

修　士　学　位　論　文

高輝度 LHC-ATLAS 実験に向けた
初段エンドキャップミューオントリガーの
ファームウェア開発と検証

令和 7 年 2 月 7 日

専攻名 物理学専攻
学籍番号 231S122S
氏　　名 水引 龍吾

神戸大学大学院理学研究科博士課程前期課程

概要

2030 年から開始される高輝度 LHC-ATLAS 実験は、ビームの輝度が現在の約 3 倍に押し上げられる。高輝度化による高統計なデータを活かした新粒子探索の感度向上などが期待されている。LHC の高輝度化に伴い、一度の衝突事象で生成される背景事象も大幅に増加するため、データを取捨選択するトリガーシステムがより一層重要となる。高輝度化に伴い、ATLAS 検出器外層にある検出器である Thin Gap Chamber (TGC) を用いた初段エンドキャップ部ミューオントリガーシステムはトリガーロジックを刷新する。初段エンドキャップ部ミューオントリガーでは TGC 検出器で得た粒子のヒットデータを Field Programable Gate Array (FPGA) に送信して粒子の飛跡再構成と横運動量の測定を行う。また、ATLAS 検出器内層のミューオン検出器の情報と統合することで背景事象を削減してトリガーレートの削減を行う。

本研究では高輝度 LHC-ATLAS 実験に向けて、内層ミューオン検出器と TGC 検出器間の双方でミューオンがあることを要求するインナーコインシデンスロジックの開発を行った。ミューオンの横運動量閾値を判別する Coincidence Window (CW) を実装して、インナーコインシデンスに内層ミューオン検出器と TGC 検出器で再構成された飛跡情報を入力し、ロジックからの出力と CW の結果が矛盾しないことを確認した。

また、効率的なトリガーロジックの開発のために新しい検証システムを開発した。開発したシステムは FPGA を実装したアクセラレータと呼ばれる拡張カードを利用する。インナーコインシデンスを開発した検証システムに実装することで、これまで時間的制約により困難だった CW の網羅的検証を行い、FPGA アクセラレータを用いた検証システムの性能を評価した。

目次

| | | |
|--------------|--|-----------|
| 第 1 章 | 序論 | 1 |
| 第 2 章 | 高輝度 LHC-ATLAS 実験 | 5 |
| 2.1 | ATLAS 検出器 | 5 |
| 2.1.1 | ATLAS 実験で使用される変数と座標系 | 5 |
| 2.1.2 | ミューオンスペクトロメータ | 6 |
| 2.1.3 | Thin Gap Chamber (TGC) 検出器 | 7 |
| 2.1.4 | RPC BIS78 検出器 | 11 |
| 2.1.5 | NSW 検出器 | 12 |
| 2.1.6 | Tile カロリメータ | 13 |
| 2.2 | TGC 検出器のエレクトロニクス | 15 |
| 2.2.1 | ASD ボード | 16 |
| 2.2.2 | Primary Processor board (PS ボード) | 16 |
| 2.2.3 | Endcap Sector Logic (SL) | 18 |
| 2.3 | 高輝度 LHC-ATLAS 実験におけるトリガーシステム | 22 |
| 2.3.1 | TDAQ システム | 22 |
| 2.4 | 高輝度 LHC-ATLAS 実験における TGC 検出器を用いたミューオントリガ | 24 |
| 2.4.1 | TGC 検出器を用いたトリガーのコンセプト | 24 |
| 2.4.2 | インナーコインシデンス | 27 |
| 2.4.3 | トリガーレートの削減と統計量の変化 | 28 |
| 第 3 章 | エンドキャップ部初段ミューオントリガーロジックの開発 | 31 |
| 3.1 | エンドキャップ部初段ミューオントリガーロジック | 31 |
| 3.1.1 | Channel Mapping | 33 |
| 3.1.2 | Station Coincidence | 33 |

| | | |
|-------|---|----|
| 3.1.3 | Segment Reconstruction | 34 |
| 3.1.4 | Wire-Strip Coincidence | 35 |
| 3.1.5 | インナーコインシデンス | 38 |
| 3.1.6 | Track Selector | 38 |
| 3.1.7 | TGC 読み出し回路 | 38 |
| 3.2 | インナーコインシデンスの構造 | 41 |
| 3.2.1 | Global Pos Decoder | 42 |
| 3.2.2 | NSW と TGC BW 間のコインシデンス | 43 |
| 3.2.3 | Tile カロリメータと BW TGC 間のコインシデンス | 45 |
| 3.2.4 | TGC EIL4 (EI) と BW TGC 間のコインシデンス | 47 |
| 3.2.5 | RPC BIS78 と BW TGC 間のコインシデンス | 48 |
| 3.2.6 | Which Inner ロジック | 48 |
| 3.3 | インナーコインシデンスの開発 | 48 |
| 3.3.1 | インナーコインシデンスのレイテンシ調節とタイミング制約の解決 | 49 |
| 3.3.2 | インナーコインシデンスに対する機能追加 | 51 |
| 3.3.3 | Vivado シミュレーションを用いた出力の確認 | 54 |
| 第 4 章 | トリガーロジック検証システム | 57 |
| 4.1 | 既存のトリガーロジック検証システム | 57 |
| 4.1.1 | 実機を用いた検証システム | 57 |
| 4.1.2 | Vivado シミュレーション | 58 |
| 4.1.3 | ビットワイスシミュレーション | 59 |
| 4.1.4 | ソフトウェアシミュレーション | 59 |
| 4.2 | アクセラレータカード | 60 |
| 4.2.1 | Alveo アクセラレータカード | 61 |
| 4.2.2 | XCU200 | 61 |
| 4.2.3 | Vivado | 64 |
| 4.2.4 | Vitis | 65 |
| 4.3 | FPGA アクセラレータを用いたトリガーロジック検証システムの開発 | 66 |
| 4.3.1 | 先行研究における開発状況 | 66 |
| 4.3.2 | Run 3 で用いられているトリガーロジックの実装と検証 | 66 |
| 4.3.3 | 先行研究と本研究との差異 | 68 |
| 4.3.4 | 開発手順 | 70 |
| 4.3.5 | 開発したトリガーロジック検証システムの詳細 | 70 |
| 4.3.6 | ホスト CPU で入力用データの生成・読み取りと FPGA への転送 . | 72 |

| | | |
|----------------|---------------------------------|----|
| 4.3.7 | FIFO buffer ロジック | 73 |
| 4.3.8 | Patch Panel ロジック | 75 |
| 4.3.9 | トリガー処理 | 76 |
| 4.3.10 | 大統計を用いたファームウェアの検証作業 | 76 |
| 4.4 | インナーコインシデンスファームウェアロジックのハードウェア実装 | 76 |
| 4.4.1 | ソフトウェアシミュレーションと検証用ミューオンデータ | 76 |
| 4.4.2 | ハードウェア実装とデバッグ作業 | 77 |
| 4.4.3 | 検出効率の評価 | 79 |
| 第 5 章 結論と今後の展望 | | 85 |
| 謝辞 | | 87 |
| 参考文献 | | 89 |

第 1 章

序論

素粒子標準模型 (Standard Model) は宇宙を構成する粒子と粒子間の相互作用を記述する物理学における基本的な理論で、現時点では素粒子実験における実験結果を最も正確に説明する [1]。標準模型は基本的な 4 つの相互作用のうち、重力相互作用を除く 3 つの相互作用 (電磁相互作用、弱い相互作用、強い相互作用) を記述する。基本粒子は物質を構成するフェルミオンと相互作用を媒介するボソンに分類される。フェルミオンとボソンの性質を表 1.1、および表 1.2 に示す。

表 1.1: 素粒子標準模型におけるフェルミオン [2]

| | 世代 | 名称 | 表記 | 質量 | 電荷 | スピン | |
|------|----|-----------|------------|-----------|----------------|---------------|--|
| クォーク | 1 | アップ | u | 2.2 MeV | $+\frac{2}{3}$ | $\frac{1}{2}$ | |
| | 2 | チャーム | c | 1.3 GeV | | | |
| | 3 | トップ | t | 173 GeV | | | |
| | 1 | ダウソ | d | 4.7 MeV | $-\frac{1}{3}$ | | |
| | 2 | ストレンジ | s | 94 MeV | | | |
| | 3 | ボトム | b | 4.2 GeV | | | |
| レプトン | 1 | 電子 | e | 511 keV | -1 | $\frac{1}{2}$ | |
| | 2 | ミューオン | μ | 106 MeV | | | |
| | 3 | タウオ | τ | 1.78 GeV | | | |
| | 1 | 電子ニュートリノ | ν_e | <2 eV | 0 | | |
| | 2 | ミューニュートリノ | ν_μ | <0.19 MeV | | | |
| | 3 | タウニュートリノ | ν_τ | <18.2 MeV | | | |

フェルミオンは、グルーオンと結合するクォークと、結合しないレプトンに分けられる。クォーク・レプトンは電荷が 1 だけ違うペアを構成しており、世代と呼ばれる同じ量子数を

表 1.2: 素粒子標準模型におけるボソン [2]

| | 名称 | 表記 | 質量 | 電荷 | スピン |
|---------|-------|----------|----------|---------|-----|
| ベクトルボソン | 光子 | γ | 0 | 0 | 1 |
| | グルーオン | g | 0 | 0 | |
| | W ボソン | W^\pm | 8.4 GeV | ± 1 | |
| | Z ボソン | Z | 91.2 MeV | 0 | |
| スカラーボソン | ヒッグス | H | 125 MeV | 0 | 0 |

持つ3つのグループに分けられる。力を媒介するボソンは強い相互作用を媒介するグルーオン、電弱相互作用を媒介する光子、弱い相互作用を媒介する W/Z ボソンの3種類がある。また、湯川相互作用によりフェルミオンに質量を与える、自発的対称性の破れにより W/Z ボソンに質量を与えるヒッグス粒子がある。標準模型は様々な素粒子実験によりその正当性が評価されてきたが、標準模型では説明できない事象も報告されている。例えば、宇宙観測で銀河の角速度分布から間接的に存在が確認された宇宙質量のおよそ 27% を占めるダークマター [3] や、標準模型からの予測値から 4.2σ のずれが報告されているミューオンの異常磁気能率 [4] などがある。LHC-ATLAS 実験では、実験開始当初から標準模型を超えた理論 (BSM:Beyond the Standard Model) の検証が行われている。

スイスのジュネーブ郊外にある欧洲原子核研究機構 (CERN) では、素粒子標準模型の検証や精密測定、BSM に繋がりうる新しい物理事象の探索を目的として大型ハドロン加速器 (LHC) を用いた実験が行われている。LHC は地下 100 m に設置されており、周長は 26.7 km である。LHC-ATLAS 実験では LHC で加速した粒子を ATLAS 検出器の中心で衝突させ、生成・散乱された粒子の軌跡から物理事象の探索・測定を行う。第三期実験 (Run 3) では陽子を中心エネルギー 13.6 TeV まで加速して 1 バンチ当たり 25 ns の間隔で衝突させる。

ATLAS 検出器の全容を図 1.1 に示す。ATLAS 検出器にはソレノイド磁石とトロイド磁石の2種類のマグネットが設置されており、磁場で粒子の軌道を湾曲させることで運動量を測定する。ATLAS 検出器のサブシステムには荷電粒子の飛跡を検出する内部飛跡検出器や、粒子のエネルギーを測定するカロリーメータがある。検出器外層部には高い透過力を持つミューオンを検出するためのミューオンスペクトロメータが設置されている。

LHC 加速器はアップグレード後、2030 年から高輝度 LHC [6] として運転が始まる。今後の LHC 加速器のスケジュールを図 1.2 に示す。現在行われている第三期実験 (Run 3) とアップグレード後の高輝度 LHC を用いた実験 (Run 4) との比較を表 1.3 に示す。LHC の高輝度化に伴い、ATLAS 実験で 1 バンチ当たりに観測される陽子-陽子衝突事象が増加し、統計量の増加による新粒子探索への感度向上が期待されている。より多くの統計量を獲得

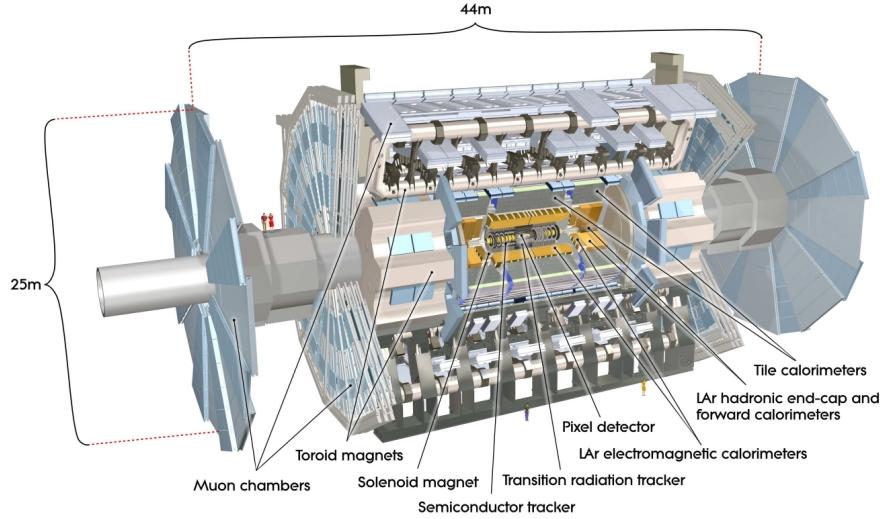


図 1.1: ATLAS 検出器の全体像 [5]。

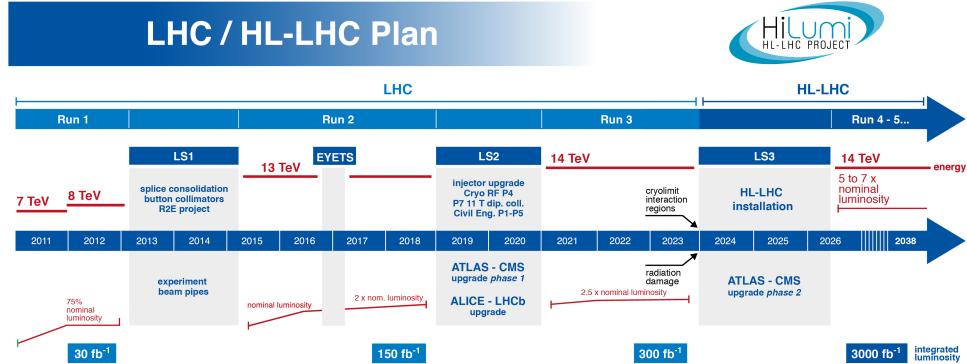


図 1.2: 今後 20 年間にわたる LHC 加速器のアップグレード計画 [7]。検出器や読み出しエレクトロニクスといった装置のアップグレードのために Long Shutdown (LS) が予定されている。現在の予定では 2030 年以降、HL-LHC を用いた第四期 LHC-ATLAS 実験が行われる予定である。

するために、初段エンドキャップ部トリガーのレイテンシは $10 \mu\text{s}$ に増強される。それに伴い ATLAS 検出器のトリガーエレクトロニクスは刷新されるため、搭載するトリガーロジックを新たに開発する必要がある。エンドキャップ部のミューオントリガーは、ATLAS 検出器内の構造と中性粒子の散乱で生成された荷電粒子などによる背景事象が多いため、内層と外層の検出器間でコインシデンスを取るロジック（インナーコインシデンス）でトリガーレートを抑える必要がある。インナーコインシデンスは現行の Run 3 実験でも採用されているが、高輝度化以後の新しい初段エンドキャップミューオントリガーにもロジックを刷新して実装される予定である。

本研究では高輝度化アップグレードに向けて、インナーコインシデンスロジックの開発

表 1.3: Run 3 と Run 4 (高輝度 LHC-ATLAS 実験) のビームスペック比較

| | Run 3 | Run 4 (高輝度 LHC) [7] |
|--|--------------------|--------------------------|
| 重心エネルギー [TeV] | 13.6 | 14 |
| 瞬間ルミノシティ [$\text{cm}^{-2}\text{s}^{-1}$] | 2×10^{34} | $5 - 7.5 \times 10^{34}$ |
| 積分ルミノシティ [fb^{-1}] | 450 | $3000 - 4000$ |

を行った。また、実機検証に近い環境を再現しつつ、より高速にトリガーロジックの検証を行うために、新しいトリガーロジック検証システムを開発・拡張し、インナーコインシデンスの検証に実用した。本論文の第二章以降の内容は以下の通りである。

第二章 高輝度 LHC-ATLAS 実験の概要について述べる。

第三章 初段ミューオントリガーロジックの説明とインナーコインシデンスの開発について述べる。

第四章 FPGA アクセラレータを用いたトリガーロジック検証システムの開発と、それを用いたインナーコインシデンスロジックの検証について述べる。

第五章 本論文のまとめについて述べる。

第 2 章

高輝度 LHC-ATLAS 実験

本章では高輝度 LHC-ATLAS で使用される座標系やミューオン検出器と TDAQ システムについて解説し、本研究のテーマである初段ミューオントリガーについて説明する。

2.1 ATLAS 検出器

ATLAS 検出器は円筒型の直径 25 m、長さ 44 m の汎用検出器である。ATLAS 検出器は複数の検出器の組み合わせで構成されており、内側から内部飛跡検出器、カロリメータ、ミューオンスペクトロメータの順に設置されている。複数の検出器を組み合わせた総合的な観測により、陽子–陽子衝突で生成される粒子の運動量や電荷、粒子の種類や質量を精密に測定することができる。

2.1.1 ATLAS 実験で使用される変数と座標系

ATLAS 検出器の座標系を図 2.1 に示す。ATLAS 実験では、直交座標系 (x, y, z) と円筒座標系 ($z, R(\theta), \phi$) が用いられる。直交座標系では、LHC リングの内側を向く方向を x 軸、地中から上空に向かう方向を y 軸とし、 z 軸はビーム軸と平行に、(x, y, z) で右手系を形成する方向に取る。円筒座標系では、ビームに沿った z 軸を基準に、動径方向を R 、方位角方向を ϕ 、 z 軸から起き上がる方向の極角を θ としている。 z 軸の正方向を A-side、負方向を C-side と定義する。一般にハドロン加速器では、衝突点から生成された粒子が散乱する方向を示すために、式 2.1 で定義される疑ラピディティ η を用いる。

$$\eta = -\ln \left(\tan \frac{\theta}{2} \right) \quad (2.1)$$

ATLAS 検出器内では、 $|\eta| < 1.05$ の領域をバレル領域、 $|\eta| > 1.05$ の領域をエンドキャップ領域と呼ぶ。

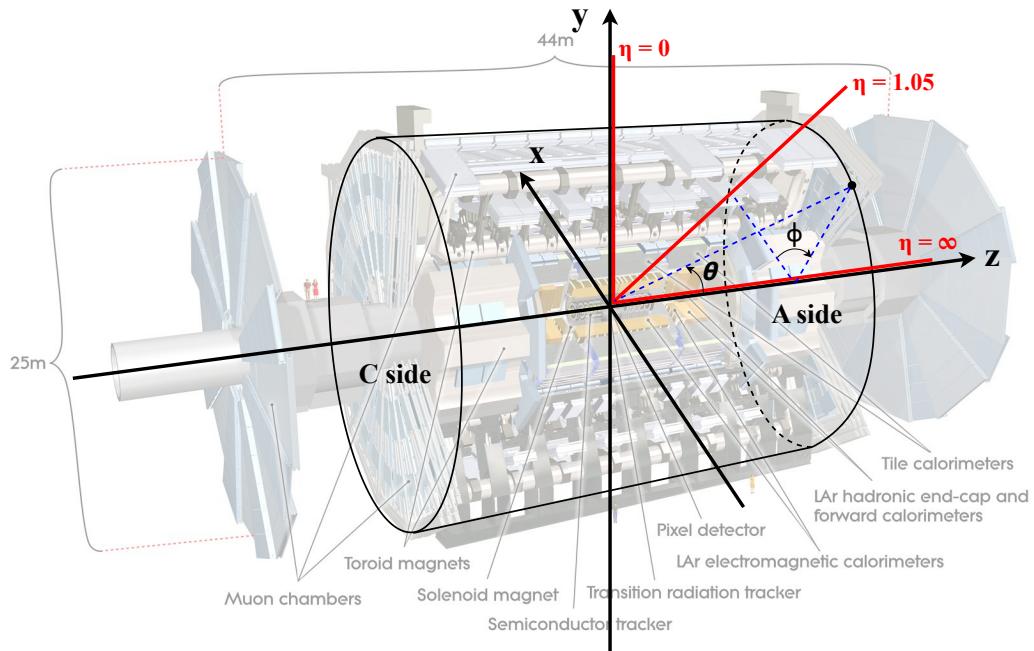


図 2.1: ATLAS 検出器で使用される座標系 [5]

内部飛跡検出器とカロリメータの間にはソレノイド型超伝導磁石が設置されており、ソレノイド磁場を形成する。同様に、カロリメータの外側にはトロイド型超伝導磁石が設置されており、トロイド磁場を構成する。超伝導磁石の配置を図 2.2 に示す。内部飛跡検出器はソレノイド磁場で曲げられた粒子の飛跡を観測し、曲率から粒子の運動量を計算する。カロリメータは、内層に電子や光子のエネルギー測定を行う電磁カロリメータ、外層にハドロンのエネルギー観測を行うハドロンカロリメータが設置された構造になっている。ミューオンスペクトロメータはミューオンの飛跡を再構成し、運動量を測定する。

2.1.2 ミューオンスペクトロメータ

ミューオンスペクトロメータは ATLAS 検出器を構成する検出器のうち、ミューオンの測定に特化した検出器の総称である。精密測定用の Monitored Drift Tube (MDT)、New Small Wheel (NSW) と、時間分解能に優れたトリガー用の検出器である Resistive Plate Chamber (RPC) および Thin Gap Chamber (TGC) から構成される。これらの検出器はミューオンの高い透過力を鑑みて ATLAS 検出器の外層部に設置されている。バレル部の xy 平面におけるミューオンスペクトロメータの配置を図 2.3 に示す。この構造のうち、バレル部のトロイド磁石が位置する領域を Small Sector、トロイド磁石間に位置する領域を Large Sector と呼ぶ。Small Sector と Large Sector におけるミューオン検出器の配置を

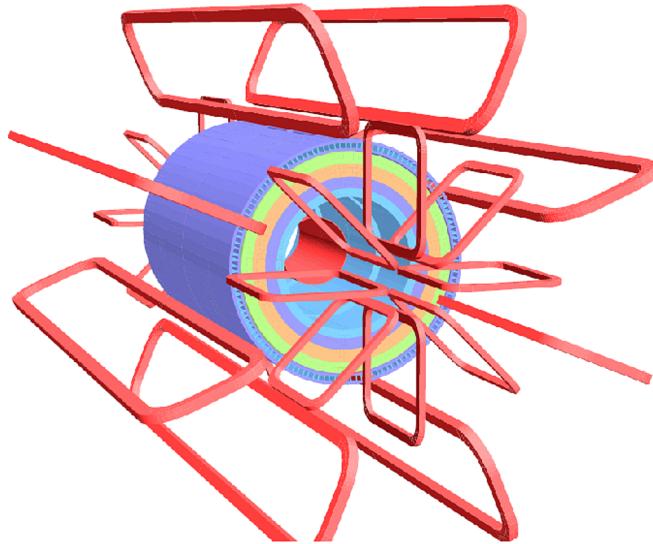


図 2.2: ATLAS 検出器に設置された超伝導マグネットの配置 [8]。ソレノイド磁石は内部飛跡検出器を囲むように設置されている。トロイド磁石はバ렐部とエンドキャップ部に設置されており、 ϕ 方向に 8 回対象となるように 22.5 度ずつ回転して設置されている。

図 2.4 および図 2.5 に示す。

本論文ではトロイド磁石の内側にある New Small Wheel (NSW) 検出器と Resistive Plate Chamber (RPC) 検出器の RPC BIS78 モジュール、そして EIL4 TGC といったミューオン検出器を内層ミューオン検出器と呼ぶ。また、Tile カロリメータの外側にあるセルもミューオン検出器として使用することが可能なので内層ミューオン検出器の一つとして扱う。2.1.3 節以降で本研究のエンドキャップミューオントリガーの主体となる TGC 検出器と、各内層ミューオン検出器について説明する。

2.1.3 Thin Gap Chamber (TGC) 検出器

TGC は Multi Wired Proportional Chamber (MWPC) の一つである。チェンバーの断面図を図 2.6 に示す。アノードワイヤー電極とカソードストリップ電極によってそれぞれ $R(\eta)$ 、 ϕ 方向の読み出しを行う。TGC のガスギャップは厚さ 2.8 mm で、55%:45% の比率で $\text{CO}_2/n - \text{C}_5\text{H}_{12}$ 混合ガスが充填されている。ガスギャップは厚さ 1.6 mm の FR4 によって挟まれており、FR4 の表面に張られた銅箔がストリップとして機能する。アノードワイヤーは直径 50 μm の金メッキされたタンクステン製ワイヤーで、1.8 mm 間隔で並んでいる。ミューオンの電離作用によりガスギャップ内で生成されたイオンは、ワイヤーに印加した 2.7 kV の高電圧が形成する電場によって加速と電離を繰り返し、電子雪崩を引き起こす。電子雪崩で生成された正イオンのドリフトによる電磁誘導で、ワイヤーには負電荷が、ストリップでは正電荷が検出される。TGC はトリガー用検出器なので高い時間分解

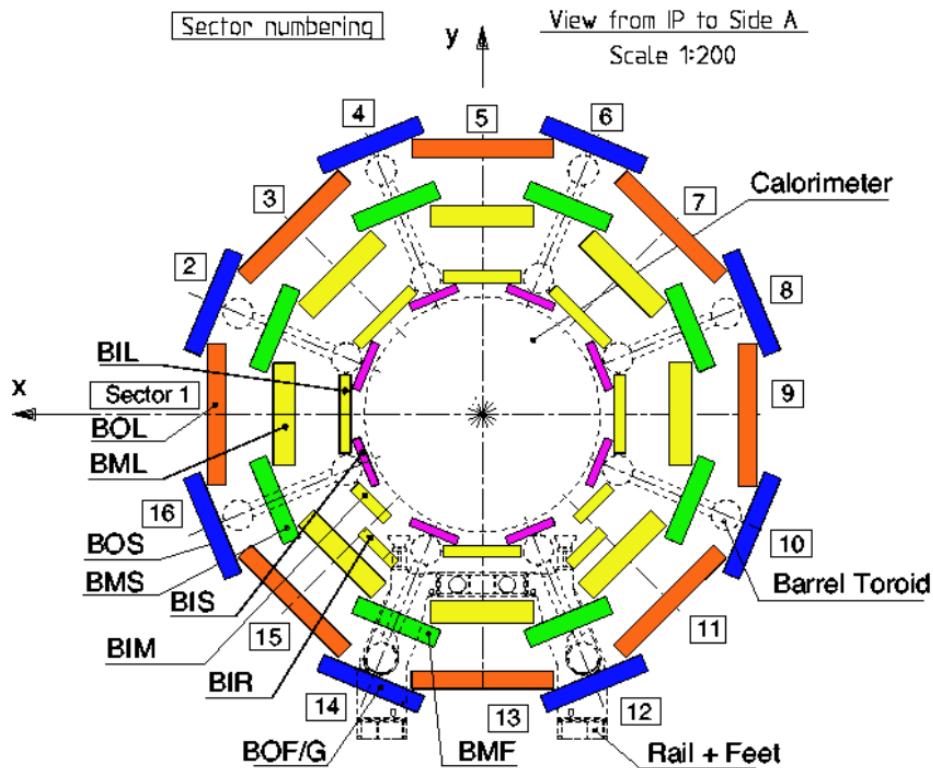


図 2.3: ビーム軸方向から見たバレル部におけるミューオンスペクトロメータの断面図 [5]。トロイド磁場に合わせて、トロイド磁石や支持支柱(図内点線)と干渉しないように ϕ 方向に 8 回対称で配置されている。

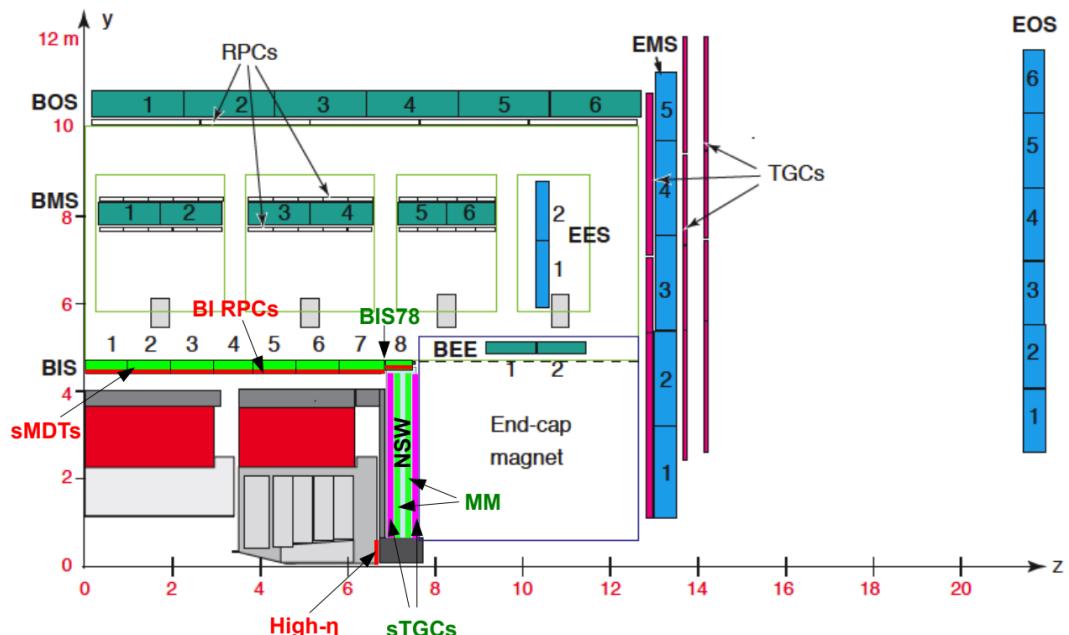


図 2.4: Small Sector におけるミューオン検出器の配置 [9]。図中の NSW, BIS78 が内層ミューオン検出器である。Tile カロリメータは図中の赤く塗りつぶされている領域にある。

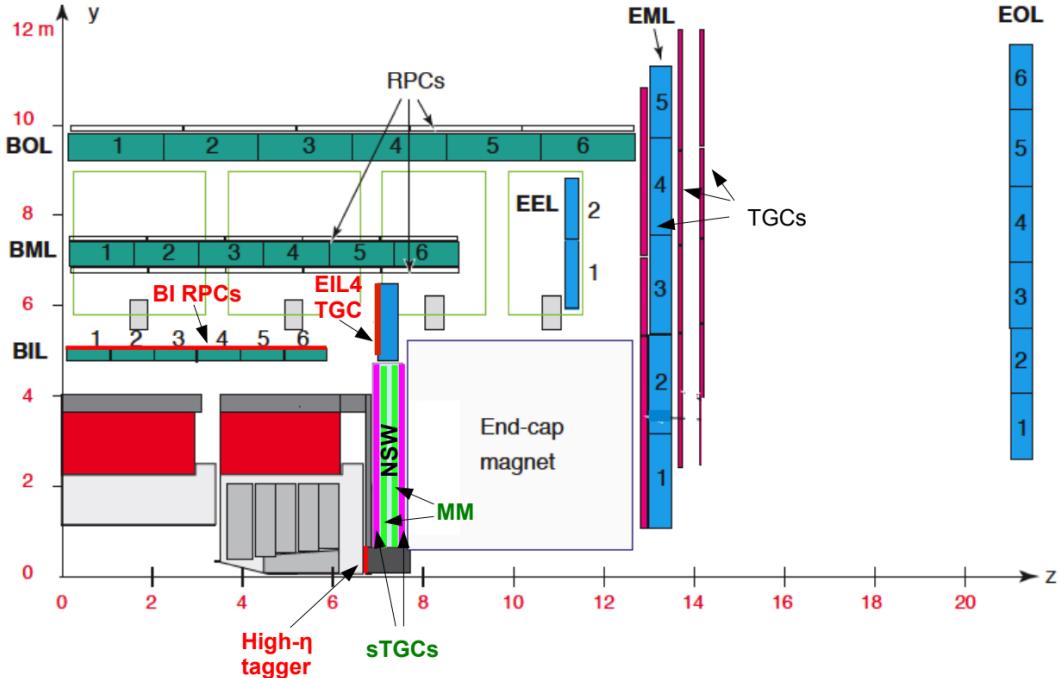


図 2.5: Large Sector におけるミューオン検出器の配置 [9]。図中の NW と EIL4 TGC が内層ミューオン検出器。Tile カロリメータは図中の赤く塗りつぶされている領域にある。

能が要求されている。そのため、電離して生成されたイオンのドリフト時間を短縮するために、ワイヤーとストリップの距離は 1.4 mm と決められている。TGC のガスギャップは 3 層構造と 2 層構造を持つ。概要を図 2.7 に示す。ワイヤーとストリップにより二次元読み出し可能な TGC を多層構造することで、通過した粒子の飛跡を三次元で再構築することができる。

ATLAS 検出器に設置された TGC 検出器の概要を図 2.8 に示す。LHC-ATLAS 実験において TGC 検出器は $1.05 < |\eta| < 2.4$ をカバーする。トロイド磁石の外側にある Big Wheel (TGC BW) と内側にある TGC EIL4 から構成されている。 $R - z$ 平面における TGC 検出器の配置を図 2.9 に示す。TGC BW は 3 つのステーションから構成されており、衝突点に近いほうから順に M1、M2、M3 と呼ぶ。M1 は 3 層構造、M2、M3、EIL4 は 2 層構造を持つ。TGC EIL4 は高輝度 LHC-ATLAS 実験から 3 層構造を持つチェンバーが使用される。高輝度 LHC-ATLAS 実験では 3 層中 2 層のヒットでトリガー判定を行うシステムが用いられる。仮に 1 層が機能しなくなってしまっても 2 層中 1 層のヒットを要求するトリガーシステムに切り替えることが可能となっており、冗長性の強化が図られる。

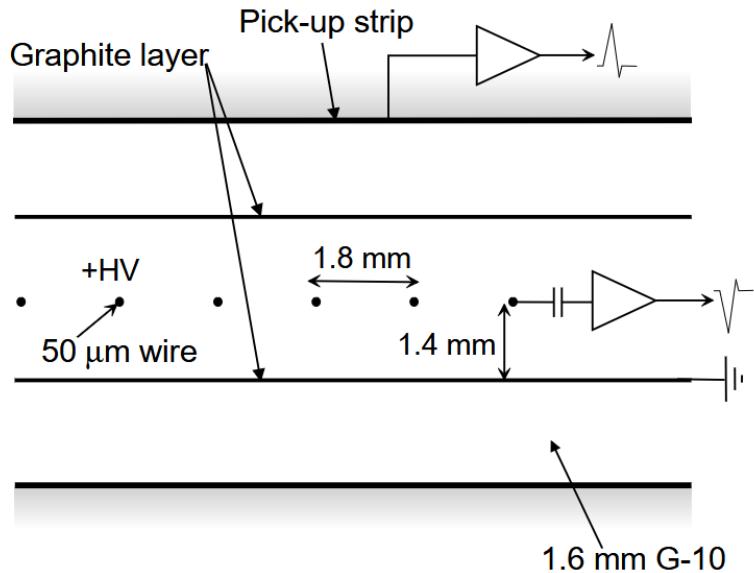


図 2.6: TGC のチャンバーの断面図。FR4 のガスギャップに接する面にはグラファイトがコーティングされており、グラウンドに接地されている。[10]

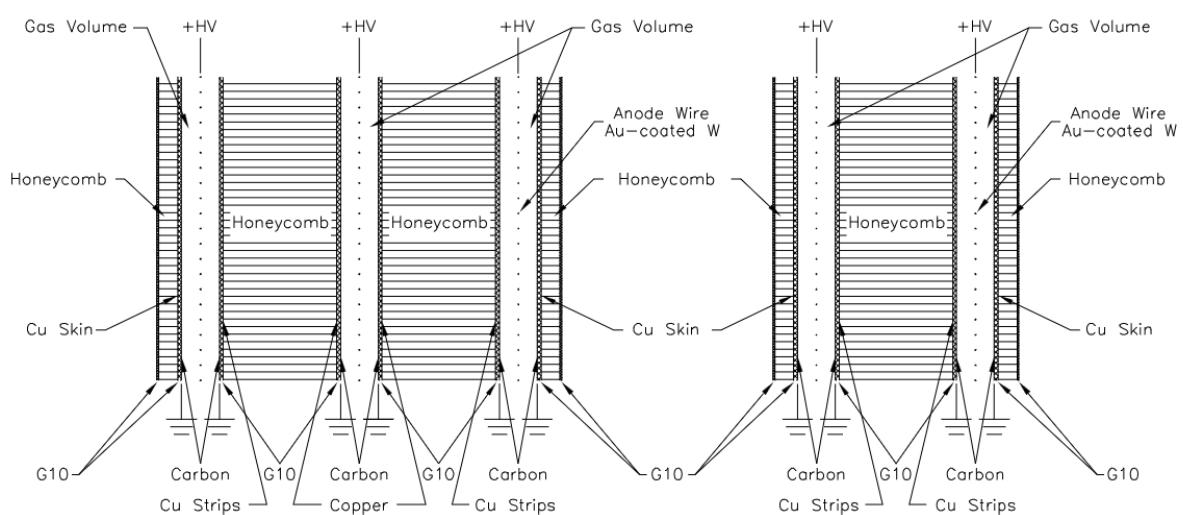


図 2.7: TGC 検出器の 3 層、2 層構造。[10]。ガスギャップを挟む FR4 はハニカム構造に支えられている。

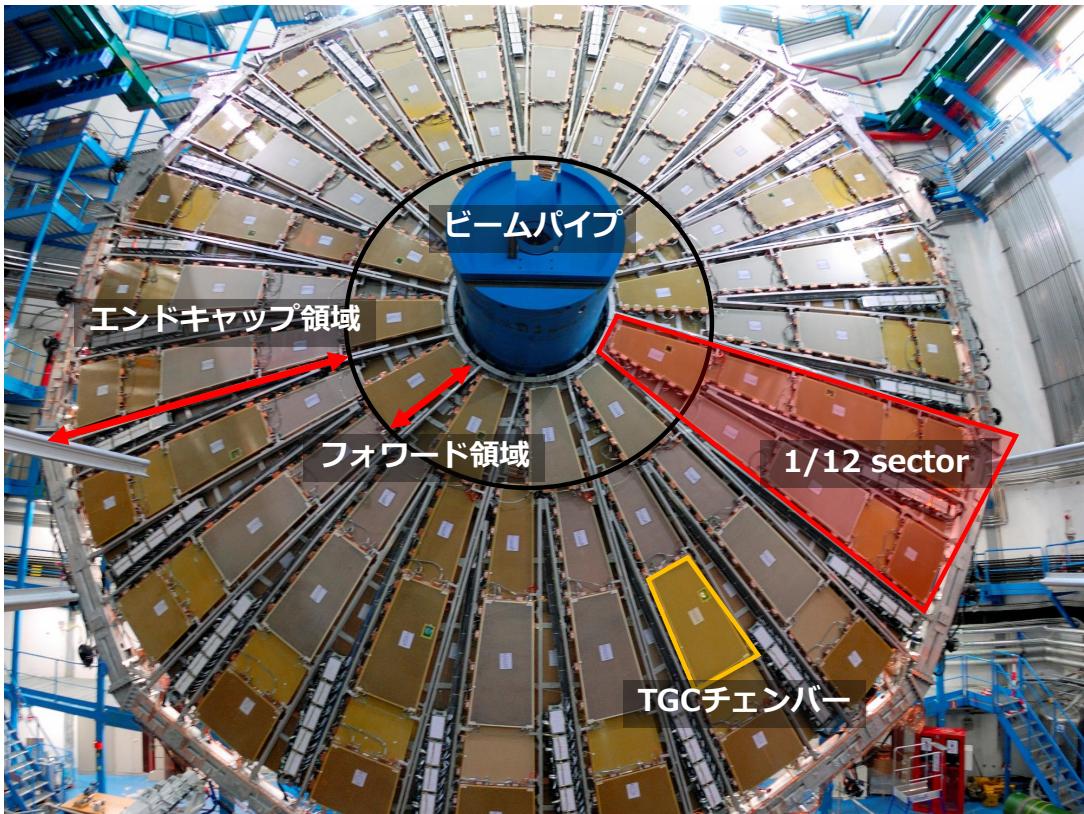


図 2.8: TGC BW 検出器の M1 ステーションを正面から見た写真 [11]。TGC BW は台形型のチェンバーを敷き詰めた構造となっており、直径 25 m の円盤構造をなす。TGC BW のうち、 $1.05 < |\eta| < 1.92$ の領域はエンドキャップ部と呼ばれ、 ϕ 方向に 48 回対称の形でチェンバーが設置されている。 $1.92 < |\eta| < 2.4$ はフォワード部と呼ばれ、チェンバーは ϕ 方向に 24 回対称の形をとる。エンドキャップ部のチェンバー 2 つをそれぞれ ϕ_0 , ϕ_1 と交互に命名し、フォワード部の一つと組み合わせた構造を 1/24 セクターと呼ぶ。1/24 セクターでは、それぞれ独立したトリガー回路で処理が行われるためトリガーセクターとも呼ばれる。

2.1.4 RPC BIS78 検出器

RPC BIS78 は Run 3 実験から small sector 領域 ($1.05 < |\eta| < 1.25$) の A-side 側にのみ設置された新しい RPC 検出器である。高輝度化アップグレード後の Run 4 からは C-side にも設置される。図 2.10 に示すように、厚さ 1.2 mm の高抵抗プレートで厚さ 1 mm のガスキャップを挟み込む構造をしている。高抵抗プレートには 5.8 kV の高電圧が印加されており、合計 3 層のガスギャップを持つ。ガスギャップの両端に 1 mm 間隔で直行するストリップが設置されており、 (η, ϕ) の二次元位置座標の読み出しを行う。空間分解能は 1 mm、時間分解能は 0.4 ns である。

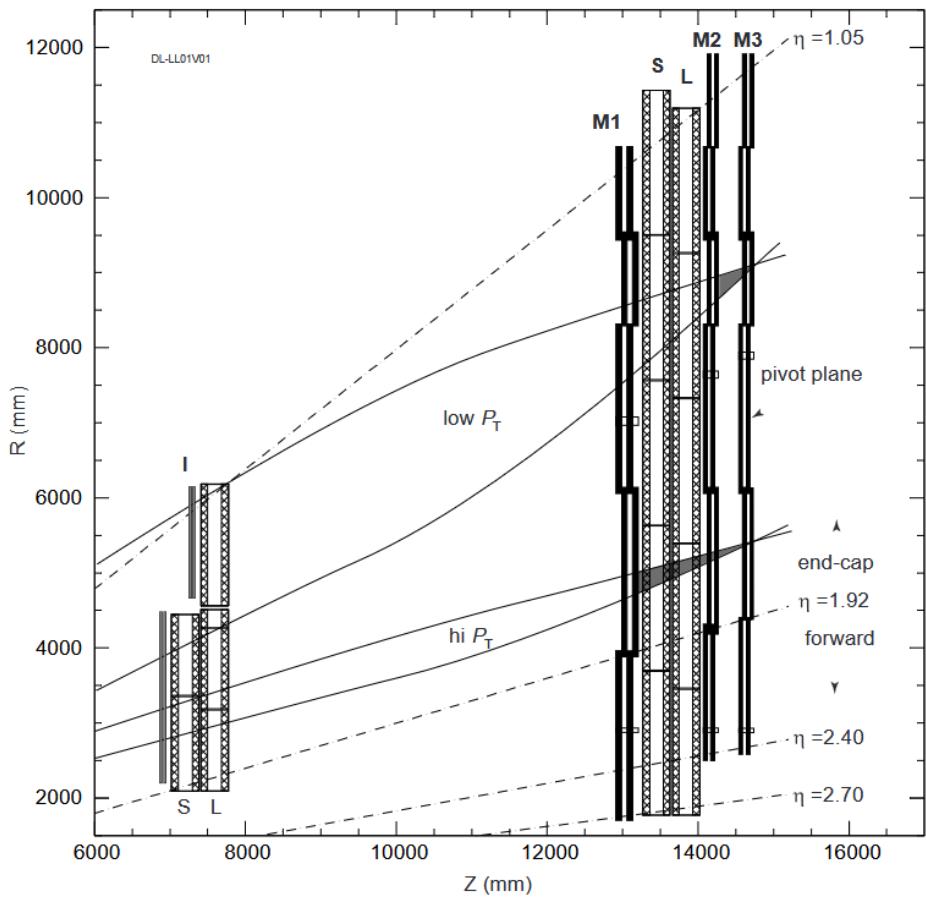


図 2.9: TGC 検出器の $R - z$ 平面における配置 [10]。衝突点に近い方から M1, M2, M3 ステーションが設置されている。

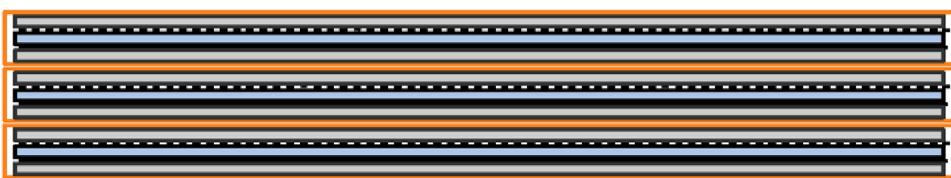


図 2.10: RPC BIS78 検出器の構造 [12]。RPC BIS78 検出器は 3 層のガスギャップを持っており、ハニカムによって構造を支えている。

2.1.5 NSW 検出器

NSW 検出器は $1.3 < |\eta| < 2.7$ をカバーする、トロイド磁石の内側に設置されたミューオン検出器である。NSW の構造を図 2.11 に示す。NSW はトリガーと精密測定両方の用途で使用される。4 層構造の small-strip TGC (sTGC) で 4 層構造の Micromegas (MM) を 2 つ挟み込んだ、16 層構造をもち、位置情報及び角度情報を取得する。MM はドリフト面と

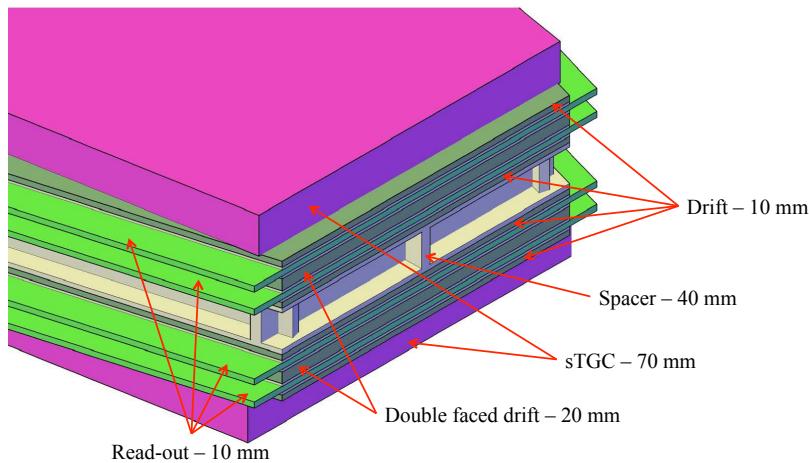


図 2.11: NSW 検出器の断面図。上下の赤紫の部分は sTGC であり、MM は sTGC に挟まれた構造となっている。[9]。

金属メッシュ面と読み出し用のストリップから構成されている。ドリフト面とメッシュ面に高電圧を加えて狭い増幅領域を形成し、素早い読み出しが可能となっている。sTGC の概要を図 reffig:sTGC に示す。sTGC は TGC と同じ検出器だが、TGC BW より短い 1.8 mm 間隔で設置されたワイヤーによる ϕ 方向読み出しと、3.2 mm 間隔で設置されたストリップで R 方向の位置座標読み出しを行う。電荷情報を用いた重心計算により、 $60\sim150 \mu\text{m}$ の R 方向位置分解能をもつ。パッドと呼ばれる幅の広い読み出しカドードにより粗い位置情報の読み出しを行う。MM は平面電極と金属メッシュで構成されたガス検出器である。イオンのドリフト時間を信号の時間差から測定することで、入射したミューオンの角度を見積もることが可能であり、高い位置分解能をもつ。MM は 40° で入射してきたミューオンに対して $90 \mu\text{m}$ の位置分解能を持つ。NSW 全体の分解能のデザイン値は、 η 方向に 0.005、 ϕ 方向に 10 mrad、ビーム軸に対する角度分解能は 1 mrad となっている。

2.1.6 Tile カロリメータ

Tile カロリメータはハドロンのエネルギーを測定する検出器の一つで、図 2.13 に示すような吸収体である鉄とプラスチックシンチレータを交互に組み合わせた形となっている。シンチレータ内でハドロンが発したシンチレーション光を波長変換ファイバーで読み出し、PMT に入力して電気信号として読み出す。 ϕ 方向に対して 64 個、 R 方向に A、BC、D の 3 層に分割した単位をセルと呼び、各セルは独立して信号読み出しを行う。各セルの配置図を図 2.14 に示す。ハドロンや電子、 γ 線などほとんどの粒子は最外層の D 層手前で停止する。そのため、D 層を通過する粒子は透過力の高いミューオンがほとんどであり、ミュー

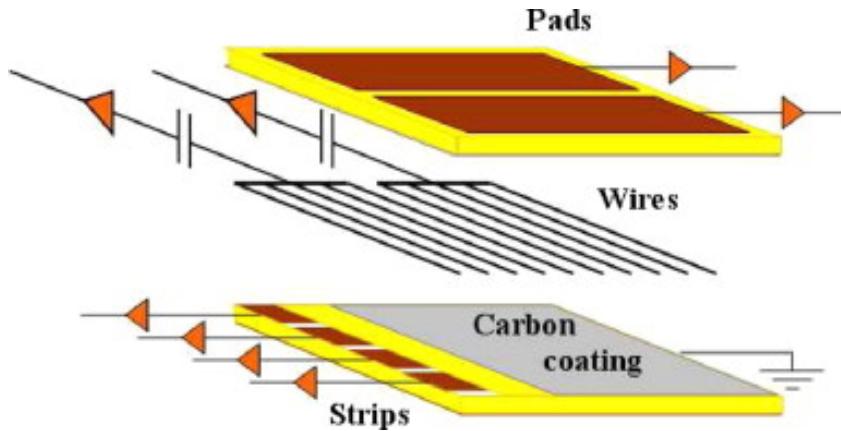


図 2.12: sTGC の概要図 [13]。

オン検出器としてトリガー判定に用いることが可能となっている。エンドキャップ部初段ミューオントリガーでは D5, D6 セルの情報を用いる。

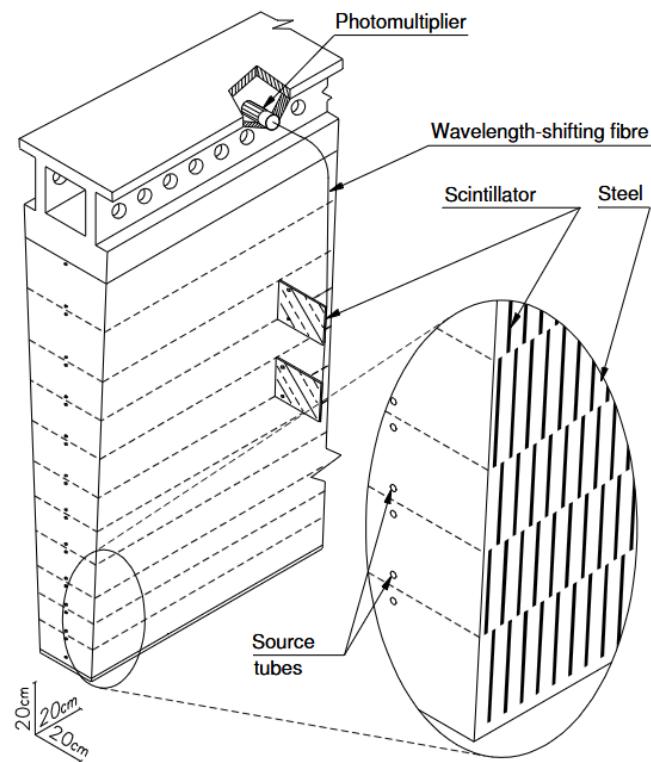


図 2.13: Tile カロリメータの構造 [5]。鉄とタイル状のシンチレータが重なった構造を持つ。

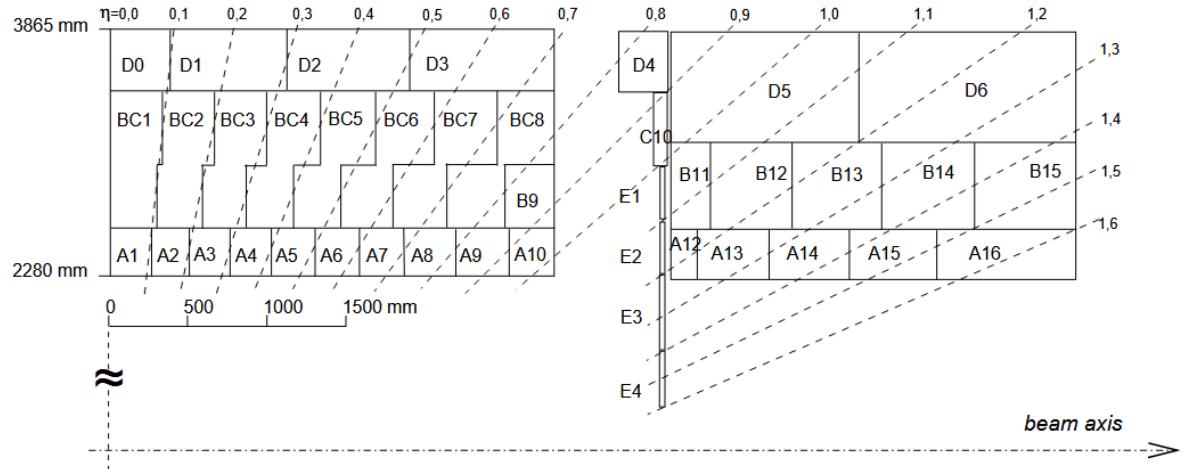


図 2.14: Tile カロリメータのセルの配置図 [5]。エンドキャップ部のミューオントリガーには D5,D6 セルを利用する。 $\eta < 1.2$ の領域では D5+D6 セルの情報を、 $\eta > 1.2$ では D6 セルだけの情報を扱う。

2.2 TGC 検出器のエレクトロニクス

現在の Run 3 で使用されている TGC 検出器の読み出しエレクトロニクスは、PS ボードのバッファーサイズの都合で $10 \mu\text{s}$ のトリガーレイテンシに対応していない。そのため、Run 4 以降の高輝度 LHC-ATLAS 実験に向けて、大幅なアップデートが計画されている。図 2.15 に、アップデート後の TGC 検出器読み出しエレクトロニクスの概要を示す。荷電粒子が TGC 検出器内を通過すると、電流信号が発生する。この信号は Amplifier Shaper Discriminator (ASD) を通して電圧信号に変換され、整形増幅される。取得した電圧信号はコンパレータによるノイズを除去を経てデジタル信号に変換され、後段の PS board に送信される。PS board には Patch Panel ASIC (PP ASIC) が搭載されており、デジタル信号を 40 MHz の LHC クロックに同期させる。この同期により、信号がどの陽子バンチ交差に対応するかを正確に同定することができる。バンチ交差を同定されたヒット情報は PS board 上の FPGA に送信され、ステータス情報などのヘッダーを追加して後段の Sector Logic (SL) に送信される。SL は ATLAS 検出器とは隔離された回路室に設置される。

SL は最大 31 枚の PS board からの情報を受信し、ボードに搭載した大規模 FPGA に入力する。FPGA に入力された信号情報はトリガーロジックとリードアウトロジックに転送される。リードアウトロジックではヒット情報を一時的にバッファーに保存する。トリガーロジックでミューオンのヒット判定を行い、後段の MUCTPI にトリガー判定の結果を送信する。トリガー判定を受けたバンチ交差はリードアウトロジックのバッファーから取り出され、後段の FELIX に送信される。

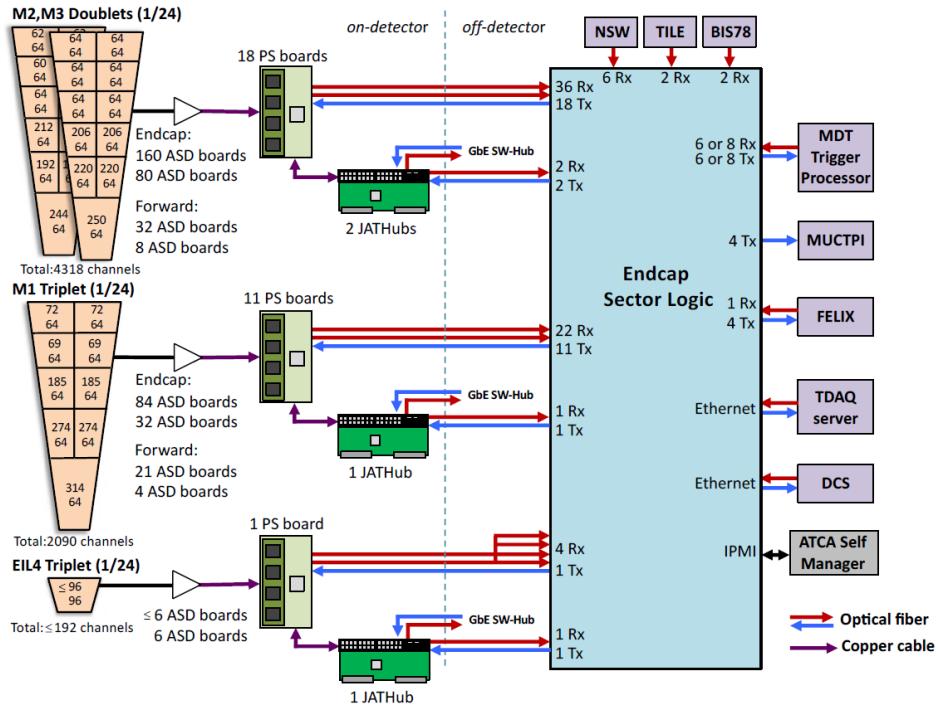


図 2.15: 高輝度 LHC-ATLAS 実験における TGC 検出器システムの概要 [9]。

2.2.1 ASD ボード

ASD ボードの概要を図 2.16 に示す。ASD ボードに搭載された ASD チップは 4 つの読み出しチャネルを処理する。ASD ボード 1 枚に ASD チップは 4 つ搭載されており、16 チャンネルの処理が可能である。TGC の読み出しチャネルの総数は 32 万チャネルなので、約 2 万枚の ASD ボードが ATLAS 検出器に使用される。ASD チップでは、TGC からの電流信号をチャージアンプであるプリアンプで電圧信号に変換する。このプリアンプは放電時定数 16 ns、電圧変換効率 0.8 V/pC であるエミッタ設置回路である。プリアンプから出力された電圧信号は利得 7 倍のメインアンプで増幅し、コンパレータで閾値電圧との比較を行う。コンパレータは、Low Voltage Differential Signaling (LVDS) 規格に従い、閾値を上回った電圧を受け取る間は High を、下回っている間は Low を出力してデジタル信号に変換する。整形増幅したデジタル信号は後段の PS ボードに送信される。

2.2.2 Primary Processor board (PS ボード)

PS ボードの概要を図 2.17 と図 2.18 に示す。PS ボードは 16 枚の ASD ボードと接続可能で、各 ASD ボードからの出力を処理する。A-side、C-side 合わせて 1434 枚の PS ボードを実験で使用する。PS ボードの主なコンポーネントは光通信用モジュールである SFP+

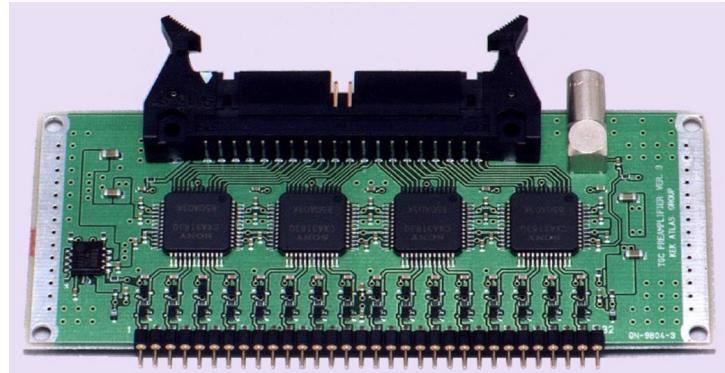


図 2.16: ASD ボードの全体像 [9]。

が 2 つ、PP ASIC が 8 枚と FPGA が 1 つとなっている。4 枚の PP ASIC はマザーボードに搭載されており、残りの 4 枚はメザニンボードに搭載されている。それぞれの SFP+ モジュールは 4 つの送信用リンクと受信用リンクを持つ。PS ボードと後段の SL は 3 本の光ファイバーで接続されており、2 本は PS ボードから SL に送信するヒットデータの送信に利用する。残りの 1 本は SL から PS ボードをコントロールする信号の送信に利用される。搭載された FPGA は AMD / Xilinx の Kintex-7 シリーズのものであり、ヒット情報にステータス信号などのヘッダーを追加し、320 bit のデータにして SL に送信する。また、ADC を 16 チャンネル分搭載しており、ASD のモニタリングに使用する。

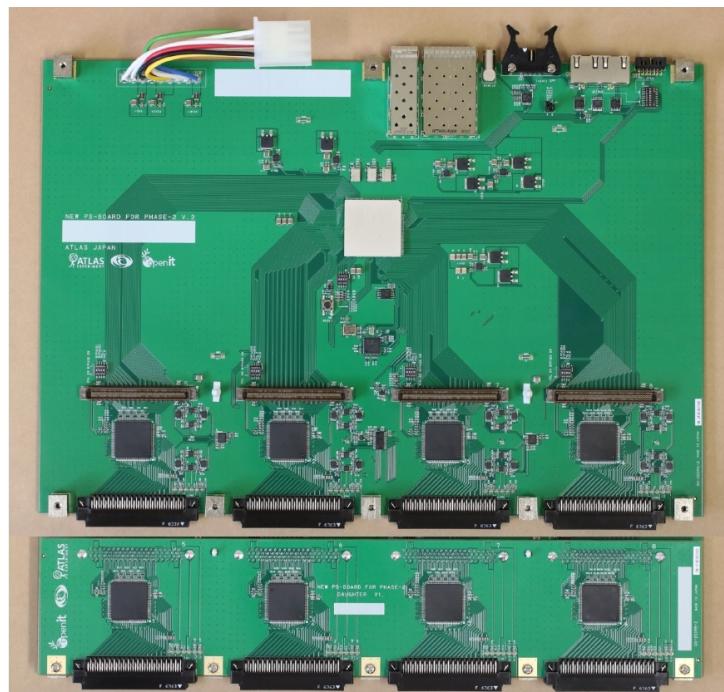


図 2.17: PS ボードの全体像 [14]。

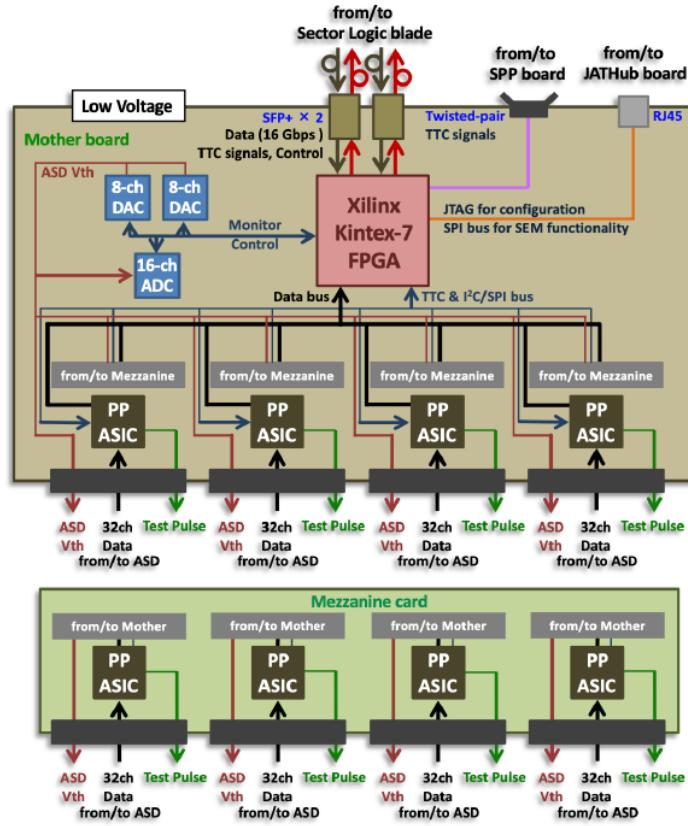


図 2.18: PS ボードのブロック図 [14]。

PP ASIC

PP ASIC は ASD ボードから PS ボードに入力されたヒット情報のバンチ交差を識別する回路である。可変遅延回路と陽子バンチ識別回路で構成されている。LHC-ATLAS 実験における陽子バンチ交差は 40 MHz で実行されるが、衝突点から TGC 検出器にミューオンが飛来するまでの時間である Time of Flight (TOF) や TGC 検出器から PS board に信号が至るまでの時間はケーブル長の違いにより各 PS board 每に異なる。そのため、PP ASIC では可変遅延器により適切な信号遅延を与え、陽子バンチ交差ごとに信号を同期させる。

2.2.3 Endcap Sector Logic (SL)

SL の概要を図 2.19 に示す。SL は Advanced Telecommunications Computing Architecture (ATCA) 規格 [15] で設計されたブレードであり、Intelligent Platform Management Controller (IPMC) インターフェイス [16] によるモニタリング機能を搭載している。主なコンポーネントは光通信用モジュールである FireFly [17]、フラッシュメモリとの通信を制御する QSPI+ [18]、FPGA や SL の状態チェックに加え、ボード全体の制御に使

用する Zynq MPSoC [19]、トリガーロジックやリードアウトロジックを搭載する Virtex UltraScale+ FPGA [20] である。SL は TGC 検出器や磁場領域内部にある TGC EIL4、NSW、Tile カロリメータのヒット情報を取得し、ミューオンの再構成と情報の統合によるトリガー判定の発行や、TGC BW 検出器からのヒット情報読み出しを行う。SL はミューオン候補と NSW の飛跡情報を MDT Trigger Processor (MDT-TP) に送信する。MDT TP では SL からの出力と MDT 検出器の情報を統合して高い精度でミューオン候補の飛跡再構成を行って再び SL に送信する。最終的に SL から MUCTPI にトリガー情報が送信される。図 2.20 に示すように、SL は 1 つで TGC BW の 1 つのトリガーセクターを担当する。SL は A-side と C-side のそれぞれで 24 枚、合計 48 枚が使用される。トリガーセクター内の全ての PS board の情報を 1 つの SL で処理するため、包括的なトリガーロジックの搭載が可能である。

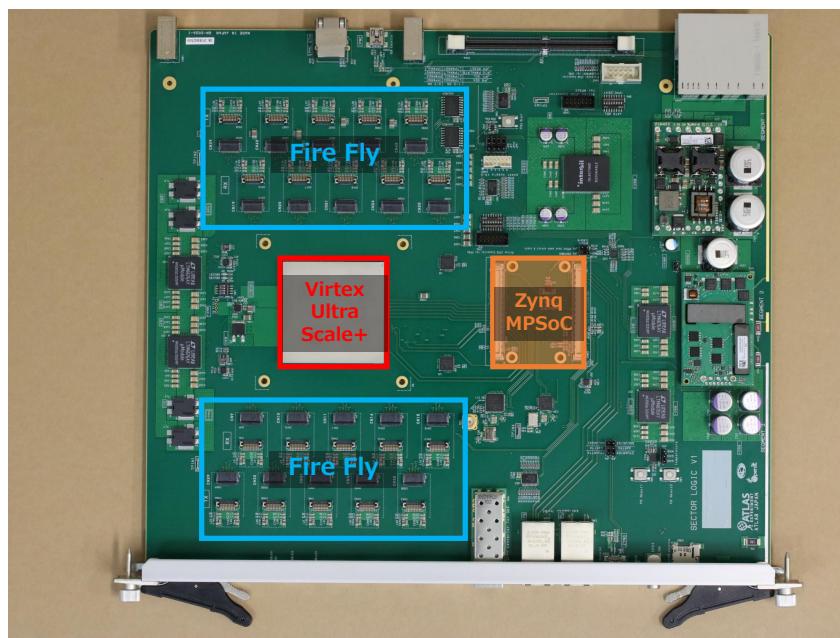


図 2.19: SL ボードの全体像 [21]。Zynq MPSoC は取り外されている。

Fire Fly

Fire Fly は光通信用モジュールであり、光信号と電気信号を変換する。Virtex UltraScale+ FPGA の入出力ピンと光ファイバー間の信号変換のために用意されている。SL は送受信用にそれぞれ 10 個の Fire Fly を搭載しており、それぞれ 12 チャンネルのリンクを持つ。そのため、Virtex UltraScale+ FPGA と光ファイバー間で 120 チャンネル分の送受信を行うことが可能となっている。

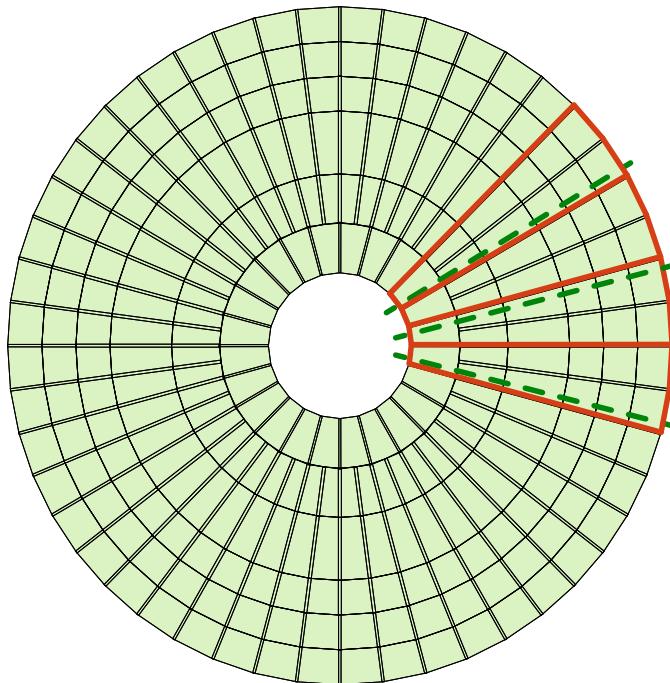


図 2.20: 1 つの SL がカバーするトリガーセクター [9]。赤線の領域は 1 つの SL が処理する領域で、緑の点線は MDTTP がカバーする範囲である。SL と MDTTP カバレッジが異なるため 1 つの SL は最大 3 つの MDTTP ボードに飛跡情報を出力する。

Zynq MPSoC

Zynq MPSoC はメモリや Arm プロセッサから構成される Processing System (PS) 部と FPGA からなる Programmable Logic (PL) 部からなるシステムである。PS 部のプロセッサには Linux OS がインストールされる。Zynq MPSoC は TDAQ システムなどや Detector Control System (DCS) とのインターフェイスの役割を担うだけでなく、Virtex Ultrascale+ FPGA に実装された読み出し回路の出力を確認するシステムが実装されている。

Virtex Ultrascale+ FPGA

SL は AMD / Xilinx 社の Virtex Ultrascale+ FPGA (XCVU13P-FLGA2577-1-e) を搭載している。XCVU13P は非常に大規模な FPGA で、複雑かつ大規模なトリガーロジック実装のために重要な役割を果たす。XCVU13P には大規模なメモリの構築を可能にする BRAM/URAM や、ロジックの途中でデータを保持する役割をもつレジスタが実装されている。XCVU13P デバイスの両端にはギガビットトランシーバ用のバンクである GTY トランシーバ [22] が並んでおり、光ファイバーのバンドリンクなどで用いられる。XCVU13P

は回路の高集積化のために 4 つの Super Logic Region (SLR) に分割された構造を持つ。XCVU13P のスペックと 1 つの SLR で利用可能なリソースを表 2.1 に示す。

表 2.1: XCVU13P のスペック [23]

| | |
|------------|-------------|
| ロジックセル | 3,780,000 個 |
| LUTs | 1,728,000 個 |
| BRAM | 94.5 Mbit |
| URAM | 360.0 Mbit |
| GTY トランシーバ | 128 個 |

■Super Logic Region (SLR) SLR [24] は AMD / Xilinx 社の 2.5 次元パッケージング技術の呼称であり、シリコンダイで物理的に分割された領域のことである。図 2.21 に示すように、XCVU13P は同じ性能の SLR を 4 つ組み合わせて構成されている。隣接する SLR は Super Long Line (SLL) [25] と呼ばれる専用のワイヤーで信号の送受信を行う。XCVU13P ではそれぞれの SLR 間に 23,040 本の SLL が実装されている。SLL を介した通信は通常のレジスタ間通信よりレイテンシが大きくなる、物理的な位置も固定されており、レジスタ配置の自由度が下がる。そのため、ファームウェアを実装する際には

- SLR を跨ぐロジックを極力設計しない。
- SLL を介した信号線の数をできる限り少なくする。
- クロックやリセット信号といった制御信号は同方向に流れるように配置する。

という原則を守ることで 1 クロック周期の間に処理が完了しないロジックの搭載を防ぐ。

■Block RAM・Ultra RAM (BRAM·URAM) Block RAM (BRAM) と Ultra RAM (URAM) はメモリの一種であり、それぞれ 36Kb と 288Kb のサイズを持ったメモリとして利用可能である。BRAM は 2 つの読み書きポートを持ち、18Kb のメモリ 2 つとして稼働させることができる。URAM はアドレス長 12bit、出力値 72bit で固定されており、2 つの独立したポートを持つ。SL のトリガーロジック内部では再構成されたミューオンの飛跡情報を用いて LUT によるミューオンの運動量計算が行われる。LUT はメモリの一種であり、アドレス番地に特定の出力を割り振っておくパターンリストのことである。入力であるアドレス値に対してあらかじめ登録された値を出力するため、複雑な演算を避けて高速で結果を出力することができる。本トリガーロジックでは BRAM や URAM を用いた大規模な LUT により高速な演算を実現している。

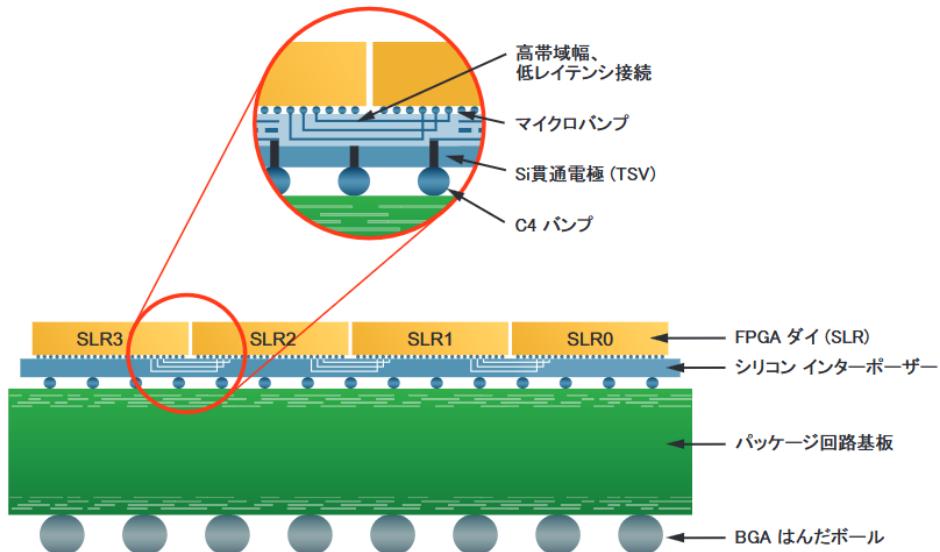


図 2.21: SLR の概略図 [26]。

2.3 高輝度 LHC-ATLAS 実験におけるトリガーシステム

LHC-ATLAS 実験では 40 MHz のレートで陽子バンチ衝突が発生し、同時に大量の散乱事象が発生するため、膨大なデータ (1 バンチ交差あたり 10 Mbit オーダー) が生成される。衝突で生じた全データを保存することは、ストレージの容量や通信帯域の問題で困難を極めるため、物理的に有用と思われるデータのみを保存するトリガーシステムによりデータの取捨選択を実行する。トリガー判定により有用とみなされたイベントを選択的に保存するためには、データ取得システムとトリガーシステムが緊密に連携したシステムを構築する必要がある。ATLAS 実験において、トリガーシステムとデータ取得システムを一括システムのことを、TDAQ (Trigger and Data Acquisition) システムと呼ぶ。

2.3.1 TDAQ システム

2030 年度より予定されている高輝度 LHC-ATLAS 実験ではビームの高輝度化に伴い、背景事象が大幅に増大する。Run 3 で使用している現行の TDAQ システムのままでは、データの読み出し帯域の制約により厳しいトリガー条件を設定せざるを得ず、結果的に物理的興味の高いデータも棄却することとなる。そのため、高輝度化に向けた TDAQ システムのアップグレードが行われる。

図 2.22 は高輝度 LHC-ATLAS 実験における TDAQ システムの概要図である。初段トリガーは Level-0 Trigger と呼び、後段トリガーは Event Filter と呼ぶ。以下で高輝度

LHC-ATLAS 実験における TDAQ システムについて述べる。

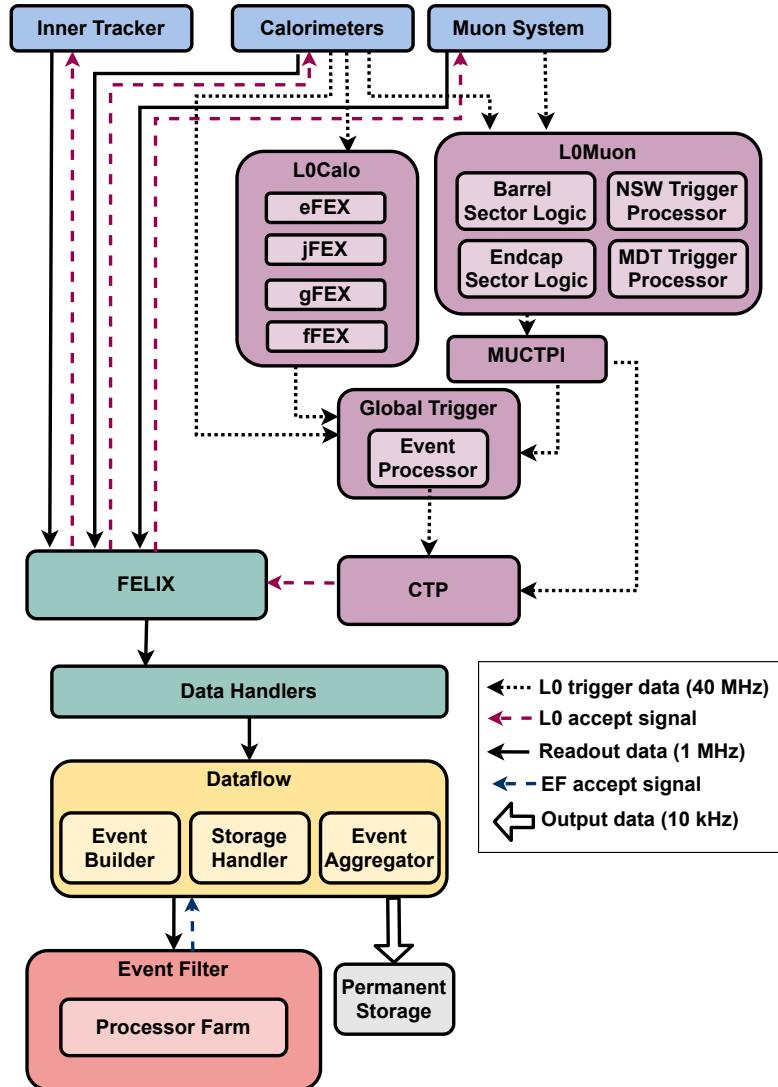


図 2.22: 高輝度 LHC-ATLAS 実験における TDAQ システム [27]。

Level-0 Trigger (L0 Trigger)

L0 Trigger は ATLAS 検出器から送られてくる 40 MHz のデータを、 $10 \mu\text{s}$ 以内に 10 MHz まで下げる初段トリガーである。ハードウェアを用いて高速でミューオン飛跡を再構成してトリガー判定を行う。Application Specific Integrated Circuit (ASIC) や Field Programmable Gate Array (FPGA) などのデバイスに、ハードウェア記述言語 (HDL)

で開発したトリガーロジックを回路実装して高速なロジックを実現する。L0 Trigger は Level-0 Calo (L0Calo)、Level-0 Muon (L0Muon)、Level-0 Global Trigger、Central Trigger Processor (CTP) で構成される。L0Calo ではカロリメータの情報を元にトリガー判定を行い、Level-0 Global Trigger に結果を送信する。Level-0 Muon では Run 3 実験で使用されていた Level-1 Muon とは異なり、MDT を含めた各ミューオンスペクトロメータの情報や、カロリメータの情報も統合してトリガー判定を行う。Level-0 Muon は処理する $|\eta|$ により異なるトリガープロセッサに分かれており、 $|\eta| < 1.05$ のバレル部を処理する Barrel Sector Logic、 $1.05 < |\eta| < 2.4$ のエンドキャップ部を処理する Endcap Sector Logic で構成される。これらトリガープロセッサから Muon to Central Trigger Processor Interface (MUCTPI) にトリガー情報が送信され、Level-0 Global Trigger と CTP に転送される。Level-0 Global Trigger では L0Calo と MUCTPI の情報を統合してトリガー判定を行い、CTP に送信する。CTP は L0 Trigger の最終判断としてイベントの選別を行い、各フロントエンドエレクトロニクスに Front-End Link exchange (FELIX) 経由で Level-0 Accept (L0A) を発出する。全てのフロントエンドエレクトロニクスは、バンチ交差が生じてから L0A が各検出器に届けられるまでの時間 (Level-0 レイテンシ) が常に一定の値を取るように設計されている。このような設計を Fixed Latency Scheme と呼び、Level-0 レイテンシは $10 \mu\text{s}$ 以内の決められた値で固定される。

Event Filter

Event Filter ではソフトウェアを用いて L0 Trigger を通過したイベントに対してトリガー判定を行う。トリガーレートは 1 MHz から 10 kHz まで削減される。主にソフトウェアを用いたトリガーであり、内部飛跡検出器など、全検出器からの情報を統合して、高い精度でイベントを再構成する。最終的に残った事象は CERN のデータセンターのストレージに保存される。

2.4 高輝度 LHC-ATLAS 実験における TGC 検出器を用いたミューオントリガー

本節では高輝度 LHC-ATLAS 実験における TGC 検出器を用いたトリガーのコンセプトを説明し、TGC 検出器のデータ読み出しエレクトロニクスについて説明する。

2.4.1 TGC 検出器を用いたトリガーのコンセプト

ATLAS 検出器中心部にある衝突点から、 $1.05 < |\eta| < 2.4$ のエンドキャップ部に向かうミューオンはトロイド磁石による磁場の影響で軌道を曲げて ATLAS 検出器外層部にある

TGC 検出器に入射する。TGC BW の各ステーションは、ワイヤーとストリップそれぞれのヒット情報を処理して ϕ と R のヒット情報を得る。各ステーションで算出したワイヤーとストリップのヒット情報を、3つのステーション間で統合してミューオンの飛跡を再構成し、ミューオンの運動量を計算する。

TGC BW の設置場所はトロイド磁場領域の外側のため、TGC BW に入射したミューオンのヒット点は直線飛跡をなす(図 2.23 参照)。この直線飛跡と、衝突点から M3 のヒット点を結んだ直線(無限運動量飛跡)の位置座標の差分である $(d\phi, dR)$ を元に p_T の算出を行う。トロイド磁場は一般に ϕ 方向に巻き付くように構築されている。そのため衝突点由来のミューオンは R 方向に曲げられることになる。運動量の大きいミューオンは磁場の影響を受けにくいため dR は小さくなる傾向にある。 dR の大きさからミューオンの運動量を算出することができる。一方、 $d\phi$ の大きさと p_T の間には一般的に相関はなく、均一な磁場であれば曲げられることはない。実際には図 2.24 と図 2.25 に示すように、磁場が不均一なので、 ϕ 方向にミューオンは曲げられるが、大きく曲げられることは少ない。エンドキャップトロイド磁石による磁場は一様ではないため、同じ運動量を持つミューオンであっても、 R 方向や ϕ 方向に曲げられる大きさはミューオンが飛来する座標によって異なる。そのため、 (ϕ, R) 方向を等間隔に Region として区分して、各 Region ごとに $(d\phi, dR)$ と横運動量 p_T の関係を対応付けた Look Up Table (LUT) を用意してミューオンの運動量を高速で計算する。この運動量計算アルゴリズムをパターンマッチングと呼ぶ。

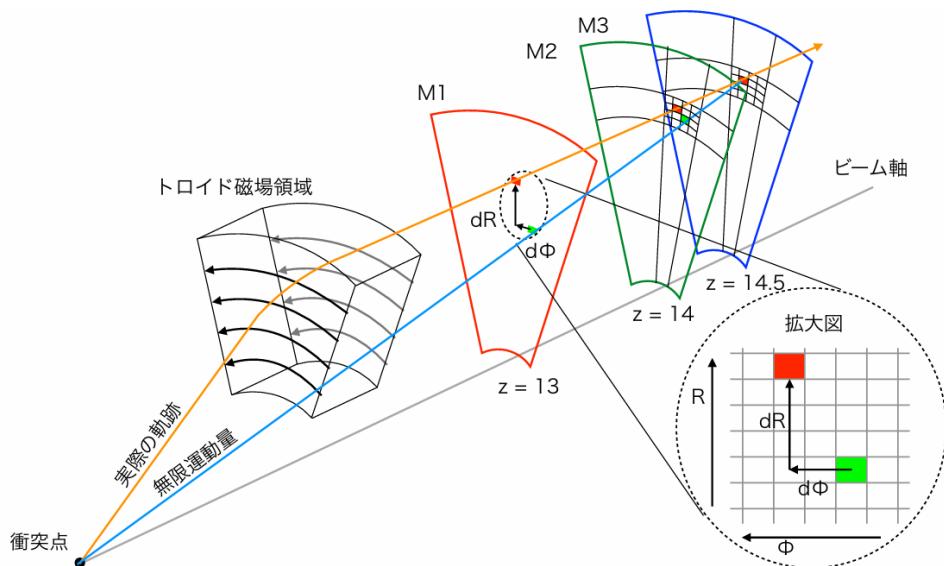


図 2.23: TGC 検出器で測定されるミューオンの位置・角度情報の模式図 [28]。

LHC-ATLAS 実験の Run 1 では TGC BW のみを用いたパターンマッチングにより運動量計算が行われていた。しかし、フェイクミューオンによるトリガーレートの増大が問題

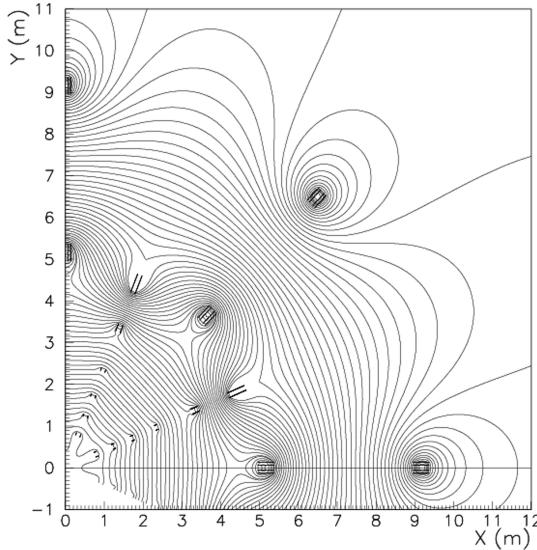


図 2.24: エンドキャップ部におけるトロイド磁場の x 、 y 依存性 [8]。

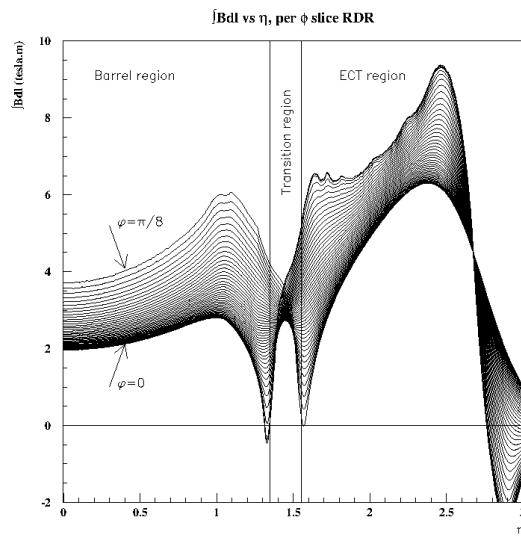


図 2.25: トロイド磁場の η 依存性 [8]。

となっていた。そのため、Run 2 以降のトリガーロジックでは磁場の内側にある検出器とコインシデンスを取るロジックであるインナーコインシデンスが実装され、フェイクミューオンの削減が行われた。図 2.26 は Run 3 中に Tile カロリメータと NSW 検出器との統合によるフェイクミューオンの削減を表したものである。

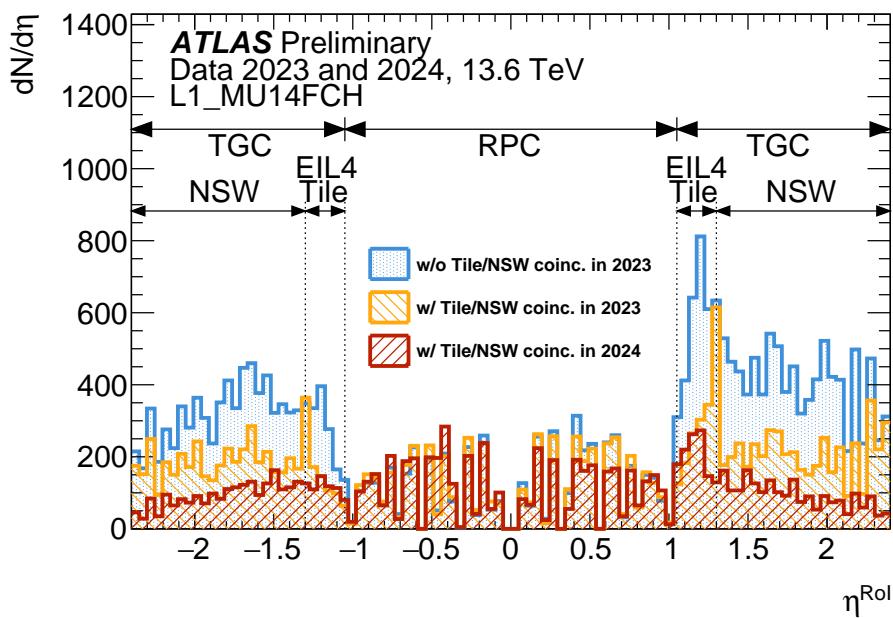


図 2.26: インナーコインシデンスロジックによるトリガーレートの削減 [29]。

2.4.2 インナーコインシデンス

インナーコインシデンスはトロイド磁場領域内部の内層ミューオン検出器と TGC BW 間でコインシデンスを取得するロジックである。内層ミューオン検出器との統合によりフェイクミューオンを棄却し、トリガーレートを削減する。フェイクミューオンとは、衝突事象に由来しない粒子が TGC BW に入射した際に衝突点由来のミューオンであると誤認する現象である。フェイクミューオン事象の一例を図 2.27 に示す。フェイクミューオンの生成原因としては、ビームパイプ由来の中性ハドロンが ATLAS 検出器内部の構造と散乱を起こすことで荷電粒子が生成される事例や、陽子–陽子衝突点やハドロンカロリメータでの散乱で生じた中性ハドロンが検出器の構造体で荷電粒子を散乱させる事例などである。生成された荷電粒子はエンドキャップトロイド磁石の磁場で軌道を曲げられ、あたかも衝突点から飛來したミューオンのように TGC BW に入射する。フェイクミューオンはほとんどが陽子で構成されており、磁場の向きが A-side と C-side で鏡像であることから、A-side 側で特に多く検出される。トロイド磁場領域内部のミューオン検出器と TGC BW の情報を統合することでフェイクミューオンと衝突点由来のミューオンを効率的に区別することができる。

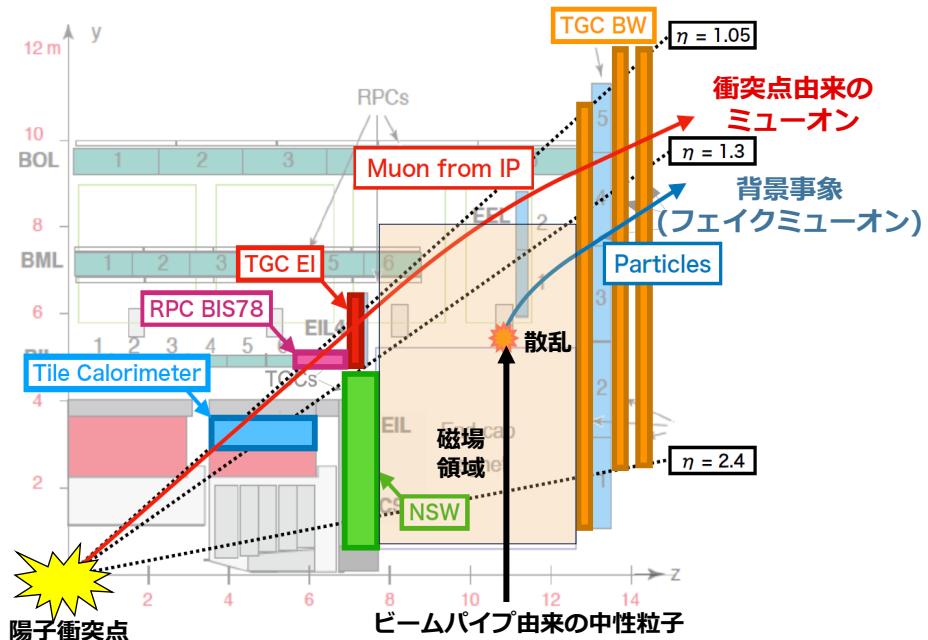


図 2.27: フェイクミューオントリガーの一例 [30]。図の事例はビームパイプ由来の中性ハドロンが ATLAS 検出器内部の構造と散乱を起こすことで荷電粒子が生成される事例である。

2.4.3 トリガーレートの削減と統計量の変化

トリガーレートを削減すると LHC-ATLAS 実験で取得可能な積分ルミノシティが増加することがわかっている。デッドタイムとは読み出しアルゴリズムの実装により、L0A の発行を中断してフロントエンド回路から信号の読み出しを一時停止する時間のことである。つまり、デッドタイム中のバンチ交差は読み出しを行わないイベントとなる。デッドタイムには simple デッドタイムと complex デッドタイムがある。バンチ交差を取得した際、取得したバンチ交差の後ろの数バンチは読み出しアルゴリズムの都合で取得不可能となる。その際に生じるデッドタイムが simple デッドタイムである。また、complex デッドタイムとは一定のバンチ交差の間で取得するバンチ数を制限することにより生じるデッドタイムである。図 2.28 は、Run 3 における初段トリガーのレートと complex デッドタイムの関係を表したグラフである。興味のあるデータはすべて取得するのが理想だが、連続してデータの取得を行うと後段トリガーのキャパシティを超えてしまう。また、ノイズにより異常な数のトリガー判定が発出されてしまう事象も起こりうる。そのため、各トリガーシステムは一定の間隔で取得するバンチ数の上限を決めている。simple デッドタイムはトリガーレートに対して線型に、complex デッドタイムは指数的に増加する傾向があり、データ取得可能な時間が急激に減少してしまう。トリガーレートを低下させることで、物理データの取得をスムーズに行い、積分ルミノシティを増加させることができる。本研究で開発するインナーコインシデンスによるフェイクミューオンの除去は、トリガーレートの削減を行い、デッドタイムを削減する。そのため、物理的価値のあるデータ収集に大きな役割を果たす。

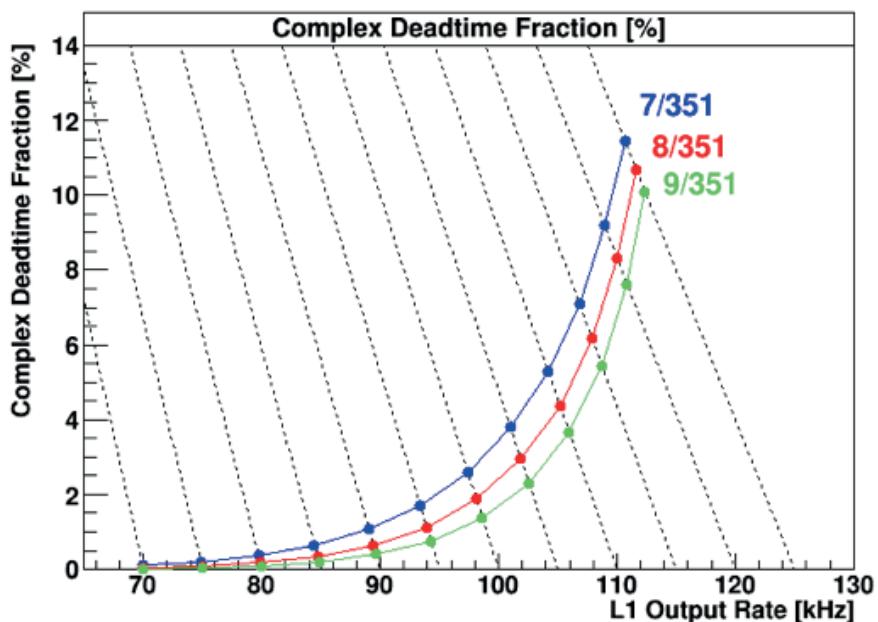


図 2.28: Run 3 における液体 Ar カロリメータの初段トリガーレートと complex デッドタイムの割合の関係 [7]。図中の青、赤、緑の曲線はそれぞれ 351 バンチ交差あたり 7, 8, 9 バンチ以上の L0A が発行された場合、読み取りを一時中断する場合を表す。デッドタイムは複合的な要因で発生し、レートに対して指数的に増加する傾向がある。Run 3 において初段トリガーのトリガーレートは 100 kHz で設計されており、トリガーレートが大きくなるにつれ急激にデッドタイムの割合が増加することがわかる。そのためトリガーレートが一定数を超過するとデータ取得できる時間が減少し、積分ルミノシティは減少する。

第 3 章

エンドキャップ部初段ミューオントリガーロジックの開発

本章では Endcap Sector Logic の Virtex UltraScale+ FPGA に実装されるトリガーロジックについて説明する。SL は TGC BW のトリガーセクターの全てのヒットマップを前段回路から受信してトリガー処理を行う。TGC BW の 7 層のヒットパターンを用いて高い位置・角度分解能で飛跡を再構成し、内層ミューオン検出器の情報も活用して、フェイクミューオンを除去しつつ高い検出効率を実現する。トリガーロジックは各バンチ交差の情報をシーケンスで処理する、いわゆるパイプライン処理を行うように実装されており、LHC クロックに同期した 160 MHz で駆動する。

3.1 エンドキャップ部初段ミューオントリガーロジック

SL に実装するエンドキャップ部初段ミューオントリガーロジックを図 3.1 に示す。トリガーロジックは Channel Mapping、Wire/Strip Station Coincidence、Wire/Strip Segment Reconstruction、Wire Strip Coincidence、インナーコインシデンス、Track Selector の 6 段階で構成されている。Channel Mapping ロジックは PS board から受信したヒットデータを後段のロジックに適した形にマッピングする。マッピングされたデータは Wire/Strip Station Coincidence で各ステーション内の代表点を決定する。Wire/Strip Segment reconstruction で代表点を結び合わせてミューオンの飛跡再構成を行い、陽子衝突点と M3 の代表点を結ぶ直線と無限運動量飛跡とのなす角を求める。

Wire Strip Coincidence では Wire/Strip Segment reconstruction で計算した角度情報を統合してミューオンの横運動量 p_T 閾値を計算し、トリガー判定を行う。Channel Mapping から Segment reconstruction に至るまで、TGC BW の情報のみを用いてミューオンの飛跡再構成を行う。一連のロジックを TGC BW Coincidence と呼び、最大 180 個のミュー

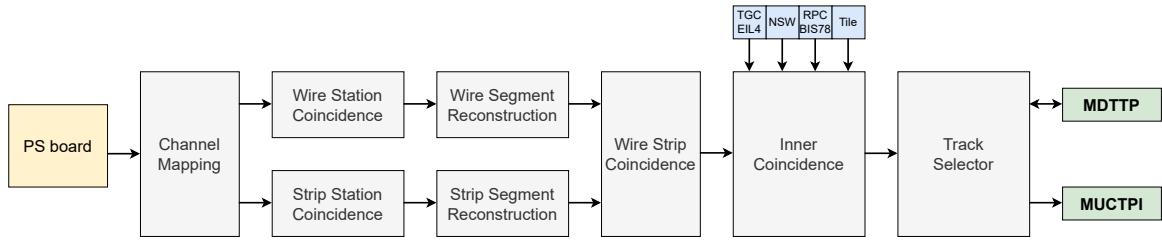


図 3.1: SL に実装されるファームウェアロジックのブロック図。PS board から送信されたヒット情報を各ロジックで順番に処理する。

オン飛跡候補を出力する。インナーコインシデンスはトロイド磁場内に設置されたミューオン検出器との間でコインシデンスを取るロジックである。最大 180 個の飛跡候補から、コインシデンスを取得したミューオン飛跡候補のみ選択して最大 112 個の飛跡候補まで絞る。ミューオン飛跡候補は最終的に Track Selector で 6 個に絞り込まれる。絞り込まれたミューオン飛跡候補のうち 3 つが後段の MDTTP に送信され、高い精度で p_T が計算される。MDTTP は演算後、飛跡候補を SL に返送する。SL は MDTTP 帰ってきた飛跡候補 3 つと、MDTTP に送信しなかった飛跡候補 3 つを合わせて MUCTPI に送信する。

各トリガーロジックは Virtex UltraScale+ FPGA の異なる SLR にそれぞれ実装される。各ロジックが実装される SLR の関係を図 3.2 に示す。Channel Mapping から始まり、Wire-Strip Coincidence に至る TGC BW Coincidence は SLR0, 2, 3 に実装されている。TGC のトリガーセクターのうち、Endcap ϕ_0 , ϕ_1 からの信号をそれぞれ SLR0,2 で、Forward からの信号を SLR3 で処理する。インナーコインシデンスと Track Selector は SLR1 に実装される。SL と、MDTTP や MUCTPI の間でミューオン飛跡候補を送受信するインターフェイスも SLR1 に実装される。

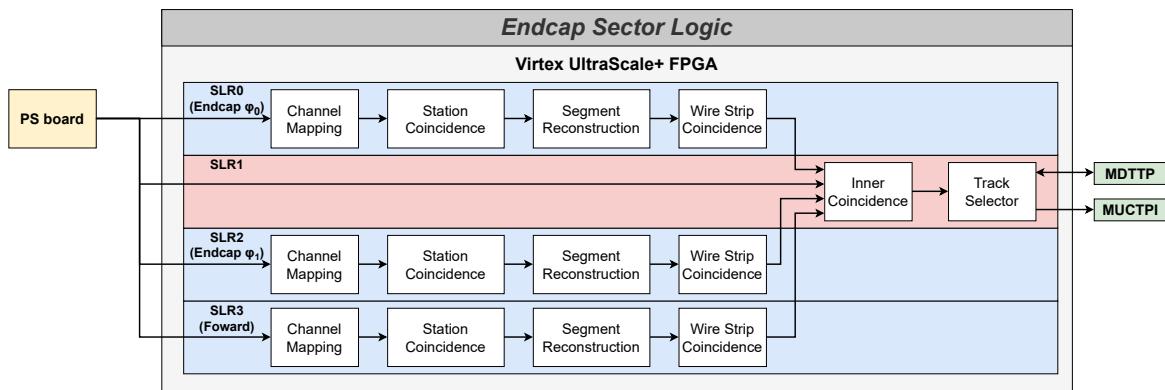


図 3.2: 各 SLR に実装されるトリガーロジックの概要。4 つの SLR のうち、3 つの SLR で TGC BW の飛跡再構成を実行する。残る 1 つの SLR では内層ミューオン検出器とコインシデンスを取るロジックが実装される。

3.1.1 Channel Mapping

Channel Mapping では PS board から SL に転送されたデータを、Station Coincidence 以降のロジックで処理しやすいフォーマットにマッピングする。TGC BW 検出器は η 方向対して、M1 は 4 つ、M2、M3 は 5 つずつチェンバーが設置されている。各チェンバーは不感領域を生み出さないために、わずかに重ねて設置されている。トリガーロジックで処理する上で重複領域が存在することは不都合なので、OR を取って信号の重複を解消する。ワイヤーに関しては重複しているチャンネル間で OR を取る。ストリップは各チェンバーで処理されるが、 η 方向に曲げられたミューオンは複数のチェンバーに跨ってヒットを残す場合がある。このような場合には、ステーション間でコインシデンスを取る際に M3 のチェンバーに対して適切な領域の M1、M2 の情報を参照することが難しくなる可能性がある。そのため、図 3.3 のように、M1、M2 のチェンバー間で OR を取って広い領域の情報を取得可能な状態にする。

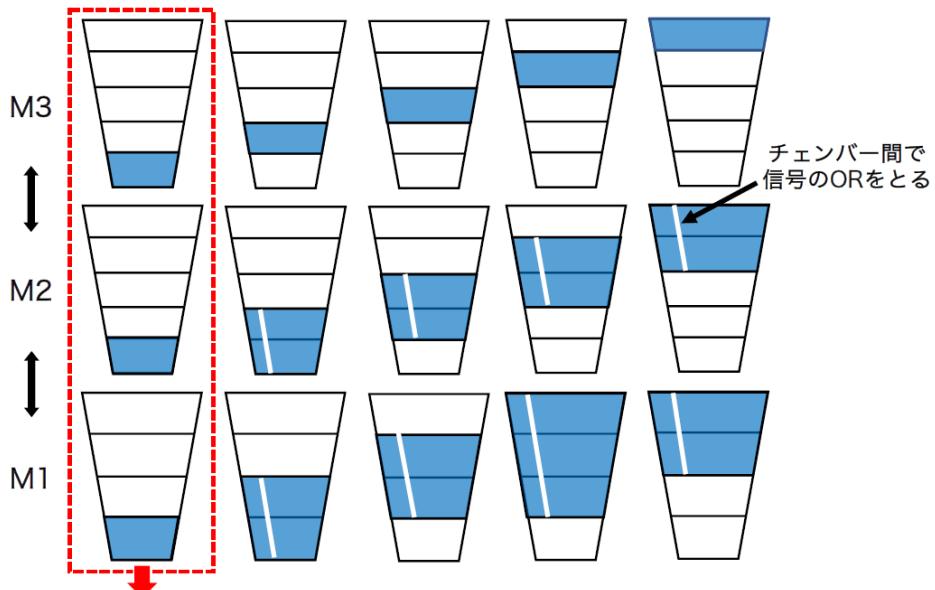


図 3.3: チェンバー間のストリップに対する OR の取得方法 [30]。M1、M2 で広い範囲を参照できるように、OR を取った情報を後段に送る。M3 は OR を取らない。

3.1.2 Station Coincidence

Station Coincidence は各ステーション内の各層のヒット点から代表点を算出するロジックである。代表点を用いることで位置分解能の向上と、後段のロジックで扱うデータを削減することができる。TGC BW のワイヤーは η 方向に、ストリップは ϕ 方向にずらして配

置かれている。ステーションは M1 が 3 層、M2、M3 が 2 層構造を持っており、ワイヤーとストリップそれぞれで重複するチャンネル領域を代表点 (staggered channel) とする。各代表点のヒット情報からコインシデンスの取れた代表点の情報を Segment Reconstruction に転送する。一連のロジックを Station Coincidence と呼び、その概要を図 3.4 に示す。

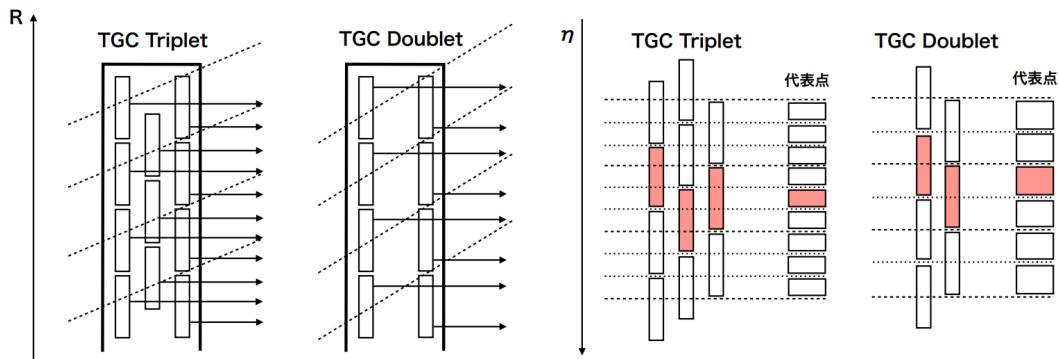


図 3.4: Station Coincidence の概要 [31]。チェンバーは η 方向に対して互い違いな配置となっており、重複部分を代表点として定義している。

3.1.3 Segment Reconstruction

Segment Reconstruction では LUT を用いたパターンマッチアルゴリズムを応用して、Station Coincidence で得た代表点の組み合わせから角度情報を算出する。算出する角度情報は図 3.5 に示すように、M3 の代表点と衝突点を結ぶ直線と、無限運動量飛跡がなす角度である。Wire Segment Reconstruction では $\Delta\theta$ を、Strip Segment Reconstruction では $\Delta\phi$ を計算する。トロイド磁場の影響から TGC BW に入射するミューオンは η 方向に曲げられる傾向がある。そのため、 $\Delta\theta$ はミューオンの p_T を再構成する際の有効な分別変数として機能する。一方、 ϕ に大きく曲げられることは少ない。そのため、 $\Delta\phi$ は衝突点由来のミューオンであることを保証するための補足的情報として利用される。

全ステーションの代表点のヒットパターンを記録し、角度情報を対応付けた LUT を作成することはリソースの消費が激しく現実的ではない。そのため、閾値未満のミューオンが作るであろう角度情報に関してはパターンマッチアルゴリズムで考慮しないこととする。具体的には、 $p_T = 4 \text{ GeV}$ のミューオンによる角度幅を上回る領域はパターンマッチアルゴリズムの考慮に入れない。ストリップは M3 にある 4 個の ID を代表点を基準とし、M2 で 32 個の ID、M1 で 128 個の ID を組み合わせてパターンマッチングを行う領域 (block) 定義する。ワイヤーは M3 が 8 個の ID、M2 は 16 個の ID、M1 は 32 個の ID で定義されている。これら block の概要を図 3.6 に示す。

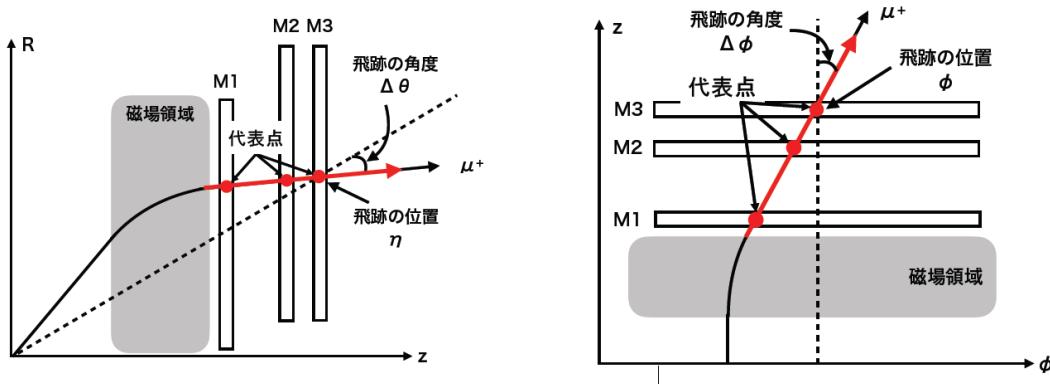


図 3.5: Segment Reconstruction におけるパターンマッチアルゴリズムで用いる角度情報 [30]。赤い線はパターンマッチアルゴリズムによって再構成される飛跡で、黒い点線は衝突点と M3 ステーションの代表点を結ぶ無限運動量飛跡を表す。再構成飛跡と無限運動量飛跡がなす角度をパターンとして LUT に保存する。

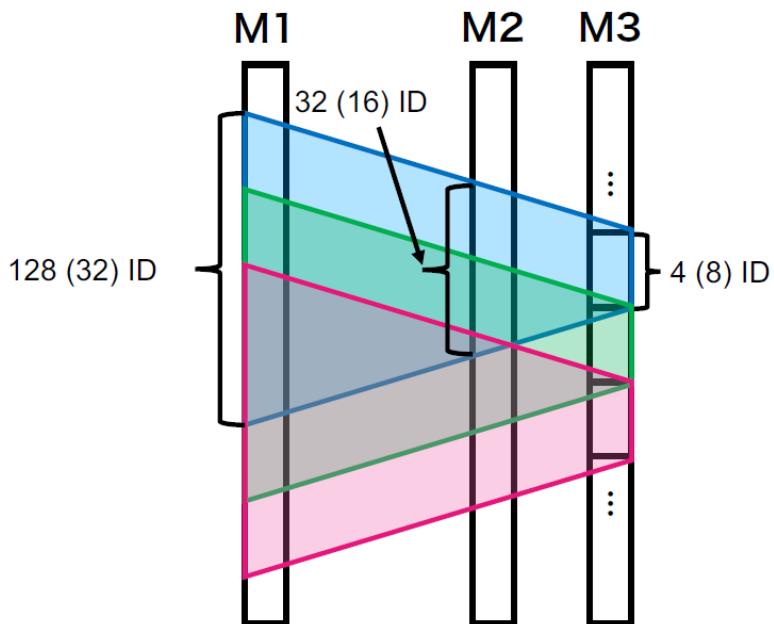


図 3.6: パターンマッチングアルゴリズムを行う領域 [30]。

3.1.4 Wire-Strip Coincidence

Wire-Strip Coincidence では Segment Reconstruction でワイヤーとストリップで独立に取得した角度情報 ($\Delta\theta, \Delta\phi$) を組み合わせて、 p_T 閾値を出力する (パターンマッチアルゴリズム)。飛跡は p_T が大きければ磁場で曲げられにくくなり、角度が小さくなる。この関係を用いて block の組み合わせごとに p_T を定義し、LUT として BRAM に実装する。こ

のように、通過した粒子の飛跡情報を元に横運動量 p_T を取得する LUT を Coincidence Window (CW) と呼ぶ。パターンマッチアルゴリズムについての詳細について、図 3.7 に示す。

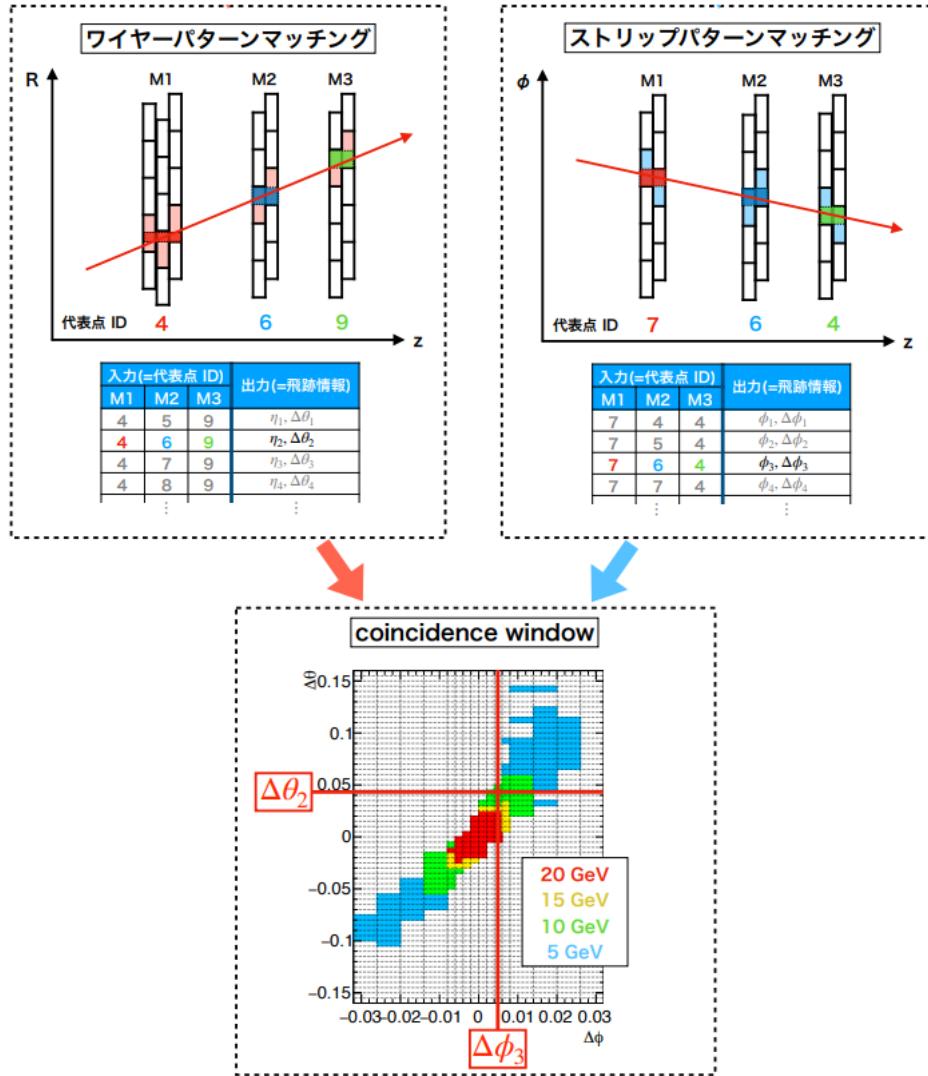


図 3.7: Wire Strip Reconstruction で使用するパターンマッチアルゴリズムの概要 [31]。

Wire-Strip Coincidence は Endcap ϕ_0, ϕ_1 をそれぞれ 35 個、Forward を 8 個に分割した領域 (Unit) で並列に演算する。Unit の模式図を図 3.8 に示す。再構成されたミューオン飛跡候補は Block Selector ロジックで選別される。8 Unit Region で 4 つのストリップと 2 つのワイヤーの組み合わせで再構成される 8 個の飛跡候補から 1 個の飛跡候補に絞る。全 Unit を 8 Unit Region として実装すると FPGA のリソースが足りなくなるため、4 つのストリップと 8 つのワイヤーで構成される 32 Unit Region を定義して 32 個の飛跡候補を 4 個に絞る。32 Unit Region で選択する 4 つの飛跡候補は、2 個は正電荷で残りの 2 個は負

電荷となるように選択する。これは J/ψ 粒子の崩壊時に生じるような、異なる電荷をもつミューオンペアに対する感度を損なわないようにするためである。44 個の 8 Unit Region と 34 個の 32 Unit Region から合計 180 個のミューオン飛跡候補をインナーコインシデンスに転送する。

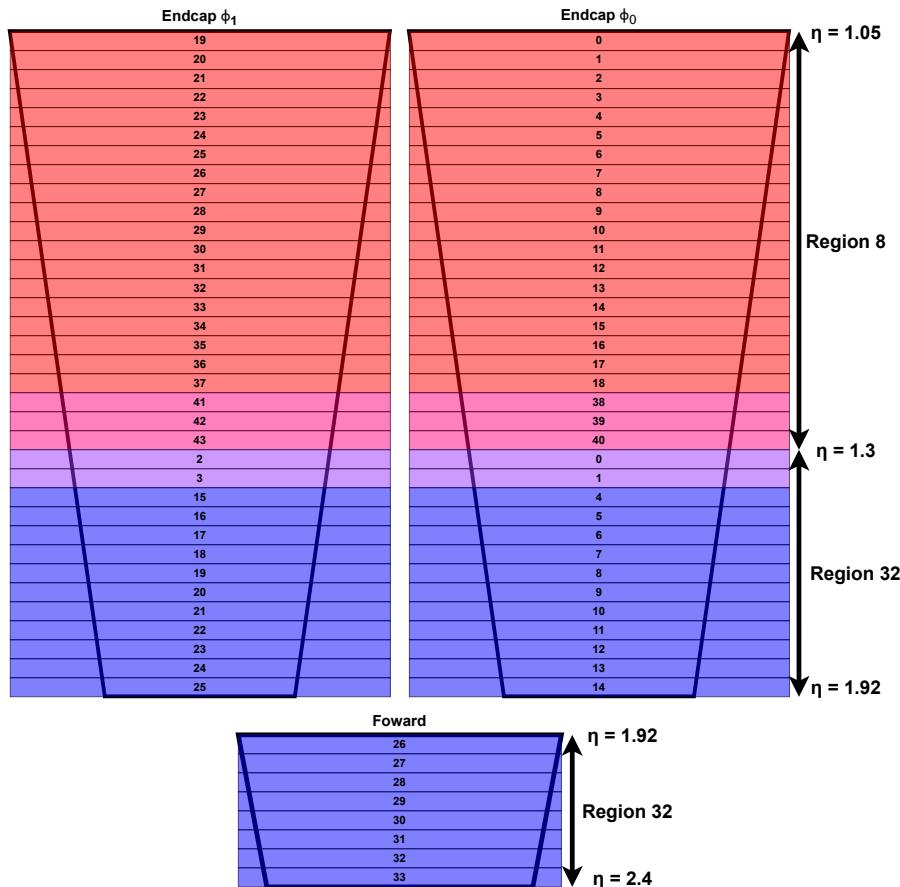


図 3.8: トリガーロジックで使用する Unit の模式図。赤は 8 Unit Region で青は 32 Unit Region である。Region に割り振られている数字はトリガーロジックで使用する Unit のナンバリングである。後段のインナーコインシデンスでも Unit ごとにトリガー判定を行う。

パターンマッチングアルゴリズムで使用する CW

CW はモンテカルロシミュレーションにより作成される。閾値を超えた運動量をもつミューオンが TGC BW に入射した時の角度情報をそれぞれ記録し、 $\Delta\theta$, $\Delta\phi$ の 2 次元分布を作成する。この 2 次元分布を 2 次元ヒストグラム化し、エントリーの多い bin を一つずつ選択する。99 % のミューオンが取得できる bin の集合が完成するまで bin を取り続ける。99 % 取得可能な bin の集合を Window と定義し、各運動量閾値でマージする。20 GeV を超えるミューオンに対しても感度を与えるために 40, 60, 80 GeV 付近のミュー

オンの Window も作成し、各 Window をマージする。現在は 5, 10, 15, 20 GeV の 4 通りで運動量閾値で定義されているが、実験時には 16 通りで実装される。

3.1.5 インナーコインシデンス

インナーコインシデンスは TGC BW と TGC EIL4 (EI)、RPC BIS78、Tile カロリメータ、NSW といった内層ミューオン検出器の情報を統合してミューオンの運動量を再構成する。内層ミューオン検出器との統合によりフェイクミューオンを棄却し、トリガーレートの削減を目的とする。実装の詳細については後述する。

3.1.6 Track Selector

Track Selector は再構成されたミューオン飛跡の p_T を参照して、112 個まで絞られたミューオン飛跡候補を 6 個まで絞るロジックである [32]。飛跡候補の選択は図 3.9 に示すような、128 個のインプットと 8 個のアウトプットで構成されるソーティングネットワークが実装されている。回路実装には高速かつ並列なソーティングアルゴリズムである、Batcher 奇偶マージソートが採用されており、16 個の 8-key sorting network と 15 個の 16-key merging network で構成される。8-key sorting network と 16-key merging network を図 3.10 と図 3.11 に示す。最終的に得られた 6 個の飛跡候補のうち、3 個は MDTTP に送信される。MDTTP で精密な測定を行った後 SL に再度転送され、6 個のミューオン飛跡候補を MUCTPI に送信する。

3.1.7 TGC 読み出し回路

TGC 読み出し回路とは、TGC BW のフロントエンドからトリガー判定を受けたバンチ交差情報の読み出しを制御するロジックである。トリガーロジックと同じ Virtex UltraScale+ FPGA に実装される。FPGA に実装される読み出し回路の SLR レイアウトを図 3.12 に示す。L0 Buffer, Derandomizer, Zero Suppressor & Serializer, Event Builder ロジックからなる。Event Builder 以外の読み出し回路は SLR0, 2, 3 に実装され、Event Builder は SLR3 に実装される。

L0 Buffer は BRAM をリングバッファとして実装し、PS ボードから入力されるビットマップを保存する。書き込みポインタのアドレスを 40 MHz の LHC クロックに併せて 1 ずつ増やし、ポインタが最後尾まで移動したら再び先頭の番地にポインタを戻す。この処理を繰り返すことで BRAM をリングバッファとして機能させることができる。読み出しポインタは 240 MHz の LHC クロックと同期して、4 バンチ交差のアドレスをポイントしたあと 2 アドレス戻り、2 クロックの間ポインタを止める。この動作により L0A が発行さ

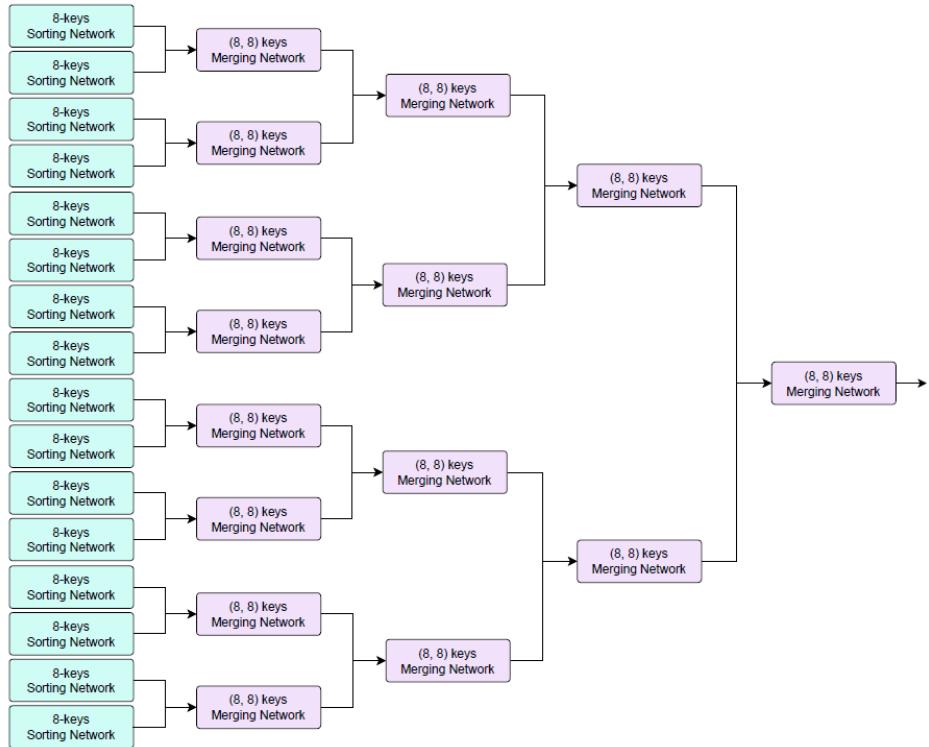


図 3.9: Track Selector の概要 [33]。16 個の 8-key sorting network と 15 個の 16-key merging network の組み合わせで構成される。最大 128 個の飛跡情報を入力して、8 個の飛跡情報まで飛跡候補を出力するロジックである。実際に入力される飛跡候補の個数は、インナーコインシデンスで再構成された 112 個が最大である。

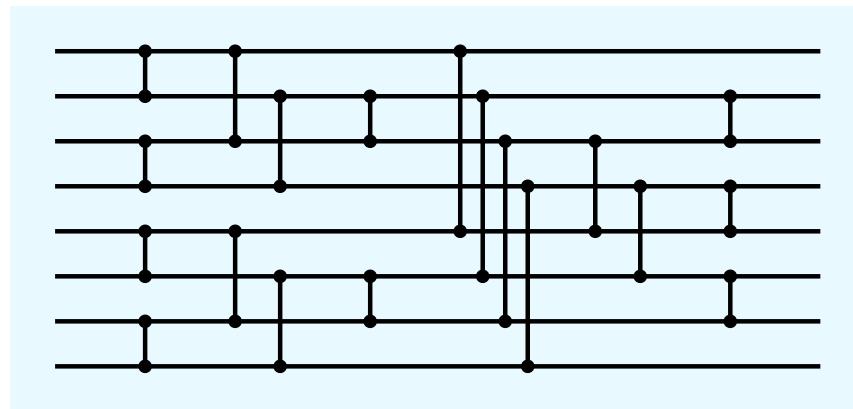


図 3.10: 8-key sorting network の概要。横線はワイヤーで、縦線はコンパレータを示す。

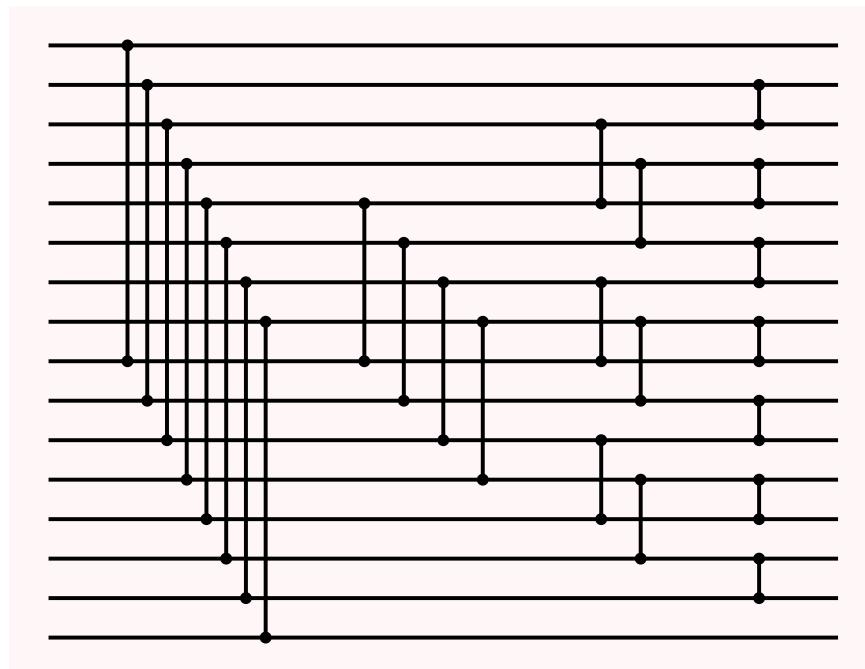


図 3.11: 16-key merging network の概要。横線はワイヤーで、縦線はコンパレータを示す。

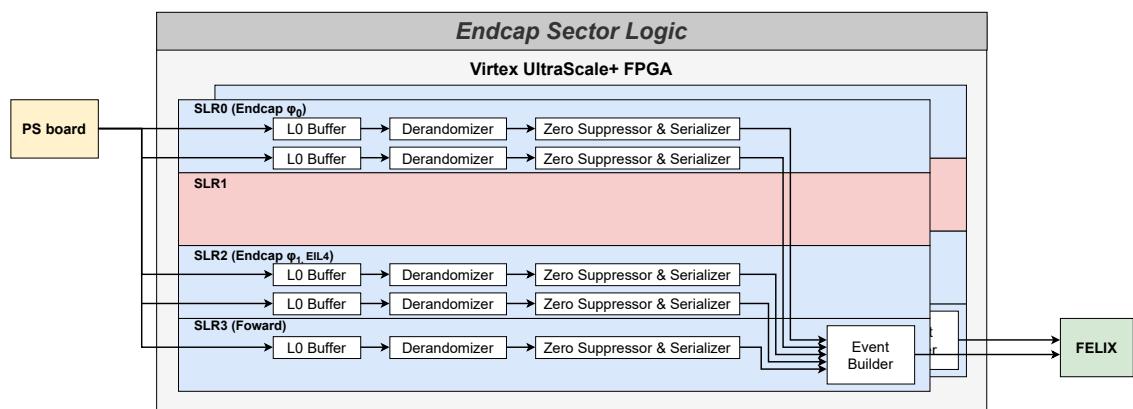


図 3.12: 各 SLR に実装される読み出し回路の概要。PS board から送られてきてビットマップを SLR0, 2, 3 に実装された 5 つのロジックチェーンで処理する。SL から FELIX に送信する読み出し用のリンクは 2 本用意されているため、2 つの読み出しロジックが並列実装されている。ロジックを経て圧縮、LOA シグナルが配布されたバンチ交差と、前後のバンチ交差と 2 つ次のバンチ交差情報を整形・圧縮し、SLR3 の Event Builder を経て FELIX へ送信する。

れたバンチ交差に加え、前後 1 バンチ交差と、2 つ後のバンチ交差の計 4 バンチ交差を読み出すことができる。L0 buffer からの読み出し以降、全ての読み出しロジックは 240 MHz で駆動する。

Derandomizer は先入れ先出しのメモリ構造を持つ FIFO に L0 buffer からの出力と L0A と一緒に発行される L0ID やバンチ交差情報などの信号を格納するロジックである。Derandomizer は後段回路の処理待ちバッファーとして機能する。FIFO の深さは 512 で、オーバーフローが発生する頻度が十分に小さくなり、かつリソースの使用量が最低限で済む値として設定されている。

Zero Suppressor は Derandomizer から 4 バンチ分の信号を読み取り、データを圧縮するロジックである。4 バンチ交差のデータ (previous · current · next · next-to-next) を 16bit 単位のセルに分割する。ヒット情報のあるセルのみを選択して、previous · current · next · next-to-next が区別可能になるように BC タグをつけて、1 ワード 32bit ごとに FIFO に入力する。TGC から送信されるビットマップのうち、99 % はヒット情報のないゼロデータなので Zero Suppressor を通すことでデータを大幅に圧縮することができる。Serializer は 240 MHz クロックに同期して FIFO からデータを出力するロジックである。Zero Suppressor & Serializer ロジックはヒット数に比例してレイテンシが増加する。

Event builder では前段のロジックから送信してきたデータを、FELIX に送信するフォーマットに変換するロジックである。また、FELIX と TGC を結ぶリンクのうち読み出し回路で使用可能なものは 2 本の 32bit リンクのみである。そのため、Event builder では前段のロジックを通過した 5 リンク分のデータを直列化する。各レーンの読み出し FIFO から順番に信号を取得していく。信号を取得した FIFO は読み出しフラグを立てる。全ての読み出しフラグがアクティベートされたことを確認してから一斉にフラグをディアクティベートする。このように処理することで異なるバンチ交差が混同することを防いでいる。直列化されたデータはプリアンブルを添付してパケット化し、FELIX に 32bit ずつ送信する。

3.2 インナーコインシデンスの構造

インナーコインシデンスのモジュールの関係を図 3.13 に示す。Global_Pos_decoder モジュールは各検出器で使用されている座標 (ローカル座標) から、ATLAS 検出器全体の座標系 (グローバル座標) に変換するモジュールである。グローバル座標に変換された飛跡情報は Decoder モジュールに転送される。Decoder モジュールでは TGC BW と各検出器との間で η や ϕ といった位置情報や $\Delta\theta$ や $\Delta\phi$ といった角度情報を参照し、差分を取るなどして Coincidence モジュールで演算しやすい形にデコードする。デコード済みのデータをもとに Coincidence モジュールでコインシデンスを取得したかどうかのフラグと、 p_T の

閾値を取得する。コインシデンスの取得方法は各検出器の特性を踏まえて別々のロジックを搭載したモジュールが準備されている。`Which_Inner` でどの検出器の飛跡統合結果を使用するか決定し、出力値を決定する。

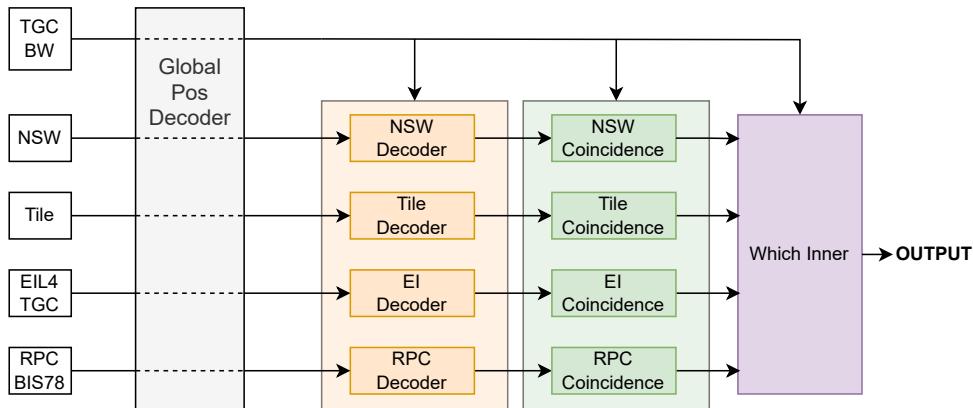


図 3.13: TGC BW と各内層ミューオン検出器の飛跡情報は `Global_Pos_Decoder` モジュールで全ての検出器の座標情報を統一される。その後 Decoder でデータを変換し、TGC BW と各検出器間でコインシデンスを取得する。最終的に `Which_Inner` でどの検出器の情報を使うか決定する。`Which_Inner` から top モジュールにデータの差し戻しが行われ、後段の `Track_Selector` モジュールにミューオン飛跡候補を転送する。インナーコインシデンスは外部からの信号で全体を無効化することができる。その際はコインシデンスを取得せず、ミューオン飛跡候補を最大 112 個まで削減してから `Track_Selector` にデータを送信する。

3.2.1 Global Pos Decoder

各検出器は各々独自の、ローカル座標を用いてトリガー処理を行う。ローカル座標を使用することで、飛跡情報に使用するビットの桁数を絞り、リソースを削減することができる。一方、インナーコインシデンスは複数の検出器の情報を統合するため、ローカル座標のままでは演算に利用することが難しい。そこで、`Global_Pos_Decoder` で ATLAS 実験全体で使用する `global` な座標系に変換する。グローバル座標は η が $-2.7 \sim 2.7$ の範囲で 14bit、 ϕ が $0 \sim 2\pi$ の範囲で 12bit と定義されている。TGC BW のローカル座標とグローバル座標の概要について、図 3.14 に示す。TGC BW におけるローカル座標は、トリガーセクター内の基準点からの差で定義されている。そのため、各トリガーセクターの座標情報とローカル座標を線型結合することでグローバル座標を取得することができる。

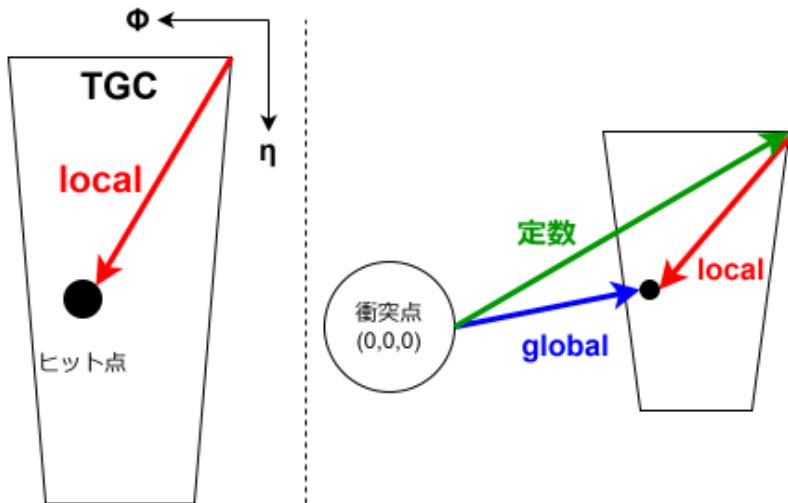


図 3.14: ローカル座標は特定の検出器を用いたトリガーロジック内でのみ使われる座標系である。図に置ける"定数"はローカル座標の起点となる座標である。ローカル座標から、ATLAS 検出器全体で使用されるグローバル座標への変換は、ローカル座標と"定数"の線形結合で実現される。

3.2.2 NSW と TGC BW 間のコインシデンス

NSW と TGC BW 間のコインシデンスについて図 3.15 に示す。NSW からインナーコインシデンスに転送される飛跡候補は最大 16 個である。NSW が送信する飛跡情報には NSW を通過したミューオン飛跡と無限運動量飛跡がなす角度の差である θ_{NSW} と、飛跡の位置情報 $\eta_{\text{NSW}}, \phi_{\text{NSW}}$ から構成される。

インナーコインシデンスではリソースやレイテンシの制約があるため、16 個の飛跡情報を最大 4 個に絞る必要がある。NSW_Decoder モジュールでは、 p_T 閾値が大きい飛跡候補を優先的に選択するために、TGC BW と NSW 間の η 情報の差、 $|d\eta_{\text{NSW}}| = |\eta_{\text{BW}} - \eta_{\text{NSW}}|$ が小さい飛跡を選択する。 $|d\eta_{\text{NSW}}|$ を用いた飛跡選択アルゴリズムについて図 3.16 に示す。また、 ϕ 方向に大きく異なる飛跡を選択しないよう、 $d\phi_{\text{NSW}} (= \phi_{\text{BW}} - \phi_{\text{NSW}})$ が閾値を超えた飛跡情報は棄却される。 $d\phi_{\text{NSW}}$ の閾値は磁場の違いを考慮して Unit ごとに設定される。上記の手順で 4 つに絞られた飛跡情報は、対応する $|d\eta_{\text{NSW}}|$ と $\Delta\theta$ を NSW_Coincidence モジュールに転送する。

NSW_Coincidence モジュールで行う運動量再構成について、図 3.17 に示す。TGC BW と NSW のコインシデンス取得と運動量再構成には $d\eta_{\text{NSW}} (= \eta_{\text{BW}} - \eta_{\text{NSW}})$ を使用する。また、衝突点で生成されたミューオンは大きな物質量を持つカロリメータなどで多重散乱を引き起こす。ミューオンが磁場で受けた影響と散乱を区別するために $\Delta\theta_{\text{NSW}}$ を加えて

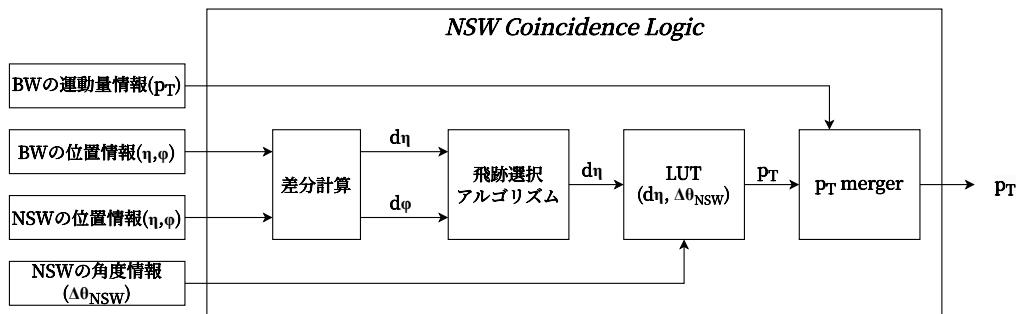
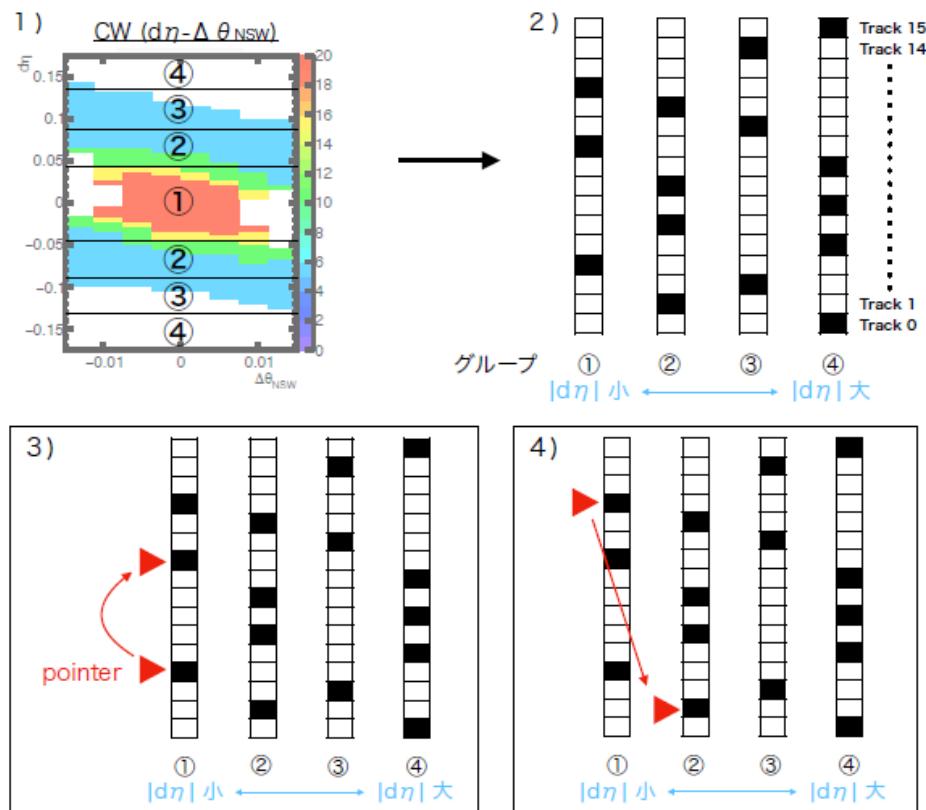


図 3.15: NSW と TGC BW 間のコインシデンスで使用するロジックのブロック図

図 3.16: NSW における飛跡選択ロジック [30]。 $|d\eta_{NSW}|$ に 4 つの閾値を設定し、各閾値を下回ったグループで飛跡候補を分類する。一番 $|d\eta_{NSW}|$ が小さいグループにポインタを合わせ、順番に飛跡を選択する。1 つの飛跡を同一グループ内に複数の飛跡が格納されている場合、ポインタを進めて 1 つずつ候補を選択する。グループ内の飛跡候補をすべて選択した場合、次に $|d\eta_{NSW}|$ が小さいグループにポインタを移し、最大 4 個の飛跡候補を出力する。

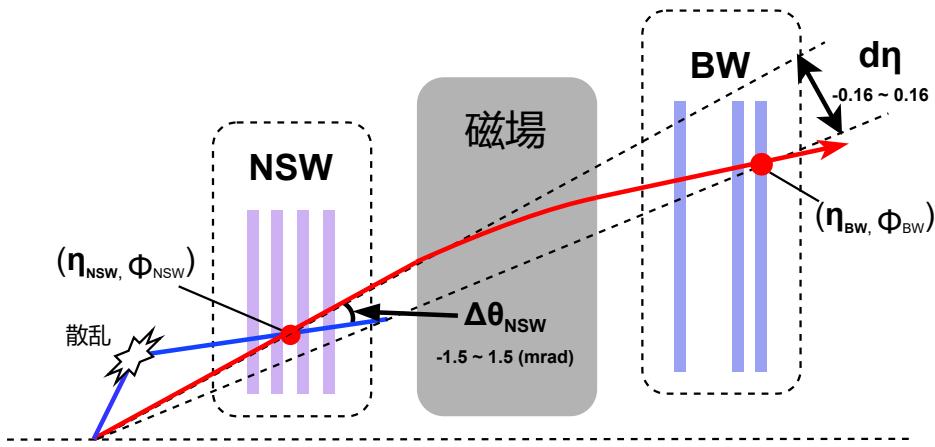


図 3.17: NSW を使用した p_T 閾値再構成の概要。NSW はトロイド磁場領域の内側にあるので η 方向に磁場によって曲げられることはない。TGC BW に入射するミューオンは既にトロイド磁場領域を通過しているので η に曲げられている。そのため、 η の差を取ることでミューオンが受けた磁場の影響を調べることができる。 p_T が大きいほど磁場による湾曲は小さくなるため、 $d\eta_{NSW}$ を用いて p_T 閾値を計算することができる。また、 θ_{NSW} を取得することで散乱によるミューオンの広がりも考慮した p_T 閾値の判定を行う。

CW を作成する。以上により作成した CW をモジュール内部に LUT として実装し、パターンマッチアルゴリズムで p_T 閾値の計算とコインシデンスの判定を行う。CW の具体的な例を図 3.18 に示す。CW は磁場を考慮して、 η, ϕ 方向に対して 9 領域と 6 領域に分けて実装されている(図 3.19、図 3.20)。NSW_Coincidence モジュールでは NSW_Decoder モジュールから送信してきた最大 4 個の飛跡候補を処理する。それぞれの飛跡候補の結果を互いに比較して、 p_T 閾値が最も高い飛跡候補 1 つに絞る。4 つの飛跡候補全てでコインシデンスが取得されなかった場合、飛跡候補を棄却する。

3.2.3 Tile カロリメータと BW TGC 間のコインシデンス

Tile カロリメータと TGC BW 間のコインシデンスのブロック図を図 3.21 に示す。Tile カロリメータに一定のエネルギー閾値を設けて、ミューオンのセル通過を要求することで TGC BW の情報と統合し、衝突点由来ではないミューオンを削減する。Tile カロリメータの D セルを通過する粒子は 99 % がミューオンであり、エネルギー閾値のみでミューオンの通過を検知することができる。

`Tile_Decoder` は使用するセル情報の判別を行い、結果を `Tile_Coincidence` モジュールに送信する。 $1.05 < |\eta| < 1.2$ の領域は Tile カロリメータの D5 セルと D6 セルの両方をミューオンが通過する。また、 $1.2 < |\eta| < 1.3$ の領域は D6 セルのみを通過する。TGC BW

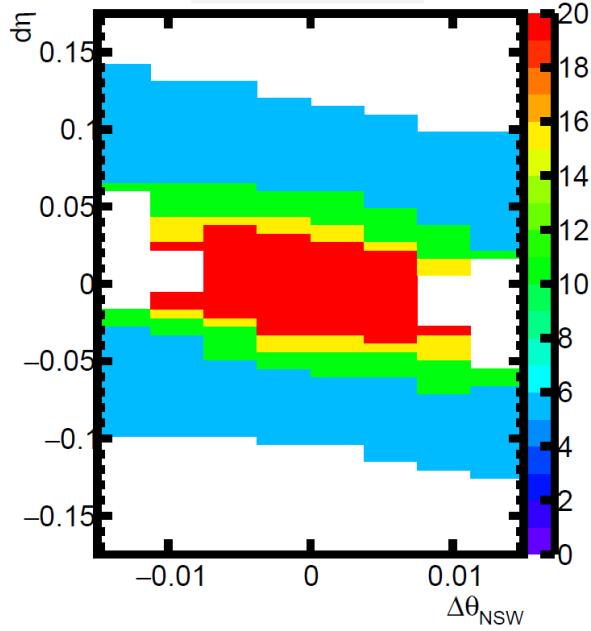


図 3.18: NSW と TGC BW 間のコインシデンスで使用する CW の一例 [31]。 $d\eta_{\text{NSW}}$ は -1.685 から $+1.685$ の範囲、 $\Delta\theta$ は -1.5 mrad から $+1.5$ mrad の範囲で定義されている。

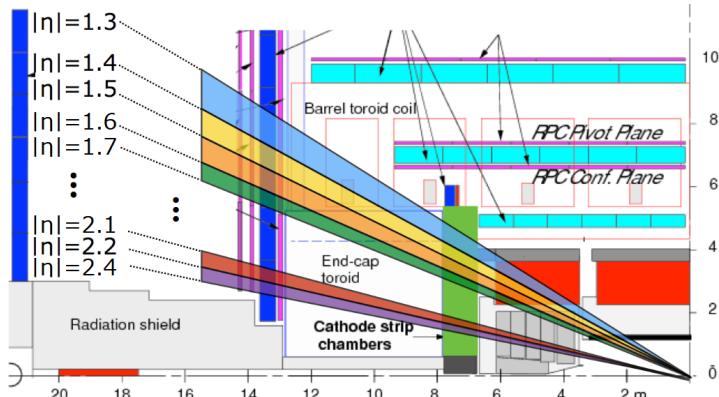


図 3.19: NSW で使用する CW の η 方向分割 [31]。 η 方向に 9 分割されている。 $\eta = 1.9 \sim 2.1$ 、 $2.2 \sim 2.4$ は 0.2 ずつ、その他の領域は 0.1 ずつ領域を分割している。領域は η が小さいほうから 0 ~ 8 の ID で定義される。

の飛跡情報が、 $1.2 < |\eta|$ であるか否かを判別して Position フラグを Tile_Coincidence モジュールに送信する。

Tile_Decoder で取得した Position フラグと TGC BW で再構成された p_T 閾値から、D5 + D6 セルあるいは D6 セルのどちらを用いるか決定する。4 つのモジュールのうちどのモジュールを使用するかの情報は Mask 情報と呼ばれ、LUT によって判定される。使用するモジュールは 2 つか 3 つである。Mask 情報で示されたモジュールのエネルギーが閾値を超えていた場合、ミューオンのヒットがあったとして TGC BW の p_T 閾値を出力する。

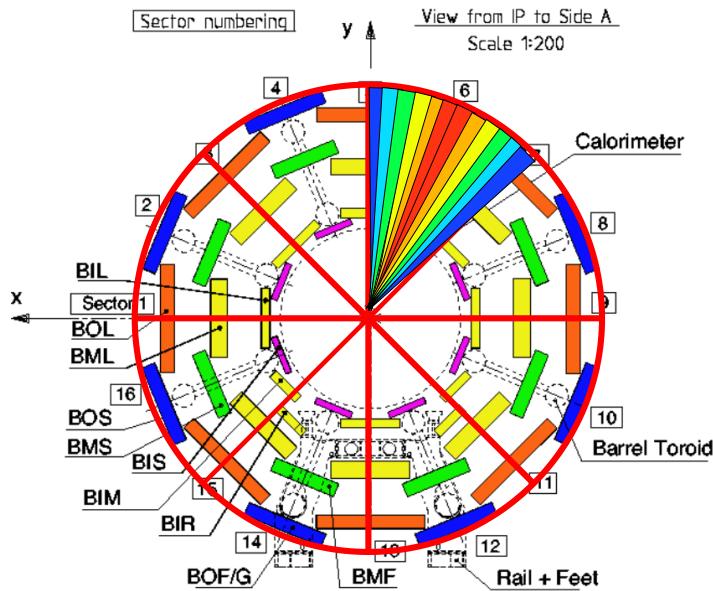


図 3.20: NSW で使用する CW の ϕ 方向分割。 ϕ 方向に 6 分割されている円周を 8 回対称となっているトロイド磁場に併せて 8 分割し、さらに領域を 12 分割した領域で定義される。12 分割した領域を左右の外側から順番に選択し、0 ~ 5 の ID を順番に与える。

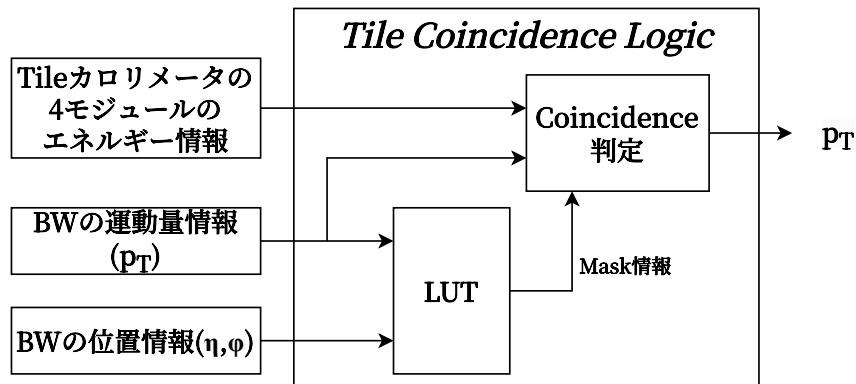


図 3.21: Tile カロリメータと TGC BW 間のコインシデンスで使用するロジックのブロック図

3.2.4 TGC EIL4 (EI) と BW TGC 間のコインシデンス

高輝度 LHC において EI はワイヤー及びストリップから読み出しを行って、全ヒット情報を用いたコインシデンスの取得が可能となっている。TGC BW の飛跡情報 1 個に対して、EI から最大 4 個の飛跡情報が Endcap SL に送信される。EI の飛跡情報は Endcap SL 内で Station Coincidence ロジックを経て TGC BW の飛跡情報と統合する。

TGC BW と EI の η の差である $d\eta_{EI} = \eta_{BW} - \eta_{EI}$ を演算する。ファームウェアでは $d\eta_{EI}$ の下位 7bit の情報を EI_Coincidence モジュールに出力する。 $d\eta_{EI}$ の 8bit 目

以降の信号を参照して飛跡候補を棄却することで、 $d\eta_{EI} > 0.2$ の飛跡候補を棄却することができ、低い p_T 閾値のミューオンを棄却する。

EI と TGC BW 間のコインシデンス取得は $|d\eta_{EI}| = |\eta_{EI} - \eta_{BW}|$ を用いる。低い p_T を持つミューオンは $|d\eta_{EI}|$ が大きくなる傾向があるため、 $|d\eta_{EI}|$ に閾値を設けて低 p_T ミューオンを削減する。

3.2.5 RPC BIS78 と BW TGC 間のコインシデンス

RPC BIS78 を通過するミューオンはトロイド磁場の不均一性から生じる攪乱により ϕ 方向にも大きく曲がる。そのため $|d\eta_{EI}| (= |\eta_{RPC} - \eta_{BW}|)$ だけでなく $|d\phi_{RPC}| (= |\phi_{RPC} - \phi_{BW}|)$ を用いた p_T の判定を行う。また、磁場領域に入射する際の η, ϕ 方向の角度情報である $\Delta\eta, \Delta\phi$ と $|d\eta_{EI}|, |d\phi_{RPC}|$ との相関関係を CW とし、出力値を相補的に活用して性能をさらに向上させている。

3.2.6 Which Inner ロジック

内部ミューオン検出器は図 3.22 に示すように $|\eta| < 1.3$ の領域において、各検出器のカバレッジは η, ϕ に依存する。そのため、Unit の座標に対応する内部ミューオン検出器を適切に選択する必要がある。これらを取りまとめるロジックを搭載したモジュールが **Which_Inner** モジュールである。**Which_Inner** モジュールは並列に走る各検出器のコインシデンスマジュールの結果を統合して、どの検出器とのコインシデンス結果を最終出力とするか決定する。実装は LUT で行う。

3.3 インナーコインシデンスの開発

本研究開始時、インナーコインシデンスは予め決められていたコンセプトに即したファームウェアの開発が進んでいた。しかし、信号間のタイミング調節や各信号線の幅を合わせる作業などが不完全で、実際に信号を入出力することができない状態であった。そこで、インナーコインシデンスのファームウェアにレジスタの追加などによるタイミングの調節と、Vivado シミュレーション [34] を用いた入出力値の比較によるロジックの検証を行った。各内層ミューオン検出器とのインナーコインシデンスのうち、NSW と Tile カロリメータに対するロジックを開発・検証した。

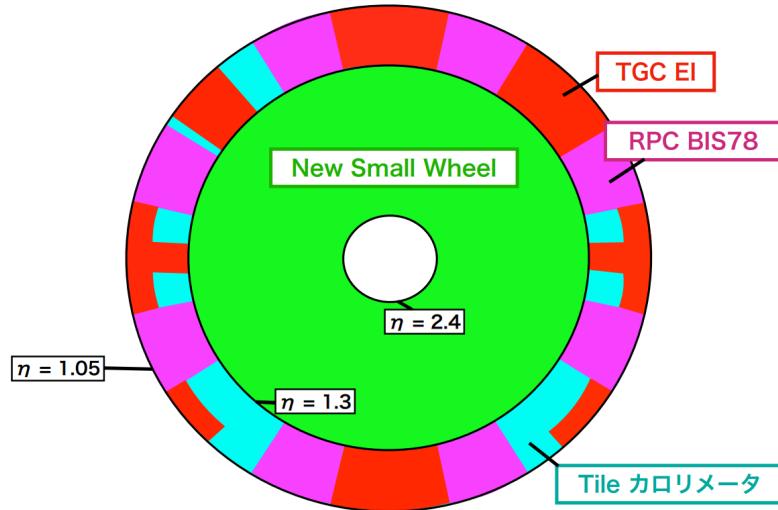


図 3.22: η, ϕ 方向に対する内層ミューオン検出器のカバレッジ [31]。各検出器がお互いを補う形で配置されている。ただし、Tile カロリメータは $|\eta| < 1.3$ 全域にわたってカバーしている。

3.3.1 インナーコインシデンスのレイテンシ調節とタイミング制約の解決

インナーコインシデンスで使用する各モジュールについて、レイテンシ調節が不完全であったため、正確な信号が得られない状況であった。そのため各モジュール間のレイテンシを調節し、信号間のタイミングのずれを解消した。NSW の Coincidence ロジックのレイテンシについて図 3.23 に、Tile の Coincidence ロジックのレイテンシについて図 3.24 に示す。Tile の Coincidence ロジックのレイテンシは NSW の Coincidence ロジックと比較して Decoder モジュールで 1 クロック、Coincidence モジュールで 1 クロック分短い。そのため、`Which_Inner` に実装したシフトレジスタで `Tile_Coincidence` モジュールの結果を遅延させて NSW の結果を得るまで待機させている。シフトレジスタは深さ N, 幅 M の 2 次元配列を実装し、N bit の信号を格納する。クロック信号を受ける度に隣の配列に移動させる。端にたどり着いたらデータを取り出すことで M クロックのレイテンシが実現する。レイテンシと格納する信号の長さは可変となっている。

p_T 閾値比較ロジックの改善

最大 4 つの飛跡候補は 1 クロックずつ Coincidence モジュールに転送される。Coincidence モジュールはパイプライン処理によって 4 つの飛跡候補を 1 クロックずつ出力する。出力された最大 4 つの飛跡候補は p_T 閾値の比較により 1 つに絞られる。従来は、図 3.25 に示すようなフィードバック処理により p_T 閾値の比較を行っていた。しかし、フィードバックを用いた手法は同じレジスタを何度も使いまわす処理が行われるため、パイプライン

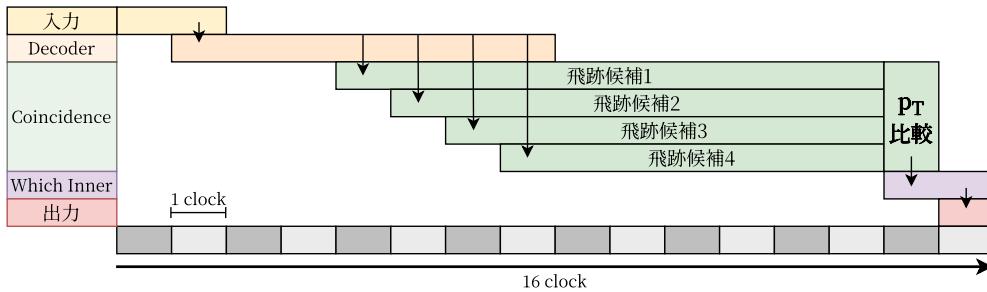


図 3.23: TGC BW と NSW 間のコインシデンス取得ロジックの各モジュールにおけるレイテンシ。NSW から送信してきた信号は TGC BW で再構成された飛跡と比較するために各 Region の Decoder へ 1 クロックかけて分配される。Decoder モジュールでは 4 クロックかけて飛跡候補を選択し、1 クロックずつ 4 つの飛跡候補を出力する。各飛跡候補は 1 クロックおきに 1 つの Coincidence モジュールに入力される。各飛跡候補はパイプライン処理で順番に Coincidence の取得が実行される。TGC BW の飛跡情報との統合を行って得られた p_T の比較を 1 クロックで行う。Coincidence モジュールのレイテンシは 11 クロックである。Which_Inner で他の内層ミューオン検出器の結果と統合して出力を行う。入力から出力までのレイテンシは 16 クロックで固定されている。

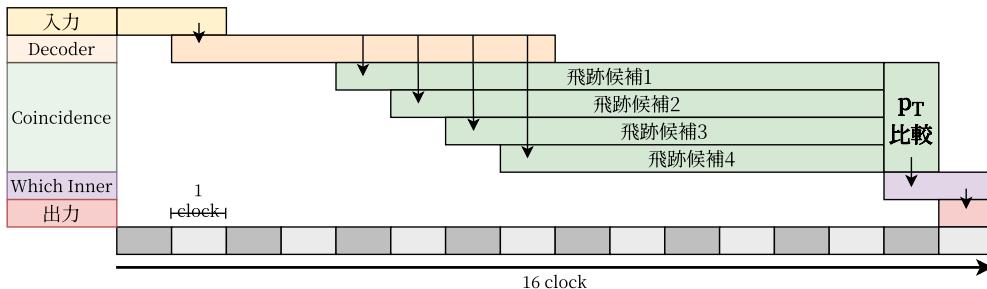


図 3.24: TGC BW と Tile 間のコインシデンス取得ロジックの各モジュールにおけるレイテンシ。NSW と同様に、Decoder を介して Coincidence で演算を行い、最大 4 つの飛跡候補間で p_T 閾値の比較を行う。遅延により、NSW の結果と足並みを揃えて Which_Inner ロジックで処理する。入力から出力までのレイテンシは 16 クロックで固定されている。

ン実装されたロジックと相性が悪く、タイミング制約も厳しくなる可能性があった。そのため、図 3.26 に示すような、フィードバックを行わないロジックに変更した。

タイミング制約違反の解決

FPGA はプログラミングによって任意のデジタル回路を構築することができる。しかし、1 クロック周期の間に処理が完了しないロジックが実装されている場合、タイミング制約違反が発生する。実際、インナーコインシデンスでタイミング制約違反が発生していた。そこで、シフトレジスタによる遅延の最適化、不要なレジスタの削除、信号線の幅の見直し、

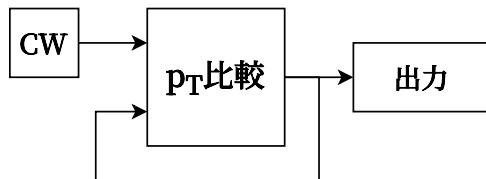


図 3.25: 変更前の p_T 閾値比較ロジック。入力された p_T 閾値は 1 つ前のクロックで得た飛跡候補による p_T 閾値と大きさを比較し、大きいほうを出力する。出力値をフィードバックし、次の飛跡候補の p_T 閾値と比較する。フィードバックを 3 回繰り返すことで 4 つの飛跡候補と p_T 閾値の比較を行う。



図 3.26: 変更後の p_T 閾値比較ロジック。それぞれの飛跡候補から計算した p_T 閾値を異なるレジスタに格納してシフトレジスタによる遅延を与えてタイミングを調節する。全ての飛跡候補の演算が完了してから比較を行う。レジスタの使用量とレイテンシが増加するが、フィードバック処理を行わないと一方通行のロジックとなっており、タイミング制約が改善される。

p_T 閾値比較ロジックの変更により、インナーコインシデンスで発生していたタイミング制約違反を解決した。図 3.27 はタイミング制約違反解決時のタイミング制約に関する情報である。

3.3.2 インナーコインシデンスに対する機能追加

インナーコインシデンスに対して機能の追加を行った。NSW の Coincidence ロジックに、新しく $d\phi$ の閾値で飛跡候補を棄却するロジックと、搭載する LUT の設定を外部から行うためのロジックを追加した。また、既存の CW を LUT の設定に使用する COE ファイル [35] に変換する手法も開発した。

$d\phi_{\text{NSW}}$ 閾値の実装

NSW_Decoder モジュールは 16 個の飛跡候補を $d\eta_{\text{NSW}}$ の大きさのみで 4 個の飛跡候補にソーティングするロジックなので、 ϕ 方向に大きく離れた飛跡候補を選択してしまう可能性があった。そこで、 $d\phi_{\text{NSW}} (= \phi_{\text{BW}} - \phi_{\text{NSW}})$ を計算し、閾値を超えた飛跡情報は棄却するロジックを追加した。閾値は各 Region ごとに設定可能となっている。

| Design Timing Summary | | |
|--------------------------------------|------------------------------------|---|
| Setup | Hold | Pulse Width |
| Worst Negative Slack (WNS): 0.039 ns | Worst Hold Slack (WHS): 0.001 ns | Worst Pulse Width Slack (WPWS): 0.512 ns |
| Total Negative Slack (TNS): 0.000 ns | Total Hold Slack (THS): 0.000 ns | Total Pulse Width Negative Slack (TPWS): 0.000 ns |
| Number of Failing Endpoints: 0 | Number of Failing Endpoints: 0 | Number of Failing Endpoints: 0 |
| Total Number of Endpoints: 1479355 | Total Number of Endpoints: 1454709 | Total Number of Endpoints: 744358 |

図 3.27: インナーコインシデンスの最適化後のタイミングに関するステータス。hold slack はいわゆる信号の待ち時間であり、開発ツールの最適化により 0 に近い数字を取る。negative slack は信号が次のクロックまでに間に合わない時間を表しており、負の値になるとタイミング制約違反となる。図のステータスは最も遅いレジスタの Negative Slack が正となっているため実装に問題がないと判断できる。Pulse Width はクロック信号の品質を確かめるもので、High と Low にどれだけの間とどまるかを示す。正の値を取っており、やはり実装に問題がないことを示す。

LUT controller ロジックの実装

インナーコインシデンスに実装された BRAM や URAM に LUT を書き込むための LUT_controller ロジックを実装した。LUT の設定には RAM ID を指定して LUT の書き込みを行うモジュールを決定する。1 つのモジュールに複数の LUT が実装されている場合は wr activate フラグで LUT を指定する。インナーコインシデンスに実装されている LUT と RAM ID、wr activate フラグの関係を表 3.1 に示す。

表 3.1: インナーコインシデンスに実装されている LUT の一覧表。 $d\eta$ 、 $d\phi$ は内層ミューオン検出器と TGC BW の η 、 ϕ の差を表す。 $\Delta\eta$ 、 $\Delta\phi$ 、 $\Delta\theta$ は飛跡の角度情報を表す。Which は内層ミューオン検出器の名前ではなく、Which_Inner モジュールを示す。

| 検出器 | 役割 | 使用する情報 | RAM ID | wr activate |
|-------|----------------------|--------------------------------------|---------------|-------------|
| Tile | Mask 情報の出力 | $p_{T_{BW}}$ 、 $p_{T_{NSW}}$ | 0x0 ~ 0x3B | 3'b001 |
| EI | CW | $d\eta_{EI}$ | 0x100 ~ 0x13B | 3'b001 |
| | BW と EI 間の p_T 統合 | $p_{T_{BW}}$ 、 $p_{T_{EI}}$ | | 3'b010 |
| RPC | CW | $d\eta_{RPC}$ 、 $d\phi_{RPC}$ | 0x200 ~ 0x23B | 3'b001 |
| | CW | $\Delta\phi_{RPC}$ | | 3'b010 |
| | CW | $\Delta\eta_{RPC}$ | | 3'b011 |
| | BW と RPC 間の p_T 統合 | $p_{T_{BW}}$ 、 $p_{T_{RPC}}$ | | 3'b100 |
| NSW | CW | $\Delta\theta_{NSW}$ 、 $d\eta_{NSW}$ | 0x300 ~ 0x38D | 3'b001 |
| | BW と NSW 間の p_T 統合 | $p_{T_{BW}}$ 、 $p_{T_{NSW}}$ | | 3'b010 |
| Which | 使用する検出器の選択 | $p_{T_{BW}}$ | 400 ~ 0x43B | 3'b001 |

CW の LUT 実装

CW は縦軸と横軸で異なる物理量を与えた 2 次元座標で定義される。一方で、LUT は RAM に実装されるため、RAM のアドレスと出力値を対応付けた実装が必要となる。そのため物理量を元に作成された CW は 1 対 1 対応で LUT を実装することができない。そのため、CW の各 bin をアドレス順に並び替える処理を行ってから、COE ファイルにフォーマットを合わせて実装する。COE ファイルは RAM に初期値を与えるためのファイルで、アドレスとデータを同時に定義して格納することができる。

トリガーロジックファームウェアにおける LUT の運用を図 3.28 に示す。2 つの物理量を組み合わせたビット列をアドレスとして RAM に入力し、出力値を得る。2 つの物理量は正負の値を取るため符号付きのビット列として定義される。しかし、結合して RAM に入力する際には、0 から始まる正の数として処理される。CW と COE ファイルの仕様の違いを踏まえた上で COE を作成するための詳しいアルゴリズムを図 3.29 に示す。

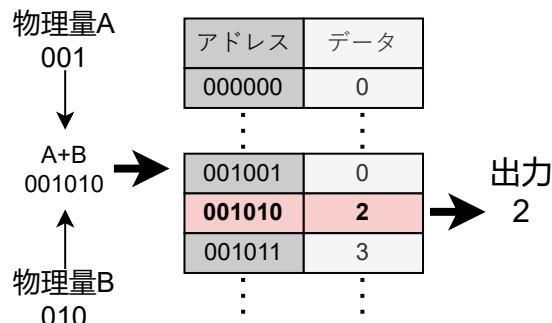


図 3.28: ファームウェアにおける LUT の運用形式。物理量 A,B のビット列を結合して 1 つのビット列とし、RAM にアドレスとして入力する。対応するアドレスに格納されているデータを選択して出力を行う。

NSW dtheta data URAM に対する LUT の実装

NSW_dtheta_data_URAM はインナーコインシデンスに実装されている LUT の一つであり、縦軸は $d\eta_{NSW}$ 、横軸は $\Delta\theta$ で、URAM を使用して実装される。 $d\eta_{NSW}$ と $\Delta\theta$ は NSW の分解能を鑑みてそれぞれ 8、5bit と定義されている。URAM には $8 + 5 = 13$ bit のアドレスが入力されることになるが、URAM はアドレス長が 12bit で固定されているため直接入力することができない。そこで $\Delta\theta$ の符号部で場合分けを行う。まず、CW を $\Delta\theta$ が正の領域と負の領域で分割し、 $8 + (5 - 1) = 12$ bit のアドレスを持った CW を 2 つ作成する。URAM は出力値が 72bit の幅を持ち、 p_T 閾値は 4bit で定義されているため、 p_T 閾

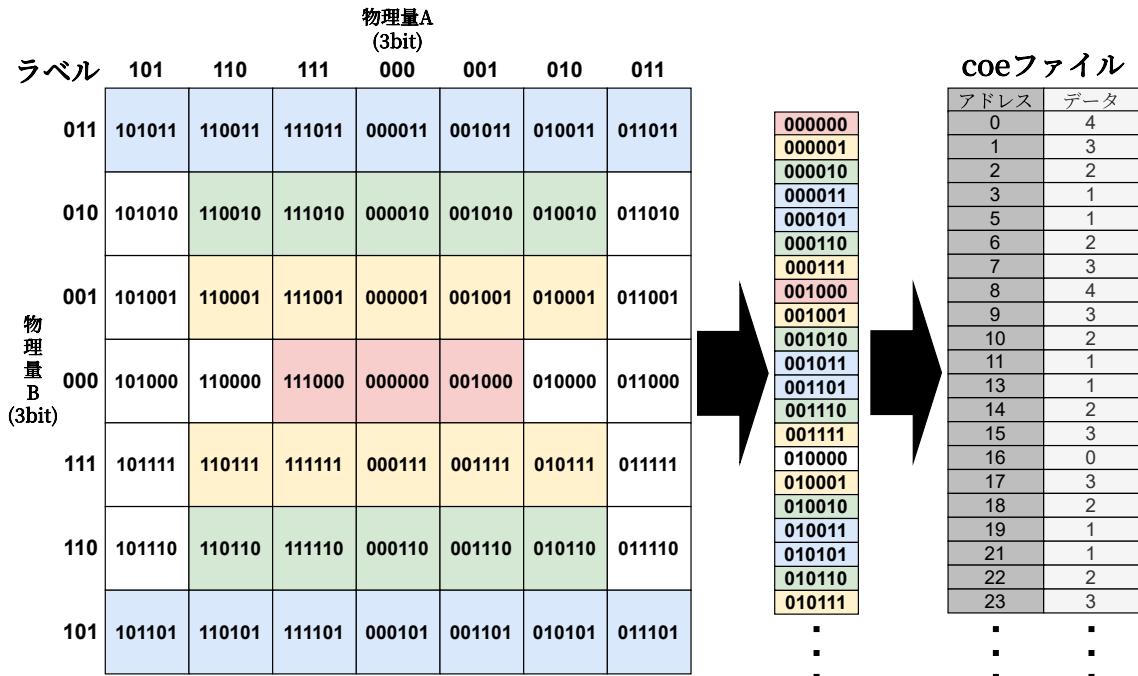


図 3.29: COE 作成手順。縦軸と横軸を bit 長に併せて分割し、ファームウェアで使用する 2 つの物理量のビット値に対応するラベルを与える。縦軸と横軸のラベルを 2 進数表記とし、結合する。結合したラベルは RAM に入力する際のアドレス値に対応する。ラベルを符号なし整数として昇順でソートし、ラベルに対応するデータを書き込むことで COE ファイルを作成する。

値 16 個を 1 つの URAM の出力値に格納することが可能である。そのため、正と負で分割された CW を同時に実装し、出力値を $\Delta\theta$ の符号で場合分けすることで URAM に CW を実装することが可能となっている。

3.3.3 Vivado シミュレーションを用いた出力の確認

ファームウェアの開発に使用する AMD / Xilinx 社の Vivado アプリケーションを用いてインナーコインシデンスの出力検証を行った。ロジックが完成している NSW 及び Tile とのコインシデンス取得ロジックに対して出力検証を行った。NSW_Coincidence モジュールには図 3.18 に示す CW を実装して検証を行った。検証時には $\Delta\theta_{NSW} = 0$ 、 $\eta_{BW} = 1.3$ で固定し、 η_{BW} を変化させることで検証した。本検証を模式的にあらわしたものを見ると図 3.30 に示す。出力される p_T 閾値は $d\eta_{NSW} = 0$ 付近で最大値を取り、 $|d\eta_{NSW}|$ が大きくなるにつれ小さくなることが予想される。CW の出力結果に関わらず、Which_Inner は NSW の出力を採用するように LUT を実装した。検証結果を図 3.31 に示す。図 3.30 と比較しても矛盾のない結果が得られた。これにより、インナーコインシデンスがファームウェアレベ

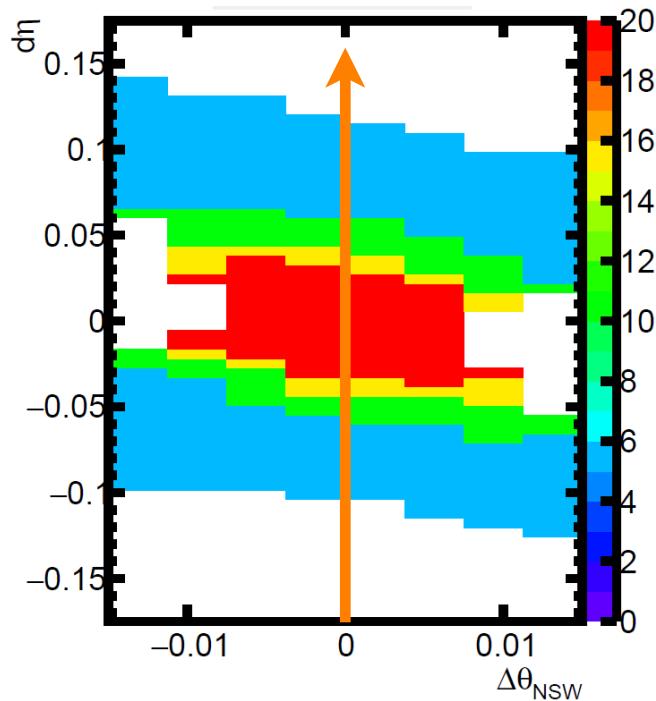


図 3.30: Vivado シミュレーションを用いた NSW コインシデンスロジックの検証方法。今回の検証ではシミュレータに入力する $d\eta$ の値をオレンジ矢印のように変化させた。 $\Delta\theta_{\text{NSW}}$ を 0 で固定しているため、 $d\eta$ 方向にのみ変化する。

ルで正常に動作することを初めて確認した。

`Tile_Coincidence` ロジックの検証では、D5 セル、D5+D6 セル、それぞれでエネルギー閾値を超えているかどうかの 4 パターン。入力する飛跡情報の η によって、D5 セル、D5+D6 セル、どちらのエネルギー閾値を使用するかの 2 パターンに分かれる。よって、`Tile` のコインシデンスロジックでは $4 \times 2 = 8$ パターンが内部で取りうる状態として考えられる。8 つのパターンと出力結果の対応を表 3.2 に示す。Vivado シミュレーションで TGC BW の p_T 閾値を規則的な値とし、8 つのパターン全てで出力を確認した。結果、対応表と矛盾する結果は得られなかったため、ロジックは正常に実装されていると判断した。

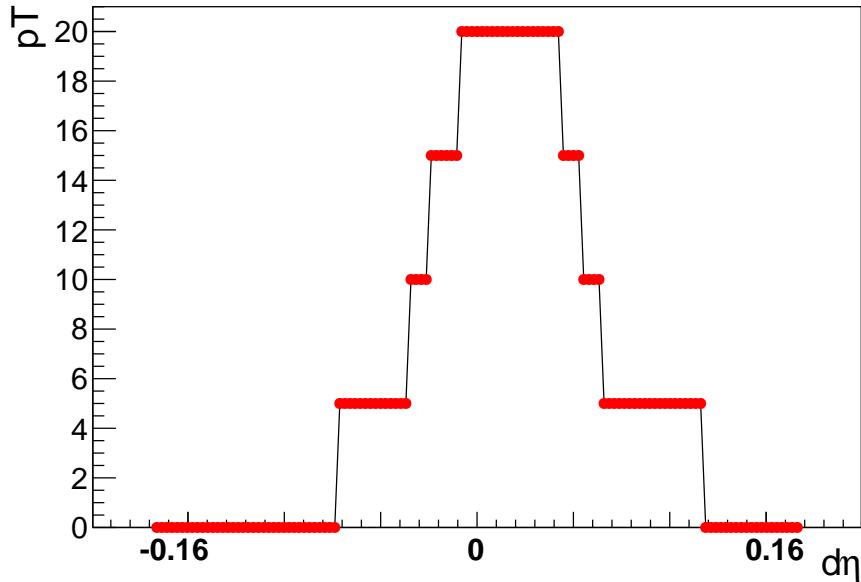


図 3.31: NSW コインシデンスロジックの検証結果。赤い点は出力した $d\eta_{NSW}$ に対して出力された p_T 閾値であり、黒い線は赤点同士を順番に繋いだものである。実装した CW と比較しても出力に矛盾は見られなかった。

表 3.2: Tile コインシデンスロジック検証で使用するセル情報と出力の対応表。

| 使用するセル [†] | D5 セルの閾値 [‡] | D5+D6 セルの閾値 [‡] | 出力結果 |
|---------------------|-----------------------|--------------------------|-----------|
| D5 セル | ○ | ○ | p_{TBW} |
| D5 セル | ○ | × | p_{TBW} |
| D5+D6 セル | ○ | ○ | p_{TBW} |
| D5+D6 セル | ○ | × | 0 |
| D5 セル | × | ○ | 0 |
| D5 セル | × | × | 0 |
| D5+D6 セル | × | ○ | p_{TBW} |
| D5+D6 セル | × | × | 0 |

[†] 使用するセルは入力する η_{BW} が $\eta = 1.2$ を上回っているかどうかで操作する。

[‡] ○ はエネルギー閾値を超過したことを表し、× は下回っていることを表す。ロジックで使用するセルが ○ の場合のみ TGC BW で再構成された飛跡候補の運動量を出力する。

第 4 章

トリガーロジック検証システム

本章では、トリガーロジックの開発を効率化するために開発したトリガーロジック検証システムについて、その要求事項および内部機構のコンセプトを説明する。また、開発したトリガーロジック検証システムを用いたインナーコインシデンスロジックの実装と検証について記述する。

4.1 既存のトリガーロジック検証システム

トリガーロジックを検証する手法は既に複数存在する。既存の検証手法について説明し、インナーコインシデンスを検証する上でどのような利点と課題があるかについて述べる。

4.1.1 実機を用いた検証システム

Endcap SL 実機を用いたトリガーロジックの検証について、概要を図 4.1 に示す。トリガーロジック及び TGC 読み出し回路の出力結果を SL 自身で検証することができる。FPGA で演算したトリガーロジックの出力結果をチップ間通信を用いて MPSoC 内の RAM に送信する。MPSoC に実装されたプロセッサがデータを読み取ることで出力を確認することができる。入力に関しても、本来 PS board から送信されるヒットデータを SL 内部から入力することができる。テストパターンは BRAM に格納されており、外部から自由に書き換えることができる。本検証手法は SL 実機に実装されたロジックを直接検証可能という点で唯一無二の機能を持つが、入力に用いるヒットパターン信号の作成が困難であること。データの出入力の柔軟性が低いこと。数の限られる実機を用いる都合で複数人の同時検証が難しいこと。などの課題点が存在する。

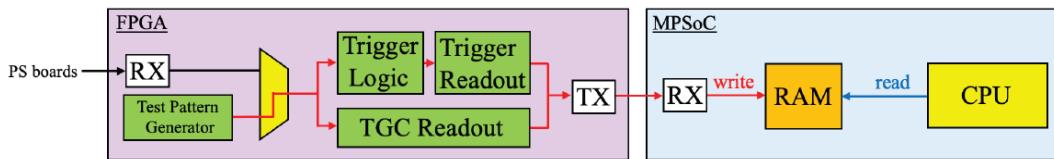


図 4.1: SL 実機を用いたトリガーロジック検証システムの概要 [14]。

4.1.2 Vivado シミュレーション

Vivado シミュレーションは AMD / Xilinx 社が提供するファームウェア開発ツールである Vivado を用いた、ファームウェアシミュレーションである。図 4.2 は Vivado シミュレーション実行時の GUI である。開発したファームウェアをテストベンチソースでインスタンス化して実行する。テストベンチシースは検証するファームウェアにとってインターフェースとして機能し、I/O の制御をすることが可能である。手順はファームウェアのコンパイル、信号線の接続やパラメータ解析を行いインスタンスを展開してスナップショットの作成を行うエラボレーション、ファームウェア内の挙動を再現するシミュレーションを経て GUI への波形表示や外部ファイルへの検証結果の出力を行う。ファームウェアの挙動をよく再現するだけでなく、タイミング制約などのデザインの問題についてもデバッグすることが可能である。ファームウェアを FPGA に実装するには数時間要するため、数分でロジックの挙動を確認できる点で非常に有用である。しかし、テストベンチを介した入出力を行う関係上、入出力の柔軟性が低い。また、検証にかかる時間も決して早くないため大統計のイベントを一斉に検証する際に時間が掛かる問題がある。

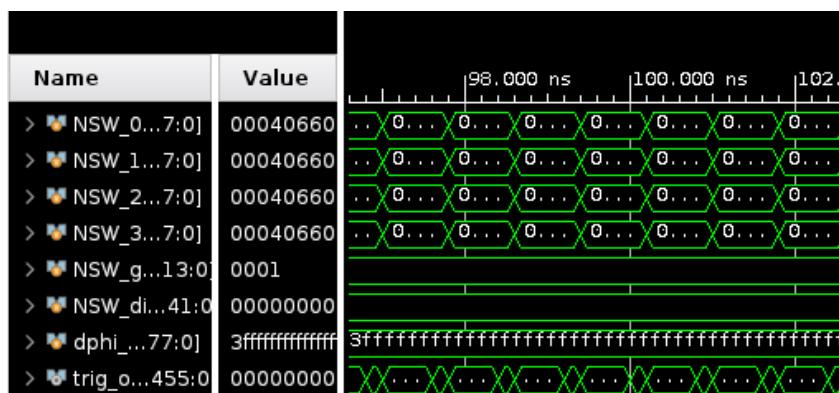


図 4.2: Vivado シミュレーションの出力結果の一例。波形情報を GUI で視覚的に確認することができる他に、テキストファイルに結果を出力してデバッグを行うことも可能である。

4.1.3 ビットワイズシミュレーション

ビットワイズシミュレーション [36] はファームウェアの挙動をソフトウェアで再現した、ビットワイズシミュレータを用いたトリガーロジック検証である。ファームウェアはハードウェア記述言語 (HDL) で実装されているが、ファームウェアを直接使用する検証手法は上記の実機検証と Vivado シミュレーションを用いた検証に限られていた。そのため、HDL を用いずにソフトウェアを用いてビット単位の処理を再現することで高速かつ柔軟な検証を実現したシミュレータがビットワイズシミュレータである。ビットワイズシミュレータは、TGC の各チャネル情報を MySQL [37] を用いたリレーショナル・データベースを用いて実装したテストパターン生成機構と、トリガーロジックの実行、そして出力結果の検証から構成される。概要を図 4.3 に示す。ビットワイズシミュレーションは高速かつ柔軟性の高い検証を行うことができる。1 イベント当たりの検証速度が高速であるため、LUT の検証といった高統計を用いたロジック検証が可能である。一方、トリガーロジックを直接利用するわけではないので、実機と結果が食い違った場合、実機かビットワイズシミュレータのどちらに不具合が存在するのか自明ではなくなる。また、本シミュレータは一部のトリガーセクターに限った実装のみ行われており、カバレッジが制限されている。TGC BW Coincidence ロジックの実装は完了しているがインナーコインシデンスのビットワイズシミュレータ化は完了しておらず、インナーコインシデンスの検証では用いることができない。

4.1.4 ソフトウェアシミュレーション

ソフトウェアシミュレーション [39] はソフトウェアシミュレータを用いたトリガーロジック検証手法である。ソフトウェアシミュレータはトリガーロジック挙動をソフトウェアで再現するシミュレータである。ビットワイズシミュレータと異なる点として、ソフトウェアシミュレータはミューオンの飛跡情報を実数値で扱い、float 演算によって処理する。トリガーロジックのコンセプトを確認するために開発された経緯を持つ。入力信号はモンテカルロシミュレーションを用いて作成される。インナーコインシデンスを含む全てのトリガーロジックが実装されている。検証速度は非常に高速で、200 万イベントを 30 分程度で処理することが可能である。ソフトウェアシミュレーションはトリガーロジックで用いる CW の開発などで応用されている。HDL を用いたトリガーロジックを直接扱うわけではないため、トリガーロジックファームウェアの検証に用いることはできない。

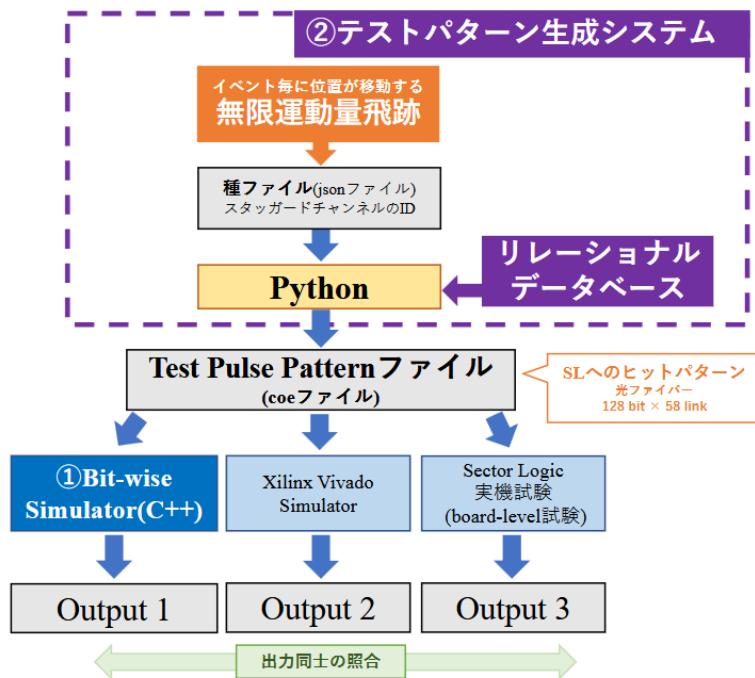


図 4.3: ビットワイズシミュレータを用いたトリガーロジック検証 [38]。テストパターン生成機構は coe ファイルの生成を行う。これは BRAM に書き込みを行う際に使用するデータフォーマットで、実機検証と同じ coe ファイルを入力することで出力結果の比較を行うことができる。

4.2 アクセラレータカード

アクセラレータカードとは FPGA コアを搭載する拡張カードのことである。PC のマザーボードに Peripheral Component Interconnect-Express (PCIe) [40] を通じて直接接続することができる。アクセラレータカードを接続した PC をホストと呼び、ホスト PC とアクセラレータの間でデータ通信を行う。主にサーバーのような大量のデータを処理する必要がある現場で利用される。FPGA は電気回路で直接演算するため、高い並列演算処理能力を持つ。そのため、プログラムを High Level Synthesis [41] と呼ばれる手法で回路実装して高速かつ省電力で演算を行うことができる。一般的に、CPU 以外のリソースを用いるヘテロジニアスコンピューティングで活用される。本研究では内蔵されている FPGA コアに着目した。アクセラレータは PC に直接接続して FPGA を操作可能であるため、トリガーロジックの検証システムを PC 一台で完結するシステムで構築可能である。その上、FPGA をそのまま使用するため柔軟性が高く、任意のトリガーロジックを回路実装することができる。実験で使用するトリガーロジックも変更を加えることなく実装可能である。実験で使用する FPGA と比較しても安価に購入可能である。アクセラレータカードは複数

種類のプロダクトが販売されており、検証システムに使用するアクセラレータを選択する必要がある。AMD / Xilinx 社が開発・販売する、Virtex UltraScale+ FPGA と同様のアーキテクチャで実装された FPGA を内蔵する Alveo データセンターアクセラレータカード [42] を用いることとした。

4.2.1 Alveo アクセラレータカード

Alveo アクセラレータカードは主に U50、U200、U250 そして U280 の型番がある。これらの性能と価格の比較図を表 4.1 に示す。Alveo U200 と U250 と、Virtex UltraScale+ FPGA の比較を表 4.2 に示す。Alveo に実装された FPGA は Endcap SL で使用する Virtex UltraScale+ FPGA と同様のアーキテクチャで開発されている。特に、Alveo U250 は EndCap SL で使用する XCVU13P と同じアーキテクチャの XCU250 が実装されている。FPGA アクセラレータはコンピューティングカードとして大量生産が行われており、競合他社との価格競争も激しい。AMD / Xilinx 社は FPGA アクセラレータにおけるシェアを広げようと販路の拡大を図っていると考えられ、同等のスペックを持つ FPGA コアを単体で購入するよりも低価格で入手できる。価格と性能を総合的に判断した結果、Alveo U200 を使用して開発を行った。Alveo U200 のイメージを図 4.4 に示す。

表 4.1: Alveo シリーズの FPGA と主な型番のリソースと価格の比較表 [43]。Alveo U200 と U250 は DDR メモリのみ、Alveo U50 と U280 は 8GB のフラッシュメモリを搭載している。

| | Aleo U200 | Alveo U250 | Alveo U280 | Alveo U50 |
|---------------|-------------|------------------|-------------|-----------|
| FPGA | XCU200 | XCU250 | XCU280 | XCU50 |
| LUTs (K) | 1,182 | 1,728 | 1,182 | 872 |
| Registers (K) | 2,364 | 3,456 | 2,364 | 1,743 |
| メインメモリ (GB) | 64 | 64 | 40 | 8 |
| value (USAD) | ¥ 1,032,643 | ¥ 1,621,084 [44] | ¥ 1,413,883 | ¥ 552,071 |

4.2.2 XCU200

Alveo U200 に搭載されている FPGA である XCVU200 は集積度を高めるために 3 つの SLR に分割された構造を持つ。SLR のレイアウトについて図 4.5 に、各 SLR ごとのリソース量を表 4.3 に示す。Virtex UltraScale+ FPGA と異なる点は、SLR1 の半分に実装された静的領域 (Static Region) である。Static Region はユーザーによる書き換えが不可能な領域であり、Direct Memory Access Controller (DMAC) [46] がビルトイン機能として実装されている。DMAC は後述する DMA を管理するアプリケーションである。SLR0、

表 4.2: XCU200/250 は Alveo U200/U250 に搭載されている FPGA。両者とも Virtex Ultra Scale+ 9 / 13 と同じアーキテクチャで、実装されているリソースも同等である。XCU200/250 の価格はアクセラレータ本体の価格である。

| | FPGA accelerator | | FPGA core (Virtex UltraScale+) | |
|------------------|------------------|-------------|--------------------------------|--------------|
| | XCU200 | XCU250 | XCVU9 | XCVU 13 |
| LUT (K) | 1,182 | 1,728 | 1,182 | 1,728 |
| Register (K) | 2,364 | 3,456 | 2,364 | 3,456 |
| URAM Blocks (GB) | 960 | 1,280 | 960 | 1,280 |
| value (USAD) | ¥ 494,450 | ¥ 1,637,129 | ¥ 8,111,800 | ¥ 11,421,800 |



図 4.4: Alveo データセンターアクセラレーションカード U200 のイメージ図 [45]。

SLR2 の全域と SLR1 の半分は自由に書き換え可能な動的領域 (Dynamic Region) となっている。本研究ではトリガーロジック検証システムと検証を行うトリガーロジックの双方を Alveo U200 の Dynamic Region に実装した。

表 4.3: XCU200 の各 SLR ごとの動的領域のリソース量 [47]。

| リソース量 | SLR0 | SLR1 | SLR2 |
|-------|-------|-------|-------|
| レジスタ | 776 K | 410 K | 770 K |
| LUTs | 388 K | 205 K | 770 K |
| BRAM | 720 | 420 | 720 |
| URAM | 320 | 160 | 320 |

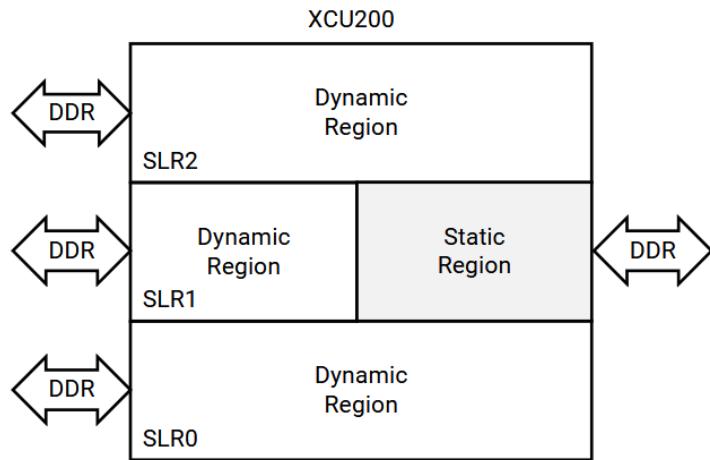


図 4.5: XCU200 の SLR レイアウト [47]。各 SLR はの Dynamic Region と SLR1 の Static Region はそれぞれ DDR と接続されており、データ通信が可能となっている。

DDR (Double Data Rate)

DDR はランダムアクセスメモリ (RAM) の規格の一つで、DDR 規格に則った RAM を DDR メモリと呼ぶ [48]。本論文中で単に DDR と呼ぶ場合は DDR メモリを指す。一般的なコンピューター向けに作成された SDRAM 規格の次世代型規格で、クロックサイクルの立ち上がりと立ち下がりの両方でデータを転送することで、SDRAM と比べて 2 倍の速度でデータを転送する。高速かつ大規模な RAM の規格として広く一般的に使用されており、現在は第五世代の DDR5 まで登場している。開発用 PC と Alveo U200 は両者とも第四世代である DDR4 規格のメモリを搭載している。DDR4 はおよそ 20 GB/s の高速転送に対応している。

ダイレクトメモリアクセス

ダイレクトメモリアクセス (Direct Memory Access、DMA) とは、メモリとメモリの間で、高速データ通信を行うためのシステムである。連続したメモリアドレスにアクセスする際、CPU の負担を軽減しつつスループットを引き上げて、高速でデータ通信が行うことができる。DMA は Direct Memory Access Controller (DMAC) と呼ばれる IC チップによって制御される。図 4.6 はホスト PC の DDR とアクセラレータの DDR 間の DMA を表した図である。

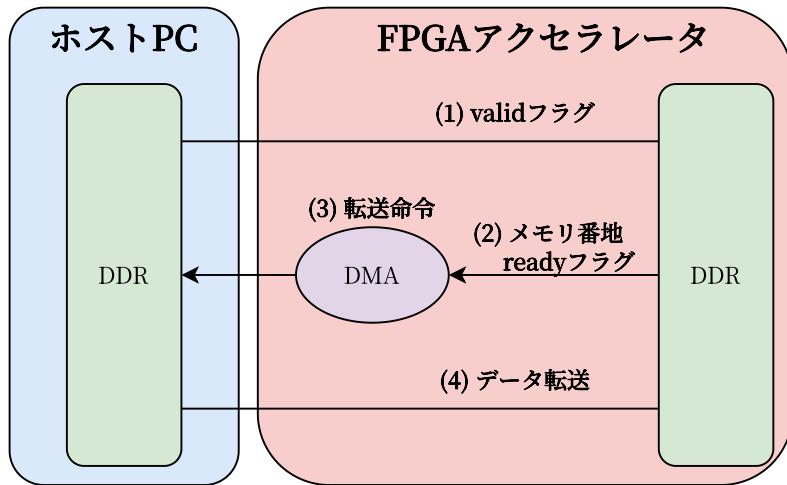


図 4.6: DMA を用いたデータ通信の流れ。ホスト PC 内の DDR は FPGA アクセラレータ内の DDR に、データの存在を外部に示すフラグ情報である valid フラグを送信する。アクセラレータ内の DDR は、データ受信が可能であることを表す ready フラグをメモリ番地の情報と一緒に DMA に送信する。DMA は指定されたメモリ番地のデータを送信するように命令を発する。DMA から命令とメモリ番地を受け取った DDR は送信先の DDR に対して直接データを送信する。FPGA アクセラレータからホスト PC にデータ転送する際も同様の手順で DDR 間の通信が実行される。

AXI4 interface

AXI4 interface は第四世代 AMBA インターフェイス規格である [49]。AMBA は Advanced Microcontroller Bus Architecture の略称で、デバイスに実装されたチップ間の通信やチップの機能管理のために、ARM 社が提供するオープンソースな規格の総称である。AXI4 interface はチップやデバイス間のハンドシェイク型通信インターフェースとして確立している。図 4.7 は AXI4 interface の働きを表した図である。32bit から最大 1024bit のデータを 1 度に送受信できる。今回の検証システムでは、1 回の転送ごとに 32bit のデータを送受信するように設定した。

4.2.3 Vivado

Vivado [50] とは、AMD / Xilinx 社が提供する、デザイン入力、合成、配置配線、検証、シミュレーションなどのツールが統合された FPGA および SoC 向けの設計ソフトウェアである。Vivado を用いたデザイン入力では、VHDL や Verilog などのハードウェア記述言語を使用してデザインを作成し、GUI ベースで IP コアを統合することが可能である。本研究ではトリガーロジックの開発と、トリガーロジック検証システムの開発で使用した。

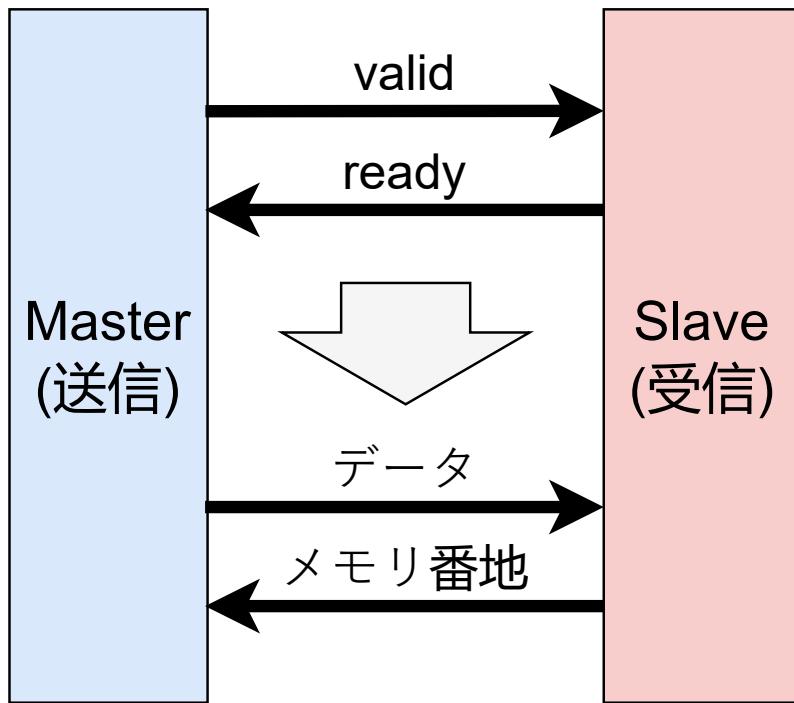


図 4.7: AXI4 interface の概略図。送受信両方に AXI を制御する AXI コントローラが実装される。送信側は Master、受信側は Slave と呼ばれる。Master は送信準備が完了次第 ready フラグを、Slave は受信準備が完了次第 valid フラグを発出する。両者のフラグが発出されるとハンドシェイク成立となる。ハンドシェイクが成立したら、Slave がデータのアドレス番地を Master に受け渡し、Master は指定されたアドレス番地にあるデータを Slave に送信する。

4.2.4 Vitis

Vitis [51] とは、AMD / Xilinx 社が提供する、FPGA および SoC 向けの統合ソフトウェアプラットフォームである。Vitis は、Vivado と組み合わせることで FPGA を用いた AI 演算機能の開発やハードウェアアクセラレーションアプリケーションの開発など、ハードウェアを用いた抽象度の高いデザインの開発を行うことができる。Vitis は、C/C++ で書かれたプログラムを回路実装することができる高位合成も含まれている。本研究では、トリガーロジック検証システムをアプリケーションとして包括し、FPGA アクセラレータに対するロジックの実装と、C++ で書かれた制御用のコードの実装を Vitis 上で行った。

4.3 FPGA アクセラレータを用いたトリガーロジック検証システムの開発

先行研究では FPGA アクセラレータを用いてトリガーロジックの検証が実現可能であることが示されていた [52]。本研究と同じく Alveo U200 の FPGA コアにトリガーロジックを実装して、検証実行にかかる時間、必要なリソース量について研究されている。先行研究の詳細について解説を行う。

4.3.1 先行研究における開発状況

先行研究におけるトリガーロジック検証システムのデザインを図 4.8 に、FPGA コアに実装されたシステムのデザインを図 4.9 に示す。ホスト PC の CPU で生成されたデータは DMA と AXI interface を通じて User IP 部に実装されたトリガーロジックに入力される。データは入力を 2 ポート、出力を 1 ポートで行っており、各ポートの幅は 512bit である。実際に User IP に図 4.10 を実装して検証を行っている。検証時の実行時間を表 4.4 に示す。

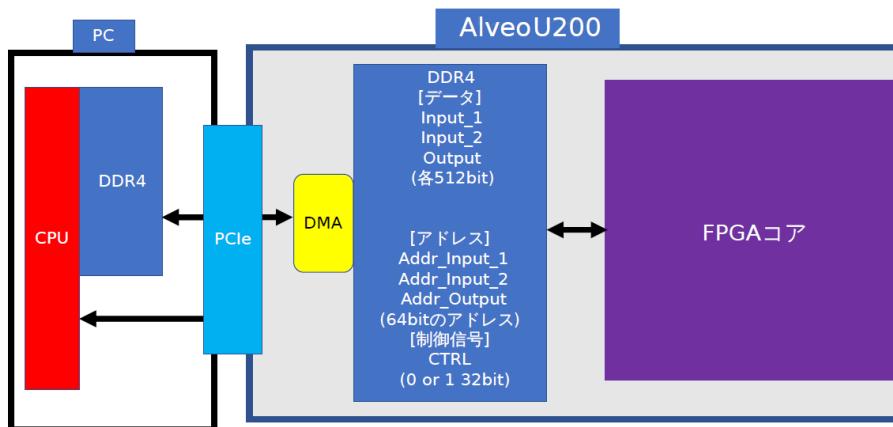


図 4.8: 先行研究におけるトリガーロジック検証システムのデザイン [52]。ホスト PC から Alveo U200 の DDR への通信は PCIe を介した DMA で行う。DDR に転送するデータは 512bit データ 2 つの合計 1024bit である。その他 DDR のアドレス情報や制御用の信号を含めて DDR に転送する。出力である Output は 512bit である。

4.3.2 Run 3 で用いられているトリガーロジックの実装と検証

上記のシステムは User IP としてトリガーロジックを実装することでトリガーロジック検証システムとして機能させることができる。先行研究では LHC-ATLAS 実験 Run 3 で

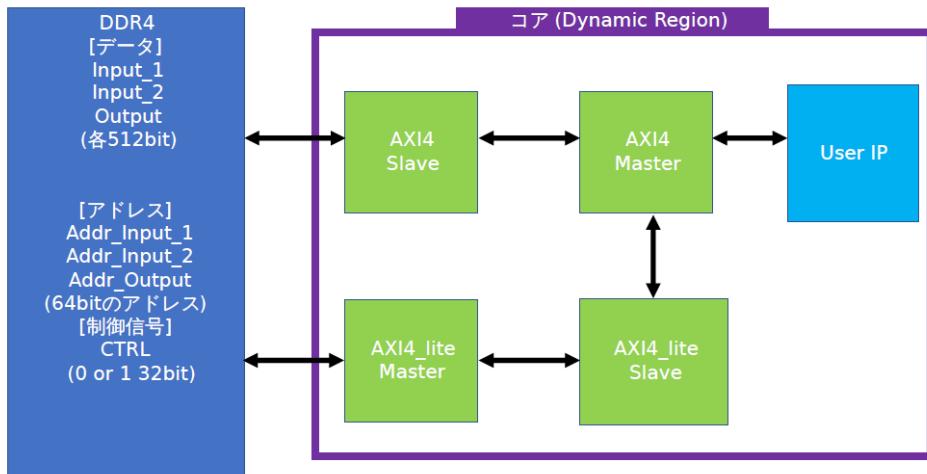


図 4.9: 先行研究における FPGA コアに実装されたシステムのデザイン [52]。Dynamic Region に実装された AXI interface で Alveo U200 内の DDR と通信を行う。User IP 部に任意のファームウェアロジックを実装することができる。

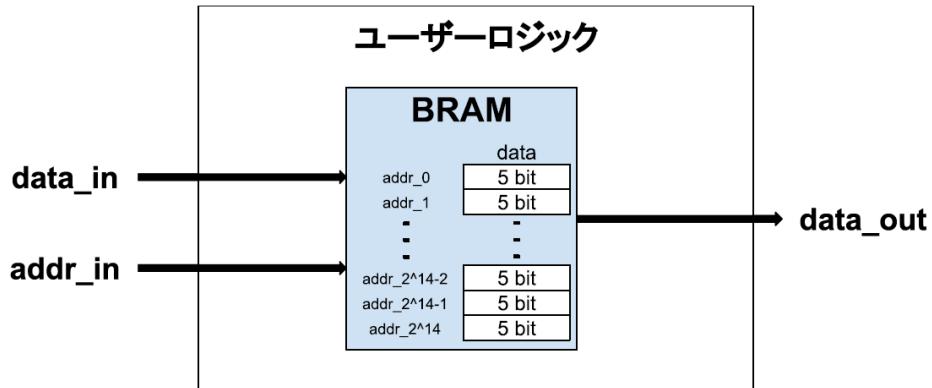


図 4.10: 先行研究において実装された User IP [52]。アドレス幅 14bit、出力幅 5bit の BRAM が実装されている。

表 4.4: アプリケーションの実行時間 [52]。

| 項目 | 実行時間 |
|---------------------------|----------|
| Alveo からのファームウェア情報の読み取り | 17 ms |
| lost と Alveo メモリのバッファ割り当て | 155 ms |
| Alveo への DMA 転送 | 96 ms |
| FPGA コアの処理時間 | 228 ms |
| Alveo からの DMA 転送 | 80 ms |
| その他 | 1,401 ms |
| 合計時間 | 1,977 ms |

使用されている CW を BRAM に書き込むことでトリガーロジックの実装可能性について議論している。BRAM に実装された CW を図 4.11 に、ロジックの概要を図 4.12 に示す。トリガーロジックを Alveo U200 の FPGA コアに実装した際のリソース使用量から、トリガーロジックの搭載可能数について考察している。実装したトリガーロジックのセクター数と BRAM 使用量の比較を図 4.13 に示す。結果として Run 3 で使用されているトリガーロジックを 7.5 セクター分実装可能である。

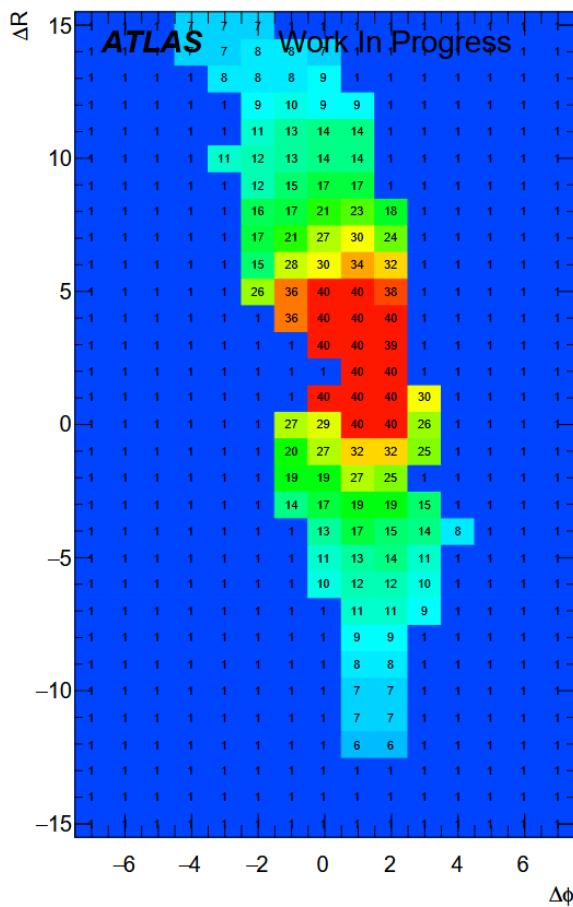


図 4.11: BRAM に実装された CW [53]。

4.3.3 先行研究と本研究との差異

先行研究では Alveo U200 に Run 3 で使用されたトリガーロジックを実装し、データの入出力を経て検証に要する時間とリソースの使用量について議論している。データセットを 1 ストリーム分準備して、ストリームの出入力でトリガーロジックの実行結果を確認していた。しかし、ストリームに格納できるデータ量は 1 つのパスあたり最大 $32 \times 4096\text{bit}$ である。さらに、ストリームは AXI interface で $32 \sim 1024\text{bit}$ ごとの転送に制限される。

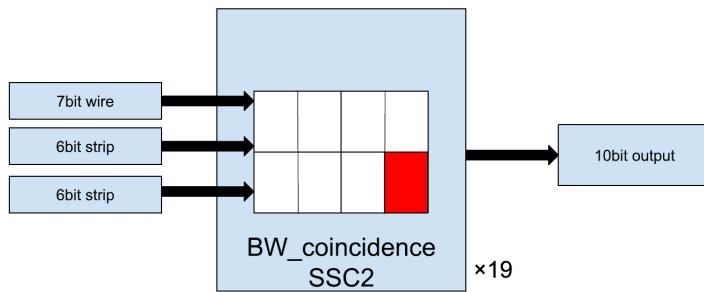


図 4.12: 先行研究で実装されたトリガーロジック [52]。SSC とは Run 3 で使用されているトリガー単位である subsector を 8 つまとめたものである。このロジックは入力として 7bit のワイヤー情報と 6bit のストリップ情報を要求している。SSC は縦 2 × 横 4 の subsector で構成されている。赤で表示された部分にヒットのある事象のみインプットとして選択して BRAM に入力している。出力値はヒットした位置情報と粒子の電荷と運動量の情報である。

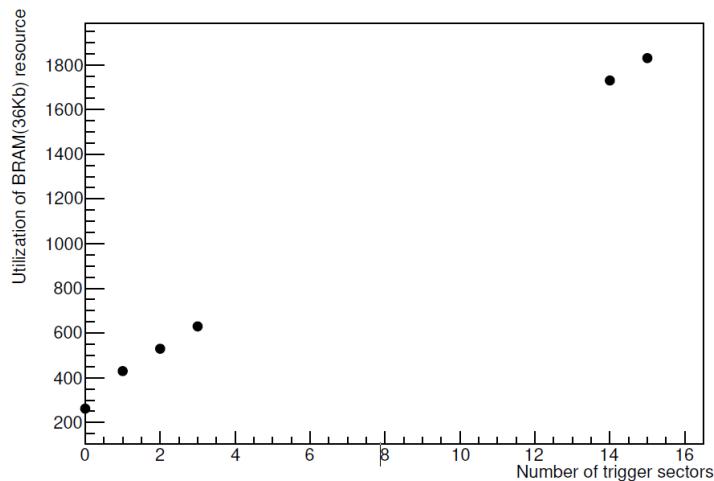


図 4.13: トリガーロジックのセクター数と使用する BRAM の数の比較 [52]。1 セクター目実装時の BRAM 使用数は 170 個で、その後 1 セクター追加する度に 100 個の BRAM を消費している。

そのため 1 クロック当たりの検証で入力可能なインプットの幅も最大 1024bit に制限されている。また、1 入力の幅を 32bit 以下に制限しても 4096 クロック分のデータのみ入力可能である。トリガーロジックはパイプライン処理で実行されるため、4096 イベントを超える連続入力が必要なデータに対するトリガーロジックの検証が困難となっている。そのため LUT の網羅的検証のような、高統計を用いたトリガーロジックの検証に応用することが難しく、FPGA アクセラレータを用いた検証システムの利点を生かすことが難しい。そのため、本研究では最大 32×4096 bit の入力幅を持つロジックを、任意のイベント数で検証可能とするシステムを開発した。

4.3.4 開発手順

開発は Alveo データセンタークセラレーションカード U200 の開発フローに従い、図 4.14 に示す手順で行った。使用したアプリケーションやプラットフォームの詳細は表 4.5 に示す。開発フローは主にファームウェア部の開発、ソフトウェア部の開発、論理合成に分かれ る。検証システムと検証を行うトリガーロジックはどちらも HDL で実装されている。検証システムにはトリガーロジックを実装できる領域をあらかじめ用意しており、配線の接続によりトリガーロジックをインスタンス化することで検証システムのハードウェア部を作成することができる。1 つのプロジェクトとして Vivado 上に登録された検証システムとトリガーロジックは Vivado に搭載された IP パッケージャーにより、1 つの IP にパッケージングされる。IP とは Intellectual Property の略称で、ファームウェアにおける特定の機能を再利用可能な設計ブロックとしてパッケージしたものであり、本研究では Xilinx Object (xo) ファイル形式で IP を出力する。アクセラレータの制御は C++ で書かれた制御用のプログラムで実行する。このプログラムをホストコードと呼ぶ。ホストコードはデータの生成と DDR との通信を制御し、アクセラレータの FPGA に実装された回路を実行する。xo ファイルと、ホストコードを Vitis にインポートして、プロジェクトを作成する。FPGA 上の回路合成と制御用プログラムのコンパイルを行い、一つのアプリケーションを生成する。開発に使用する PC は表 4.6 で示した構成となっている。CPU は Intel-core シリーズ第十世代のフラグシップモデルである Core i9 10920X を使用している。動作周波数は 3.5 GHz である。使用している OS は ATLAS Run 3 実験まで使用されていた Cent OS 7.9 である。Alveo U200 は図 4.15 に示すように、この PC のマザーボードに直接接続されている。

表 4.5: 使用したアプリケーション及びプラットフォームの詳細

| | |
|------------------|--------------------------------|
| 開発プラットフォーム | Xilinx_200_gen3x16_xdma_base_2 |
| ソフトウェアライブラリ | XRT 2022.2 |
| 検証システム開発ソフトウェア | Vivado 2022.2 |
| アプリケーション開発ソフトウェア | Vitis 2022.2 |

4.3.5 開発したトリガーロジック検証システムの詳細

図 4.16 は開発したトリガーロジック検証システムと検証システム全体の流れを表したブロック図である。検証システム全体は 1 台の PC で完結するが、PC 内部の装置間で通信を行ってトリガーロジックの検証を行う。HDL で開発・実装した検証システムはアクセラ

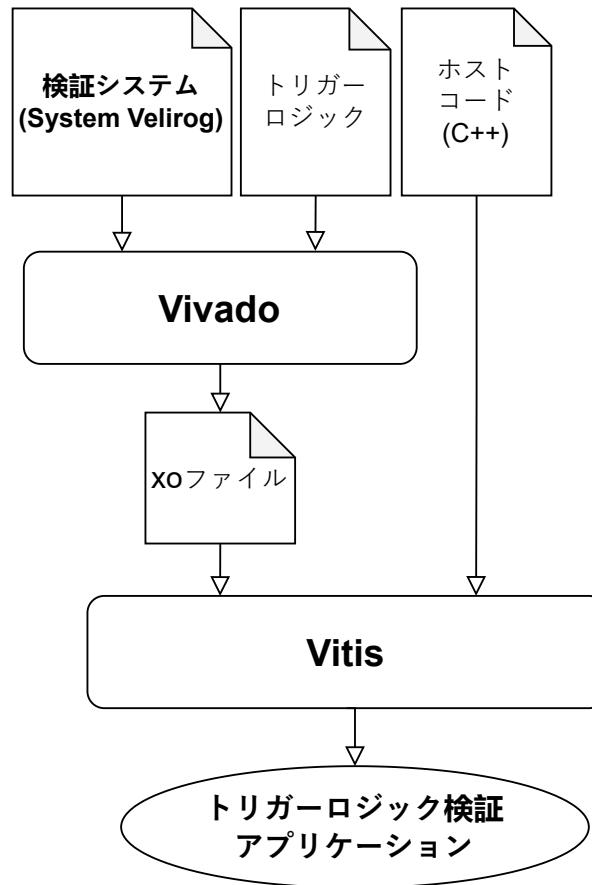


図 4.14: 開発のフロー図。FPGA に搭載する検証システムのロジックと検証したいトリガーロジックを Vivado にインポートして 1 つのプロジェクトとする。xo ファイルにパッケージされたプロジェクトは C++ で書かれた制御用のホストコードと一緒に Vitis でコンパイルして、トリガーロジック検証アプリケーションとする。

表 4.6: 開発用 PC の構成

| | |
|--------------|-------------------|
| CPU | Core i9 10920X |
| FPGA アクセラレータ | Alveo U200 |
| マザーボード | ASUS WS X299 SAGE |
| DDR メモリ (GB) | 128 |



図 4.15: 開発用 PC に接続された Alveo U200

レータ内の DDR と FPGA 間の通信と、トリガーロジックに対するデータの入出力を管理し、トリガーロジックを実行するシステムになっている。

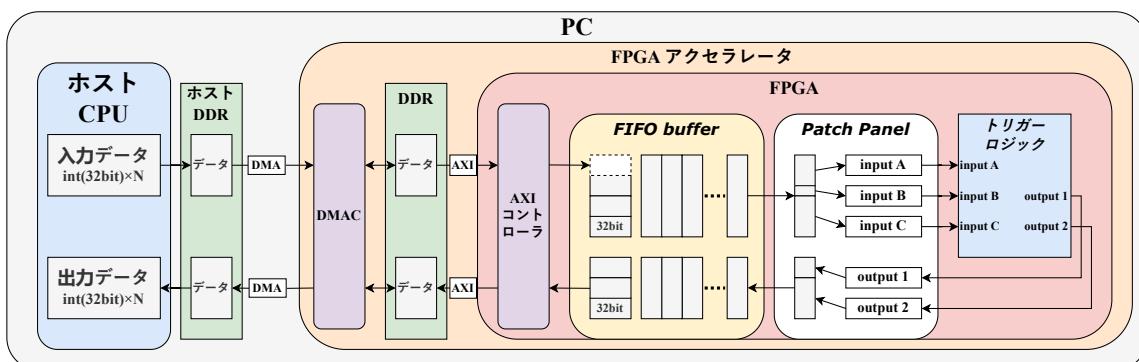


図 4.16: 開発した検証システムのブロック図。システムを構成するコンポーネントは、CPUによるデータ生成、DDR 間の DMA 通信、DDR から FPGA への AXI interface による通信、FIFO buffer ロジック、Patch Panel ロジック、検証するトリガーロジックに分けられる。データ生成は C++ で書かれたホストコードが、DDR は FPGA にビルドイン実装されている DMA が、AXI interface による通信とそれ以降のロジックは全て FPGA の Dynamic Region に実装された検証システムによって実行される。

4.3.6 ホスト CPU で入力用データの生成・読み取りと FPGA への転送

C++ で書かれたホストコードはホスト PC の CPU で実行される。演算や、テキストファイルや ROOT ファイルから読み取ることでデータを生成する。生成したデータは int 型、もしくは符号なし int 型の形式で 32bit ずつ配列に保存される。データの生成が完了したらホスト PC に搭載されている DDR のアドレスの書き込みアドレスを指定する。FPGA を制御する信号を書き込むアドレス、FPGA に入力するデータを入れるアドレス、FPGA

からの出力を受け取るアドレスを指定する。アドレス指定後、入力用のデータを送信して制御信号を送信する。CPU から制御信号を受け取ったホスト PC の DDR は FPGA の Static Region に実装された DMAC によってアクセラレータの DDR と DMA 通信を行う。DDR 間の通信バスは PCIe である。DDR-FPGA 間の通信は AXI4 interface の仕様に従って 32bit ずつデータを転送する。開発した検証システムは FPGA の Dynamic Region に AXI4 interface が実装する。受信したデータは後段の FIFO buffer ロジックに転送する。

4.3.7 FIFO buffer ロジック

FIFO buffer ロジックの概要を図 4.17 に示す。FIFO buffer は FPGA に 32bit ずつ転送されたデータを整列させて、一斉にトリガーロジックへ入力可能な形式に変換する。データの入力の概要を図 4.18 に、出力の概要を図 4.19 に示す。データの整列は 32bit のデータを結合 1 つずつ結合する形で行われる。トリガーロジックに置ける 1 クロックの入力信号分のデータの結合が完了したら、結合済みのデータをもう一つの FIFO に入力する。2 つ目の FIFO では後段のトリガーロジックの処理完了までデータを保持する役割を持つ。後段の準備が整ったら Patch Panel ロジックへデータを転送する。また、Patch Panel ロジックから出力データを受取り、32bit に分割して AXI 出力時に巨大な 1 つのデータを 32bit ずつのデータに分割して DDR に転送する。FIFO buffer は 1 度の DDR 転送で、数十回のトリガー処理に対応する入出力データを保持することができる。

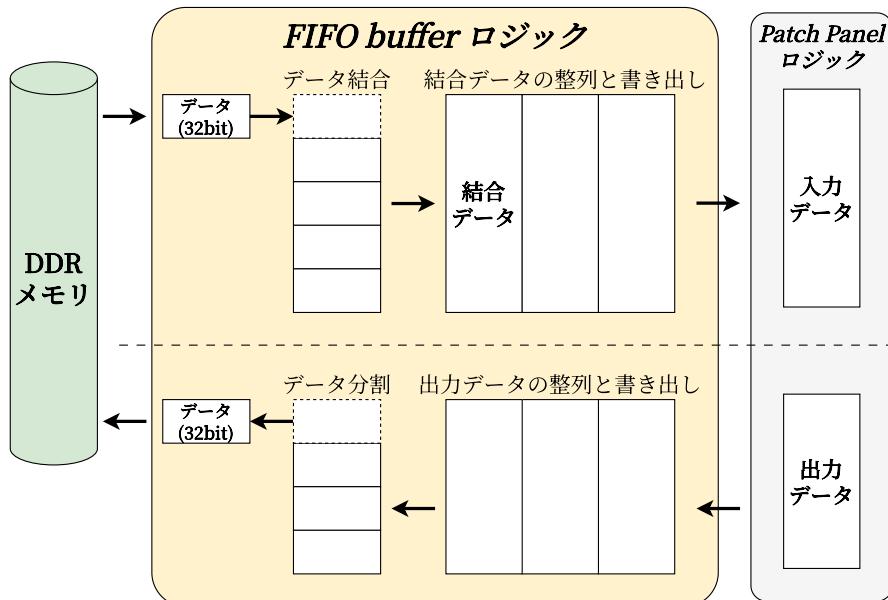


図 4.17: FIFO buffer ロジックの概要。

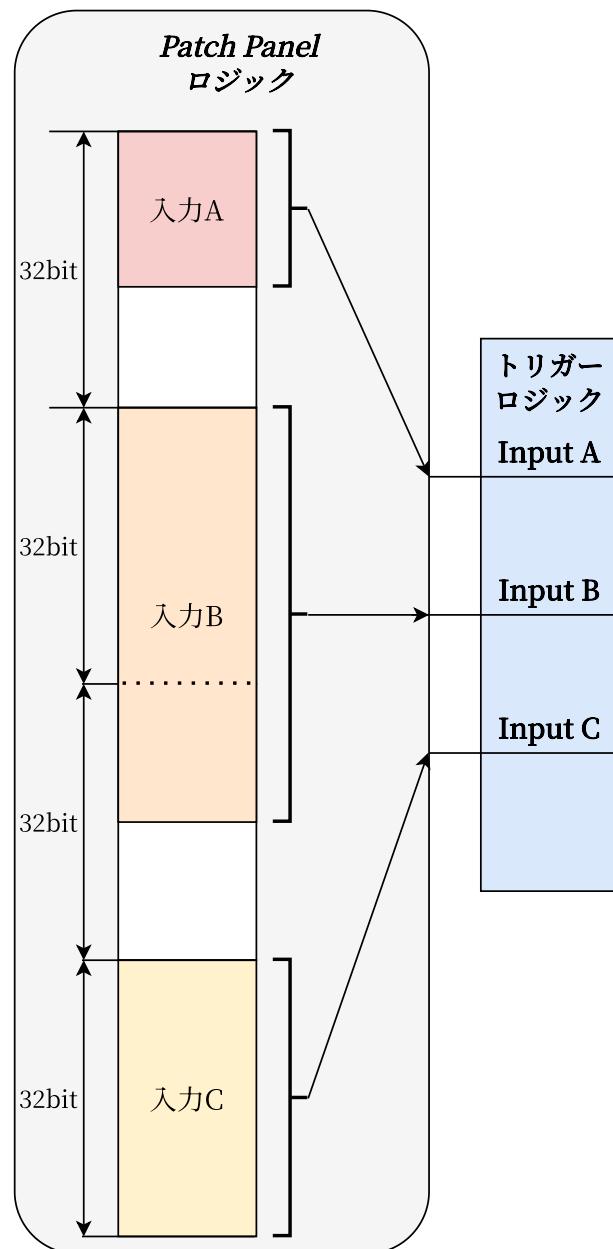


図 4.18: 入力時の Patch Panel ロジックの概要。CPU で生成された 32bit 単位のデータは FIFO buffer による結合を経て同時に Patch Panel ロジックに入力される。32bit 未満、もしくは 32bit を超過するデータをトリガーロジックに入力するために信号線を接続する。

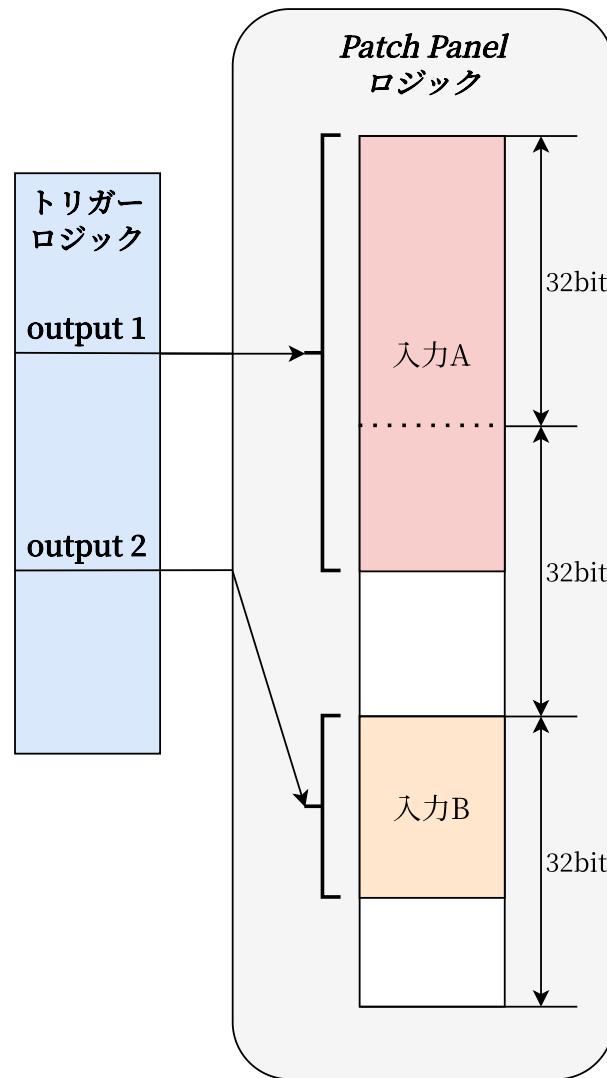


図 4.19: 出力時の Patch Panel ロジックの概要。出力されたデータを結合して FIFO buffer に送信する。この際、CPU での読み出しを考えて各データの開始点を 32bit 事に区切る。

4.3.8 Patch Panel ロジック

Patch Panel ロジックでは、FIFO buffer ロジックで生成された 1 つのデータとトリガーロジックのインプットを結び付ける役割を持つ。トリガーロジックの入力線はトリガーロジックごとに様々な幅を持ち、一様ではない。そのため Patch Panel ロジックで自

動的にデータを分配して正しい入力をトリガーロジックに与える。出力時には後段の FIFO buffer で 32 bit ずつ分割可能な形式で出力信号を一つのデータに束ねて FIFO buffer に転送する。

4.3.9 トリガー処理

Patch Panel ロジックにより入出力線が外部とつながったトリガーロジックは 1 入力ごとにクロックの立ち上がり/立下りを実行する。この際のクロック信号も CPU で生成された入力用データに含まれる。そのため、トリガーロジックに対して 2 回の入力で 1 クロック分の処理を行う。トリガーロジックへデータを書き込む操作と読み出す操作は同じタイミングで並列に実行される。このクロック処理は、単一のクロックで実装されたトリガーロジックに対してのみ有効である。

4.3.10 大統計を用いたファームウェアの検証作業

開発した検証システムはハードウェアで駆動するため、データの入出力を Vivado シミュレーションと比較して高速で行うことができる。具体的には、500 クロックの処理に Vivado シミュレーションは 1 分を要するが、アクセラレータを用いた検証では 1.5 秒で完了する。そのため、Vivado シミュレーションでは現実的ではなかった、大統計を用いた検証が可能となった。

4.4 インナーコインシデンスファームウェアロジックのハードウェア実装

開発した検証システムの性能測定と、第 3 章で説明したインナーコインシデンスの検証のために、アクセラレータ内の FPGA にインナーコインシデンスを実装して検証システムを稼働させた。システム全体の概略を図 4.20 に示す。

4.4.1 ソフトウェアシミュレーションと検証用ミューオンデータ

FPGA アクセラレータに実装されたインナーコインシデンスロジックに入力するための飛跡情報は、ソフトウェアシミュレーションで作成されたデータを使用した。ソフトウェアシミュレーションにより、Wire–Strip Coincidence を通過した粒子の飛跡情報を取得し、NSW の飛跡情報と合わせてインナーコインシデンスに入力した。ソフトウェアシミュレーションの結果は 64bit の double 型で保存されているため、 η は $-2.7 \sim 2.7$ の範囲で 14bit、 ϕ は $0 \sim 2\pi$ の範囲で 12bit に変換する。本来は TGC BW の全域で検証を進めるところだ

が、用意できた CW の都合で、A-side の $\phi_{ID} = 5$ において、 $\eta_{ID} = 0$ および $\eta_{ID} = 3$ の領域に絞って評価を行った。使用した領域を図 4.23 に示す。このヒストグラムは入射してきたミューオンの数を表している。

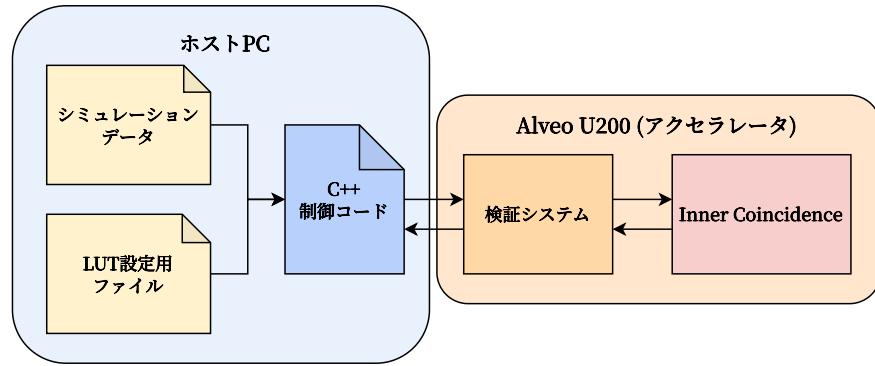


図 4.20: FPGA アクセラレータを用いたインナーコインシデンスロジック検証のブロック図。インナーコインシデンス内にある LUT と入力するミューオンデータは外部でテキストファイルの形で保存されている。C++ で書かれた制御コードで検証システムを介してインナーコインシデンスの入出力を制御する。

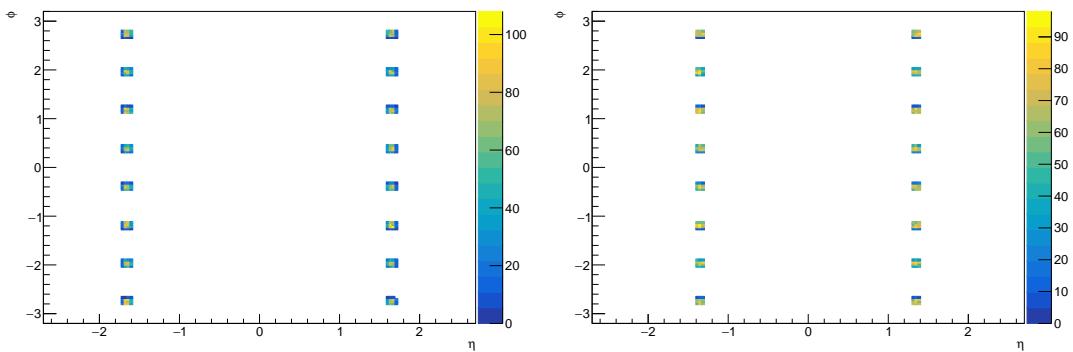


図 4.21: $(\eta_{ID}, \phi_{ID}) = (0, 5)$ の領域。

図 4.22: $(\eta_{ID}, \phi_{ID}) = (3, 5)$ の領域。

図 4.23: インナーコインシデンスロジックからの出力比較。Vivado シミュレーション(左)と FPGA アクセラレータにハードウェア実装されたもの(右)で出力が異なる。

4.4.2 ハードウェア実装とデバッグ作業

検証を開始するにあたって、インナーコインシデンスロジックがアクセラレータに正しく実装されているか検証する必要があった。そのため 3 章で行った Vivado シミュレーションによる LUT の出力検証と同じ入力を与え、出力を確認した。図 4.24 は Vivado シミュ

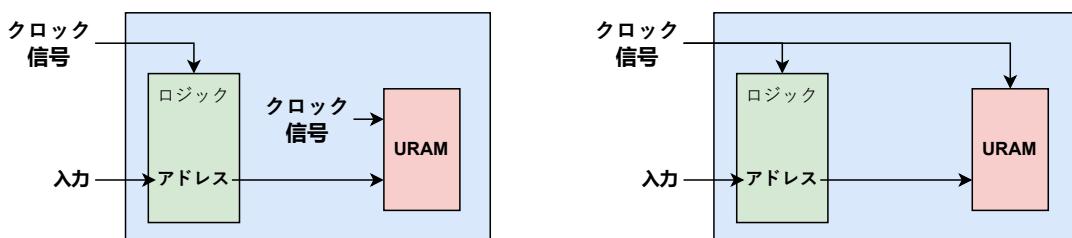
レーションとアクセラレータからの出力を同じタイミングで抜粋したものである。Vivado シミュレーションは `Inner_pT` の出力が 1、2 と変わるのでに対してハードウェアからの出力は全て 4 となっており、シミュレーションとハードウェア実装の間で結果の不一致が発生した。ファームウェア全体を再検討した結果、NSW と TGC BW との統合部に実装されている CW が他のファームウェアと異なるクロックドメインで独立して駆動していることが発覚した。クロックの同期が行われておらず、不正な値が URAM から出力されていることがわかった。この問題はクロックドメインを統一することで解消できた。図 4.25 は URAM とクロックドメインの関係を図示したものである。このことは、アクセラレータを用いた検証システムがハードウェア特有の挙動をうまく再現しており、Vivado シミュレーションでは確認できなかったファームウェアの不具合を発見することに役立つことを示した事例である。

| | |
|-----------------------------|-----------------------------|
| Inner_pT: 1 out: 12 flag: 1 | Inner_pT: 4 out: 12 flag: 1 |
| Inner_pT: 1 out: 12 flag: 1 | Inner_pT: 4 out: 12 flag: 1 |
| Inner_pT: 1 out: 12 flag: 1 | Inner_pT: 4 out: 12 flag: 1 |
| Inner_pT: 1 out: 12 flag: 1 | Inner_pT: 4 out: 12 flag: 1 |
| Inner_pT: 1 out: 12 flag: 1 | Inner_pT: 4 out: 12 flag: 1 |
| Inner_pT: 1 out: 12 flag: 1 | Inner_pT: 4 out: 12 flag: 1 |
| Inner_pT: 2 out: 12 flag: 1 | Inner_pT: 4 out: 12 flag: 1 |
| Inner_pT: 2 out: 12 flag: 1 | Inner_pT: 4 out: 12 flag: 1 |

(a) Vivado シミュレーションからの出力結果

(b) ハードウェアからの出力結果

図 4.24: インナーコインシデンスロジックからの出力比較。Vivado シミュレーション (左) と FPGA アクセラレータにハードウェア実装されたもの (右) で出力が異なる。



(a) 修正前の URAM に実装されたクロック信号。 (b) 修正後の URAM に実装されたクロック信号。

図 4.25: URAM のクロックドメインの比較。URAM と URAM をラッピングしているモジュールに、それぞれ独立したクロックドメインが存在しており、クロックの同期に失敗していた。そのため、ラッピングしているモジュールと同一のクロックを URAM に接続して、同じクロックドメインとし、クロックを同期させた。

4.4.3 検出効率の評価

FPGA アクセラレータを用いて p_T 閾値 20 GeV での検出効率を評価した。検出効率の評価は式 (4.1) で表されるフェルミ分布関数を用いる。各 p_T に対する検出効率でフィッティングを行い、検出効率と分解能、 p_T 閾値について評価を行った。

$$f(p_T) = \frac{\epsilon}{e^{\frac{p_T - \mu}{\sigma}} + 1} \quad (4.1)$$

A-side の $(\eta_{ID}, \phi_{ID}) = (3, 5)$ で使用する CW を図 4.26 に、トリガー検出効率を図 4.27 に示す。TGC BW のみ用いたトリガー判定と NSW の情報も加味したトリガー判定の検出効率について、フェルミ分布関数によるフィッティングで得られた各パラメータを表 4.7 に示す。この領域で使用する内層ミューオン検出器は NSW のみであり、出力結果がそのままインナーコインシデンスの検出効率となる。TGC BW のみを使用したトリガーに対して、NSW を加えたトリガーはプラトー領域での検出効率が 4% 低下した。これは、TGC BW の検出効率に対して NSW の検出効率が乗算されるためであると考えられる。また、 $p_T < 20$ GeV であるようなミューオンに対する検出率が低下しており、低 p_T ミューオンを効率的に棄却可能であることがわかる。これらの結果より実験時もインナーコインシデンスによって、トリガーレートを削減することが可能であると考えられる。一方で、TGC BW の情報だけで評価されたトリガー検出効率と、NSW を加えて評価した場合のトリガー検出効率の閾値が大きく異なることが分かった。原因として、CW が不完全である可能性が考えられる。図 4.28 は $p_T = 15$ GeV と判定された事象のプロットである。プロットを確認すると、 $p_T > 20$ GeV で検出効率が増加する部分が存在し、さらなるチューニングが必要であることがわかる。現状使用している CW は MDT 検出器で再構成された飛跡情報から NSW の飛跡情報をシミュレートして作成されたものである。そのため、NSW の正確な飛跡を直接利用するような手法で CW を再チューニングすることで、検出効率と閾値が改善されると考える。

また、NSW の境界部である、 $(\eta_{ID}, \phi_{ID}) = (0, 5)$ の領域について検証を行った。この領域は図 4.29 に示すように、NSW の端に当たる部分で、一方の電荷をもつ粒子だけが NSW と TGC BW の両方を通過する。本検証は A-side で行っているため、主に負電荷を持つミューオンが検知される。そのため $\eta = 1.3$ での検出効率は 50% 程度であり、 $\eta = 1.3 \sim 1.4$ の領域では 50% より少し良い程度の検出効率が期待される。CW を図 4.30 に、測定結果を図 4.31 に示す。パラメータについて表 4.8 に示す。プラトー領域の検出効率はおよそ 6 割程度となった。NSW の端であることを考えれば妥当な値であると考えられる。インナーコインシデンスロジックは NSW 以外の内層ミューオン検出器の飛跡情報を組み合わせることで検出効率を高く保つように実装されている。

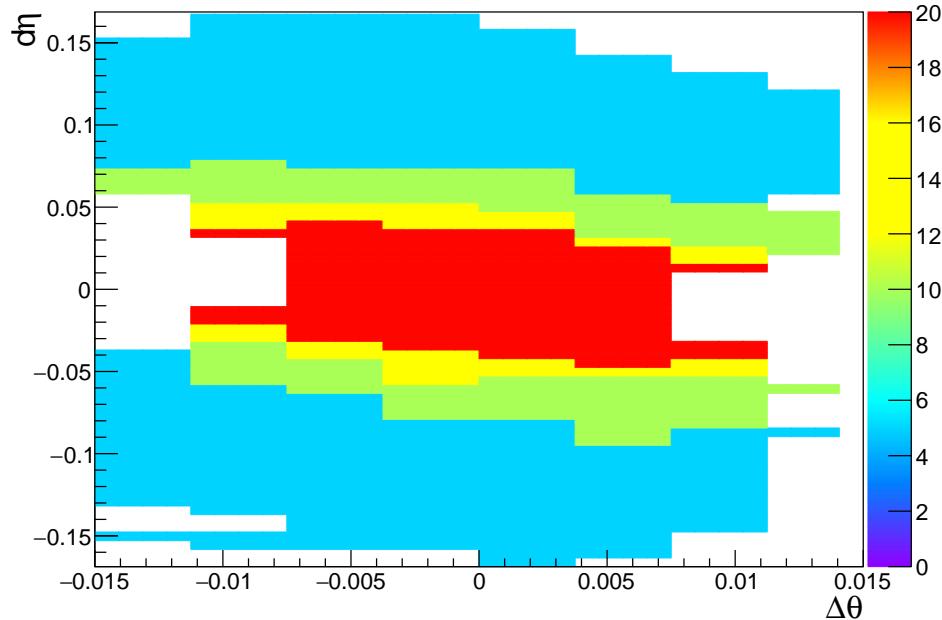


図 4.26: $(\eta_{ID}, \phi_{ID}) = (3, 5)$ で使用する CW。

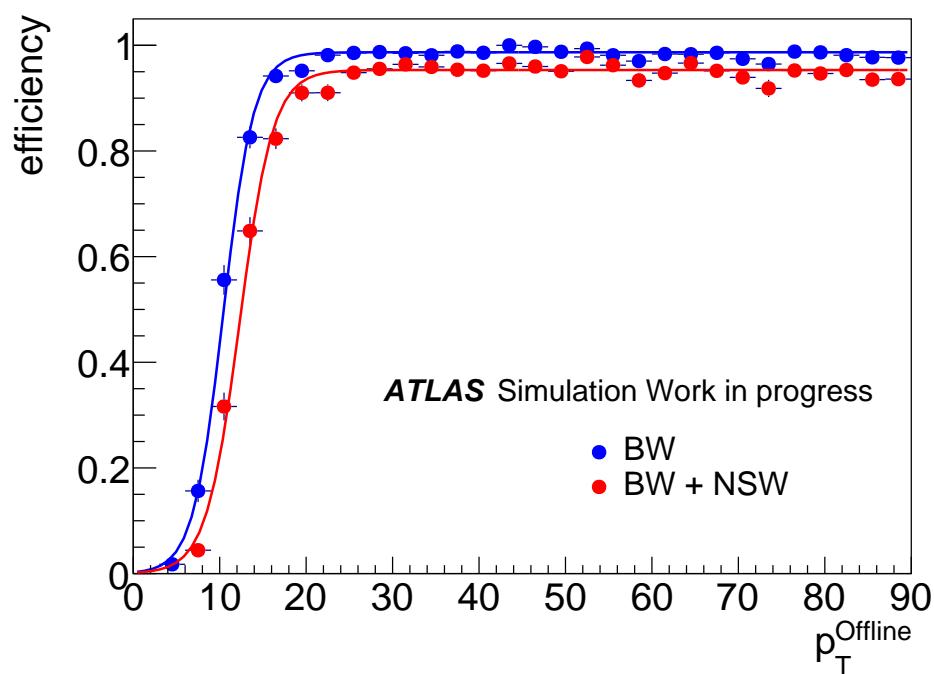
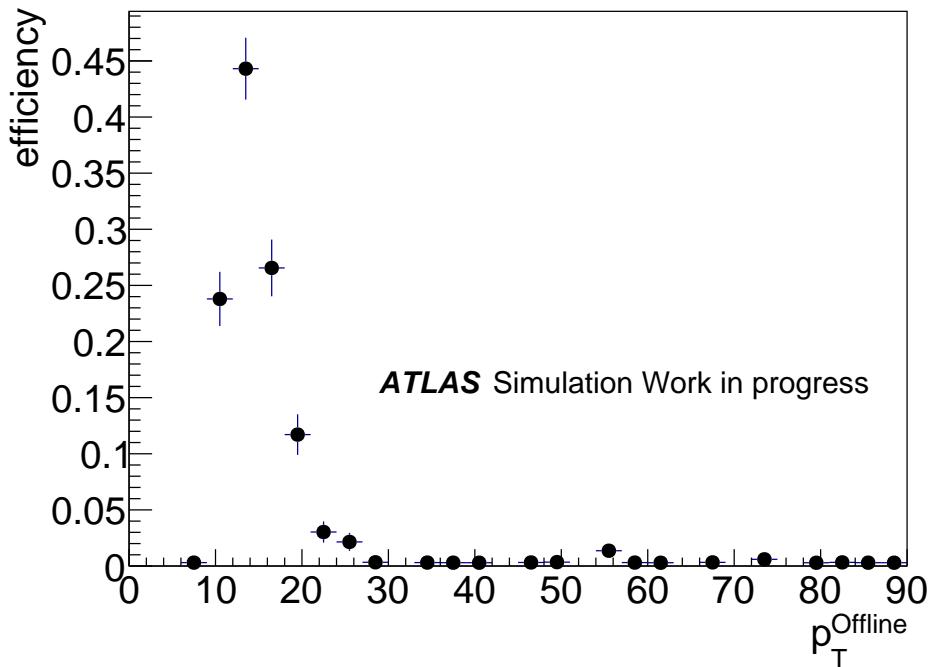


図 4.27: $(\eta_{ID}, \phi_{ID}) = (3, 5)$ における p_T 閾値 20 GeV の検出効率。青は TGC BW の情報のみ使用したトリガーの検出効率。赤線は NSW の結果も加えた検出効率である。TGC BW で検出された 9,519 イベントに対して、NSW を考慮してトリガー判定が行われたのは 8,890 イベントだった。

表 4.7: $(\eta_{ID}, \phi_{ID}) = (3, 5)$ におけるフィッティング結果のパラメータ比較。

| | TGC BW | TGC BW + NSW |
|----------|--------|--------------|
| 検出効率 | 98.7% | 94.9% |
| 閾値 [GeV] | 10.4 | 13.0 |
| 分解能 | 16% | 15% |

図 4.28: $(\eta_{ID}, \phi_{ID}) = (3, 5)$ において、 p_T 閾値が 15 GeV であると判定された割合。横軸はミュオンの p_T で、各点は検出効率を表す。 $p_T > 20$ GeV の領域に検出効率が増加する部分がある。表 4.8: $(\eta_{ID}, \phi_{ID}) = (0, 5)$ におけるフィッティング結果のパラメータ比較。 p_T 閾値は 20 GeV。

| | TGC BW | TGC BW + NSW |
|----------|--------|--------------|
| 検出効率 | 98.6% | 64.8% |
| 閾値 [GeV] | 14.3 | 16.1 |
| 分解能 | 10% | 9% |

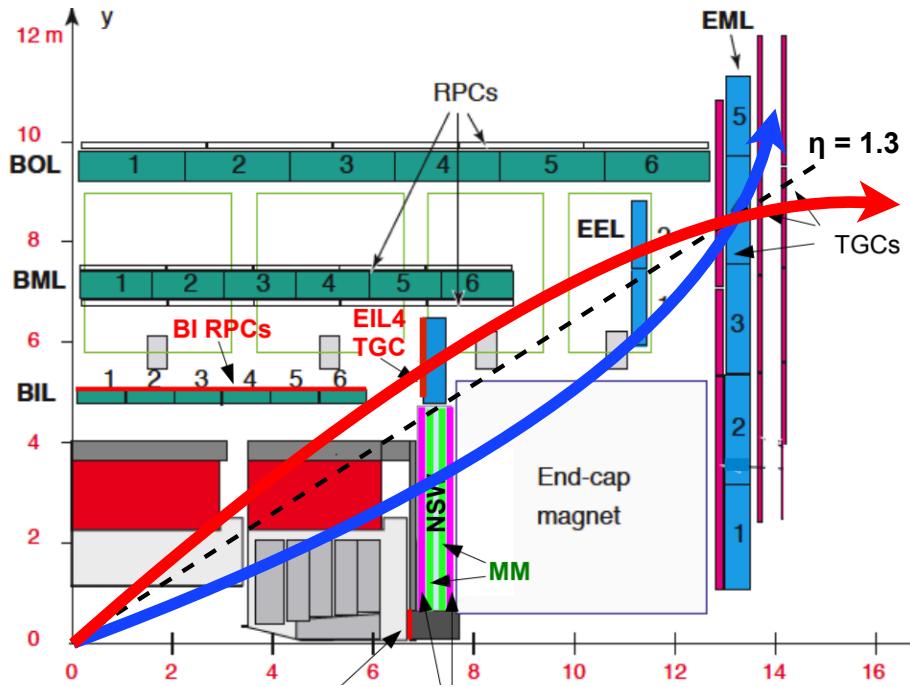


図 4.29: TGC WB の $\eta = 1.3$ に入射するミューオンの飛跡。磁場は奥から手前に向かっているため、電荷によって曲がる向きが異なる。A-sideにおいて赤線は正電荷、青線は負電荷のミューオンが描く飛跡を表す。赤線の正電荷は NSW を通過しないが、青線の負電荷は NSW を入射する。

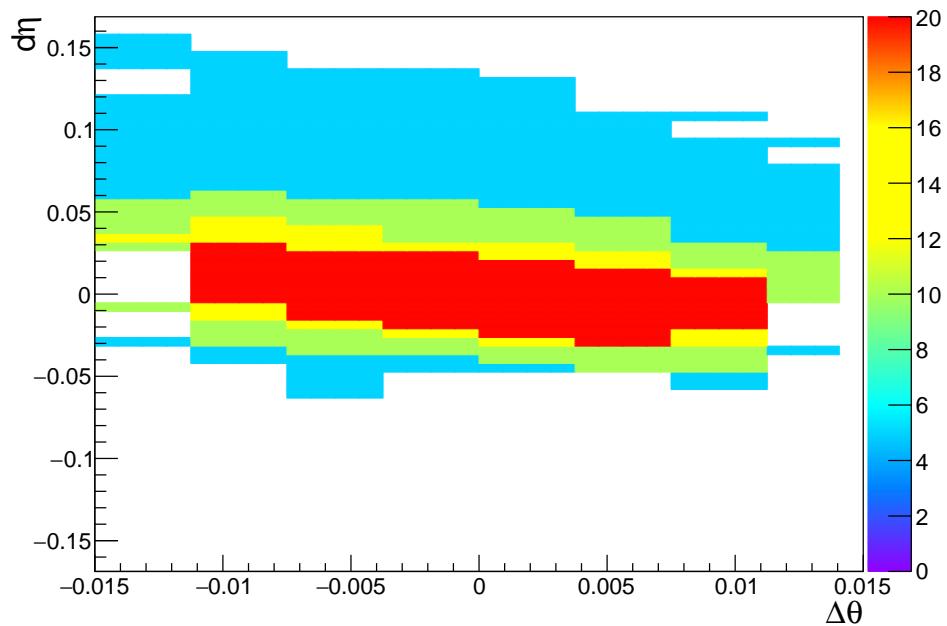


図 4.30: $(\eta_{ID}, \phi_{ID}) = (0, 5)$ で使用する CW。それぞれの色は p_T 閾値を表す。

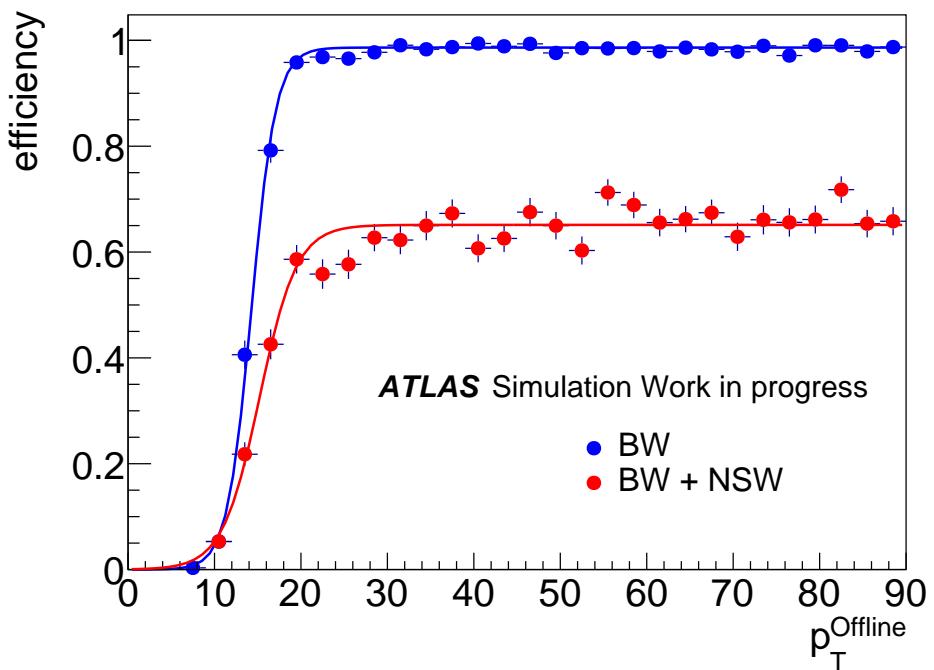


図 4.31: $(\eta_{\text{ID}}, \phi_{\text{ID}}) = (0, 5)$ における検出効率。青は TGC BW の情報のみ使用したトリガーの検出効率。赤線は NSW の結果も加えた検出効率。TGC BW で検出された 9182 イベントに対して、NSW を考慮してトリガー判定が行われたのは 5915 イベントだった。NSW を加えたトリガーについては統計数が少ないので、プラトー領域でばらつきが見られる。 p_T 閾値は 20 GeV。

第 5 章

結論と今後の展望

LHC-ATLAS 実験は 2030 年より高輝度化アップデートが行われ、ビームの瞬間輝度が約 3 倍となる。初段ミューオントリガーのトリガーレートは 100 kHz から 1 MHz に増強され、トリガーレイテンシは $2.5 \mu\text{s}$ から 4 倍の $10 \mu\text{s}$ に延長される。ATLAS 検出器の初段ミューオントリガーエレクトロニクスが刷新されるとともに、トリガーロジックも全て変更される。本研究では Endcap Sector Logic に実装する開発途中のインナーコインシデンスロジックの最適化とロジックが想定通りに動くことを、Vivado シミュレーションを通じて確認した。ロジックを最適化した結果、当時問題となっていたタイミング制約違反を解決した。

また、効率的なトリガーロジックの検証を行うために AMD / Xilinx 社が提供する FPGA アクセラレータを用いたトリガーロジック検証システムを開発した。開発したトリガーロジック検証システムは PC 一台で完結し、ATLAS 実験のファームウェア開発で使用するツールチェーンを最大限活用して、Vivado と Vitis のみで実行可能である。任意のトリガーロジックを実装でき、入出力の柔軟性も高いため、Virtex UltraScale+ FPGA アーキテクチャを用いたトリガーロジックのハードウェア検証環境を簡易化することに成功した。さらに、開発した検証システムは Vivado シミュレーションを用いた従来の手法と比較して 20 倍の速度で検証可能であることを確認した。

開発した検証システムを応用して FPGA にインナーコインシデンスロジックを実装した。MC シミュレーションで作成したデータをインナーコインシデンスに入力することで、NSW と TGC BW 間のコインシデンスロジックを検証した。検証を行った結果、検出の分解能と閾値が上昇し、トリガーレートの削減に繋がることが確認できた。また、NSW のコインシデンスロジックで使用する CW の最適化が必要であることも確認した。

今後の課題として、インナーコインシデンスは Tile カロリメータと NSW とのコインシデンス取得ロジックの開発・検証が完了したが、EI と RPC とのコインシデンス取得ロジックが未検証であるので、今後検証を進めていく必要があると考える。また、現状の FPGA

アクセラレータを用いたトリガーロジック検証システムは複数のクロックで動作するロジックに対応していないため、レジスタを用いてクロックに同期させるシステムに改良することが必要である。

本研究を通じて、アクセラレータを用いた新しいトリガーロジック検証システムを開発し、効率的なトリガーロジック開発が可能となった。より一層の複雑化、巨大化が見込まれる高エネルギー実験のファームウェアロジック開発が効率的に推し進められるようになるとを考えている。また、高輝度 LHC-ATLAS 実験の初段エンドキャップミューオントリガーに実装するインナーコインシデンスロジックの開発と検証により、トリガーロジックが完成へ一歩近づいた。インナーコインシデンスはトリガーレート削減に大きく貢献するロジックであり、高輝度 LHC-ATLAS 実験成功に繋がることを期待する。

謝辞

本論文の作成にあたり、指導教員である前田順平准教授には多大なる助力を頂きました。日々の研究のサポートからミーティング資料や申請書類のご確認、CERN 出張のご支援や各種シンポジウムに向けた発表練習、TWEPP の Proceeding の校閲及び改訂作業など、2 年間のご指導を通じて頂いたご支援は枚挙に暇がございません。心より感謝申し上げます。

神戸大学粒子物理学研究室の先生方、先輩方、後輩の皆様、そして同級生の皆様には、研究活動において心身の両面で多大なるご支援を賜りました。ATLAS グループに所属する藏重久弥教授・山崎祐司教授には、示唆に富んだご提案や、私の研究への疑問点のご指摘、物理的アプローチによる結果のご考察などにより、私の研究を推し進めてくださいました。また、数々のご質問にもご回答いただき、誠にありがとうございます。身内賢太朗准教授は学振の校閲や日々の研究生活のご支援をして頂きました。竹内康雄教授は私の研究と先行研究との差異について議論をしてくださいました。両名とも心より感謝申し上げます。東野聰研究員は同じ部屋で研究を行う傍ら、日頃から暖かい言葉を投掛けて頂き大変励みになりました。深く感謝いたします。秘書の吉田和美氏と北川順子氏には遅延しがちな私の出張申請をスムーズに処理していただきました。お礼申し上げます。また、故越智敦彦准教授と深く関わり合ったのは研究室に来てから 1 年弱の期間でしたが、ご一緒した日々は今も鮮烈に記憶に残っています。安らかにお眠りください。先輩の高橋真斗氏には博士課程での研究について私が何度も訪ねても真摯にお答えいただきました。感謝の意を表します。同級生の生井凌太氏、村田優衣氏とは同部屋の縁で、共に研究に邁進することができました。私が度々コンビニに行く際も、わざわざ御同行いただき厚く御礼申し上げます。同じく同級生の鈴木大夢氏、西将太氏、樋口流雲氏らと共に研究生活を過ごせたことで充実した研究生活を送ることができました。ご厚情に感謝いたします。後輩一同が研究室を賑やかに彩っていただいたのでいつも楽しく大学に来ることができました。ご支援に感謝いたします。

日本 ATLAS グループの皆様にも大変お世話になりました。東京大学素粒子物理国際研究センターの奥村泰幸准教授には CERN での地下作業の経験を賜っただけでなく、インナーコインシデンスの研究を着手する切掛けを頂戴いたしました。日々の研究にもミー

ティングを通じて多くのアドバイスを頂きました、ご厚意に感謝いたします。同じく東京大学素粒子物理国際研究センターの齋藤智之助教にも物理学会の発表資料のレビューやCERNでの活動のご支援をしていただきました。ご高配いただきありがとうございます。名古屋大学高エネルギー素粒子研究室の堀井泰之准教授にはSLファームウェアの実装に関するアドバイスを頂きました。ご支援に心から感謝いたします。高エネルギー加速器研究機構の青木雅人研究機関講師は地下での作業や、TGCオペレーションの作業についてご教授頂きました。また、CERNの冷蔵庫の氷が解けて洪水が起きた際に、一緒に床を拭いてくださり、神戸部屋の屋根が吹き飛んだ際は修復をお手伝いしてくださいました。ご厚志に感謝いたします。京都大学の陳詩遠准教授には学振に書く物理解析の詳細や物理解析でどのような研究を行うのかについて、お忙し中質問に答えてくださいました。ご芳情に感謝いたします

CERNに滞在する際、日本ATLASグループの皆様には大変お世話になりました。青木匠氏、林雄一郎氏には2度のCERN滞在で宿泊場所をご提供いただきました。それだけでなく、フランスを中心としたCERNでの生活について数々の貴重な教えを賜りました。ありがとうございます。河本地弘氏は私の研究に対するご助言やご指導など、多大なるご支援を頂きました。それだけでなく、CERNでの空港までのお迎えやお部屋を借してのパーティもして頂きました。ご支援に厚く感謝申し上げます。古川真林氏には遠く離れた施設への車での送り迎えや週末の食事会へのお誘いなど、大変お世話になりました。成川佳史氏とはと一緒にグラスゴーで開催されたTWEPPに参加させていただいた際に、右往左往する私を導いてくださいました。ハイキングにも誘っていただき、ご一緒に断崖絶壁を這い進んだのは良き思い出でございます。ご厚意に深く感謝いたします。橋本大輔氏ともと一緒にTWEPPに参加させていただきました。海外での身の振り方に加え、研究に対する直向な姿勢を教えてくださいました。ありがとうございます。長坂鍊氏は私がシャドウとしてTGCオペレーションを学ぶ際、懇切丁寧にTGCオペレーションを教えてくださいました。ありがとうございます。

本論文の解析に使用したシミュレーションデータを提供してくださった中川徹郎氏には大変感謝しております。ご尽力に深く感謝いたします。また中川徹郎氏を含め、須部実咲氏、牧田藍瑠氏、近藤翔太氏、石川諒氏といった神戸大学外のATLASグループの同級生の皆さんとのシンポジウムや学会等を通じた交流は、研究への大きな活力になっていました。大変感謝しています。博士課程後期課程でも仲良くしていただけますと幸いです。

最後に、24年間私を支えて頂いた両親に感謝いたします。両親の助力無くしては本論文の執筆はかねませんでした、ありがとうございます。

参考文献

- [1] S. Weinberg, The making of the Standard Model, Eur. Phys. J. C, Vol. 34, p. 5, 2004.
- [2] S. Navas and C. Amsler and T. Gutsche and C. Hanhart and J. J. Hernández-Rey and C. Lourenço and A. Masoni and M. Mikhasenko and R. E. Mitchell and others, Review of Particle Physics, Physical Review D, Vol. 110, No. 3, p. 030001, Aug 2024.
- [3] CERN, Dark Matter, <https://home.cern/science/physics/dark-matter>, Accessed: 2025-02-02.
- [4] B. Abi and T. Albahri and S. Al-Kilani and D. Allspach and L. P. Alonzi and A. Anastasi and A. Anisenkov and F. Azfar and K. Badgley et al. (Muon g-2 Collaboration), Measurement of the Positive Muon Anomalous Magnetic Moment to 0.46 ppm, Phys. Rev. Lett., Vol. 126, No. 14, p. 141801, April 2021.
- [5] ATLAS Collaboration and Aad, G. and Abat, E. and others, The ATLAS Experiment at the CERN Large Hadron Collider, Journal of Instrumentation, Vol. 3, No. 08, pp. S08003–S08003, 2008.
- [6] ATLAS Collaboration, High-Luminosity Large Hadron Collider (HL-LHC): Technical Design Report, No. CERN-2020-010 in CERN Yellow Reports: Monographs. CERN, 2020.
- [7] ATLAS Collaboration, Technical Design Report for the Phase-II Upgrade of the ATLAS Trigger and Data Acquisition System - Event Filter Tracking Amendment Technical Report CERN-LHCC-2022-004, CERN, Mar 2022.
- [8] ATLAS Collaboration, ATLAS MAGNETIC FIELD, <http://atlas.web.cern.ch/Atlas/GROUPS/MUON/magfield>, Accessed: 2025-01-27.
- [9] ATLAS Collaboration, Technical Design Report for the Phase-II Upgrade of the ATLAS Muon Spectrometer Technical report, CERN, 2017.
- [10] ATLAS Collaboration, ATLAS level-1 trigger: Technical Design Report Technical Report CERN-LHCC-98-014, CERN, 1998 Accessed: 2025-02-02.

- [11] C Marcelloni et al., Installation of the first of the big wheels of the ATLAS muon spectrometer, a thin gap chamber (TGC) wheel, Accessed: 2025-01-27.
- [12] L. Pizzimento, Performance of the BIS78 RPC detectors: a new concept of electronics and detector integration for high-rate and fast timing large size RPCs, Journal of Instrumentation, Vol. 15, No. November 2020, p. C11010, 2020 XV Workshop on Resistive Plate Chambers and Related Detectors (RPC2020).
- [13] Kawamoto, T. and Vlachos, S. and Pontecorvo, L. and Dubbert, J. and Mikenberg, G. and Iengo, P. and Dallapiccola, C. and Amelung, C. and Levinson, L. and Richter, R. and others, New Small Wheel Technical Design Report Technical report, CERN. Geneva. The LHC experiments Committee, June 2013 ATLAS New Small Wheel Technical Design Report.
- [14] 三島章熙, 高輝度 LHC-ATLAS 実験に向けたミューオントリガー論理回路の開発・実装・試験—読み出し・制御系の構築とトリガー系の統合, 東京大学修士論文, 2023.
- [15] PICMG, AdvancedTCA, <https://www.picmg.org/openstandards/advancedtca>, Accessed: 2025-02-02.
- [16] Intel Co. Ltd, Intelligent Platform Management Interface Specification v2.0 rev. 1.1, <https://www.intel.co.jp/content/www/jp/ja/products/docs/servers/ipmi/ipmi-second-gen-interface-spec-v2-rev1-1.html>, Accessed: 2025-02-02.
- [17] Samtec, FireFly™ Optical Transceivers, <https://www.samtec.com/optics/systems/firefly/>, Accessed: 2025-02-02.
- [18] Nordic Semiconductor, QSPI — Quad Serial Peripheral Interface, https://docs.nordicsemi.com/bundle/ps_nrf52840/page/qspi.html, October 2024 Technical Documentation, Accessed: 2025-02-02.
- [19] AMD, Zynq UltraScale+ Device Technical Reference misc (UG1085), <https://docs.amd.com/r/en-US/ug1085-zynq-ultrascale-trm/Zynq-UltraScale-Device-Technical-Reference-misc>, Accessed: 2025-02-02.
- [20] AMD, Virtex UltraScale+ FPGA Product Brief, <https://www.amd.com/content/dam/amd/en/documents/products/adaptive-socs-and-fpgas/fpga/ultrascale-plus/virtex-ultrascale-product-brief.pdf>, Accessed: 2025-02-02.
- [21] ATLAS Collaboration, Phase-II TGC Sector Logic, <https://twiki.cern.ch/twiki/bin/view/Atlas/Ph2TgcSectorLogic>, Accessed: 2025-01-29.
- [22] AMD, UltraScale Architecture GTY Transceivers User Guide, <https://docs.amd.com>.

- com/v/u/en-US/ug578-ultrascale-gty-transceivers, Accessed: 2025-02-02.
- [23] AMD, UltraScale Architecture and Product Data Sheet: Overview, <https://docs.amd.com/v/u/en-US/ds890-ultrascale-overview>, Product Specification, Accessed: 2025-02-02.
- [24] AMD, UltraFast Design Methodology Guide for FPGAs and SoCs (UG949), <https://docs.amd.com/r/en-US/ug949-vivado-design-methodology/Super-Logic-Region-SLR>, Accessed: 2025-02-02.
- [25] AMD, Versal Adaptive SoC Hardware, IP, and Platform Development Methodology Guide, <https://docs.amd.com/r/en-US/ug1387-acap-hardware-ip-platform-dev-methodology/Super-Long-Line-SLL-Routes>, Super-Long-Line (SLL) Routes, Accessed: 2025-02-02.
- [26] AMD, Large FPGA Methodology Guide (UG872 v14.3), https://www.xilinx.com/htmldocs/xilinx13_4/ug872_largefpga.pdf, Accessed: 2025-02-02.
- [27] ATLAS Collaboration, Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System Technical Report CERN-LHCC-2017-020, CERN, June 2018.
- [28] 赤塚 駿一, LHC-ATLAS 実験 Run-3 に向けたミューオントリガーの改良, 京都大学修士論文, 2017.
- [29] Aoki Masato, L1Muon Endcap Inner-Coincidence Performance plots in 2024 Technical report, High Energy Accelerator Research Organization (JP), 2024.
- [30] 河本 地弘, 高輝度 LHC ATLAS 実験に向けた初段ミューオントリガーアルゴリズムの実装と検出器全体への拡張, 京都大学修士論文, 2023.
- [31] 三野 裕哉, 高輝度 LHC ATLAS 実験に向けた初段ミューオントリガーアルゴリズムの開発及びハードウェアへの実装, 京都大学修士論文, 2020.
- [32] 鍋山 友希他 ATLAS 日本トリガーグループ高輝度 lhc-atlas 実験 初段ミューオントリガーに用いる飛跡選別回路の開発第 77 回年次大会 (2022 年) オンライン開催. 日本物理学会, 2022 講演番号: 15aA572.
- [33] 成川 佳史, 高輝度 LHC-ATLAS 実験に向けたミューオントリガー論理回路の実装及び性能評価 —トリガー系の統合と試験システムの構築—, 東京大学修士論文, 2024.
- [34] AMD, Vivado Logic Simulation Overview, <https://docs.amd.com/r/en-US/ug900-vivado-logic-simulation/Overview>, Accessed: 2025-01-29.
- [35] AMD, Vivado Design Suite User Guide: Designing with IP (UG896), <https://docs.amd.com/r/en-US/ug896-vivado-ip/COE-File-Syntax>, Ac-

- cessed: 2025-02-02.
- [36] Erika Yamashita and ATLAS Collaboration, A comprehensive firmware validation machinery for the Level-0 Endcap Muon trigger for LHC-ATLAS Phase2 upgrade Technical Report ATL-DAQ-PROC-2024-012, University of Tokyo (JP), September 2024 Publication date: 19 Sep 2024.
 - [37] MySQL, MySQL, <https://www.mysql.com>, Accessed: 2025-02-02.
 - [38] 山下 恵理香, 高輝度 LHC-ATLAS 実験に向けた初段ミューオントリガーの開発と検証, 東京大学修士論文, 2023.
 - [39] Ren Kobayashi and ATLAS Collaboration Upgrade of the ATLAS Level-0 Endcap Muon Trigger for HL-LHC In *International Conference on Technology and Instrumentation in Particle Physics (TIPP 2021)*, pp. ATL-DAQ-SLIDE-2021-215, Online, Canada, May 2021 Submitted by ren.kobayashi@cern.ch on 10 Jun 2021.
 - [40] Intel, PCIe Support, <https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-guidance/pcie-support.html>, Accessed: 2025-02-02.
 - [41] AMD, Vitis High-Level Synthesis User Guide (UG1399), <https://docs.amd.com/r/en-US/ug1399-vitis-hls>, Accessed: 2025-02-02.
 - [42] AMD, Getting Started with Alveo Data Center Accelerator Cards User Guide (UG1301), Accessed: 2025-02-02.
 - [43] AMD, Alveo Product Selection Guide, https://www.mouser.com/pdfDocs/alveo-product-selection-guide_2021.pdf, Accessed: 2025-01-29.
 - [44] DigiKey, A-U250-P64G-PQ-G, <https://www.digikey.jp/en/products/detail/amd/A-U250-P64G-PQ-G/9645683>, Accessed: 2025-01-29.
 - [45] FUJISOFT, Alveo U200 Data Center Accelerator Card, https://www.fsi-embedded.jp/product_detail/11672/, Accessed: 2025-01-29.
 - [46] Techopedia, Direct Memory Access (DMA), <https://www.techopedia.com/jp/definition/direct-memory-access-dma>, Accessed: 2025-01-29.
 - [47] AMD, U200 and U250 Data Center Accelerator Cards Data Sheet, <https://www.avnet.com/opasdata/d120001/medias/docus/190/XLX-A-U200-P64G-PQ-G-Datasheet.pdf>, Accessed: 2025-01-29.
 - [48] Crucial, Difference Among DDR2, DDR3, DDR4, and DDR5 Memory, <https://www.crucial.jp/articles/about-memory/difference-among-ddr2-ddr3-ddr4-and-ddr5-memory>, Accessed: 2025-01-29.
 - [49] AMD, AXI Intellectual Property, <https://www.xilinx.com/products/>

- intellectual-property/axi.html, Accessed: 2025-01-29.
- [50] AMD, Vivado Revision Control Tutorial, <https://docs.amd.com/v/u/en-US/ug1198-vivado-revision-control-tutorial>, Accessed: 2025-01-29.
- [51] AMD, Vitis Accelerated Reference Design Guide (UG1702), <https://docs.amd.com/r/en-US/ug1702-vitis-accelerated-reference/Navigating-Content-by-Design-Process>, Accessed: 2025-02-02.
- [52] 丸元 星弥, FPGA 搭載型アクセラレータカードによるファームウェア開発・検証機構の構築, 神戸大学修士論文, 2023.
- [53] 塩見 崇宏, LHC-ATLAS 実験における第三期運転に向けた初段ミューオントリガーアルゴリズムの開発, 神戸大学修士論文, 2021.