). Les stubs des graines sont notés A et B sur la figure. Si le stub contenu dans la couche de projection, noté C, appartient à la trajectoire de la particule, il est facile de montrer que les surfaces S_1 et S_2 sont identiques. Par conséquent, le paramètre Δ_ϕ donné par $S_1 - S_2$

devrait être compatible avec 0 pour les stubs sélectionnés. De la même manière, on peut calculer Δ_z.

Cette technique originale permet d'effectuer la sélection sans faire aucune division, ce qui est particulièrement intéressant pour un FPGA. Une technique plus classique aurait nécessité un routage beaucoup plus complexe, ce qui aurait eu un impact sur les performances de l'algorithme.

Pour être sélectionnées, les valeurs Δ_z et Δ_ϕ du stub doivent être inférieures à des fenêtres définies au préalable avec des événements simulés. Si plusieurs stubs sont sélectionnés, on conserve seulement celui qui a le $|\Delta_\phi|$ le plus faible (cette information est beaucoup plus précise que Δ_z), l'idée étant encore une fois de privilégier les traces de plus haute impulsion transverse.

7.2.2.3/ ENREGISTREMENT DU CANDIDAT TRACE

À la fin de la procédure d'extrapolation, on dispose d'une liste de traces candidates (TC). Seuls les candidats qui n'ont pas plus d'un stub manquant sont conservés. Si plusieurs candidats sont obtenus pour un motif donné, seul le meilleur candidat est conservé. La sélection se fait de la manière suivante : si il y en a un, la priorité est donnée au candidat sans aucun stub manquant. Dans le cas contraire, le TC qui contient le plus de stubs dans les modules PS est conservé.

7.2.2.4/ DÉFINITION DES FENÊTRES D'EXTRAPOLATION

L'étalonnage est effectué à l'aide de particules seules (muons ou électrons). Pour chaque type de graine on calcule, pour chaque couche sur laquelle est susceptible de se trouver un stub, la largeur optimale de la fenêtre. La Figure 7.4 montre, par exemple, les valeurs $|\Delta_\phi|$ de la quatrième couche du tonneau (première couche de modules 2S), basées sur les graines construites à partir des 2 couches les plus internes.

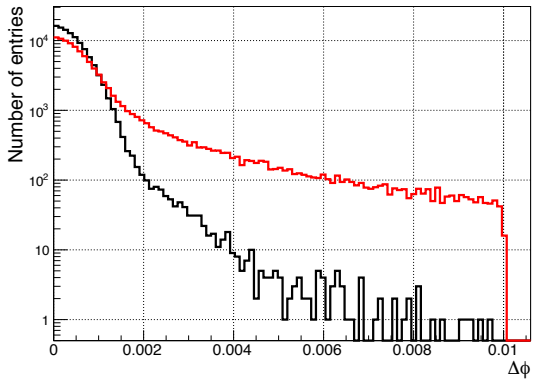


FIGURE 7.4 – Valeurs de $|\Delta_\phi|$ pour des muons (noir) et des électrons (rouge) simulés. Les graines sont issues des couches 1 et 2 du tonneau. La couche de projection est la 4.

Comme prévu, les résidus sont relativement gaussiens pour les muons, avec des queues résiduelles sensiblement plus grandes pour les électrons. Pour les muons, on peut facilement définir une fenêtre d'extrapolation stricte pour laquelle seuls 1% des résidus

seront rejetés. Pour les électrons, on voit qu'une fenêtre plus large devra être définie. Tout comme pour la définition de la fenêtre de sélection des stubs, le choix qui est fait au final est un compromis entre efficacité et pureté.

7.2.3/ PERFORMANCE ET INTEGRATION À L'ALGORITHME COMPLET

7.2.3.1/ AVANT PROPOS

L'étude dont nous présentons les résultats ici constitue la suite de celle introduite lors du chapitre précédent. Les mêmes échantillons de données ont été utilisés, mais cette fois avec un algorithme complet our la reconstruction des traces. Outre l'algorithme de filtrage décrit plus haut, nous avons ajouté les algorithmes d'ajustement des paramètres et de suppression des doublons (une trace est en effet généralement contenue dans plusieurs motifs). Ces algorithmes ont été développés par d'autres groupes, et on peut trouver une description détaillée dans les documents suivants [58, 59].

7.2.3.2/ DEFINITIONS

Comme expliqué précédemment, les fenêtres sont déterminées à l'aide d'un échantillon spécial de muons. Pour chacune des zones d'étalonnage, nous avons choisi des valeurs de Δ_ϕ et Δ_z permettant de conserver 98 % des stubs induits par des muons primaires d'impulsion transverse $p_T \geq 3GeV/c$.

Pour la définition de l'efficacité, nous ajoutons trois termes supplémentaires à la formule définie dans le chapitre précédent :

$$\epsilon = \epsilon^{SR \to AM} \cdot \epsilon^{CB} \cdot \epsilon^{TF} \cdot \epsilon^{DR} \tag{7.4}$$

où :
- $\epsilon^{SR \to AM}$: efficacité jusqu'au filtrage AM, définie dans le chapitre précédent.
- ϵ^{CB} : efficacité de l'algorithme de filtrage. Comme pour les autres paramètres on mesure ici la proportion de traces passant l'étape précédente (ici AM), et qui contiennent au moins 4 stubs dans 4 couches différentes dans au moins une combinaison produite par le TC builder.
- ϵ^{TF} : efficacité d'ajustement des pistes. En principe ce paramètre doit être très proche de 1, on doit pouvoir mesurer les paramètres d'un TC.
- ϵ^{DR} : efficacité de la suppression des doublons.

La définition du taux de faux est légèrement différente de celle utilisée pour la partie AM. En effet, si un motif activé peut contenir un certain nombre de stub ne provenant pas de la trace considérée, ce n'est pas le cas pour un candidat trace ou une trace L1, dont la pureté doit être bien meilleure.

Un candidat ou une trace L1 contenant N stubs est associée à une particule si plus de $N-1$ stubs de ce candidat appartiennent à la particule. Sur la base de cette hypothèse, le candidat est un faux si :
- Il n'est associé à aucune particule.
- Il est associé à une particule qui sort du cahier des charges ($p_T < 3GeV/c, ...$).
- Il contient plus d'un faux stub

Il convient de noter que cette définition est plus restrictive qu'une définition plus classique basée sur la comparaison de paramètres mesurés de la trace avec ceux attendus. En effet, avec notre définition, un candidat trace associé à une vraie particule de 2GeV/c sera considéré comme un faux.

7.2.3.3/ RÉSULTATS

La table 7.1 présente les résultats obtenus, sur la même base que le tableau du chapitre précédent, en incluant cette fois la valeur de l'efficacité totale.

	Muons		Electrons		$t\bar{t}$+PU140		$t\bar{t}$+PU200	
	$3 \leq p_T \leq 8$	$8 \leq p_T \leq 50$	$3 \leq p_T \leq 8$	$8 \leq p_T \leq 50$	$3 \leq p_T \leq 8$	$8 \leq p_T$	$3 \leq p_T \leq 8$	$8 \leq p_T$
ϵ^{CB}	99.0 ± 0.05	99.1 ± 0.05	93.6 ± 0.1	94.6 ± 0.1	97.9 ± 0.05	98.8 ± 0.1	97.4 ± 0.05	98.8 ± 0.1
ϵ^{TF+DR}	99.9 ± 0.05	99.9 ± 0.05	99.8 ± 0.1	99.3 ± 0.1	99.5 ± 0.05	99.5 ± 0.1	99.3 ± 0.05	99.4 ± 0.1
ϵ	96.0 ± 0.05	97.4 ± 0.05	67.8 ± 0.1	85.4 ± 0.1	90.5 ± 0.1	94.5 ± 0.2	89.9 ± 0.1	94.3 ± 0.2

TABLE 7.1 – Efficacité de reconstruction totale, pour des particules primaires de p_T ≥3 GeV/c

On voit que les efficacités obtenues avec l'algorithme de filtrage sont très bonnes, même pour les électrons. Ce n'est pas totalement surprenant car les électrons dont les trajectoires sont fortement déviées sont déjà supprimés au niveau des mémoires associatives. On voit également que cette efficacité dépend peu de l'empilement.

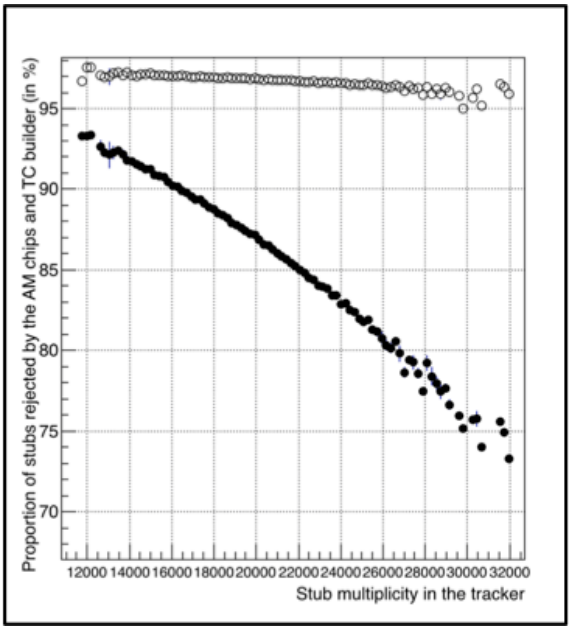


FIGURE 7.5 – Proportion de stubs filtrés, en fonction du nombre total de stubs, après les mémoires associatives (cercles noirs) et après le TC builder (cercles blancs)

.

La Figure 7.5 illustre ce point de manière remarquable. On y voit la proportion de stubs rejetée après les différentes étapes de filtrage, en fonction de la multiplicité en stubs de l'événement, paramètre directement lié à l'empilement moyen.

On voit clairement que le pouvoir de filtrage des mémoires associatives diminue avec

l'empilement, conformément à nos observations précédentes. Au contraire, l'impact de l'empilement sur le module de filtrage est beaucoup moins important, et celui-ci conserve un fort pouvoir de sélection même dans des cas extrêmes.

Comme prévu également, l'efficacité de l'ajustement et de la suppression des doublons est très proche de 1. Ceci est dû au principe de fonctionnement de ces algorithmes, mais aussi à la qualité des informations fournies par le module de filtrage.

Ce point est illustré sur la Fig.7.6, où le rapport entre le nombre de combinaisons et le nombre de candidats filtrés par événement est représenté pour les 2 valeurs d'empilement considérées. On voit que le nombre de candidats est approximativement inférieur d'un ordre de grandeur au nombre de combinaisons et dépend moins de l'empilement moyen.

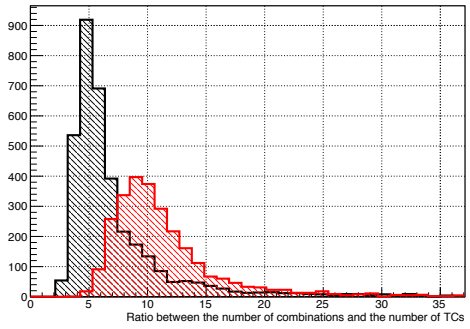


FIGURE 7.6 – Rapport entre le nombre de combinaisons et le nombre de candidats filtrés pour des événements $PU140 + t\bar{t}$ (noir) et $PU200 + t\bar{t}$ (rouge)

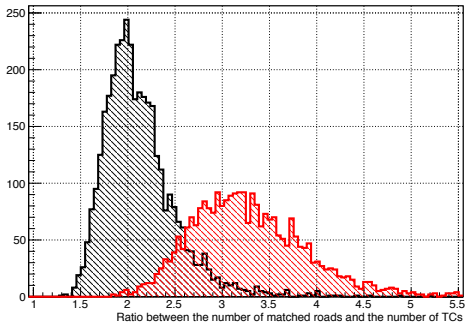


FIGURE 7.7 – Rapport entre le nombre de motifs activés et le nombre de candidats filtrés pour des événements $PU140 + t\bar{t}$ (noir) et $PU200 + t\bar{t}$ (rouge)

La limitation à un maximum de un candidat trace par motif activé est une contrainte forte, mais l'excellent pouvoir de sélection du module de filtrage joue également un rôle important. En effet, comme indiqué sur la Fig.7.7, le module de filtrage supprime en moyenne au moins la moitié des motifs activés, et cette proportion augmente avec l'empilement, sans aucun impact sur l'efficacité, comme nous l'avons déjà vu précédemment. Cela confirme l'indépendance du module de filtrage par rapport à l'empilement.

Les nombres moyens de candidats filtrés par événement et par tour de trigger sont résumés dans le tableau 7.2. Ils confirment l'observation précédente : les taux sont nettement inférieurs à ceux obtenus après AM (voir le tableau 6.2). Autre point important, les multiplicités obtenues pour les différentes tours sont assez équivalentes et la dépendance à l'empilement est beaucoup moins marquée (elle correspond désormais à l'augmentation attendue du nombre de traces).

	Nombre de TC		Taux de faux (en %)		Taux de faux avec coupure à 1.5 GeV/c (en %)	
Empilement moyen	140	200	140	200	140	200
Tonneau	6.0 (24)	7.4 (26)	5.2	8.6	1.8	4.7
Hybride	8.2 (33)	10.3 (37)	9.9	16.9	3.2	9.5
Bouchon	6.9 (28)	8.9 (31)	16.1	21.1	3.1	6.3

TABLE 7.2 – Nombre de candidats filtrés en sortie et taux de faux correspondants. Les limites à 95% sont fournies entre parenthèses.

Les deux dernières colonnes du tableau montrent le taux de faux candidats si on considère les traces provenant de particules de bas p_T comme des bonnes traces. Cela confirme bien que la plupart des faux candidats sont en réalité dus à des particules primaires avec un p_T compris entre 1.5 et 3 GeV/c, qui sont en quelque sorte des faux positifs. Le reste, qui peut être considéré comme le contenu réellement faux, est inférieur à 4% à PU140 et inférieur à 9% à PU200. Comme nous le verrons par la suite, cela est très proche de la pureté finale des traces envoyées au L1.

Les limites à 95% pour les multiplicités des candidats sont rappelées dans le tableau 7.3, et comparées aux multiplicités de combinaisons correspondantes. Ces valeurs confirment l'intérêt du module de filtrage pour le temps de traitement de l'étape d'ajustement.

	Nombre de TC		Nombre de combinaisons	
Empilement moyen	140	200	140	200
Tonneau	24	26	129	261
Hybride	33	37	195	389
Bouchon	28	31	151	267

TABLE 7.3 – Comparison des limites à 95% des multiplicités candidats filtrés et des combinaisons brutes.

Enfin, les multiplicités et la pureté des traces après le fit et la suppression des doublons sont fournis dans le Tableau 7.4. Le nombre de traces transmis au L1 trigger est tout à fait correct, et la pureté très bonne. De plus, comme on peut le voir sur la Fig. 7.8, qui présente le nombre total de traces L1 produites par événement, les queues de distributions sont relativement réduites, confirmant ainsi l'efficacité du module de filtrage. En assurant un très faible taux de faux, on contrôle la qualité et la taille du flux de données envoyé au système de déclenchement L1.

	Nombre de traces envoyées au L1		Taux de faux (en %)		Taux de faux avec coupure à 2GeV/c (en %)	
Empilement moyen	140	200	140	200	140	200
Tonneau	1.2 (5)	1.5 (5)	5.7	9.1	1.6	4.3
Hybride	1.2 (4)	1.6 (5)	10.5	17.9	3.2	9.4
Bouchon	1.0 (4)	1.3 (4)	11.6	16.2	2.9	5.8

TABLE 7.4 – Nombre moyen de traces L1 et taux de faux par tour et par évenement. Les limites à 95% sont fournies entre parenthèses.

Une image valant toujours mieux qu'un long discours, la Fig.7.9 résume de manière claire l'efficacité de la méthode présentée. L'image de gauche montre les stubs contenus dans les bouchons pour un événement à un taux d'empilement d'environ 200. La figure du milieu montre les stubs restants après les mémoires associatives, les traces commencent à apparaitre mais on voit qu'il reste du bruit, surtout dans les zones où l'activité est importante. La figure de droite montre le contenu restant après le module de filtrage. Seules les traces subsistent.

L'algorithme complet fonctionne donc parfaitement en simulation, reste à prouver qu'il fonctionne également sur le démonstrateur.

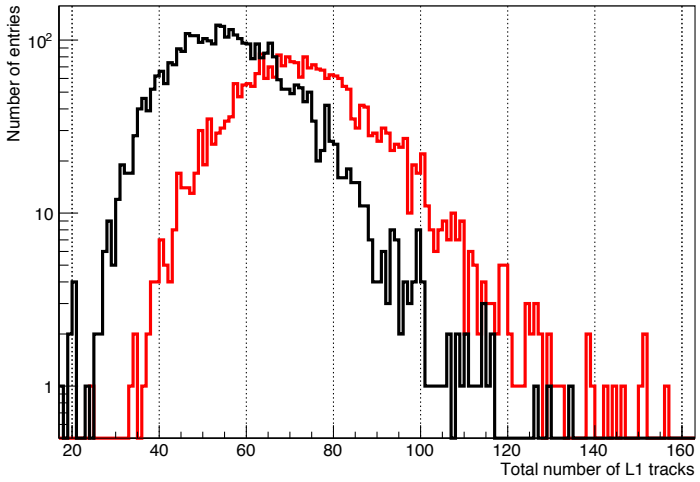


FIGURE 7.8 – Nombre final de pistes L1 par événement, pour les événements PU140+ttbar (ligne noire) et PU200+ttbar (ligne rouge)

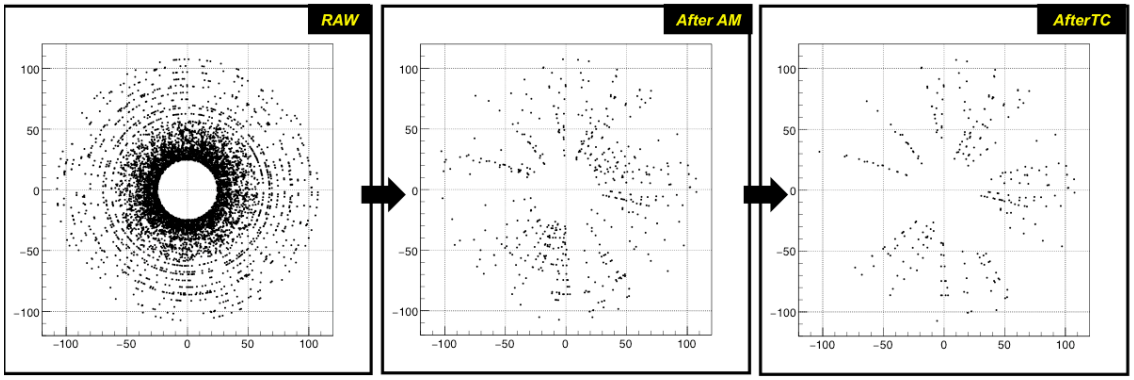


FIGURE 7.9 – Le filtrage en action sur un événement complexe.

7.3/ LE DÉMONSTRATEUR

7.3.1/ DESCRIPTION GÉNÉRALE

Le démonstrateur de poche qui a été mis en place dans le cadre de FastTrack est schématisé sur la Figure 7.10.

Il ne s'agit pas d'un système complet car on effectue uniquement la reconstruction des traces, pas la distribution des données. On part d'un système dans lequel le multiplexage en temps et en espace a déjà été effectué. Cela signifie que les données qui rentrent dans le démonstrateur (cadre nommé FEDs sur la Figure) correspondent aux données d'une tour de trigger, reçues toutes les microsecondes.

Les stubs entrants sont réceptionnés dans une gare de triage (Data Organizer) où ils sont stockés et convertis en superstrips. C'est également le point où les adresses des stubs sont converties en coordonnées globales dans le plan du trajectographe. C'est une étape complexe et très importante, qui doit être réalisée le plus rapidement possible. Nous y reviendrons plus en détails dans la section dédiée à la description du firmware du banc

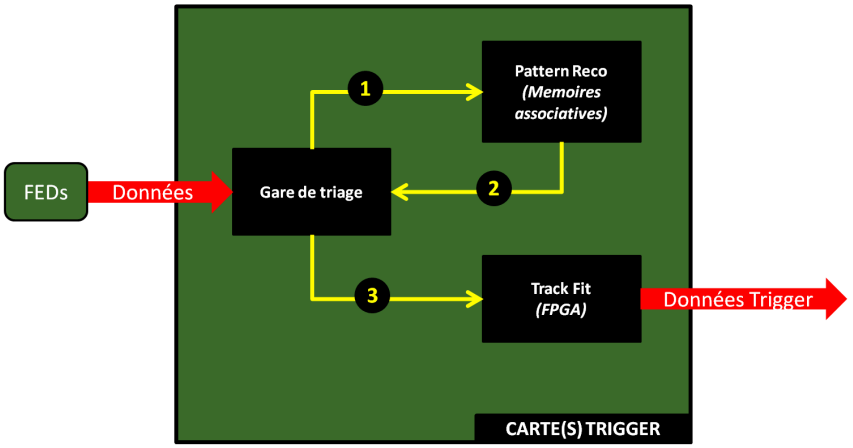


FIGURE 7.10 – Le système FastTrack : principe

de test.

Les superstrips, une fois créés, sont envoyés au chip AM pour identification. Dès que le dernier superstrip est entré dans le chip, les motifs activés retournent à la gare de triage où les coordonnées des stubs contenus dans ces motifs sont récupérés et transmis à l'étape suivante d'estimation des paramètres des traces. Les traces ainsi créées sont ensuite extraites du démonstrateur pour être transmises au système de déclenchement de niveau 1 central.

Avec ce démonstrateur, nous avons tout d'abord cherché à confirmer la faisabilité de la méthode développée et testée par simulation. Ensuite, nous avons pu fournir une première estimation de la latence totale de la procédure, qui correspond au temps compris entre l'arrivée du premier stub et la sortie de la dernière trace, tout en sachant que cette mesure était forcément provisoire.

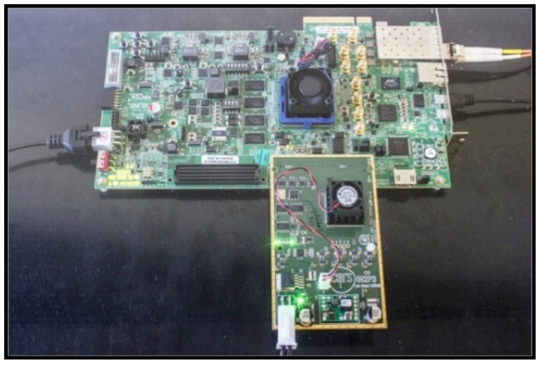


FIGURE 7.11 – Banc de test FastTrack

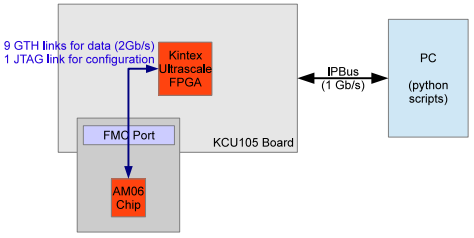


FIGURE 7.12 – Principe de fonctionnement du banc de test

Concrètement, le banc qui a été effectivement développé est présenté sur la Figure 7.11, son principe étant décrit sur la Figure 7.12. Il est basé sur une carte d'évaluation Xilinx KCU105 hébergeant un FPGA Kintex UltraScale. Un des connecteurs FMC de la carte est occupé par une mezzanine FMC développée à l'IPNL hébergeant une puce AM06. Un PC est connecté à la carte KCU105 via une connexion Ethernet optique à 1 Gb/s. Les

communications sont établies à l'aide de scripts Python et du protocole IPBus [60].

La puce AM06 [61] utilisée a été conçue pour le système de reconstruction de traces rapide d'ATLAS. Il s'agit d'un ASIC gravé en 65nm contenant 2^{17} motifs (environ 130000), composés de 8 superstrips chacun. Il intègre 8 lignes d'entrée à 2 Gb/s et une ligne à 2 Gb/s pour la sortie. La fréquence nominale du chip est de 100 MHz. La minimisation de la latence n'était pas le principal problème à l'origine de la conception de cette puce. Il n'est donc pas prévu de répondre à nos besoins en temps de latence mais reste très intéressant pour nos tests car il inclut toutes les fonctionnalités nécessaires et une densité de motif déjà élevée.

7.3.2/ Implémentation de l'algorithme dans le démonstrateur

Le diagramme bloc du firmware implémenté sur le FPGA du démonstrateur est présenté sur la Figure 7.13. Tout comme pour le CIC, celui-ci se divise en sous-blocs (entourés par des tirets) ayant chacun un rôle précis.

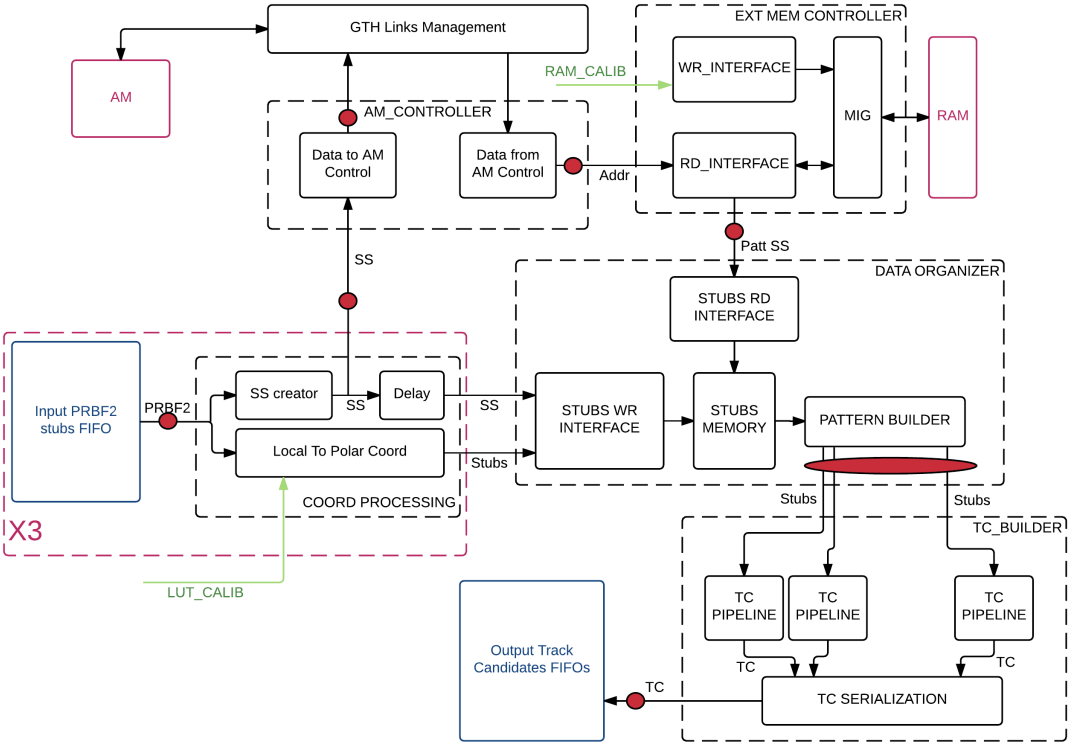


FIGURE 7.13 – Principe de fonctionnement du firmware de test

Les coordonnées locales des stubs (position dans le module, format de sortie du CIC) arrivent tout d'abord dans le bloc *COORD PROCESSING* où ils sont convertis simultanément en 2 formats différents : en superstrips et en coordonnées globales dans le repère du trajectographe. Les superstrips sont envoyés immédiatement au bloc *AM CONTROLLER* qui fait l'interface entre le firmware et le chip AM. Les coordonnées globales sont envoyées plus tard au bloc *DATA ORGANIZER*.

A partir du moment où les superstrips commencent à entrer dans le chip AM on peut commencer à récupérer des motifs activés en sortie. Ces adresses sont transmises

immédiatement au bloc *EXT MEM CONTROLLER* qui permet de récupérer la liste des superstrips du motif (via une table de hachage pré-enregistrée dans la RAM de la carte de test). Les motifs sont alors transmis au *DATA ORGANIZER* sous forme de listes de superstrips.

Le *DATA ORGANIZER* a un rôle fondamental car il permet de remettre en correspondance les stubs et les motifs activés. Pour chaque motif, on récupère à partir des superstrips du motif les coordonnées globales des stubs. On dispose alors, pour chaque motif, de la liste des stubs correspondants, sous forme de coordonnées (r, z, ϕ). On peut donc les envoyer au *TC BUILDER*.

Ce dernier correspond à l'implémentation physique de l'algorithme décrit précédemment. On trouvera une description plus détaillée de cette implémentation dans [62]. Le nombre de TC modules fonctionnant en parallèle dépend du FPGA utilisé. Notre banc de test contenait par exemple 12 modules.

Les candidats traces reconstruits sont ensuite envoyés vers le programme d'analyse permettant d'étudier les résultats obtenus.

7.3.3/ PERFORMANCES ET COMPARAISON DONNÉES/SIMULATION

Afin de comparer au mieux les résultats obtenus avec le TC builder une implémentation réaliste de l'algorithme a été développée. Au lieu d'utiliser des nombres flottants, nous utilisons, comme dans le FPGA, des nombres entiers encodés sur un certain nombre de bits. Des études préliminaires nous ont permis de constater que le meilleur compromis pour minimiser le taux de faux TCs était d'encoder les coordonnées sur 18 bits : 1 bit pour le signe, 7 bits pour la partie entière, et 10 bits pour la partie fractionnaire. Avec cet encodage, le taux de faux pour des événements complexes passe de 1.8 à 2.3%, ce qui reste acceptable.

Bien évidemment, il ne suffit pas d'encoder les nombres de cette manière pour émuler le comportement de l'implémentation FPGA, il faut également effectuer les opérations (additions et multiplications) de la même manière. En procédant ainsi, le code développé doit reproduire au bit près le comportement du TC builder. C'est effectivement ce qui a été constaté sur notre système.

Nous avions déjà pu vérifier auparavant que l'émulation AM était également très réaliste. Nous disposions donc d'une émulation au bit près de la chaîne complète. En pratique, cela signifie que pour des événements simples sans empilement, l'émulateur et le banc de test ont fournit exactement les même motifs matchés et les mêmes candidats traces. Pour des événements plus complexes des petites différences ont été observées, mais celles-ci étaient attendues et provenaient des limitations du système (queues de distribution).

De la même manière nous avons mesuré la latence totale de la procédure, à partir d'un modèle de latence du TC builder que nous avons validé au préalable (la latence des autres blocs est simple à estimer). Nous avons mesuré cette information à partir d'un échantillon de 934 événements $t\bar{t}$ avec un empilement moyen de 140. En moyenne, nous avons mesuré 420 coups d'horloge entre la lecture du premier stub et la lecture du dernier candidat trace. La Figure 7.14 donne des détails sur le temps moyen passé à chaque étape. On note que sur 420 cycles d'horloge, 228 sont liés à la puce AM06 et pourraient donc être grandement améliorés avec une nouvelle génération de puces. Les

cycles d'horloge restants (en dehors de la puce AM) sont inférieurs à 200. La Fig. 7.15 donne la distribution de la latence pour l'ensemble des événements. La fréquence du firmware étant de 125 MHz pour nos tests (pour correspondre à la fréquence du chip AM), on voit que la latence totale est inférieure à 4.2μs pour 95% des événements.

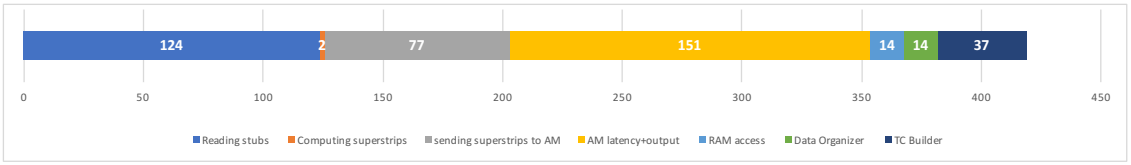


FIGURE 7.14 – Nombre moyen de coups d'horloge passés dans les différentes étapes de la procédure de reconstruction, pour des événements $t\bar{t}$ pour un empilement moyen de 140.

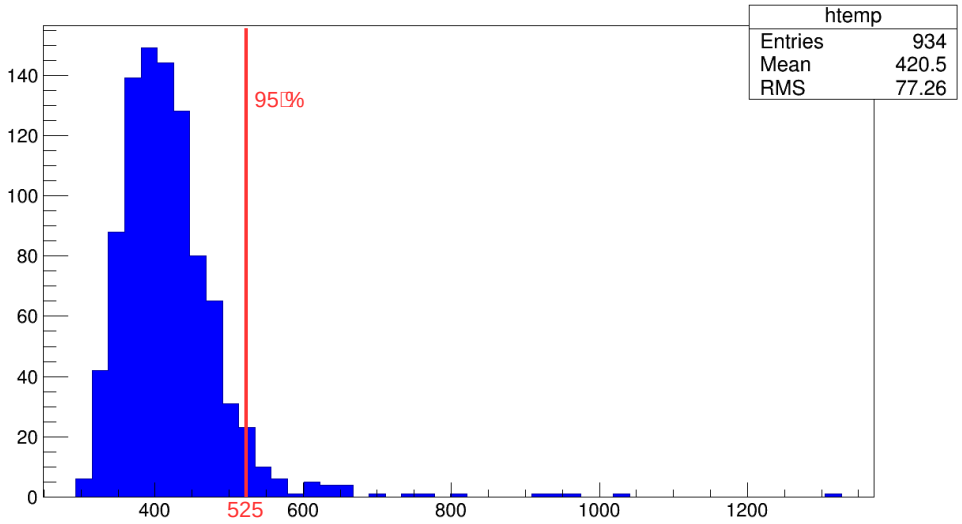


FIGURE 7.15 – Nombre total de coups d'horloge nécessaire au traitement d'événements $t\bar{t}$, pour un empilement moyen de 140, mesuré avec le banc de test.

Si l'on considère qu'une microseconde est nécessaire à l'étape d'ajustement, on peut extrapoler la latence totale de notre démonstrateur à environ 5μs. C'est un résultat tout à fait honorable, surtout lorsque l'on tient compte du fait que le chip AM06 utilisé n'a pas été optimisé pour fonctionner dans des latences réduites.

Ce résultat important a permis de démontrer que moyennant un ASIC plus rapide, une approche basée sur les mémoires associatives dans CMS était tout à fait envisageable. Ce constat a été renforcé avec le développement d'un système plus complet, incluant l'étape de distribution des données.

7.4/ VERS UN SYSTÈME COMPLET

Notre banc de test, qui embarquait uniquement un chip AM, il a servi de base à un système plus complet développé par l'INFN. Ce système comportait un firmware presque identique à celui de notre démonstrateur, mais il utilisait une mezzanine contenant 12 chips, permettant de modéliser un secteur entier. Ce système a pu être complété et testé avec succès en 2017 [63], permettant ainsi de démontrer que notre banc de test pouvait être étendu à un secteur de déclenchement complet.

Le point critique de ces systèmes était clairement le fait que l'ASIC AM06 n'était pas adapté à notre problématique. Cependant, le développement d'ASICs plus rapides était justement un des délivrables du projet FastTrack. Cette étape a été franchie avec succès en 2017 avec la caractérisation du chip AM07, qui fut un des premiers ASICs développés dans le domaine de la physique des hautes énergies dans la technologie 28 nm [64]. Ce succès a ouvert la voie au développement de versions plus complètes (AM08 puis AM09 [65]), développement en cours aujourd'hui pour l'upgrade du FTK d'ATLAS.

Concernant la distribution des données, le fonctionnement du système décrit dans le chapitre 5 avait été mis en place dans un démonstrateur dédié en 2016 [66].

7.5/ BILAN

De 2012 à 2017 un travail énorme a été effectué par la collaboration AM. En 2017, toutes les pièces du puzzle étaient réunies pour pouvoir développer un système complet adapté à CMS. Compte tenu de la complexité du projet, c'est un résultat majeur. Par le biais de notre émulateur, puis du projet FastTrack, nous avons contribué de manière significative à ces développements et surtout à leur aboutissement. La faisabilité de l'approche AM a été démontrée, c'est un fait, mais l'approche n'a pas survécu aux conclusions de la revue de projet de la fin de l'année 2016, qui a tranché en faveur d'une approche purement basée sur des FPGAs.

Il était difficile, voire impossible, de continuer sans le soutien de CMS. Certains des groupes sont partis grossir les rangs des approches FPGA, d'autres, comme nous, ont décidé de s'arrêter là.

D'un point de vue personnel, je n'ai pas souhaité poursuivre l'aventure, et ce pour plusieurs raisons. Tout d'abord, le projet concentrateur nécessitait en 2017 un investissement à plein temps. En tant que responsable, il m'aurait été difficile de mener de front ces 2 projets majeurs. Le concentrateur étant sur le chemin critique, il devenait de facto prioritaire. L'autre raison est plus intime ; quand on a investi plusieurs années dans un projet en étant convaincu qu'il était le meilleur, il est difficile d'aller travailler avec des personnes qui ont, pour certaines, passé plusieurs années à dire le contraire...

En particulier, j'ai trouvé discutable que le système de distribution de données que nous proposions n'ait pas été retenu. Il était en effet tout à fait compatible avec les autres approches, et il était le seul dont le fonctionnement avait été démontré en 2016. Au lieu de cela, il a été décidé de développer un autre système, inexistant à l'époque. Je ne suis pas naïf et je sais que d'autres facteurs doivent être pris en compte dans ce genre de situation, mais les arguments utilisés sont loin de m'avoir convaincu.

Localement, même si le projet n'a pas abouti vers l'issue escomptée, les bons résultats

obtenus ont eu des retombées positives. Le CDD embauché via l'ANR FastTrack a obtenu un poste d'ingénieur de recherche à l'IP2I. Quant à l'ingénieur d'études (IE) avec lequel j'ai démarré cette activité en 2012, son rôle de responsable technique du projet, qu'il a assuré avec beaucoup de compétence, lui a permis de bénéficier d'une promotion IR en 2017.

En tant que responsable, ces points sont importants et montrent qu'un projet peut apporter beaucoup même s'il ne se conclut pas comme on l'aurait souhaité.

V

Conclusion

CONCLUSION

J'ai tenté de retracer à travers ce document la majeure partie de mes activités au cours de ces quinze dernières années. Tout a commencé par l'alignement d'un VELO dans LHCb, et se termine aujourd'hui par un extracteur de mégots dans CMS. Au-delà des activités, c'est également mon métier qui a évolué. De simple post-doctorant dans LHCb, je suis peu à peu devenu chef de projet, tout d'abord dans ATLAS, et plus concrètement dans CMS, où j'ai pu goûter aux joies des revues de projets et autres chemins critiques.

J'espère avoir pu retranscrire la beauté et la complexité du futur trajectographe de CMS. Si je fais ce métier, c'est pour prendre part à ces projets qui semblent avoir été créés pour cette fameuse phrase de Mark Twain : *"Ils ne savaient pas que c'était impossible, alors ils l'ont fait"*.

J'ai pu vérifier personnellement cette maxime pour le concentrateur. Si j'avais eu en 2013 les quelques connaissances que j'ai aujourd'hui dans le domaine, je n'aurais jamais repris ce projet. Et finalement, à l'heure où j'écris ces lignes, la première version du CIC est un succès. Alors évidemment, quand on se lance dans ce genre d'aventure, il n'y a pas que des moments agréables, mais quand les choses finissent par aboutir, on oublie une grande partie des difficultés rencontrées.

Même si le projet pour lequel nous nous sommes battus n'est finalement pas choisi, comme ce fut le cas du système de reconstruction de traces rapides par mémoires associatives, d'utiles enseignements peuvent être retirés. Il y a toujours quelque chose à gagner, c'est le message qu'il faut retenir au final.

BIBLIOGRAPHIE

[1] L. Evans. *The Large Hadron Collider : a marvel of technology*. Fundamental Sciences. EPFL Press, Lausanne, 2009. https://cds.cern.ch/record/1193034.

[2] Georges Aad et al. Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. *Phys. Lett.*, B716 :1–29, 2012. https://arxiv.org/abs/1207.7214.

[3] Serguei Chatrchyan et al. Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC. *Phys. Lett.*, B716 :30–61, 2012. https://arxiv.org/abs/1207.7235.

[4] S. Viret. *Etude des désintégrations radiatives des mésons B dans le détecteur ATLAS*. PhD thesis. https://tel.archives-ouvertes.fr/tel-00007066.

[5] The ATLAS Collaboration. ATLAS Level-1 Calorimeter Trigger : Timing Calibration with 2009 LHC Beam Splashes. Technical Report ATL-DAQ-PUB-2010-001, 2010. http://cds.cern.ch/record/1262858.

[6] The CMS Collaboration. Technical Proposal for a MIP Timing Detector in the CMS Experiment Phase 2 Upgrade. Technical Report CERN-LHCC-2017-027, 2017. https://cds.cern.ch/record/2296612.

[7] A. Augusto Alves, Jr. et al. The LHCb Detector at the LHC. *JINST*, 3 :S08005, 2008. https://iopscience.iop.org/article/10.1088/1748-0221/3/08/S08005/pdf.

[8] R. Aaij et al. Performance of the LHCb Vertex Locator. *JINST*, 9 :P09007, 2014. https://iopscience.iop.org/article/10.1088/1748-0221/9/09/P09007/pdf.

[9] S. Blusk, O. Buchmuller, A. Jacholkowski, T. Ruf, J. Schieck, and S. Viret, editors. *Proceedings, first LHC Detector Alignment Workshop*, number CERN-2007-004, 2007. http://cdsweb.cern.ch/record/970621.

[10] S. Viret, C. Parkes, and M. Gersabeck. Alignment procedure of the LHCb Vertex Detector. *Nucl. Instrum. Meth.*, A596 :157–163, 2008. https://arxiv.org/abs/0807.5067 et Annexe A.1.

[11] M. Gersabeck, S. Viret, and C. Parkes. Performance of the LHCb Vertex Detector Alignment Algorithm determined with Beam Test Data. *Nucl. Instrum. Meth.*, A596 :164–171, 2008. https://arxiv.org/abs/0807.5069 et Annexe A.2.

[12] J M Amoraal. Alignment of the LHCb detector with Kalman filter fitted tracks. Technical Report LHCb-PROC-2009-026, 2009. http://cds.cern.ch/record/1205066.

[13] G. Aad et al. The ATLAS Experiment at the CERN Large Hadron Collider. *JINST*, 3 :S08003, 2008. https://iopscience.iop.org/article/10.1088/1748-0221/3/08/S08003/pdf.

[14] G. Aad et al. Readiness of the ATLAS Tile Calorimeter for LHC collisions. *Eur. Phys. J.*, C70 :1193–1236, 2010. https://arxiv.org/abs/1007.5423.

[15] E Starchenko et al. Cesium monitoring system for ATLAS Tile Hadron Calorimeter. In *Proceedings, 8th International Conference on Instrumentation for colliding beam physics (INSTR02) : Novosibirsk, Russia*, 2002. https://cds.cern.ch/record/685349.

[16] J. Abdallah et al. The Laser calibration of the Atlas Tile Calorimeter during the LHC run 1. *JINST*, 11(10) :T10005, 2016. https://arxiv.org/abs/1608.02791.

[17] K. Anderson et al. Design of the front-end analog electronics for the ATLAS tile calorimeter. *Nucl. Instrum. Meth.*, A551 :469–476, 2005. http://inspirehep.net/record/699395.

[18] Ziad J Ajaltouni et al. The Tilecal Laser Monitoring. Technical Report ATL-TILECAL-94-039, 1994. https://cds.cern.ch/record/683494.

[19] M. Anfreville et al. Laser monitoring system for the CMS lead tungstate crystal calorimeter. *Nucl. Instrum. Meth.*, A594 :292–320, 2008. https://cds.cern.ch/record/1073694.

[20] P. Gris. Upgrade of the Laser Calibration System of the Atlas Hadron Calorimeter. In *Proceedings, 2015 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC 2015) : San Diego, California, United States*, number ATL-TILECAL-PROC-2015-019. http://cds.cern.ch/record/2103219.

[21] L. Rossi and O. Brüning. Introduction to the HL-LHC Project. *Adv. Ser. Direct. High Energy Phys.*, 24 :1–17, 2015. https://cds.cern.ch/record/2130736.

[22] M. Prydderch et al. CBC3 : a CMS microstrip readout ASIC with logic for track-trigger modules at HL-LHC. Technical Report CMS-CR-2017-383, 2017. https://cds.cern.ch/record/2293142.

[23] D. Ceresa et al. Characterization of the MPA prototype, a 65 nm pixel readout ASIC with on-chip quick transverse momentum discrimination capabilities. Technical Report CMS-CR-2018-279, 2018. https://cds.cern.ch/record/2658203.

[24] A. Caratelli et al. Characterization of the first prototype of the Silicon-Strip readout ASIC (SSA) for the CMS Outer Tracker phase-2 upgrade. Technical Report CMS-CR-2018-286, 2018. https://cds.cern.ch/record/2650963.

[25] B. Nodari et al. A 65 nm Data Concentration ASIC for the CMS Outer Tracker Detector Upgrade at HL-LHC. Technical Report CMS-CR-2018-278, 2018. http://cds.cern.ch/record/2650712.

[26] P. Moreira. The LpGBT Project, Status and Overview. In *ACES 2016 - Fifth Common ATLAS CMS Electronics Workshop for LHC Upgrades*, 2016. https://cds.cern.ch/record/2137809.

[27] C. Soós and others. Versatile Link PLUS transceiver development. In *Proceedings, Topical Workshop on Electronics for Particle Physics (TWEPP 2016) : Karlsruhe, Germany*, volume 12, page C03068, 2017. https:/inspirehep.net/record/1518507.

[28] S. Viret. Stubs. Technical Report CMS DN-2018/003, 2018. http://cms.cern.ch/iCMS/jsp/openfile.jsp?tp=draft&files=DN2018_003_v2.pdf et Annexe C.1.

[29] S. Viret. Stub windows tuning : a tutorial. https://www.dropbox.com/s/sc1ifhfbcv5pts3/Stub%20windows%20tuning-%20a%20tutorial.pdf, 2017.

[30] R. Turchetta. Spatial resolution of silicon microstrip detectors. *Nucl. Instrum. Meth.*, A335 :44–58, 1993. https://doi.org/10.1016/0168-9002(93)90255-G.

[31] B.Nodari S.Scarfi S. Viret L. Caponetto, G. Galbit. CIC1 technical specification. https://espace.cern.ch/Tracker-Upgrade/Electronics/CIC/Shared%20Documents/Specifications/CIC_specs_v3.pdf et Annexe C.3, 2018.

[32] S. Viret. Data transmission efficiency of the phase II tracker front-end system for the tilted geometry. https://espace.cern.ch/Tracker-Upgrade/Electronics/CIC/Shared%20Documents/Simulation%20studies/FE_inneff_tilted.pdf et Annexe C.2, 2018.

[33] F. De Canio et al. Characterization of SLVS Driver and Receiver in a 65 nm CMOS Technology for High Energy Physics Applications. In *Proceedings, Topical Workshop on Electronics for Particle Physics (TWEPP17) : Santa Cruz, CA, USA*, 2017. http://cds.cern.ch/record/2673784.

[34] K. E. Batcher. Sorting Networks and Their Applications. In *Proceedings of the April 30–May 2, 1968, Spring Joint Computer Conference*, 1968. https://doi.org/10.1145/1468075.1468121.

[35] M. Smith. *Application-Specific Integrated Circuits*. Addison Wesley, 1997. Lien pdf.

[36] A. Baird et al. A Fast high resolution track trigger for the H1 experiment. *IEEE Trans. Nucl. Sci.*, 48 :1276–1285, 2001. https://arxiv.org/abs/hep-ex/0104010.

[37] M. Dell'Orso and L. Ristori. VLSI structures for track finding. volume A278, pages 436–440, 1989. http://inspirehep.net/record/276421.

[38] J. Adelman et al. Real time secondary vertexing at CDF. *Nucl. Instrum. Meth.*, A569 :111–114, 2006. https://doi.org/10.1016/j.nima.2006.09.022.

[39] Evelyn J. Thomson et al. Online Track Processor for the CDF Upgrade. *IEEE Trans. Nucl. Sci.*, 49 :1063–1070, 2002. https://doi.org/10.1109/TNS.2002.1039615.

[40] F. Morsani et al. The AMchip : A VLSI associative memory for track finding. *Nucl. Instrum. Meth.*, A315 :446–448, 1992. http://inspirehep.net/record/346554.

[41] M. Shochet, L. Tompkins, V. Cavaliere, P. Giannetti, A. Annovi, and G. Volpi. Fast TracKer (FTK) Technical Design Report. Technical Report CERN-LHCC-2013-007, 2013. https://cds.cern.ch/record/1552953.

[42] Fabrizio Palla and Giuliano Parrini. Tracking in the trigger : From the CDF experience to CMS upgrade. *PoS*, VERTEX2007 :034, 2007. https://pos.sissa.it/057/034/pdf.

[43] Edward Bartz et al. FPGA-Based Tracklet Approach to Level-1 Track Finding at CMS for the HL-LHC. *EPJ Web Conf.*, 150 :00016, 2017. https://arxiv.org/abs/1706.09225.

[44] I. Tomalin et al. An FPGA based track finder for the L1 trigger of the CMS experiment at the High Luminosity LHC. *JINST*, 12 :P12019, 2017. http://cds.cern.ch/record/2287640.

[45] A. Annovi et al. A new "Variable Resolution Associative Memory" for High Energy Physics. In *Proceedings, 2nd International Conference on Advancements in Nuclear Instrumentation, Measurement Methods and their Applications (ANIMMA 2011) : Ghent, Belgium*, 2011. https://cdsweb.cern.ch/record/1352152.

[46] G. Baulieu and S. Viret. Associative memory l1 track trigger camp. https://indico.cern.ch/event/325924/, 2014.

[47] G. Vidale. A study of l1 track trigger algorithms for the cms detector at hl-lhc. http://tesi.cab.unipd.it/57489/1/VidaleGiorgio_tesi.pdf, 2017.

[48] G. Baulieu & S. Viret. System performance, from default geometry to titled geometry. https://indico.cern.ch/event/591143/contributions/2385618/, 2016.

[49] G. Baulieu & S. Viret. Killing tails and combinatorics :an AM update. https://indico.cern.ch/event/354998/contributions/835298/, 2015.

[50] G. Baulieu & S. Viret. Latest Simulation results from Lyon. https://indico.cern.ch/event/521286/contributions/2134756/, 2016.

[51] G. Baulieu & S. Viret. AM Simulation Status Overview. https://indico.cern.ch/event/348920/contributions/1748967/, 2014.

[52] G. Baulieu & S. Viret. Bank optimization studies. https://indico.cern.ch/event/354336/contributions/1759770/, 2014.

[53] G. Baulieu & S. Viret. Recent ideas for AM optimization. https://indico.cern.ch/event/612363/contributions/2468787/, 2017.

[54] G. Baulieu & S. Viret. Stub bend in AM approach. https://indico.cern.ch/event/625269/contributions/2524812/, 2017.

[55] Développement d'un système électronique de reconstruction de traces pour les expériences du grand collisionneur de hadrons. https://anr.fr/Projet-ANR-13-BS05-0011, 2016.

[56] H Wind. Principal component analysis, Pattern recognition for track finding, Interpolation and functional representation. Technical Report CERN-EP-INT-81-12-REV, 1982. https://cds.cern.ch/record/139228.

[57] S. Amerio et al. The GigaFitter : Performance at CDF and Perspectives for future Applications. *J. Phys. Conf. Ser.*, 219 :022001, 2010. https://iopscience.iop.org/article/10.1088/1742-6596/219/2/022001/pdf.

[58] W. Ashmanskas et al. Performance of the CDF Online Silicon Vertex Tracker. *IEEE Trans. Nucl. Sci.*, 49 :1177–1184, 2002. https://inspirehep.net/record/584840.

[59] E. Clement et al. A High-performance Track Fitter for Use in Ultra-fast Electronics. *Nucl. Instrum. Meth.*, A935 :95–102, 2019. https://arxiv.org/abs/1809.01467.

[60] Thomas Stephen Williams. IPbus A flexible Ethernet-based control system for xTCA hardware. Technical Report CMS-CR-2014-334, 2014. https://cds.cern.ch/record/2020872.

[61] Alberto Annovi et al. AM06 : the Associative Memory chip for the Fast TracKer in the upgraded ATLAS detector. Technical Report ATL-DAQ-PROC-2016-017, 2016. https://cds.cern.ch/record/2228284.

[62] G. Galbit. Clustering et Filtrage rapide de traces pour le L1 track trigger de CMS. https://indico.in2p3.fr/event/12860/contributions/11805/, 2016.

[63] G. Fedi et al. A Real-Time Demonstrator for Track Reconstruction in the CMS L1 Track-Trigger System Based on Custom Associative Memories and High-Performance FPGAs. *PoS*, TWEPP-17, 2018. https://cds.cern.ch/record/2673659.

[64] A. Annovi et al. Characterization of an Associative Memory Chip in 28 nm CMOS Technology. In *Proceedings, 2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2018. https://doi.org/10.1109/ISCAS.2018.8351801.

[65] A. Stabile et al. Phase-II Associative Memory ASIC Specifications. Technical Report CERN-OPEN-2018-003, 2018. http://cds.cern.ch/record/2320701.

[66] M. Vaz et al. A Framework for Development and Test of xTCA Modules with FPGA Based Systems for Particle Detectors. In *Proceedings, 11th International Workshop on Personal Computers and Particle Accelerator Controls (PCaPAC 2016) : Campinas, Brasil*, 2017. http://inspirehep.net/record/1649458/.

TABLE DES FIGURES

1.1 Le LHC 7
1.2 Le détecteur LHCb 10
1.3 Le VELO de LHCb 11
1.4 Principe de fonctionnement du VELO 11
1.5 Vue détaillée des strips d'un module du VELO 11
1.6 Alignement et reconstruction de traces 12
1.7 Vertex reconstruits avant alignement 13
1.8 Vertex reconstruits après alignement 13
1.9 Le détecteur ATLAS 14
1.10 Vue des fibres et des photomultiplicateurs d'un secteur du calorimètre à tuiles 15
1.11 Calibration bas niveau du calorimètre à tuiles 16
1.12 Description de la boite LASER 17
1.13 Système de distribution de lumière du système LASER 17
1.14 Déroulement d'un évenement LASER 18

2.1 Les différentes phases du projet LHC 20
2.2 Taux d'occupation et granularité : illustration 20
2.3 Flux de données dans le trajectographe phase II 21
2.4 Le futur trajectographe de CMS 22
2.5 Vue de la répartition des modules de détection dans le plan R-Z 22
2.6 Vue d'un module 2S (ensemble et détail). 23
2.7 Vue d'un module PS (ensemble et détail). 23
2.8 Stub : principe de base 24

3.1 Readout : principe de base 27
3.2 Principe de lecture détaillé d'une moitié de module 2S 28
3.3 Principe de lecture détaillé du module PS 29
3.4 Stub : principe de base 30
3.5 Définition de base d'un stub provenant d'une particule primaire. 31

3.6 Corrections appliquées pour prendre en compte la planéité des modules. . 31
3.7 Paramètres définissant le stub dans le cas d'un module incliné par rapport à l'axe du faisceau. 32
3.8 Résolution des stubs provenant de muons primaires dans le tonneau 33
3.9 Résolution des stubs provenant de muons primaires dans les bouchons . . 33
3.10 Les différents types de stubs associés à une particule primaire 33
3.11 Régions du trajectographe nécessitant une fenêtre de sélection distincte . . 35
3.12 Fenêtres larges pour un empilement moyen de 140 interactions par collision 36
3.13 Fenêtres larges pour un empilement moyen de 200 interactions par collision 36
3.14 Résolution des stubs de mauvaise qualité provenant de muons primaires dans le tonneau 38
3.15 Résolution des stubs de bonne qualité provenant d'électrons primaires dans le tonneau 38
3.16 Largeur de stubs de bonne qualité induits par des muons primaires en fonction de l'impulsion transverse du muon. 39
3.17 Largeur de stubs de mauvaise qualité induits par des muons primaires en fonction de l'impulsion transverse du muon. 39

4.1 Principe de fonctionnement du CIC 42
4.2 Flux de données entrant et sortant du CIC 44
4.3 Principe de fonctionnement de l'outil d'analyse des pertes dynamiques . . . 47
4.4 Proportion de stubs de bonne qualité perdus au niveau du concentrateur (en %), pour un empilement moyen de 250 interactions par événement, avec une vitesse de sortie de 320MHz. 48
4.5 Exemple d'evolution de la profondeur de la FIFO du CIC dans une zone à haute occupation. 49
4.6 Diagramme bloc du CIC1 51
4.7 Principe du réalignement des données dans le cas du module 2S 52
4.8 Traitement des données de la voie L1 53
4.9 Les étapes de la conception d'un ASIC [35] 55
4.10 Floorplan du CIC2 56
4.11 Grille d'alimentation du CIC2 56
4.12 CIC2 après routage complet 57
4.13 Analyse IR drop pour le CIC2. L'IR drop maximal (cellules rouges) correspond ici à environ 10mV, soit moins de 1% de la tension nominale. 58
4.14 Système de test du CIC 59
4.15 Vue d'un CIC1 micro-câblé sur la mezzanine de test. 60
4.16 Profil de consommation du CIC1 60

5.1 Principe de fonctionnement d'un chip AM. 66
5.2 Distribution de données : multiplexage en espace 68
5.3 Distribution de données : multiplexage en temps. 68

6.1 Identification de traces avec des mémoires associatives. 71
6.2 Optimisation d'une banque de motifs 72
6.3 Exemple de motif dans un détecteur simple. Les traces vertes seront associées au motif, pas les rouges. 73
6.4 Définition géométrique d'un superstrip. 74
6.5 Encodage d'un superstrip et représentation d'un motif dans le tonneau (6 couches). 74
6.6 Définition d'un motif de résolution variable à partir de motifs de haute résolution. 75
6.7 Conversion du motif en bits ternaires (cas ou 1 seul bit ternaire est utilisé). 75
6.8 Comparaison du motif de résolution variable avec celui de basse résolution. 75
6.9 Codage de 15 positions avec 2 bits ternaires (encodage classique). 76
6.10 Codage de 15 positions avec 2 bits ternaires (encodage Gray). 76
6.11 Efficacité d'une banque produite avec (a) un spectre plat en p_T, (b) un spectre plat en $1/p_T$, (c) 3 zones à spectre plat en $1/p_T$ 79
6.12 Chaque superstrip de la banque pointe vers un superstrip du détecteur virtuel. 79
6.13 Motif à largeur constante (gauche) et motif projectif (droite) 81
6.14 Propriétés de la banque tonneau. Les parties du détecteur impliquées sont en vert. La taille correspondante des superstrips (en nombre de strips) et les nombres de bits ternaires maximum sont indiqués. 81
6.15 Propriétés de la banque hybride. Les parties du détecteur impliquées sont en vert. La taille correspondante des superstrips (en nombre de strips) et les nombres de bits ternaires maximum sont indiqués. 82
6.16 Propriétés de la banque bouchon. Les parties du détecteur impliquées sont en vert. La taille correspondante des superstrips (en nombre de strips) et les nombres de bits ternaires maximum sont indiqués. 82
6.17 Efficacité de reconstruction des traces pour des muons primaires sans empilement 84
6.18 Efficacité de reconstruction des traces pour des électrons primaires sans empilement 84

7.1 Nombre de combinaisons par événement après le filtrage AM, pour des événements $PU140 + t\bar{t}$ (noir) et $PU200 + t\bar{t}$ (rouge) 90
7.2 Principe de fonctionnement du TC builder 91
7.3 Principe de sélection des stubs dans le module TC 92

7.4 Valeurs de $|\Delta_\phi|$ pour des muons (noir) et des électrons (rouge) simulés. Les graines sont issues des couches 1 et 2 du tonneau. La couche de projection est la 4. 93

7.5 Proportion de stubs filtrés, en fonction du nombre total de stubs, après les mémoires associatives (cercles noirs) et après le TC builder (cercles blancs) 95

7.6 Rapport entre le nombre de combinaisons et le nombre de candidats filtrés pour des événements $PU140 + t\bar{t}$ (noir) et $PU200 + t\bar{t}$ (rouge) 96

7.7 Rapport entre le nombre de motifs activés et le nombre de candidats filtrés pour des événements $PU140 + t\bar{t}$ (noir) et $PU200 + t\bar{t}$ (rouge) 96

7.8 Nombre final de pistes L1 par événement, pour les événements PU140+ttbar (ligne noire) et PU200+ttbar (ligne rouge) 98

7.9 Le filtrage en action sur un événement complexe. 98

7.10 Le système FastTrack : principe 99

7.11 Banc de test FastTrack 99

7.12 Principe de fonctionnement du banc de test 99

7.13 Principe de fonctionnement du firmware de test 100

7.14 Nombre moyen de coups d'horloge passés dans les différentes étapes de la procédure de reconstruction, pour des événements $t\bar{t}$ pour un empilement moyen de 140. 102

7.15 Nombre total de coups d'horloge nécessaire au traitement d'événements $t\bar{t}$, pour un empilement moyen de 140, mesuré avec le banc de test. 102

LISTE DES TABLES

3.1 Limites d'extraction du frontend, en nombre moyen de stubs par module et par collision. 35

3.2 Proportions de traces reconstructibles mesurées avec différents types de fenêtres de sélection. 37

3.3 Proportion d'électrons reconstruits, avec des fenêtres larges et strictes optimisées pour un empilement de 140. 37

3.4 Proportion de traces reconstructibles avec ou sans coupures sur la largeur des stubs. 39

4.1 Caractéristiques techniques du CIC. 50

4.2 Comparaison des consommation maximales estimées pour le CIC1 et le CIC2. 62

6.1 Efficacités d'identification des traces (en %), pour des particules primaires d'impulsion transverse $p_T \geq 3$GeV/c 84

6.2 Nombre moyens de motifs activés et taux de faux pour des événements $t\bar{t}$. Pour le taux de sortie la limite à 95% est donnée entre parenthèses. 85

6.3 Nombre moyen de stubs et pureté des stubs par tour de déclenchement après le stade AM. Un bon stub est un stub induit par une particule avec $|d_0| < 1cm$ et $p_T > 3GeV/c$ 86

7.1 Efficacité de reconstruction totale, pour des particules primaires de $p_T \geq$3 GeV/c 95

7.2 Nombre de candidats filtrés en sortie et taux de faux correspondants. Les limites à 95% sont fournies entre parenthèses. 96

7.3 Comparison des limites à 95% des multiplicités candidats filtrés et des combinaisons brutes. 97

7.4 Nombre moyen de traces L1 et taux de faux par tour et par évenement. Les limites à 95% sont fournies entre parenthèses. 97

REMERCIEMENTS

Je me rends compte en écrivant ces lignes qu'il est beaucoup plus compliqué d'écrire des remerciements de HDR que des remerciements de thèse. C'est sans doute pour cette raison qu'on trouve rarement cette section dans les HDRs (on trouve rarement les HDRs en général de toute façon...).

C'est sûr que c'est plus compliqué d'oublier personne après 20 ans qu'après 3, très compliqué même. Il y a eu beaucoup de belles rencontres au cours de cette période, c'est ce que je retiens.

Beaucoup de ces bons moments ont une corrélation importante avec les membres de mon jury, un grand merci à eux pour avoir accepté d'en faire partie. Vous avez tous, à un moment ou à un autre de ma carrière, exercé une influence positive de part vos qualités humaines et scientifiques. Je suis heureux d'avoir pu croiser des chercheurs comme vous, et j'espère qu'on pourra un jour en dire autant de moi.

Il y a beaucoup d'autres personnes que je devrais remercier, à commencer par tous les collègues des différents laboratoires que j'ai pu côtoyer en 20 ans. La liste serait trop longue et je risquerais d'oublier des gens, donc je ne me vais pas ma lancer dans cet exercice périlleux.

Néanmoins, une grande partie du travail décrit dans ce document n'aurait pas vu le jour sans les contributions fondamentales de quelques ingénieurs de l'IP2I. Un grand merci donc à Guillaume, Geoffrey, Luigi, et Benedetta, car sans vous j'aurais eu beaucoup moins de chapitres à écrire.

Merci enfin à ma famille qui me supporte au quotidien. Vivre avec un chercheur n'est pas une sinécure, et le principal intéressé ne s'en rend pas toujours compte... Avec l'âge, çà commence à venir, mais je comprends que çà puisse vous paraître long parfois.

VI

Annexes

A

Notes LHCb

A.1/ ALIGNMENT PROCEDURE

University of Glasgow | Experimental Particle Physics

GLAS-PPE/2008-06
10th July 2008

Department of Physics and Astronomy
Experimental Particle Physics Group
Kelvin Building, University of Glasgow,
Glasgow, G12 8QQ, Scotland
Telephone: +44 (0)141 330 2000 Fax: +44 (0)141 330 5881

arXiv:0807.5067v1 [physics.ins-det] 31 Jul 2008

Alignment procedure of the LHCb Vertex Detector

S. Viret, C. Parkes, M. Gersabeck

Department of Physics and Astronomy, University of Glasgow
University Avenue, Glasgow, G12 8QQ, United Kingdom

Abstract

LHCb is one of the four main experiments of the Large Hadron Collider (LHC) project, which will start at CERN in 2008. The experiment is primarily dedicated to B-Physics and hence requires precise vertex reconstruction. The silicon vertex locator (VELO) has a single hit precision of better than 10 μm and is used both off-line and in the trigger. These requirements place strict constraints on its alignment. Additional challenges for the alignment arise from the detector being retracted between each fill of the LHC and from its unique circular disc r/ϕ strip geometry. This paper describes the track based software alignment procedure developed for the VELO. The procedure is primarily based on a non-iterative method using a matrix inversion technique. The procedure is demonstrated with simulated events to be fast, robust and to achieve a suitable alignment precision.

A.2/ ALIGNMENT TESTS

University of Glasgow | Experimental Particle Physics

GLAS-PPE/2008-07
10th July 2008

Department of Physics and Astronomy
Experimental Particle Physics Group
Kelvin Building, University of Glasgow,
Glasgow, G12 8QQ, Scotland
Telephone: +44 (0)141 330 2000 Fax: +44 (0)141 330 5881

arXiv:0807.5069v1 [physics.ins-det] 31 Jul 2008

Performance of the LHCb Vertex Detector Alignment Algorithm determined with Beam Test Data

M. Gersabeck*[a],S. Viret*[a],C. Parkes*[a] *et al.*
(the LHCb VELO group)
[a] University of Glasgow, Glasgow, United Kingdom
* *corresponding authors*

Abstract

LHCb is the dedicated heavy flavour experiment at the Large Hadron Collider at CERN. The partially assembled silicon vertex locator (VELO) of the LHCb experiment has been tested in a beam test. The data from this beam test have been used to determine the performance of the VELO alignment algorithm. The relative alignment of the two silicon sensors in a module and the relative alignment of the modules has been extracted. This alignment is shown to be accurate at a level of approximately 2 μm and 0.1 mrad for translations and rotations, respectively in the plane of the sensors. A single hit precision at normal track incidence of about 10 μm is obtained for the sensors. The alignment of the system is shown to be stable at better than the 10 μm level under air to vacuum pressure changes and mechanical movements of the assembled system.

B

Notes ATLAS

B.1/ LASER DESCRIPTION

ATLAS NOTE

ATL-TILECAL-INT-2010-001

January 25, 2010

ATLAS tile calorimeter LASER calibration system

R. Alves[b], D. Calvet[a], J. Carvalho[b], R. Chadelas[a], M. Crouau[a],
C. Crozatier[a], R. Febbraro[a], D. Lambert[a], R. Marchionini[a],
J. Oliveira[b], A. Pereira[b],C. Santoni[a], L.P. Says[a],
F. Vazeille[a], S. Viret[a]

[a] *Laboratoire de Physique Corpusculaire, Clermont-Ferrand, France*
[b] *Dept of Physics, Coimbra, Portugal*

Abstract

This note provides a detailed description of an important part of the ATLAS tile calorimeter (TILECAL) calibration framework: the LASER monitoring. The main role of this system is to monitor the stability and the linearity of the 9852 photomultiplier tubes (PMTs) and associated electronics used in the detector. This will be of particular importance for physics analysis, as (TILECAL) performance will be closely related to our understanding of the PMT response.

B.2/ LASER COMMISSIONING

ATLAS NOTE

ATL-COM-TILECAL-2010-014

June 7, 2011

Commissioning of the ATLAS tile calorimeter LASER calibration system

D. Calvet[a], J. Carvalho[b], R. Febbraro[a],
B. Galhardo[b], C. Santoni[a], F. Vazeille[a],
S. Viret[a]

[a] *Laboratoire de Physique Corpusculaire, Clermont-Ferrand, France*
[b] *Dept of Physics, Coimbra, Portugal*

Abstract

This note provides details on the performance of the ATLAS tile hadronic calorimeter (TILECAL) LASER monitoring system. The main role of this system is to monitor the stability and the linearity of the 9852 photomultiplier tubes (PMTs) and associated electronics used in the detector.

C

Notes CMS

C.1/ STUBS STUDY

Available on CMS information server

CMS Internal Note DN-2018/003

CMS Internal Note

The content of this note is intended for CMS internal use and distribution only

January 17, 2020

Stub definition and properties

S. Viret [a)]

Abstract

This document describes in details an important component of the future CMS Phase-2 tracker: the stub. The stub is the main building block of the future level-1 track trigger, and understanding its properties is therefore of particular importance.

[a)] Institut de Physique des 2 Infinis, UCBL, CNRS-IN2P3, Lyon, France

C.2/ FRONT END DYNAMIC INEFFICIENCIES

Available on CMS information server

CMS Internal Note IN-2017/XXX

CMS Internal Note

The content of this note is intended for CMS internal use and distribution only

January 4, 2017

Data transmission efficiency of the phase II tracker front-end system for the tilted geometry.

S. Viret [a)]

Abstract

Phase II outer tracker of the CMS experiment, due to the inclusion of the tracking at the first level of the trigger, will be a key component of CMS future triggering strategy. L1 tracking will be a very challenging task requiring the fast treatment of a huge data troughput. In order to meet this challenge, data rate reduction will be performed directly at the frontend level. The trigger signal will be compressed in two stages before being extracted from the detection module. The second compression step, performed by a concentrator chip (CIC), will heavily rely on average occupancy in the tracker. If this occupancy becomes to large, one has to evaluate how large the risk of data losses becomes. This document is an update of the previous losses studies, based on the future baseline tilted geometry.

[a)] Institut de Physique Nucléaire, UCBL, CNRS-IN2P3, Lyon, France

C.3/ CIC1 SPECIFICATIONS

CIC1 technical specification

***L. Caponetto*[(1)], *G. Galbit*[(1)], *B.Nodari*[(1)], *S.Scarfi*[(2)], *S. Viret*[(1)]**

(1)-Institut de Physique Nucléaire de Lyon (FR)

(2)-CERN

Version 3.1

10/12/2018

C.4/ TRACK TRIGGER DEVELOPMENT

Available on CMS information server

CMS Internal Note DN-2015/025

CMS Internal Note

The content of this note is intended for CMS internal use and distribution only

May 18, 2020

Emulation of a track reconstruction system based on associative memories

G. Baulieu [a)], G. Galbit [a)], W. Tromeur [a)], S. Viret [a)]

Abstract

The role of this note is to study the feasibility of an L1 tracking system based on associative memories (AM) for the CMS experiment Phase-2 upgrade. This work is based on simulations performed using a complete software emulation of AM-based tracking. A detailed description of this emulation, along with an estimation of the system parameters, is provided. A small hardware demonstrator is also presented.

[a)] Univ Lyon, Univ Claude Bernard Lyon 1, CNRS/IN2P3, IP2I Lyon, UMR 5822, F-69622, Villeurbanne, France

Document réalisé avec LaTeX et :
le style LaTeX pour HDR créé par S. Galland — http://www.multiagent.fr/ThesisStyle
la collection de paquets `tex-upmethodology` — http://www.arakhne.org/tex-upmethodology/

So long, and thanks for all the fish