修士論文

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 理学研究科 物理学・宇宙物理学専攻

物理学第二教室　高エネルギー物理学研究室

学生番号 0530-27-7351

赤塚 駿一

2017年1月26日

概要

LHC は欧州原子核研究機構 (CERN) に設置された陽子-陽子衝突型加速器である。ATLAS 検出器は LHC に設置された大型汎用検出器の一つであり、陽子衝突により発生する粒子を計測する。衝突により発生した未知の新粒子の崩壊を捉える、あるいはヒッグスボソンやトップクォーク等の標準理論粒子を精密に測定することにより、標準理論を超える新物理の発見を目指す。

LHC における陽子-陽子衝突の頻度は 40 MHz であるが、ATLAS 実験では記録レートの制限のため全事象を記録することはできない。そのため、物理として重要な事象を選択 (トリガー) し、データを取得する必要がある。本研究で扱うミューオントリガーは、ミューオンの横運動量に閾値を設けることで W ボソンの崩壊等による高運動量ミューオンを終状態に含む事象を選択する。トリガーは 2 段階に分けて行われ、初段のレベル 1 トリガー (L1) では全事象に対しトリガー判定を行う。ここでは 2.5 μs 以内に 100 kHz 以下までイベントレートを落とす、という厳しい制約がある。このため、L1 は高速処理が可能なハードウェアベースで実装されている。

LHC は 2018-2021 年に加速器の改良を行い、重心系エネルギーを現状の 13 TeV から 14 TeV に、瞬間ルミノシティを現在の約 2 倍の 3 $\times 10^{34}$ cm^{-2}s^{-1} に増加させ運転を再開する (LHC Run-3)。ルミノシティが増加する一方で、L1 トリガーレートに対する制約は変わらない。重要な物理事象を最大限有効に集めるために、対象となる物理事象の取得効率を維持したまま、2 倍のバックグラウンド除去が可能な新しい L1 トリガーの開発が緊急課題である。私は、現行のトリガー検出器 TGC と、Run-3 で ATLAS 検出器内層に導入される新検出器 New Small Wheel (NSW) の情報を組み合わせるトリガー判定ロジックの有用性に着目した。100 μ m の位置分解能と 1 mrad の角度分解能を持つ NSW からの情報を用いた新しい L1 トリガー実現のために、以下の 2 つの研究を行いミューオントリガーの改良を行った。

1 つ目の研究はハードウェア開発で、Run-3 ミューオントリガー判定ボード New Sector Logic (NewSL) の開発と、そのファームウェアのデザインを行った。新ロジックの実装のために、開発した NewSL には 10 Gbps の高速シリアル通信を実装した最新の FPGA を用いた。私は NewSL に実装されたチップ及び I/O インターフェイスの試験を行い、Run-3 で要求される性能を満たしていることを示した。さらに、Run-3 で NewSL に実装するファームウェアを開発し、これをビームテストで使用して、高速シリアル通信で受信した情報を用いたトリガー判定、受信したデータの保持・成型・圧縮、及び TCP/IP での読み出しに成功した。

2 つ目の研究はシミュレーションデータを用いた NewSL に実装する新しいトリガーロジックの開発である。私は、NSW の角度情報を用いた新ロジックを考案し、その性能評価を行った。その結果、角度情報を適切に利用するようにロジックを改善することで、改善前に比べさらに最大約 7.5%のトリガーレート削減が可能であることを示した。

目 次

第 1 章　イントロダクション **1**

1.1　標準理論と新物理 1

1.2　LHC 加速器 1

1.3　ATLAS 実験 2

1.4　ATLAS トリガーシステム 13

1.5　ATLAS phase-1 Upgrade 16

第 2 章　ATLAS エンドキャップミューオントリガー **22**

2.1　現行の L1 Endcap ミューオントリガー 22

2.2　ミューオントリガーのアップグレード 36

第 3 章　新トリガー判定ボード New Sector Logic の開発 **42**

3.1　New Small Wheel からの受信データ 42

3.2　New Sector Logic に対する要求 45

3.3　New Sector Logic のデザイン 46

3.4　NewSL における新技術 51

3.5　基礎性能評価 57

第 4 章　New Sector Logic Firmware の開発 **63**

4.1　Firmware とは 63

4.2　Firmware の概要 64

4.3　GTX/G-Link 送受信 Firmware 64

4.4　読み出し Firmware 69

4.5　Trigger Firmware 76

4.6　性能評価 76

第 5 章　New Sector Logic におけるトリガーロジックの開発 **82**

5.1　$d\eta$、$d\phi$、$d\theta$ によるカットロジック 82

5.2　位置のマッチングを用いたロジック 83

5.3　角度マッチングを用いた新ロジック 90

5.4　トリガーレートの評価 92

5.5　Firmware への実装 97

第 6 章　ビームテストによる性能評価 **101**
6.1　目的 101
6.2　セットアップ 101
6.3　NewSL に実装した Firmware 103
6.4　ビームテスト結果 106

第 7 章　結論と展望 **113**

謝辞 **114**

参考文献 **116**

付 録 A GTX Eye Patten **118**

第1章　イントロダクション

1.1　標準理論と新物理

標準理論は、物質を構成する最小単位である素粒子とその相互作用を記述する理論である。標準理論では 12 種類のフェルミオンと 4 種類のゲージボソン、ヒッグス粒子の計 17 種類の粒子が導入されており、2012 年にヒッグス粒子が LHC-ATLAS 実験と CMS 実験により発見され、標準理論で予言された粒子は全て発見された。

標準理論は現在までの実験結果のほとんどを説明することに成功している一方で、重力の導入、ダークマターの存在の説明、ヒッグス粒子の階層性問題、ニュートリノの質量の起源等、説明できない現象も観測されている。これらの現象を説明するために、標準理論を超えた新しい物理理論が必要であり、世界中で様々な方法で新物理の手がかりを探索する実験が行われている。

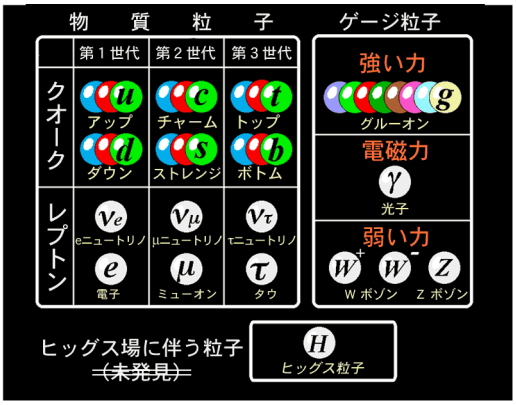


図 1.1　標準理論で導入されている粒子の図。

1.2　LHC 加速器

これらの新物理を発見するための基本的なアプローチとして、エネルギーフロンティアでの衝突実験がある。高いエネルギーで陽子を衝突させることにより、高い質量の新粒子を見つける、またはヒッグス粒子やトップ粒子のような質量の重い粒子を大量に生成し、結合定数等の精密な測定を行うことが狙いである。

以上のような目的のためにスイス・ジュネーブにある欧州原子核機構（CERN）の地下に建設された加速器が Large Hadron Collider （LHC）である。LHC は、周長 27 km の陽子-陽子衝突型の加速器であり、2017 年現在、重心系エネルギー 13 TeV の世界最高エネルギーで運転を続けている。

LHC は 2010 年より本格的な運転を開始し、2010 年 - 2012 まで運転を続けた。この間の陽子-陽子の重心系エネルギーは 7 TeV から 8 TeV であり、瞬間ルミノシティは最大で 0.77 $\times 10^{34}$ $\mathrm{cm}^{-2}\mathrm{s}^{-1}$ であった。この期間を LHC Run-1 と呼ぶ。その後、LHC は 2013 年 - 2015 年まで運転を休止し、加速器のアップグレードを行った。2015 年から現在に至るまで、LHC Run-2 が行われており、2018 年末まで継続予定である。Run-2 で重心系エネルギーは 13 TeV まで増加し、瞬間ルミノシティは 2016 年に過去最高値を更新し 1.37 $\times 10^{34}$ $\mathrm{cm}^{-2}\mathrm{s}^{-1}$ に達した。これまでに Run-1 と Run-2 で LHC が供給した統計量は約 75 fb^{-1} である。2021 年から開始予定の Run-3 では重心系エネルギーを 14 TeV まで増加させ、さらに瞬間ルミノシティ を 3 $\times 10^{34}$ $\mathrm{cm}^{-2}\mathrm{s}^{-1}$ まで上げて 3 年間運転する予定である。Run-3 終了までには 300 fb^{-1} 以上の統計量が得られると予想されており、新粒子へのさらなる感度向上や高統計での精密測定により新物理に迫ることが期待される。

1.3　ATLAS 実験

1.3.1　ATLAS 実験で目指す物理

ATLAS 実験の目的は、LHC による高エネルギー陽子-陽子衝突を用い、TeV スケールまでに期待される物理事象を幅広く調べることである。その中でも特に重要な目的の一つが、超対称性粒子（Super Symmetry Particle, SUSY 粒子） を始めとする標準理論を超える新粒子の発見である。LHC は世界最高エネルギーを持つ加速器であるため、重い新粒子まで探索感度を持つことができる。もう一つの重要な目的は、標準理論の粒子のうち、特に Higgs 粒子や top 粒子の生成断面積・崩壊分岐比を精密に測定することである。現在 Higgs や top を直接生成できる加速器は LHC だけであるので、ATLAS 実験による精密測定は非常に重要である。以下では、これらの物理目標のうち SUSY 粒子探索と Higgs 粒子の精密測定に絞って説明する。

SUSY

SUSY 粒子は、超対称性（Super Symmetry、SUSY）を仮定した際に導入される粒子の総称である。図 1.2 に示したように、超対称性理論では標準理論粒子に対する超対称性パートナーが存在するとされる。このような SUSY 粒子が存在すると、ヒッグス粒子の階層性問題や暗黒物質の候補等、標準理論では解決できない幾つかの問題を説明する事ができるため、非常に有力な理論である。特に、SUSY が階層性問題を説明するならば、数 TeV の質量領域に最も軽い SUSY 粒子が存在すると予想されるため LHC で直接生成により発見できる可能性がある。

Higgs 精密測定

Higgs 粒子の性質、特に他の粒子との結合定数の精密な測定は非常に重要である。Higgs の結合定数は標準理論ではフェルミオンの質量と比例するとされるが、標準理論を超える物理の寄与があればこの結合定数にズレが生じる可能性がある。実験で Higgs 粒子の結合定数を精度良く測る事で、新物理に対する手がかりを得る事ができる。

図 1.3、図 1.4 に LHC における Higgs の生成過程、及び主な Higgs の崩壊過程を示した。これらの生成過程と崩壊過程ごとに分類して計測・解析を行うことで、Higgs と他の粒子との結合定数を求める事ができる。図 1.5 に Run-1 における Higgs 粒子の崩壊の信号強度の測定結果のまとめを示した。このように Higgs 粒子の精密測定は、

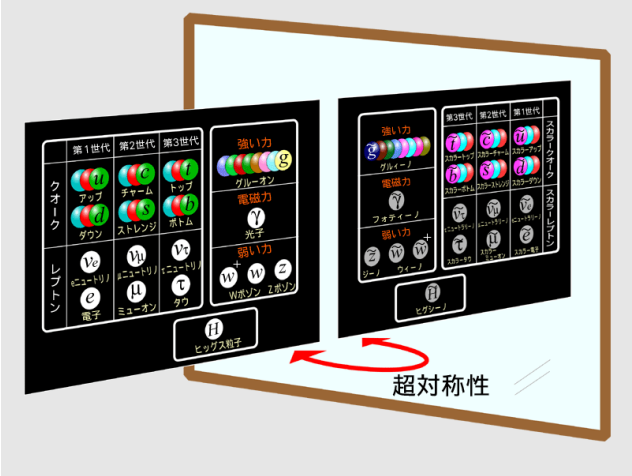


図 1.2　SUSY 粒子の図。標準理論粒子に対して超対称性パートナーが存在することが予言される。

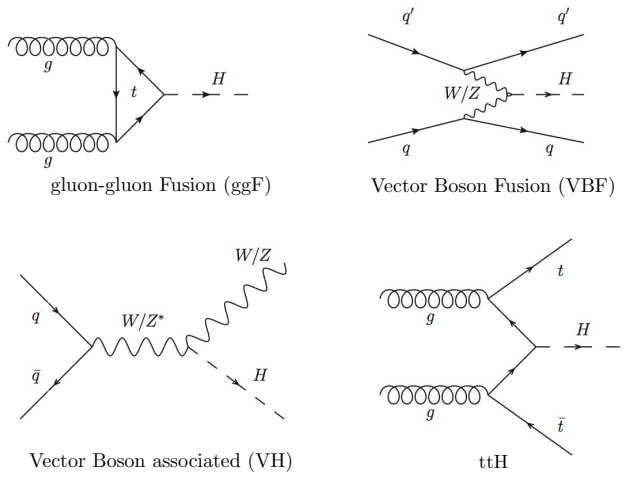


図 1.3　LHC における Higgs 粒子の生成過程。

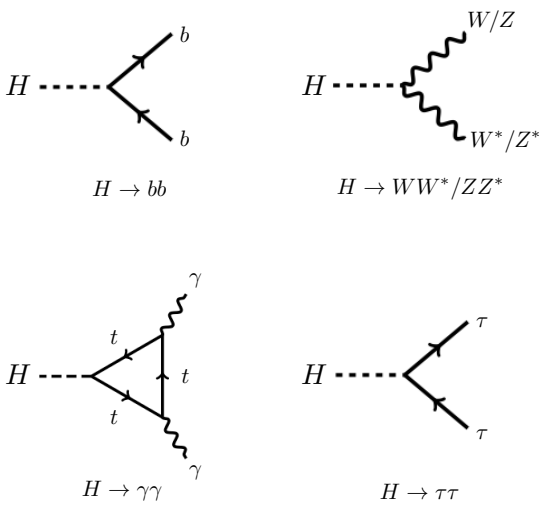


図 1.4　Higgs 粒子の主な崩壊過程。

様々な生成・崩壊チャンネルを用いて多角的に解析が進められている。より高い統計量のデータを用いてこれらの測定の誤差を小さくすることで、標準理論からの有意なズレの観測を目指す。

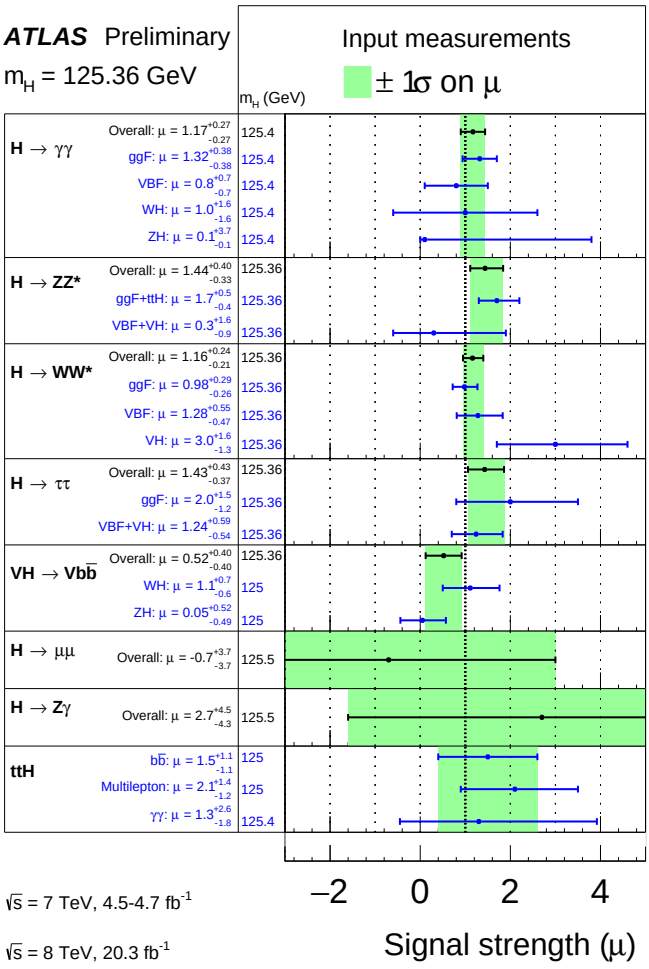


図 1.5　Run-1 における Higgs 粒子の崩壊信号強度の測定結果のまとめ。Signal Strength μ の定義は $\mu =$（観測された事象数）/（標準理論で予想される事象数）である。標準理論が正しければ μ は 1 となる。現在のところ標準理論予想との有意な差は見えていない。

1.3.2　ATLAS 検出器

ATLAS 検出器は、LHC の衝突点の一つに設置された、直径 25 m、長さ 44 m の汎用検出器である。図 1.6 に示したように、ATLAS 検出器は内側から飛跡検出器、カロリメータ、ミューオン検出器という順に構成されている。これらの検出器からの情報を組み合わせることで衝突点で生じた粒子の種類、エネルギー、運動量を計測する。本節では、ATLAS 検出器の概要を説明する。

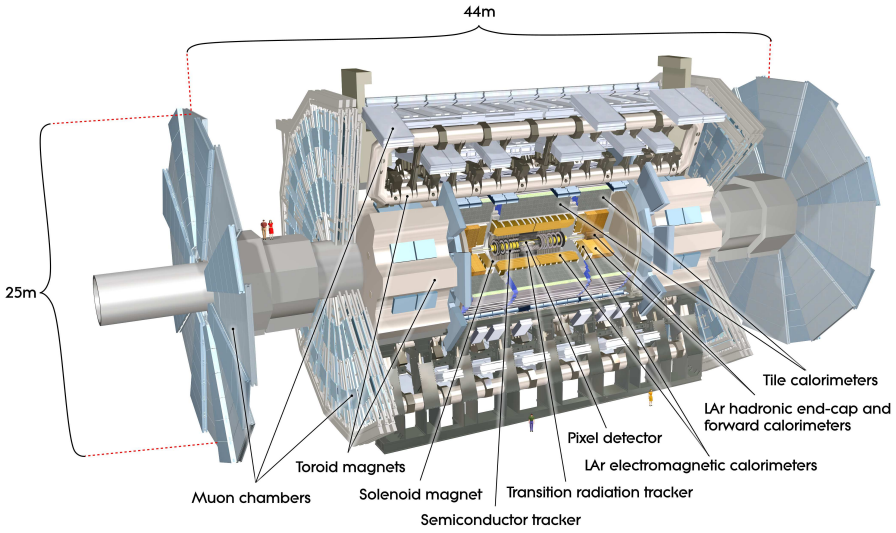


図 1.6　ATLAS 検出器の側面図 [1]。直径 25 m、長さ 44 m、重さ 7000 トン の円筒形汎用検出器。

LHC-ATLAS 実験で使用される座標系・変数

まず、ATLAS 実験で使用される座標系を導入しておく。ビーム軸方向に z 軸を取り、検出器の中心を原点とする。また、地面に垂直に y 軸を取り、上方向を正に取る。x 軸は LHC のリングの中心を正の向きに取る。ATLAS 検出器は円筒形をしているため、直交座標系と共に円筒座標系もよく用いられる。図 1.7 のように動径方向を R、方位角を ϕ で表す。

また、図 1.7 の θ 方向を表す際には擬ラピディティ η がよく用いられ、$\eta = -\ln(\tan\frac{\theta}{2})$ で定義される。ATLAS 検出器は円筒形をしており、側面部分と、底面部分で検出器の配置や種類が大きく異なる。この側面の部分を Barrel 領域、底面の部分を Endcap 領域と呼ぶ。例えば、ミューオン検出器においては Barrel 領域は $|\eta| < 1.0$、End-cap 領域は $|\eta| > 1.0$ に対応する。

粒子のエネルギー・運動量を表す際にビーム軸に垂直な成分 E_{T}、 p_{T} を利用する。これは、陽子-陽子衝突実験において、衝突する quark・gluon の z 軸方向のエネルギー・運動量は不定であるため、保存則を用いることができない一方で、ビーム軸に垂直な方向にはエネルギー・運動量の保存則がほぼ成り立つためである。ビーム軸に垂直な成分の保存則を用いると、ニュートリノ等検出できなかった粒子によるエネルギーの 2 次元的なベクトルの和を得ることができる。この見えないエネルギーの和を $E_{\mathrm{T}}^{\mathrm{miss}}$（missing E_{T}、MET）と呼ぶ。

超電導磁石

ATLAS 検出器では、荷電粒子の運動量測定のために内部に磁場がかかっている。磁場を発生させるための超電導磁石は 2 種類あり、一方は衝突点付近で発生した荷電粒子の運動量測定のためのソレノイド磁石を、他方はミューオンの運動量測定のためのトロイド磁石である。図 1.8 に、ATLAS 検出器内の磁石の配置を示した。

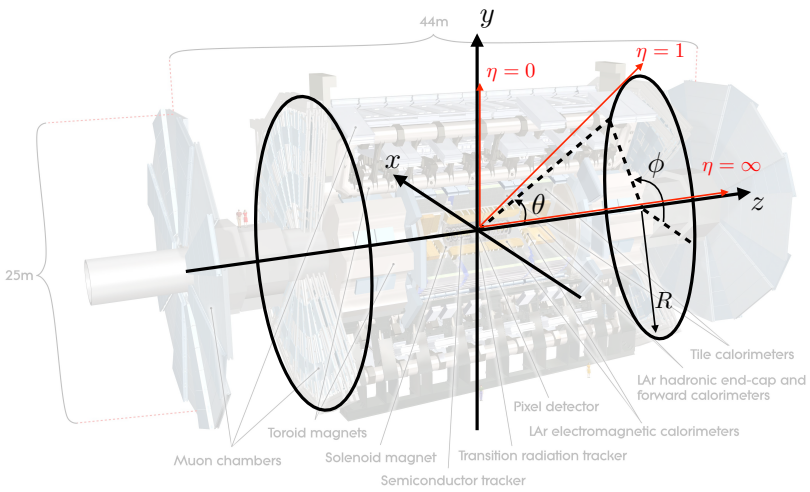


図 1.7　ATLAS で用いられる座標系。このように x, y, z 軸を取った直交座標系と、R, z, ϕ を用いた円筒座標系が用いられる。θ 方向を表す量としては擬ラピディティ $\eta = -\ln(\tan\frac{\theta}{2})$ が用いられる。

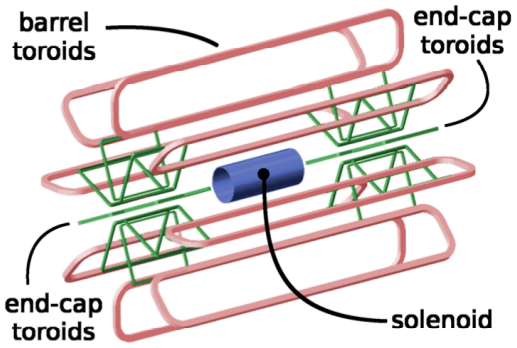


図 1.8　超電導磁石の配置。[1] 衝突点付近のソレノイド磁石と、外側を覆うトロイド磁石の 2 種類がある。

内部飛跡検出器

内部飛跡検出器は、Insertable B-Layer (IBL)、Pixel 検出器、Semiconductor Tracker (SCT) と Transition Radiation Tracker (TRT) からなり、衝突点で発生した荷電粒子の飛跡を計測する。内部飛跡検出器のある領域にはソレノイド磁場がかけてあり、荷電粒子の飛跡の曲がり具合から粒子の運動量を計算することができる。高いルミノシティ環境では 1 回の陽子バンチ交差で複数の陽子衝突事象が発生するため、精密な飛跡測定によりどの衝突点から生じた粒子であるかを識別することは非常に重要である。また、B 中間子等の長寿命の粒子の場合、再構成された飛跡から 2 次崩壊点を観測することができるので、これらの長寿命粒子を識別する際にも重要な役割を果たす。

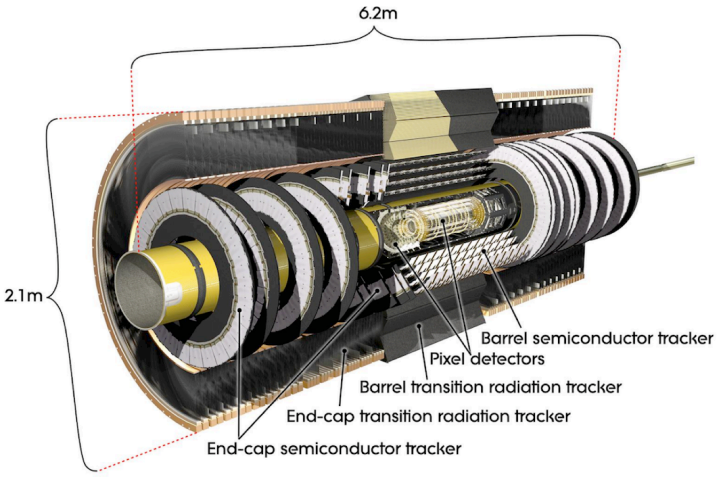


図 1.9　内部飛跡検出器の断面図 [1]。IBL は最内層に設置された検出器で、この図では書かれていない。その他の飛跡検出器は内側から Pixel、SCT、TRT の順に配置されており、それぞれ Endcap と Barrel の 2 種類に分けられる。

カロリメータ

カロリメータは電磁カロリメータとハドロンカロリメータの 2 種類からなる。電磁カロリメータでは、主に電子及び光子のエネルギーを計測し、ハドロンカロリメータでは強い相互作用によるシャワーを生じるハドロンのエネルギーを計測する。図 1.10 に、カロリメータの配置を示した。

LAr 電磁カロリメータ　電磁カロリメータは飛跡検出器のすぐ外側に設置された、鉛と液体アルゴンによるサンプリング型カロリメータである。電子と光子のエネルギーの測定に用いられる検出器で、Barrel 領域用と Endcap 領域用の検出器を合わせて $|\eta| < 3.2$ の領域を覆っている。検出器の厚さは Barrel 部・Endcap 部でそれぞれ放射長の 22 倍・24 倍以上になるように設計されている。

ハドロンカロリメータ　ハドロンカロリメータは電磁カロリメータの外側に設置されており、π中間子等のハドロンのエネルギーを測定し、quark や gluon から生じるジェットの再構成を行う。ハドロンカロリメータは $|\eta| < 1.7$

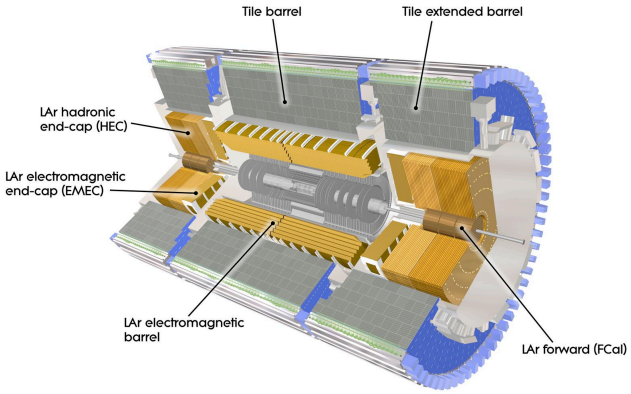


図 1.10　カロリメータの断面図 [1]。電磁カロリメータは Barrel・Endcap の 2 種類、ハドロンカロリメータは Barrel 領域の Tile、Endcap 領域の HEC と FCal の 3 種類で構成される。

の Barrel 部を覆う Tile Calorimeter と、$1.5 < |\eta| < 4.9$ を覆う Endcap 部の LAr ハドロンカロリメータの 2 種類に分けられる。Tile Calorimeter は鉄とプラスチックシンチレータによるカロリメータである。LAr カロリメータはさらに $1.5 < |\eta| < 3.2$ を覆う Hadronic End-cap Calorimeter （HEC） と、$3.1 < |\eta| < 4.9$ を覆う Forward Calorimeter （FCal）に分けられ、これらは吸収体としては銅またはタングステンを使用したカロリメータである。ハドロンカロリメータの厚さは、最も薄い $\eta = 0$ でも相互作用長の 11 倍になっている。

ミューオン検出器

ミューオン検出器は ATLAS 検出器の最外層に位置する。カロリメータを通過してミューオン検出器に入射する粒子は、標準理論の範囲内ではミューオンとニュートリノのみである。ミューオン検出器は MDT、 RPC、 TGC、CSC の 4 種類からなる。MDT、CSC は精密測定用の検出器であり、高い精度でミューオンの飛跡を再構成するのに用いられる。一方、RPC と TGC は主にデータ取得時の事象選別（トリガー）を行うために用いられる。

図 1.11、 1.12、 1.13 に、ミューオンシステムの配置図を示す。Endcap 領域のミューオントリガーには TGC が、Barrel 領域のミューオントリガーには RPC が用いられている。精密測定用の検出器としては Endcap 領域では MDT と CSC の 2 種類の検出器が、Barrel 領域では全域で MDT が用いられている。Endcap 領域では検出器は 3 つの円板状のステーション上に配置されていて、衝突点に近い方から Inner、Middle、Outer と呼ぶ。Barrel 領域では検出器は 3 層の筒状のステーション上に配置されており、これも衝突点に近い方から Inner、Middle、Outer と呼ぶ。これらのステーションを示す時には、図 1.10 に示したように Endcap-Inner ならば EI、Barrel-Outer ならば BO という様に略して表記する。また、図 1.13 の Endcap の MDT と CSC では、ϕ 方向の全域を覆うために大きなチェンバー（Large Chamber、図中で紫色で示されている）と小さいチェンバー（Small Chamber、図中で黄色で示されている）を組み合わせている。Barrel 部でも図 1.14 の様に、Large Chamber と Small Chamber を組み合わせている。これらを表す際には、先ほど説明した表記の最後に ”L” または ”S” をつけて表記する。例えば、Endcap-Outer の Large Chamber を示す際は EOL、Barrel-Inner の Small Chamber を示す際は BIS と表す。

図 1.12 には、正電荷のミューオン描く飛跡のイメージを示した。超電導磁石の節で説明した通り、ATLAS ではトロイド磁場を利用してミューオンの運動量を測定する。図 1.12 では紙面に垂直奥向きにトロイド磁場がかかっている。このため、磁場領域に入射したミューオンの軌道は η 方向に曲げられる。正電荷の場合は、図にあるように η が大きくなる方向に軌道が曲げられ、負電荷の場合は逆に η が小さくなる方向に曲げられる。運動量が高いほど飛跡は曲がりにくいので、これを利用して運動量の測定を行う。Inner、Middle、Outer の 3 ステーションにおける飛跡からトロイド磁場中での曲がり具合を評価し、運動量を計算する。

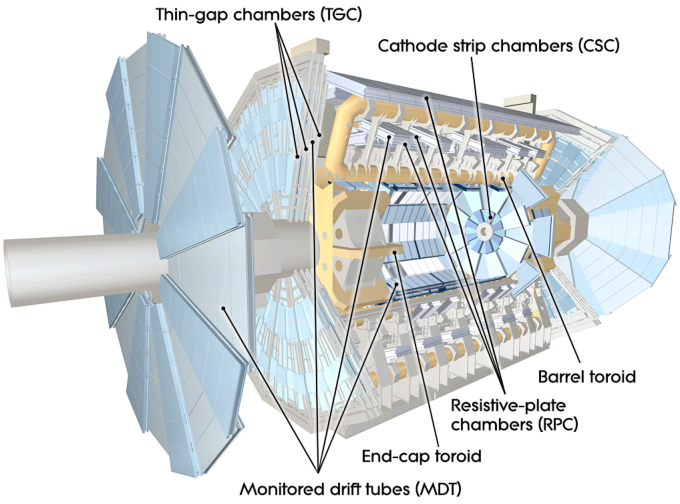


図 1.11　ミューオン検出器の断面図 [1]。ATLAS 検出器の最外層に位置し、トロイド磁場の前後に複数層の検出器を配置することでミューオンの飛跡を測定する。

以下に、4 種類のミューオン検出器 MDT、CSC、RPC、TGC の構造を説明する。

Monitored Drift Tube (MDT)　MDT は、ミューオンの運動量を精密に測定するための検出器である。図 1.15 のようなドリフトチューブを多数並べた構造になっている。図 1.16 のようにチューブを 3 層（または場所によっては 4 層）構造にしたものを 2 つ並べ、1 つの MDT チェンバーを構成する。図 1.17 のように各層のチューブの信号から直線を再構成し、高精度の方向測定を行う。Barrel 領域では z 方向、Endcap 領域では R 方向を精密に計測し、磁場の前後での方向の差から、運動量を測定する。MDT におけるイオンのドリフト時間は最大で 700 ns と長く、さらにハードウェアレベルでの飛跡の再構成は難しいためトリガーには用いられていない。また、ドリフト時間が長いことにより、高レート環境下では MDT の位置分解能が悪化することがわかっており、分解能が許容範囲に収まるイベントレートの上限値は 150 Hz/cm^2 である。

Cathode Strip Chamber (CSC)　CSC は、前方領域（$2.0 < |\eta| < 2.7$）で用いられている精密測定用の検出器である。前方領域では粒子の飛来レートが高く、MDT を使用した場合はレートの上限である 150 Hz/cm^2 を超えてしまう。そのため、応答が速く高いイベントレートに耐えうる CSC が用いられている。CSC は MWPC の構造のガスチェンバーで、図 1.18 に示したように約 5 mm 幅のカソードストリップにおける電荷の情報から重心を計算することで、 45 μm の分解能を達成している。アノードワイヤー同士の間隔、及びカソード面とアノード

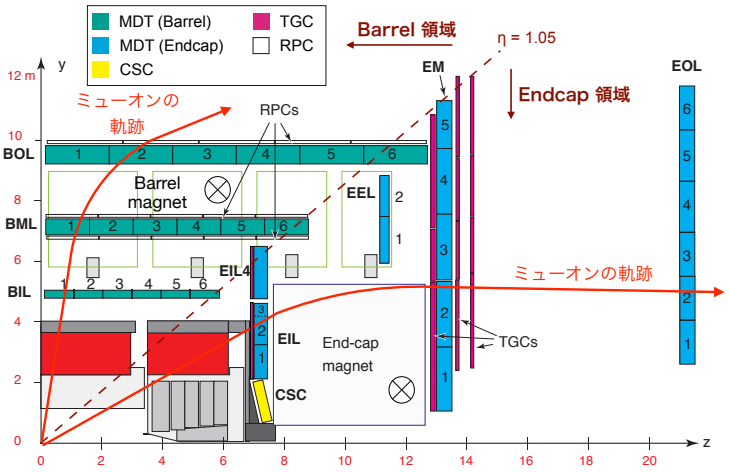


図 1.12　ミューオン検出器の y-z 平面上での配置。$\eta = 1.05$ を境に、検出器の配置が大きく異なることがわかる。赤の実線で正電荷のミューオンの描く飛跡を示した。紙面に垂直にトロイド磁場がかかっているため、ミューオンの飛跡は η 方向に曲げられる。ミューオン検出器ではその飛跡を磁場前後で測定することで運動量を計算する。

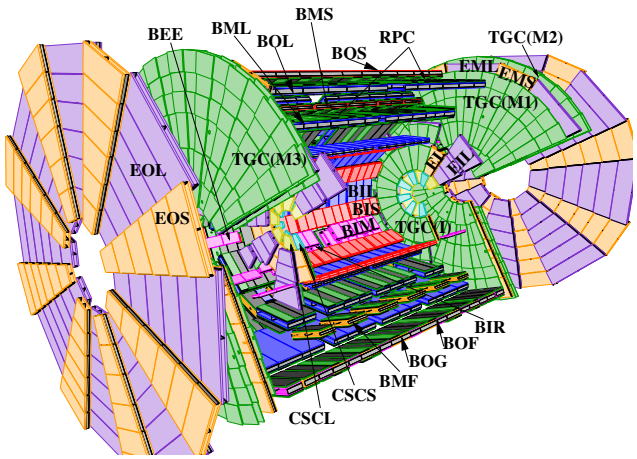


図 1.13　ミューオン検出器の配置図。チェンバー単位での命名規則が示されている。また、検出器の種類によって色分けされている。

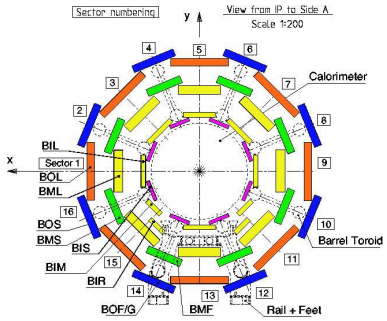


図 1.14　Barrel 部ミューオン検出器の配置。Small Chamber と Large Chamber を組み合わせて配置することで ϕ 方向を隙間なく埋めている。

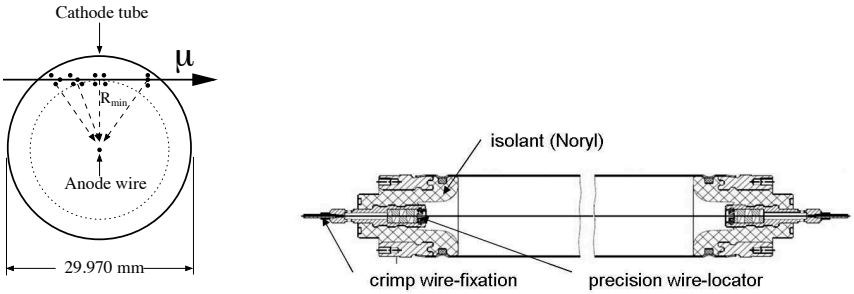


図 1.15　MDT チューブの断面図 [1]。左図: チューブの直径方向の断面図。右図: チューブの長さ方向の断面図。左図のようにチューブを通過したミューオンがガスを電離し、発生した電子はアノードワイヤーに回収される。ドリフト時間からミューオンの飛跡とアノードワイヤーとの距離を測定することができる。

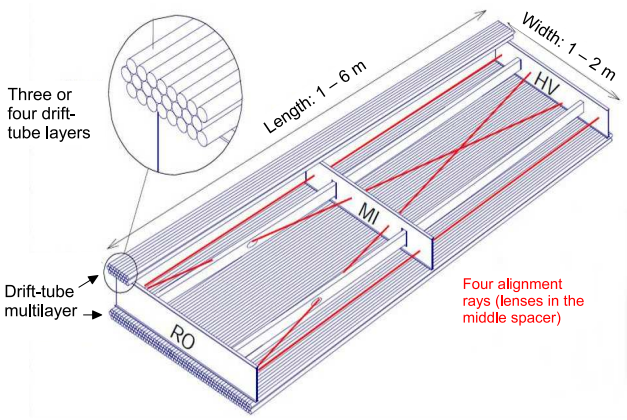


図 1.16　MDT チェンバーの構造 [1]。チューブを 3 層構造にしたものを、2 層並べた構造になっている。

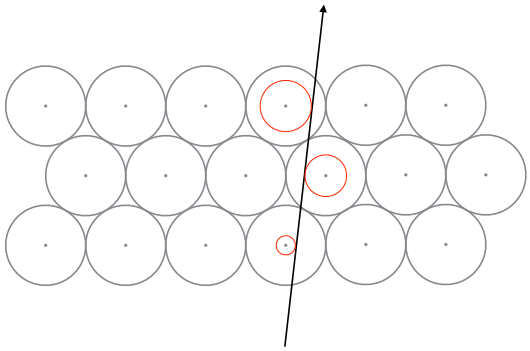


図 1.17　MDT によるトラックの再構成。6 層のチューブのヒットの時間情報を用いて直線を再構成することができる。

面の間隔は 2.5 mm と短いため信号の応答は速く、最大許容信号レートは 1 kHz/cm^2 である。

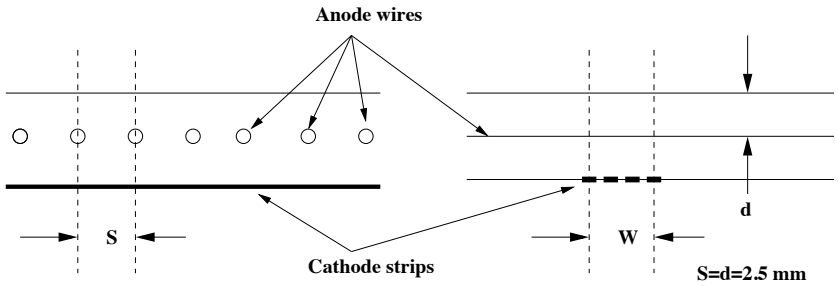


図 1.18　CSC 検出器の模式図 [1]。左図はアノードワイヤーに垂直な面から、右図は平行な面から見た断面図である。カソードストリップから読み出した電荷情報の重心を取ることで高い分解能を実現する。

Resisive Plate Chamber (RPC)　RPC は Barrel 領域のトリガー用検出器である。RPC は 2 mm のガスギャップに 9.8 kV の高電圧をかけたガスチェンバーで、それを直行するストリップで読み出す。図 1.19 に RPC の構造を示す。

Thin Gap Chamber (TGC)　TGC は、ATLAS 検出器エンドキャップ部に設置されたトリガー用検出器である。TGC は図 1.20 に示したように、ガスギャップ 2.8 mm の Multi Wired Proportional Chamber （MWPC）の構造をしている。動作ガスは CO_2 55%、n-pentane 45% の混合ガスで、アノードワイヤーにかける電圧は約 2.8 kV、この時のガスゲインは 3×10^5 である。アノードワイヤーとして、直径 50 μm の金メッキタングステンワイヤーを使用している。ワイヤー間隔は 1.8 mm と小さく設計されており、電子のドリフト距離を小さくすることで時間応答を向上させている。カソードはエポキシガラス板の片面に 1 MΩ/cm^2 のカーボンを塗布したものを使用している。エポキシガラスの反対の面には銅のストリップがアノードワイヤーと直行するように配置されている。ストリップ間隔は検出器上の位置により異なり、15 mm $\sim$ 53 mm である。ワイヤー・ストリップの情

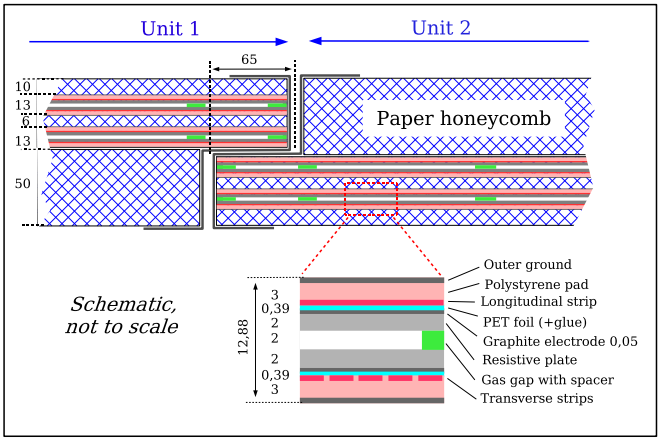


図 1.19　RPC 検出器の構造及び配置 [1]。高抵抗のプレートによりガスギャップが形成されている。ストリップはガスギャップの両面に、互いに直行するように設置されている。

報を用いて、粒子の飛跡の 2 次元読み出しが可能である。ガスギャップ及びワイヤー間隔が小さいことで検出器の時間応答は高く、特にワイヤーでは 90% 以上の確率で粒子が通過してから 25 ns 以内に信号が得られる。

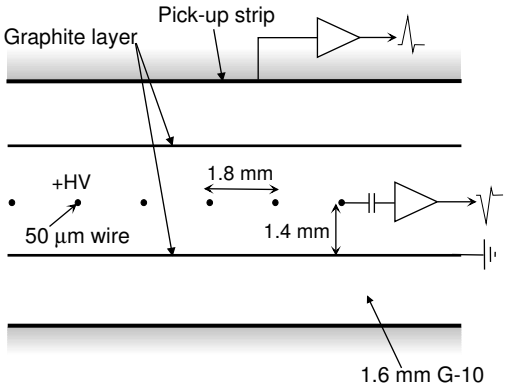


図 1.20　TGC 検出器の構造 [1]。ガスギャップ 2.8 mm、ワイヤー間隔 2.8 mm の MWPC の構造をしている。

図 1.21 に示すように、TGC チェンバー には 2 層構造の doublet チェンバーと、3 層構造の triplet チェンバーの 2 種類がある。doublet はワイヤー面 2 層・ストリップ面 2 層、triplet はワイヤー面 3 層・ストリップ面 2 層からなる。

1.4　ATLAS トリガーシステム

ATLAS 実験の高ルミノシティ環境下では、時間あたりに処理・記録できるイベント数の上限から、すべての衝突事象を記録して物理解析に用いることはできない。そのため、膨大な情報の中から如何に興味のある事象を選

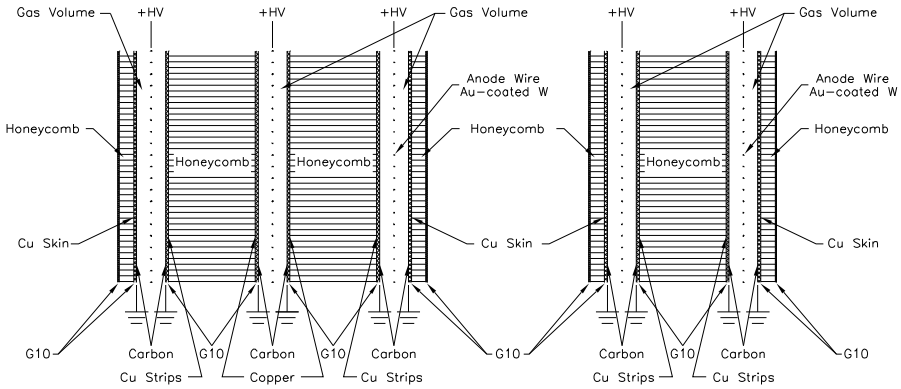


図 1.21　TGC 検出器の doublet・triplet の構成 [1]。各ガスギャップの間はハニカム構造のパネルで隔てられている。

別してデータとして記録するかが重要である。このデータ取得時に事象選別を行う部分をトリガーと呼ぶ。以下では、ATLAS のトリガーシステムについて説明する。

1.4.1　ATLAS トリガーシステムの概要

ATLAS 実験における陽子バンチの衝突頻度は 40 MHz である。一方で、データとして記録できるイベントレートは Run-2 においては約 1 kHz である。この制限を満たすために、ATLAS 実験では図 1.22 に示すように Level-1 Trigger、High-Level Trigger の大きく 2 段階に分けてトリガーをかけている。

1.4.2　Level-1 Trigger

Level-1 Trigger (L1 Trigger) では、図 1.22 に示すように ATLAS から 40 MHz で送られてくるすべてのイベントに対してトリガー判定を行い、2.5 μs 以内に 100 kHz までイベントを選別することが求められている。これを満たすために、L1 Trigger システムは高速処理が可能な ASIC 及び FPGA などの論理集積回路からなるハードウェアで実装されている。FPGA とは、Field Programmable Gate Array の略称で、使用者が中の論理回路を自由に書き換えることができる集積回路である。L1 システムがトリガー判定を行っている間、データは検出器上のフロントエンド回路（FE）の Buffer に保持されている（L1 Buffer）。

L1 Trigger はカロリメータの情報を用いて発行される L1 Calo、ミューオン検出器の情報を用いて発行される L1 Mu に加え、それらを組み合わせた複合的なトリガー L1 Topo の 3 種類に分類される。図 1.23 に、L1 Trigger の概要を示す。

L1 Trigger では、まずカロリメータ及びミューオン検出器の情報からそれぞれ単独にトリガー判定を行う。このとき用いることができる情報は検出器からの全情報ではなく、トリガー用に用意された情報のみである。ミューオンのトリガーは Barrel と End-cap 別々に判定されるので、それらの情報を MuCTPi （Muon to CTP Interfate）に送り、まとめた後に判定を行う。L1 Calo と L1 Mu はそれぞれ Central Trigger Processor （CTP）に送られると同時に、Topology Processor （L1 Topo）に送られる。L1 Topo では L1MU と L1Calo の情報を組み合わせ

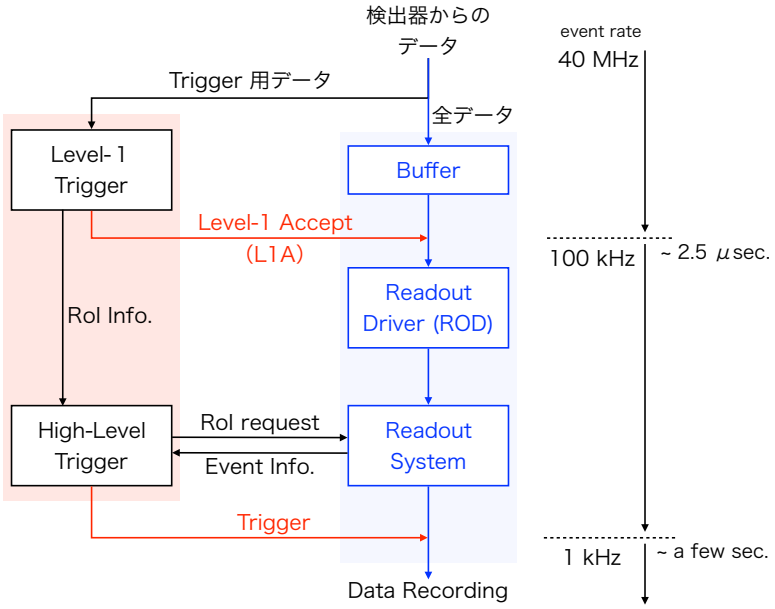


図 1.22　ATLAS トリガーシステムの概要。Level-1 Trigger では ∼ 2.5μs 以内にイベントレートを 100 kHz まで落とす。High-Level Trigger ではさらに詳しいトリガーを行い数秒以内にイベントレートを 1 kHz まで落とし、これを通過したものをデータとして記録する。

て、それぞれの数や位置関係から複合的な L1 トリガーを発行する。最後に L1 Mu、L1 Calo、L1 Topo の情報は CTP に集められ、100 kHz に収まるようにプリスケールをかけられた後に L1 Accept （L1A）としてトリガーが発行される。

L1 Trigger と Fixed Latency システム

L1 Trigger は、Fixed Latency のシステムを採用している。つまり、ある衝突事象が起きてから、その事象に対するトリガー判定結果が得られるまでの時間を常に一定にするような実装が行われている。これにより、前述のフロントエンド上の Buffer は常に一定時間データを保持して、トリガーが発行されていればデータを後段の ROD へ送り、されていなければ捨てる、というような実装をすればよいことになる。L1 Trigger 発行 2.5 μs 以内、というのはこの Buffer のサイズにより規定されている。

L1 Trigger の出力が Fixed Latency である、ということはつまり、それより上流のシステム全てが Fixed Latency であるということになる。すなわち、L1Mu、L1Calo、L1Topo、MuCTPi、CTP の全てのシステムがそれぞれ情報を受けてから結果を出すまでの Latency を一定にしている。

Latency を示す単位としてよく用いられるのが、Bunch Crossing（BC）である。LHC の陽子陽子衝突が 40 MHz であることから、衝突間の時間 25 ns を 1 BC と定義する。例えば、L1 Trigger 発行までの Latency は 2.5 μs = 100 BC ある。

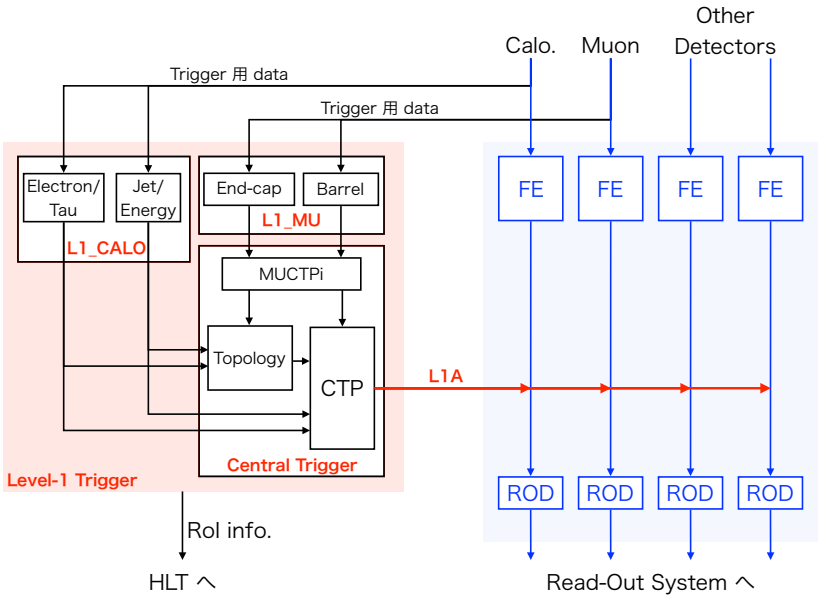


図 1.23　レベル-1 トリガーシステムの概要。

1.4.3　High-Level Trigger

High-Level Trigger（HLT）は、L1A が発行されたイベントに対し、ソフトウェアを用いて高精度な粒子再構成を行いトリガーを発行する。HLT では、Inner Detector の情報も用いて荷電粒子の飛跡再構成を行い、検出器の全情報を用いて横運動量や横エネルギーを計算し、トリガーレート削減を行う。HLT は衝突から 約 1 s の時間内でイベントレートを 1 kHz まで落とし、物理データとして保存する。

1.4.4　Trigger Menu

上で説明した L1 及び HLT のそれぞれのトリガーに、どれくらいのレートを配分するかを決めるのが、Trigger Menu と呼ばれるものである。Trigger Menu には様々な Luminosity の値に対して各トリガーに配分されるレートが規定されている。Run-2 におけるトリガーメニューの一例を、図 1.24 に示す。これを見ると、例えば赤枠で示したミューオンのシングルレプトントリガーでは、L1 で p_{T} 閾値 15 GeV のトリガーに 7 kHz 配分されているということがわかる。

1.5　ATLAS phase-1 Upgrade

1.5.1　Phase-1 Upgrade の概要

LHC のアップグレードによりルミノシティが上がるが、その恩恵を十分に受けるためには ATLAS 検出器のアップグレードも必要である。2021 年より開始するルミノシティ 3 $\times 10^{34}$ $\mathrm{cm}^{-2}\mathrm{s}^{-1}$ での運転、LHC Run-3 に向けたこの ATLAS 検出器のアップグレードを Phase-1 Upgrade と呼ぶ。

Trigger	Typical offline selection	Trigger Selection		Level-1 Rate [kHz]	HLT Rate [Hz]
		Level-1 [GeV]	HLT [GeV]	$L = 5 \times 10^{33}$ cm^{-2}s^{-1}	
Single leptons	Single iso μ, $p_T > 21$ GeV	15	20	7	130
	Single e, $p_T > 25$ GeV	20	24	18	139
	Single μ, $p_T > 42$ GeV	20	40	5	33
	Single τ, $p_T > 90$ GeV	60	80	2	41
Two leptons	Two μ's, each $p_T > 11$ GeV	2×10	2×10	0.8	19
	Two μ's, $p_T > 19, 10$ GeV	15	18, 8	7	18
	Two loose e's, each $p_T > 15$ GeV	2×10	2×12	10	5
	One e & one μ, $p_T > 10, 26$ GeV	20 (μ)	7, 24	5	1
	One loose e & one μ, $p_T > 19, 15$ GeV	15, 10	17, 14	0.4	2
	Two τ's, $p_T > 40, 30$ GeV	20, 12	35, 25	2	22
	One τ, one μ, $p_T > 30, 15$ GeV	12, 10 (+jets)	25, 14	0.5	10
	One τ, one e, $p_T > 30, 19$ GeV	12, 15 (+jets)	25, 17	1	3.9
Three leptons	Three loose e's, $p_T > 19, 11, 11$ GeV	$15, 2 \times 7$	$17, 2 \times 9$	3	< 0.1
	Three μ's, each $p_T > 8$ GeV	3×6	3×6	< 0.1	4
	Three μ's, $p_T > 19, 2 \times 6$ GeV	15	$18, 2 \times 4$	7	2
	Two μ's & one e, $p_T > 2 \times 11, 14$ GeV	2×10 (μ's)	$2 \times 10, 12$	0.8	0.2
	Two loose e's & one μ, $p_T > 2 \times 11, 11$ GeV	2×8, 10	2×12, 10	0.3	< 0.1
One photon	One γ, $p_T > 125$ GeV	22	120	8	20
Two photons	Two loose γ's, $p_T > 40, 30$ GeV	2×15	35, 25	1.5	12
	Two tight γ's, $p_T > 25, 25$ GeV	2×15	2×20	1.5	7
Single jet	Jet ($R = 0.4$), $p_T > 400$ GeV	100	360	0.9	18
	Jet ($R = 1.0$), $p_T > 400$ GeV	100	360	0.9	23
E_T^{miss}	$E_T^{miss} > 180$ GeV	50	70	0.7	55
Multi-jets	Four jets, each $p_T > 95$ GeV	3×40	4×85	0.3	20
	Five jets, each $p_T > 70$ GeV	4×20	5×60	0.4	15
	Six jets, each $p_T > 55$ GeV	4×15	6×45	1.0	12
b−jets	One loose b, $p_T > 235$ GeV	100	225	0.9	35
	Two medium b's, $p_T > 160, 60$ GeV	100	150, 50	0.9	9
	One b & three jets, each $p_T > 75$ GeV	3×25	4×65	0.9	11
	Two b & two jets, each $p_T > 45$ GeV	3×25	4×35	0.9	9
B−physics	Two μ's, $p_T > 6, 4$ GeV plus dedicated J/ψ-physics selection	6, 4	6, 4	8	52
Total				70	1400

図 1.24　トリガーメニューの一例。2015 年、ルミノシティ 0.5 $\times 10^{34}$ cm^{-2}s^{-1} におけるトリガーメニューである [2]。

ルミノシティの増加により、大きく影響を受けるのがトリガーである。イベントレートが増加する一方で記録レートは変わらないため、従来と同じトリガーシステムでは閾値を上げる/ プリスケールをかける等の対策を講じる事でトリガーレートを下げなければならなくなる。これでは、増加したルミノシティを十分に活かす事ができず、物理アクセプタンスを失う事となる。

Phase-1 Upgrade の第一の目標は、ミューオン・電子単独のトリガー、すなわちシングルレプトントリガーの閾値を維持することである。W/Z/H の電弱スケールの粒子の崩壊からのレプトンの p_{T} は数十 GeV と比較的低い。閾値を上げてしまうとこれらの重要な事象を大きく失うため、閾値の維持は非常に重要となる。シングルレプトントリガーに対しては電子・ミューオンそれぞれのトリガーシステムに改良を加えることでレートを抑え、閾値を維持する。

第二の目標は、低い p_{T} の jet や比較的小さな MET を作るような電弱スケールの物理のアクセプタンスを維持することである。Run-3 の L1 トリガーでは、 jet や MET のトリガーうち特に低閾値のトリガーは大きくレートを上げることが予想される。しかし、これらの低閾値のトリガーを捨ててしまうと低いエネルギースケールの物理へのアクセプタンスを大きく失う。このため、Run-3 では、jet/ MET に加えて他のオブジェクト（例えば、レプトン）や位置的な関係性を要求することでトリガーレートを抑制する。これを実現するために、複合的なオブジェクトによるトリガーを発行する L1 topo に現在以上にトリガーレートを配分することが必要になる。

以上の目標を達成するために、Phase-1 Upgrade では大きな改良として以下に説明する 2 つの変更が行われる。

1.5.2　LAr カロリメータ読み出し回路

1 回のバンチ交差で起こる陽子衝突の数を pile-up という。Run-2 での平均 pile-up は 40 程度であるが、Run-3 では 60 以上になると予想されている。pile-up が大きくなると、 1 イベントあたりに興味のある反応以外の衝突からの粒子でカロリメータ検出器に残るヒット情報が多くなるため、エネルギーの分解能が悪くなる。従ってカロリメータによるトリガーのレートは、ルミノシティに比例した増加ではなく、pile-up の影響により急激に増加することが予想される。

現在の LAr カロリメータのレベル 1 トリガーへの読み出し情報は $\Delta\eta \times \Delta\phi = 0.1 \times 0.1$ という、検出器の持つ本来の位置精度よりも粗い単位で読み出している。レベル 1 トリガーの段階で検出器の最大の細分化情報を用いることができれば、トリガー段階でのエネルギー分解能を改善でき、トリガーレートを削減できる。これを実現するために、Phase-1 Upgrade では LAr カロリメータの読み出し回路が刷新される予定である。本研究ではこのアップグレードは扱わないので、結果の予想のみを図 1.25 に示す（電子トリガーの例）。Phase-1 Upgrade 後のシステムで種々のカットをかけた場合と、Run-2 トリガーシステムとを比較すると、同じ閾値を設定した場合のトリガーレートは 50% 削減される。もしくは、同じトリガーレートを要求した場合には閾値を 10% ほど下げることができる。

1.5.3　New Small Wheel

Endcap 領域にある磁場の内側の検出器ステーションのことを Small Wheel と呼ぶ。Phase-1 Upgrade では、この Small Wheel に変えて新たな検出器 New Small Wheel （NSW）が導入される。New Small Wheel 導入の主な目的は、高レート環境での飛跡測定精度の向上と、次章で説明するミューオントリガーのアップグレードである。

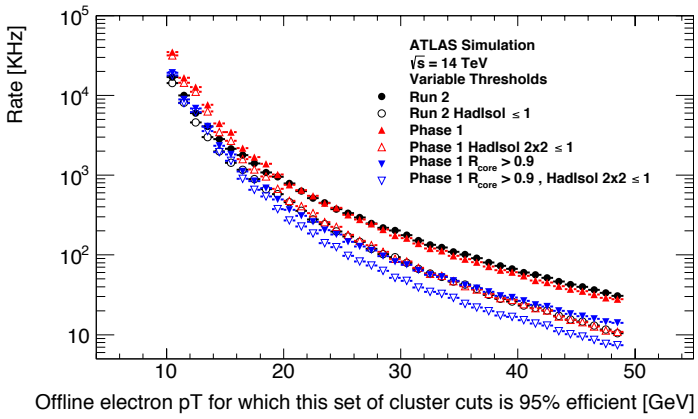


図 1.25　Phase-1 Upgrade 前後の電子トリガーのレート [3]。横軸はトリガー効率を 95% に保つように設定した閾値。

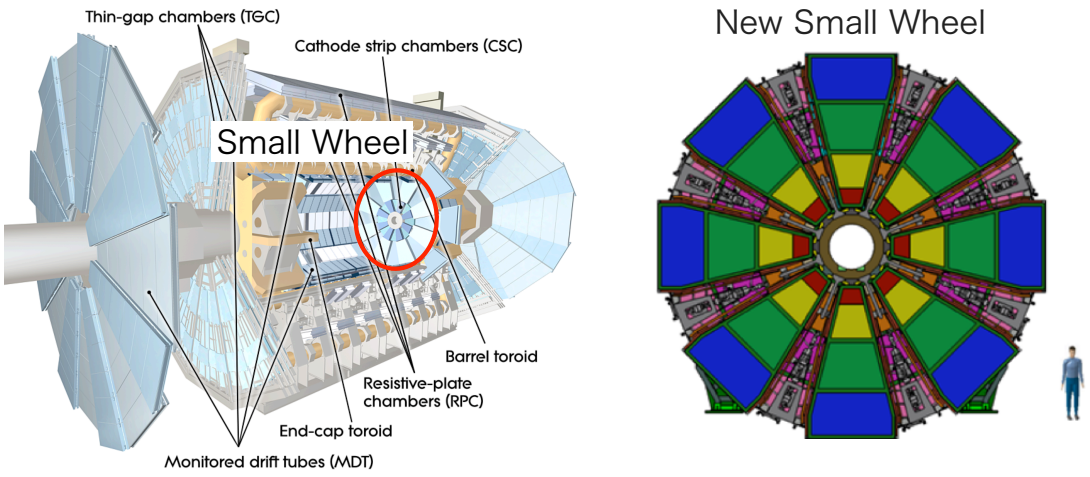


図 1.26　New Small Wheel の概要 [4]。

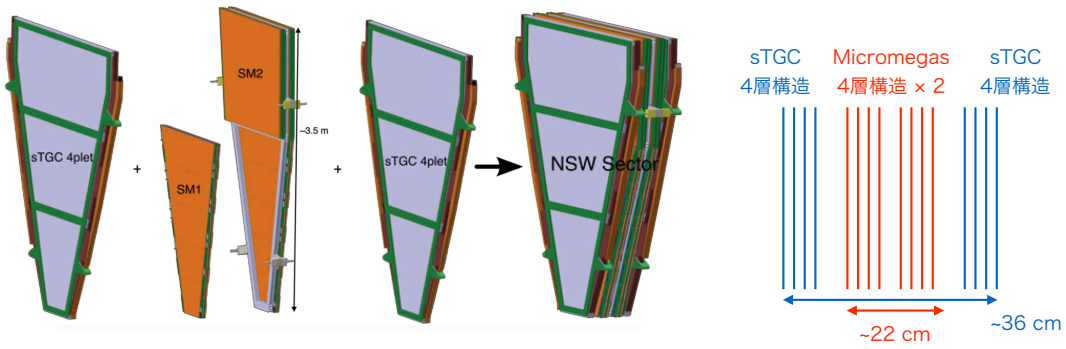


図 1.27　New Small Wheel 1 セクターの構成 [4]。 4 層構造の sTGC の間に 8 層構造の Micromegas が挟まれた構造になっている。

現在の Small Wheel の位置と、New Small Wheel の概要図を図 1.26 に示した。New Small Wheel は 8 回対称にデザインされており、Large セクターと Small セクターを組み合わせて ϕ 方向に隙間のない円盤型の構造を成す。この構造を Wheel 構造と呼んでいる。（磁場の外側は Big Wheel と呼ばれる。）それぞれのセクターは、主にトリガー検出器として用いられる small-strip TGC （sTGC）と精密測定用検出器として用いられる Micromegas（MM）の 2 種類の検出器で構成されている。図 1.27 に示すように、New Small Wheel は 4 層構造の sTGC（sTGC quadruplet）2 つの間に 4 層構造の MM が 2 つ挟まれた構造になっている。以下で、それぞれの検出器の構造及び性能を説明する。

1.5.4　small-strip TGC （sTGC）

トリガー用検出器にはイベントが起こった BC の ID のために速い応答が求められる。同時に、Phase-1 Upgrade においては 数 100 μm 以下の高い位置分解能も求められる。

sTGC は、時間応答を良くするために現行の TGC と同じくガスギャップ 2.8 mm、アノード間隔 1.8 mm の MWPC 構造を採用している。現行の TGC と sTGC とで大きく異なる点は、ストリップの情報を用いて η 方向を測るという点である。MWPC では、ストリップに誘起される信号の電荷情報用いて重心を計算する事で、ストリップ間隔よりもはるかに高い位置分解能を得る事ができる。sTGC は、ストリップ間隔が 3.2 mm と、現行の TGC でのストリップ間隔 15 mm に比べて小さく設計されている。この細かいストリップの電荷情報を用いて位置分解能を向上させる事ができる。sTGC quadruplet によるストリップ方向の位置分解能は粒子の飛来する角度に依り、 60 ~ 150μm である [4]。

もう一つ現行の TGC と異なる点は、パッドと呼ばれる読み出しカソードがあるという点である。図 1.28 に示すように、sTGC はアノードワイヤーを 2 枚のカソードで挟む構造になっており、片側は 3.2 mm ストリップを用いて、もう片方はパッドを用いて読み出す。パッドの大きさは η の値によって異なるが、典型的には 80 mm 程度であり、ストリップに比べると粗い読み出し単位となる。sTGC ではまずパッドを用いて粗いコインシデンスを取ることで粒子の通った大まかな位置を計算し、その後、その領域のストリップの情報のみを用いて精密な位置計算を行う。狭い領域に限った飛跡再構成をすれば良いため、短時間に精密な位置計算することが可能になる。

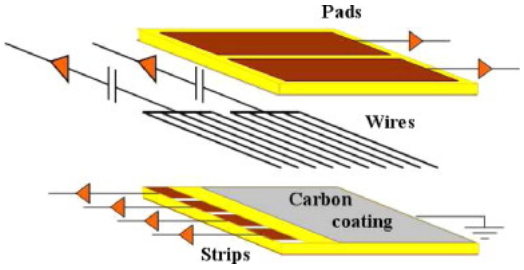


図 1.28　sTGC の検出器の構造 [4]。アノードワイヤーと垂直にストリップとパッドの読み出しが配置されている。

1.5.5　Micromegas （MM）

Run-3 で予想されるルミノシティ 3×10^{34} cm^{-2}s^{-1} において、$|\eta|$ の大きい領域の MDT では、粒子の飛来レートは優に 300 Hz/cm^2 を超えると予想される [4]。MDT の分解能が保証されるレートは 150 Hz/cm^2 であるため、新たな検出器を導入する必要がある。Phase-1 Upgrade では NSW 領域に MDT と同等の位置分解能を持ち、レート耐性の高い検出器として、Micromegas が導入される予定である。

Micromegas は図 1.29 に示すような検出器である。Micromegas の特徴は、増幅を mesh の後の短い区間で行うため、発生したイオンが mesh に到達するまでの時間が短くなることにある。これによりレートに対する耐久力が高い。また、増幅された電子による信号の時間差を用いることでトラックの z 方向の情報を再構成できるため、検出器に垂直に入射しなかった粒子に対する位置分解能を向上することができる点も特徴である。時間情報・電荷情報を組み合わせ、磁場による影響も考慮した場合の Micromegas 1 層での位置分解能は 90 μm である [4]。

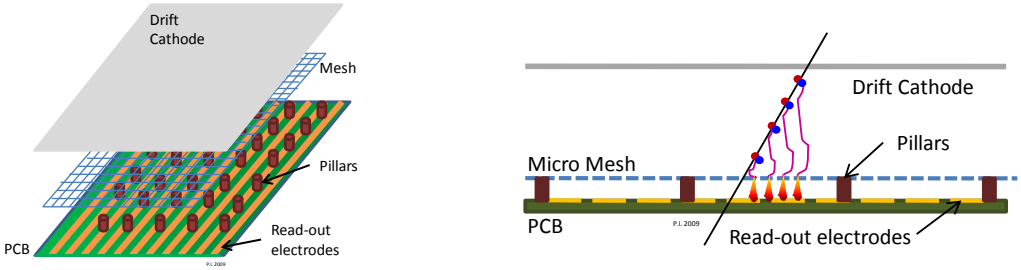


図 1.29　Micro Megas の検出器及び検出原理の概要 [4]。mesh 後の短い区間で電子増幅行われるため、レート耐性が高い。また、信号の到達する時間差を用いることで z 方向の情報を再構成できる。

第2章　ATLAS エンドキャップミューオントリガー

1 章で説明したように、ミューオントリガーは高い横運動量 p_{T} を持つミューオンを含む事象を選別するものである。ここでは、現行の Endcap 部 （$|\eta| > 1.05$）の Level-1 ミューオントリガーシステムについて説明し、その後同システムの Phase-1 Upgrade の方針について説明する。

2.1　現行の L1 Endcap ミューオントリガー

2.1.1　TGC の配置

1.3.2 項で述べた通り、Engcap 部のミューオントリガーは TGC 検出器が担っている。図 2.1 に、TGC の配置を示す。TGC は磁場の内側のステーションに 1 層、外側のステーションに 3 層の合計 4 層設置されている。磁場の内側のステーションは η 方向に分割された 2 つのチェンバーから構成され、$|\eta|$ が小さい（R の大きい）方のチェンバーを EI チェンバー、$|\eta|$ が大きい方のチェンバーを FI チェンバーと呼ぶ。磁場の外側のステーションの 3 層は衝突点に近い側から順に M1、M2、M3 と呼ぶ。

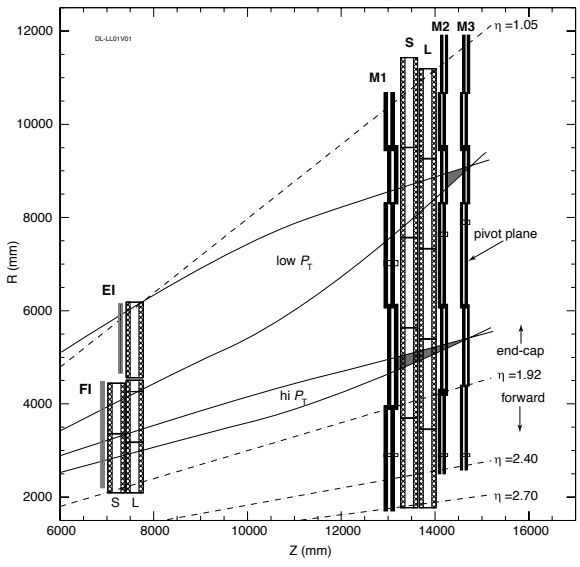


図 2.1　TGC の配置 [5]。磁場の内側に EI 及び FI チェンバー、外側に M1、M2、M3 が設置されている。S、L で示されているのは Endcap 部の MDT が設置されている位置である。

EI/FI、および M2・M3 は doublet チェンバーであり、ワイヤー・ストリップ面を共に 2 層持っている。M1 は TGC triplet チェンバーで、ワイヤー層を 3 層、ストリップ層を 2 層持っている。

図 2.2 および図 2.3 に TGC のチェンバーの配置を示した。M1-M3、及び FI は全方位角を覆っているのに対し、EI はトロイド磁石との干渉により、全方位角を覆っていない。

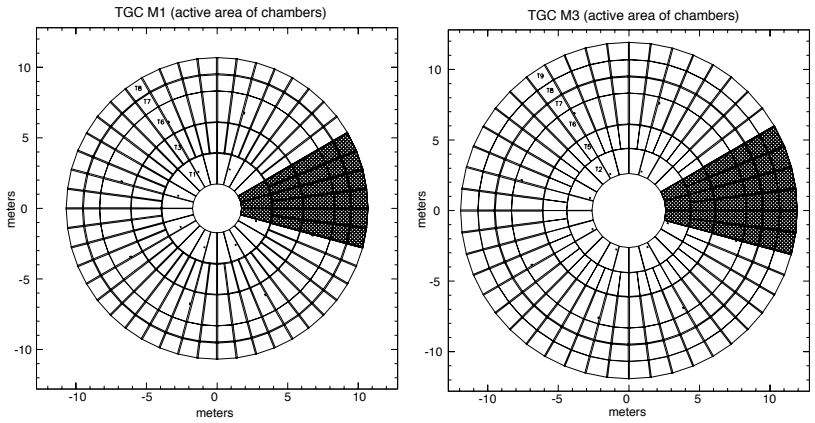


図 2.2　TGC M1、および M3 の R − ϕ 面でのチェンバーの配置。実線で囲まれた 1 マスが 1 つのチェンバーである [5]。M2 チェンバーも同様に方位角を全て覆うような円盤状の構造をしている。

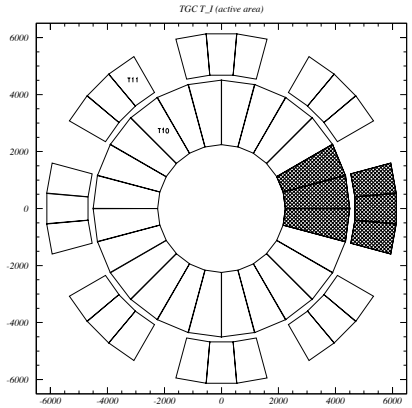


図 2.3　TGC EI、および FI の R − ϕ 面でのチェンバーの配置 [5]。FI チャンバーは全方位角を覆っているのに対し、トロイド磁石との干渉により、EI チェンバーは全方位角を覆っていない。

2.1.2　トリガー単位

TGC でのトリガー発行は、トリガーセクターと呼ばれる単位ごとに行われる。トリガーセクターは $1.05 < |\eta| < 1.9$ の領域を ϕ 方向に 48 分割、$1.9 < |\eta|$ の領域を ϕ 方向に 24 分割したものである。以下では、$1.05 < |\eta| < 1.9$ のものを Endcap トリガーセクター、$1.9 < |\eta|$ では Forward トリガーセクターと呼ぶことにする。トリガーセクターをまたぐ情報の共有は行われない。

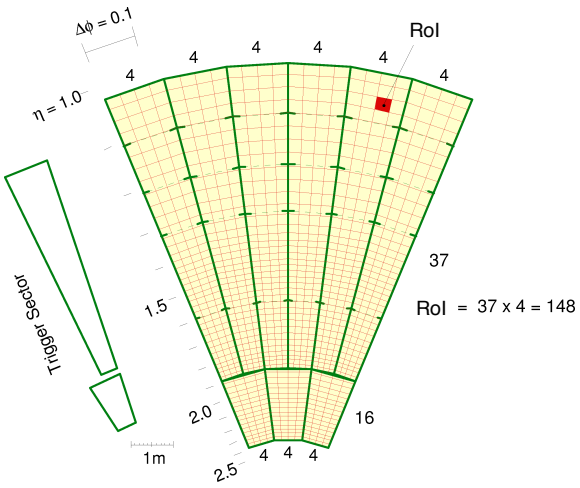


図 2.4　TGC のトリガーセクターと RoI [1]。緑の線で囲まれた部分が 1 トリガーセクターを表す。赤の線で囲まれたマスが 1 つの RoI を表す。

1 つのトリガーセクターは、η、ϕ 方向にさらに分割され、Region of Interest （RoI）という単位に分割される。Endcap トリガーセクターの RoI は 1 トリガーセクターを η 方向に 37 分割、ϕ 方向に 4 分割したものであり、大まかに $\Delta\eta \times \Delta\phi = 0.02 \times 0.03$ に対応する。Forward トリガーセクターの RoI は 1 トリガーセクターを η 方向に 16 分割、ϕ 方向に 4 分割したものであり、大まかに $\Delta\eta \times \Delta\phi = 0.03 \times 0.06$ に対応する。RoI が L1 ミューオントリガーの最小単位であり、これより細かい分解能の情報は L1 では用いることができない。

2.1.3　L1 Endcap ミューオントリガーの概要

Run-2 におけるエンドキャップ部のミューオントリガーのコンセプトを、図 2.5 に示す。

衝突点で生成されたミューオンは磁場の内側の検出器を通った後に、トロイド磁場領域に侵入する。トロイド磁場は ϕ 方向にかかっているため、ミューオンの飛跡は η 方向に曲げられる。飛跡の曲がり具合によって p_{T} を計算し、それに閾値を設けることでトリガー判定を行う、というのが基本的な手法である。

ここで注意したいのが、ミューオンの軌道が ϕ 方向へも曲げられているということである。衝突点付近のソレノイド磁場に加え、トロイド磁場も完全に ϕ 方向の成分のみを持つわけではなく、特に磁石の近くでは R 方向の磁場成分も持つ。これらの影響で、得られるミューオンの飛跡は ϕ 方向・η（R）方向の両方に軌道の変更を受ける。

2.1.4　TGC エレクトロニクス

TGC のエレクトロニクスには、トリガーロジックの実装と、データの読み出しという 2 つの役割がある。以下では、それぞれの役割がどのように回路上で実装されているかを説明する。図 2.6 に TGC エレクトロニクスの全体像を示す。

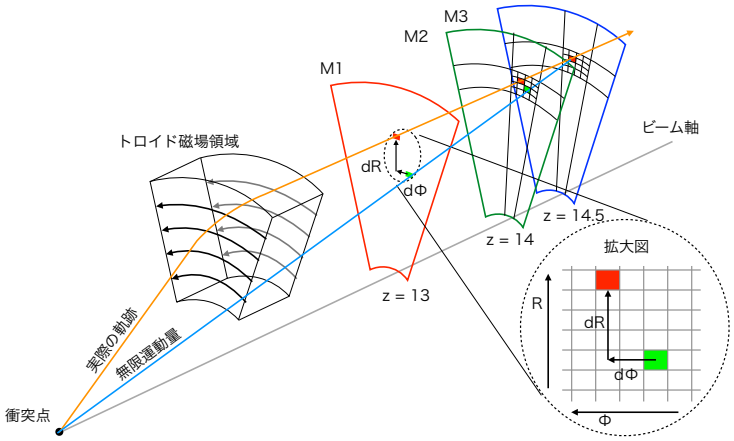


図 2.5　エンドキャップ部のトリガーのコンセプト。M1・M2・M3 でのヒットの位置からミューオンの飛跡の曲がり具合を測定し、その情報を用いてトリガー判定を行う。

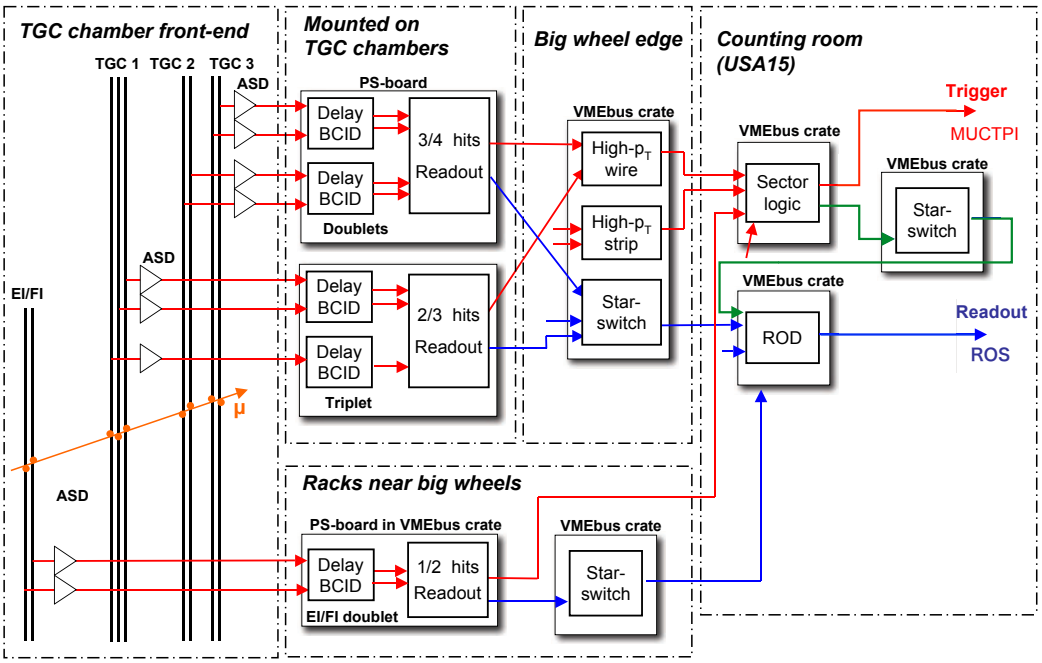


図 2.6　TGC エレクトロニクス [1] の全体像。赤線はトリガー情報の流れを、青線はデータ読み出しの流れを示している。また、緑線はトリガー判定情報の読み出しラインを示している。中央の PS-board はワイヤー用のもののみを示しているが、実際はストリップ用のものも存在する。High p_{T} strip や Star-switch、Sector logic に入力されている、逆側の繋がっていない矢印はストリップ用のデータラインを示している。

2.1.5　トリガーライン

トリガーラインは、トリガー用の情報からレベル 1 トリガーを発行するまでの信号の流れである。図 2.7 にトリガーラインの概要を示す。このように、データは ASD → PP → SLB → HPT → SL という順に受け渡され、トリガー判定結果は MuCTPi に送信される。以下ではそれぞれのエレクトロニクスの機能を説明する。

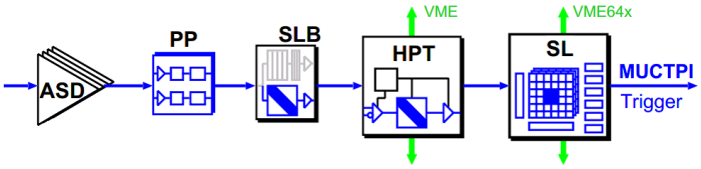


図 2.7　TGC エレクトロニクスのトリガーラインの概念図 [9]。検出器からの信号は ASD でデジタル化された後に PS ボード上の PP ASIC でタイミング合わせが行われる。その後、同じ PS ボード上の SLB ASIC →HPT→SL と段階的にコインシデンス処理が行われ、トリガー判定結果は MuCTPi へ送信される。

Amplifier Shaper Descriminator (ASD)

ASD は TGC で得られる電流信号を電圧信号に変換し、増幅したのちにコンパレータ回路を通してデジタル化する回路である。ASD ボードは TGC 検出器上に設置されており、図 2.8 のように、4 つの ASD ASIC を搭載している。ASD の信号は LVDS レベルで出力される。

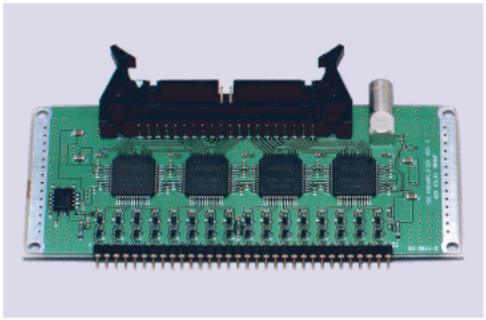


図 2.8　ASD ボードの写真 [10]。

Patch Panel and SLB ASIC ボード （PS ボード）

PS ボードは、以下で説明する Patch Panel ASIC と SLB ASIC を搭載したボードである。TGC 検出器上に設置されており、トリガー情報の送信のための HPT ボードと、読み出しのための SSW と接続されている。

Patch Panel (PP)

PP ASIC は、PS ボード上に搭載された ASIC である。PP ASIC は ASD から受信した LVDS レベルの信号のタイミング調整を行う ASIC である。ASD から受信した信号のタイミングは粒子の TOF、ASD PP ASIC のケーブルの長さの違い等によってチャンネル毎に異なるため、この PP ASIC でそれらのタイミング差を吸収し、同じ陽子衝突由来の信号が同時に次の SLB ASIC に入力されるように細かい遅延調整を行う。

SLB (トリガーライン)

SLB ASIC も PP ASIC と同じく PS ボード上に搭載されている。SLB ASIC には読み出しと、トリガーという 2 つの役割があり、図 2.9 に示したように、信号はそれぞれのために 2 つの経路に分けられる。ここではトリガーラインに注目して説明する。

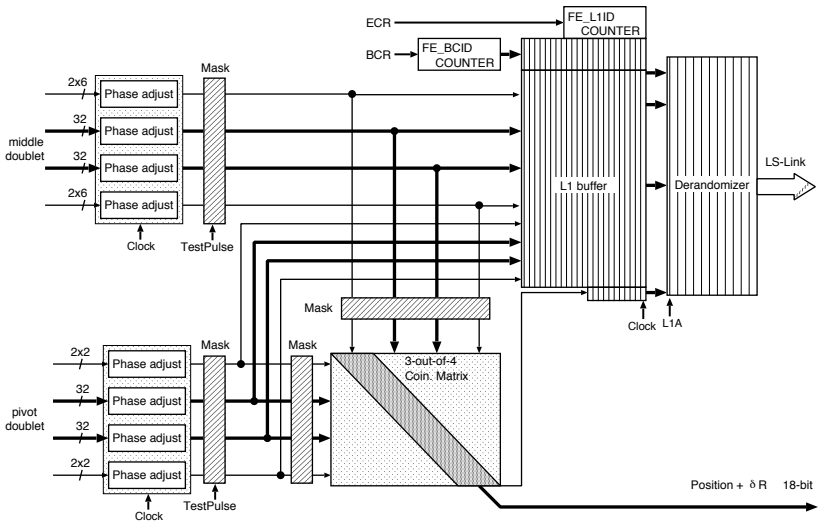


図 2.9　doublet ワイヤー用 SLB ASIC の全体のブロック図 [5]。上半分が読み出しライン、下半分がトリガーラインである。

SLB では、M1 の 2-out-of-3 （ストリップは 1-out-of-2）コインシデンス、M2-M3 の 3-out-of-4 コインシデンス、または EI/FI doublet の 1-out-of-2 コインシデンスが行われる。コインシデンスは図 2.10 に示すようにコインシデンスマトリックスを用いて行われる。この例は、M2 チェンバー 2 層分、M3 チェンバー 2 層分の情報から、3-out-of-4 を取ると同時に、それらの位置の差の小さいもの、すなわち対角線に近いもののみ送信するコインシデンスマトリックスである。M1 および M2-M3 のコインシデンス結果は LVDS で HPT ボードに送信される。この段階で位置情報は RoI 単位にエンコードされる。EI/FI の情報は G-Link 通信で直接 Sector Logic に送信される。G-Link 通信とは、専用のチップを用いたシリアル通信のことを言う。TGC システムでは G-Link でシリアル化した信号を光信号に変換して利用しており、通信速度は 640 Mbps または 680 Mbps である。従って、G-Link 光ファイバー 1 本あたり、1 BC （1/40 MHz）毎に 16-bit または 17-bit の情報を転送できる。

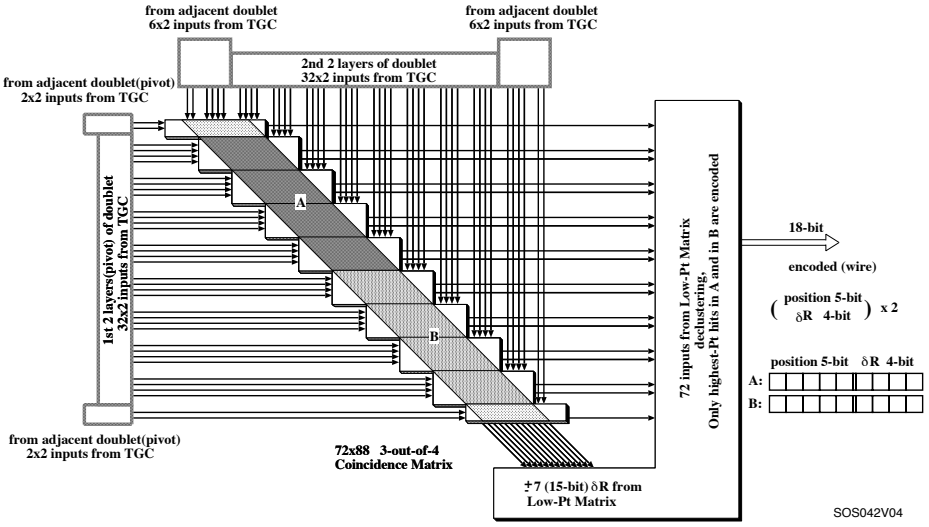


図 2.10　doublet ワイヤー用 SLB ASIC のトリガーラインのブロック図 [5]。横軸に M2 チェンバーのチャンネルを 2 層分、縦軸に M3 チェンバーのチャンネル 2 層分を並べコインシデンスマトリックスに入力し、両者の差の小さいものを後段に送信する。

HPT ボード

HPT ボードは、M1 の SLB と M2-M3 の SLB の出力情報を用いて、M1-M3 コインシデンスを行うモジュールである。実装には SLB と同じようにコインシデンスマトリックスを用いている。M1 のチェンネル情報および M3 のチャンネル情報から両者の位置の差 （ΔR またはΔϕ）を計算し、それが小さいもの （コインシデンスマトリックスの対角線に近いもの）の情報を Sector Logic に送信する。図 2.11 にワイヤー HPT ロジックのブロック図を示した。

HPT のコインシデンス結果は位置情報と ΔR またはΔϕ の情報、および HPT コインシデンスが取れたかどうかのフラグを合わせてワイヤー情報は 10-bit、ストリップ情報は 9-bit にエンコードされ、表 2.1 に示すようなデータフォーマットで G-Link 通信で Sector Logic へ送信される。

Sector Logic (SL)

SL はエンドキャップミューオントリガーの最終段のモジュールで、HPT ワイヤーおよびストリップと、EI/FI 用の PS ボード、Tile Calorimeter から受信した信号を合わせて、最終的な p_{T} 計算およびトリガー判定を行うボードである。SL ボードには Endcap トリガーセクター用と Forward トリガーセクター用の 2 種類があり、どちらも Endcap または Forward の 2 トリガーセクターの分トリガー判定を担当する。図 2.12 に、Endcap SL ボードの写真および主なチップを示した。

SL では R $-$ ϕ コインシデンスおよび Inner Coincidence ロジックが実装されている。R $-$ ϕ コインシデンスについてはのちに詳しく説明するが、図 2.14 のような Coincidence Window を、2 枚の FPGA 上に実装する。FPGA は書き換え可能であるため、検出器のアラインメントの変化や検出器の状態 （例えば閾値や HV 電圧、ガ

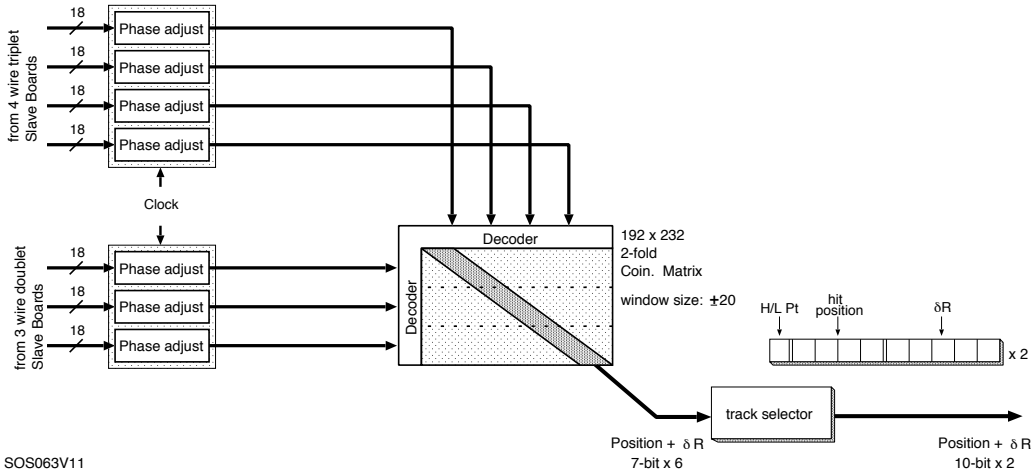


図 2.11　doublet ワイヤー HPT ロジックのブロック図 [5]。M1 と M3 の位置の差をコインシデンスマトリックスで計算して Sector Logic に送る。

表 2.1　HPT ボードが SL に送信するデータのフォーマット [8]。ΔR（$\Delta\phi$）の情報は符合も含めて 4-bit（3-bit）で送信される。H/L は HPT コインシデンスが取れたかどうかの情報で、1-bit で送信される。POS および HIT ID はヒット位置に関する情報で、4-bit で送信される。

Bit	Wire	Strip
0	ΔR[0]	$\Delta\phi$[0]
1	ΔR[1]	$\Delta\phi$[1]
2	ΔR[2]	$\Delta\phi$[2]
3	ΔR[3]	Sign
4	Sign	H/L
5	H/L	POS
6	POS	HITID[0]
7	HITID[0]	HITID[1]
8	HITID[1]	HITID[2]
9	HITID[2]	Not used

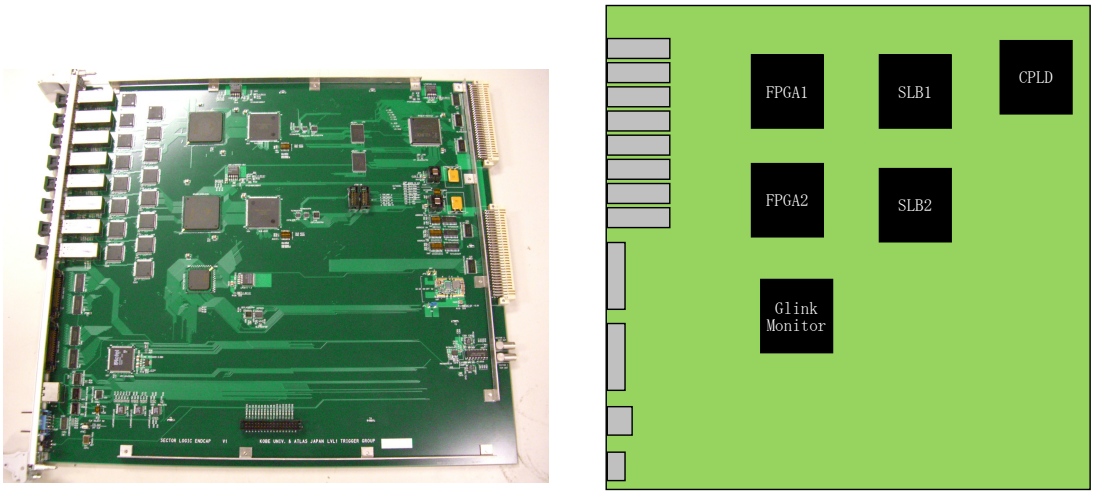


図 2.12　Sector Logic の写真（左）と、主なチップ（右）[8]。

スの状態）の変化などに応じて、Coincidence Window を変化させることにより最適なトリガー判定を行うことができる。R $-\phi$ コインシデンスを通過したイベントに対し、最後に磁場の内側の検出器とのコインシデンス、Inner Coincidence を取る。EI/FI および Tile Calorimeter の情報を用いて位置のマッチングを取り、通過したものを最終的に MuCTPi に送信する。

SL はトリガー判定に用いる信号を全て G-Link 通信で受け取る。Endcap の SL は図 2.13 に示すように 1 トリガーセクターあたりワイヤー用の HPT ボードから 4 本、ストリップ HPT ボードから 2 本のファイバーを用いて情報を受信するため、合計 $(4+2)$[本/トリガーセクター] $\times 2$ [トリガーセクター] の計 12 本の光ファイバーで TGC Big Wheel の情報を受信する。また、Endcap 部では後で説明する Inner Coincidence のために EI/FI の情報を、2 トリガーセクターをカバーする分としてファイバー 3 本で受信している。さらに、Tile Calorimeter からも Inner Coincidence のための情報をファイバー 1 本で受信しており、合計で $12+3+1=16$ 本のファイバーを用いて情報を受信している。Forward トリガーセクターの SL は 1 トリガーセクターあたりワイヤー HPT ボードから 2 本、ストリップ HPT ボードから 1 本のファイバーで信号を受信するため、合計 $(2+1)$[本/トリガーセクター] $\times 2$ [トリガーセクター] の計 6 本の光ファイバーで TGC Big Wheel の情報を受け取る。1 BC 毎に 1 枚の SL が受け取るデータ量は、Endcap SL では TGC BW から 202-bit、EI/FI から 64-bit、Forward SL では TGC BW から 50-bit である。

SL では HPT ボードから表 2.1 のフォーマットで受け取った ΔR、$\Delta\phi$ を運動量に変換しトリガー判定を行うが、その変換に用いる $(\Delta R,\ \Delta\phi) \rightarrow p_{\mathrm{T}}$ へのマップは RoI 毎に個別に定義される。これは、主に磁場の非一様性と、検出器のアラインメントの理想位置とのズレからくるもので、検出器上のヒット位置によって $\Delta R : \Delta\phi$ のコインシデンス条件を変化させることが望ましい。即ち、SL 上に図 2.14 のような $\Delta R : \Delta\phi$ コインシデンス条件を規定した Coincidence Window （CW）をすべての RoI に対して定めておく必要があるということである。SL ではこれを FPGA 上に Look Up Table （LUT）という形で実装している。LUT とは入力のパターンに対して出力パターンをあらかじめ定めた表のことであり、SL においては ΔR と $\Delta\phi$ のパターンに対して p_{T} 計算結果を

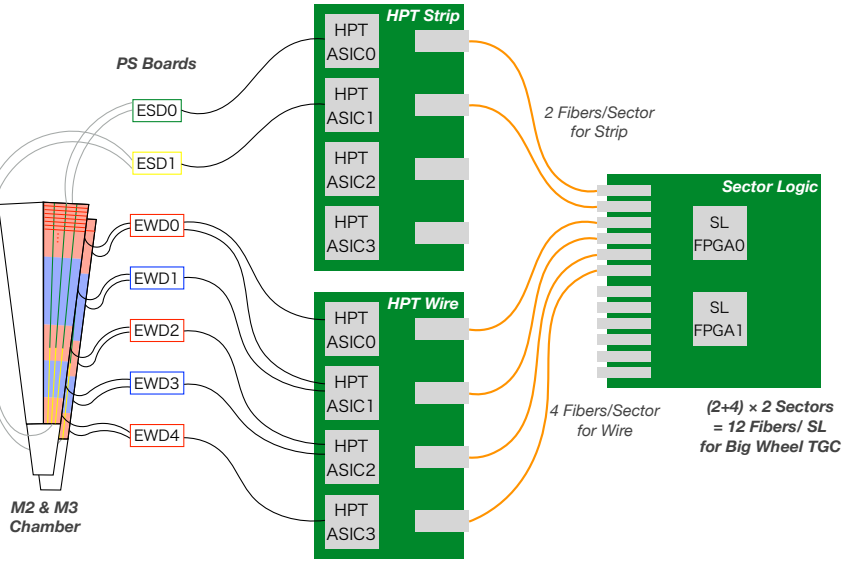


図 2.13　1 トリガーセクター分の、PS ボード、HPT ボード、SL の Big Wheel 信号の接続の概略。ESD、EWD はそれぞれ Endcap Strip Doublet、Endcap Wire Doublet を表している。HPT ボードでは実際は TGC M1 の対応する位置の PS ボード （Endcap Wire Triplet 等）からの情報を受け取っているが、図では省略した。この他に、SL は磁場の内側の検出器 （EI/FI、Tile Calorimeter）からデータを受信している。

出力する。1 つのチップで 1 トリガーセクターを担当するので、1 つの FPGA が持つ LUT の数は 1 トリガーセクターにおける RoI の数と一致し、Endcap 領域では 148 個、Forward 領域では 64 個である。

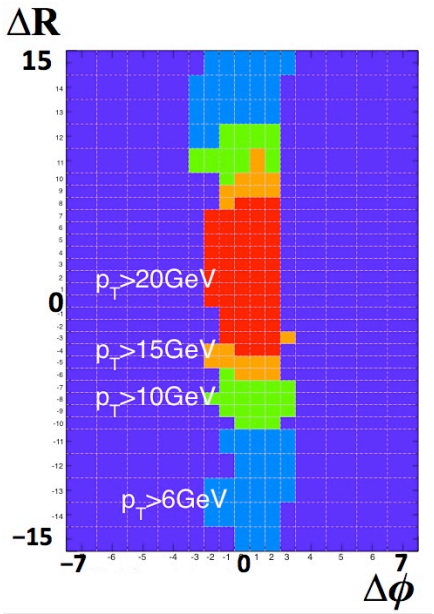


図 2.14　ある RoI における ΔR × Δϕ の Coincidence Window [6]。ΔR、Δϕ により出力結果を色分けして示しており、ΔR および Δϕ の小さい部分で高い p_{T} であると判定していることがわかる。p_{T} 計算の結果は 3-bit にエンコードされて出力される。

SL 上でのトリガーロジックの実装方法を詳しく説明する。図 2.15 に 1 トリガーセクター分の実装方法の概念図を示した。6 本の G-Link 信号で受信した信号は Decoder 部分でトラックの位置の情報 （HIT ID、POS）と方向の情報 （ΔR、Δϕ 、Sign および H/L 情報）とに分けられる。位置の情報を用いてどの RoI においてヒットがあったのかを判定し、それらの方向情報 11-bit 分を対応する LUT に配分する。LUT では H/L 、ΔR、Δϕ およびそれらの符合の情報を用いて、p_{T} と電荷情報の計算を行い、その結果を 4-bit で出力する。その後、EI/FI および Tile Calorimeter の情報を合わせてヒット位置のコインシデンス、Inner Coincidence を取り、最後に 1 トリガーセクターに 2 イベントまでトラック候補を絞り、後段の MuCTPi に送信する。

ここで、Inner Coincidence について説明しておく。これまでに説明したコインシデンスロジックは全て磁場の外側の TGC Big Wheel によるものである。Run-1 では実際に TGC Big Wheel 単体でトリガーを発行していたが、図 2.16 のように、特に $|\eta|$ の大きい領域においてオフラインでミューオンが再構成されないにも関わらずトリガー判定してしまっている場合が多いことが判明した。

この原因は、図 2.17 で「フェイク」で示されているようなバックグラウンド事象に対し、トリガー判定をしてしまうことであった。このようなフェイク事象を排除するために、Run-2 では図 2.17 に示されるように磁場の内側のいくつかの検出器とのコインシデンスを取っている。$1.0 < |\eta| < 1.3$ では EI および Tile Calorimeter と、$1.3 < \eta < 1.9$ では FI とコインシデンスを取ることで、フェイクトリガーを大きく削減することができる [7] [8]。現在、EI および FI コインシデンスは実際に ATLAS データ取得に用いられており、Tile Calorimeter コインシ

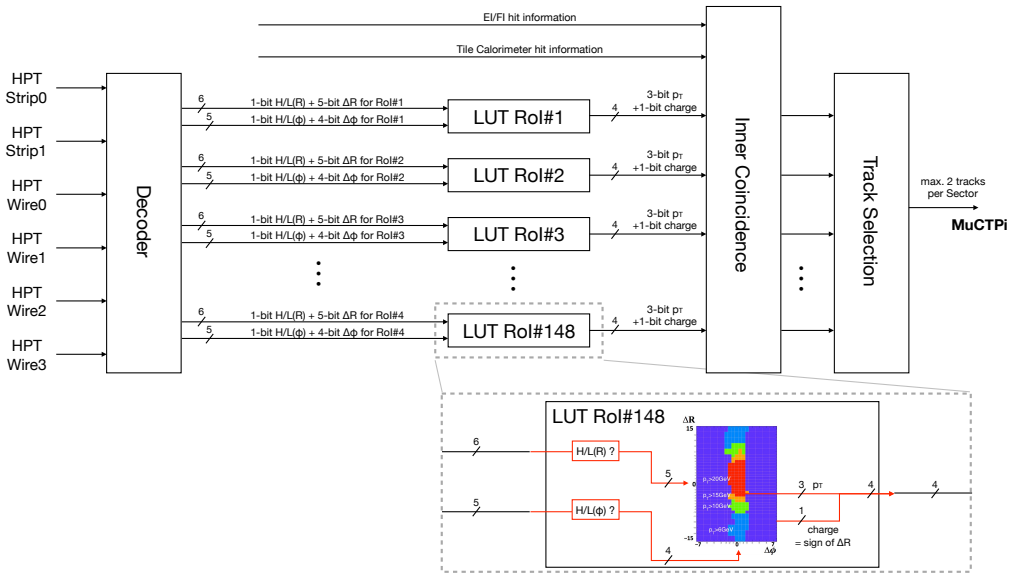


図 2.15　Endcap Sector Logic における 1 トリガーセクター分のトリガーロジックの実装方法の概要。また、1 つの RoI に対応する LUT の処理についても図の下部に示した。

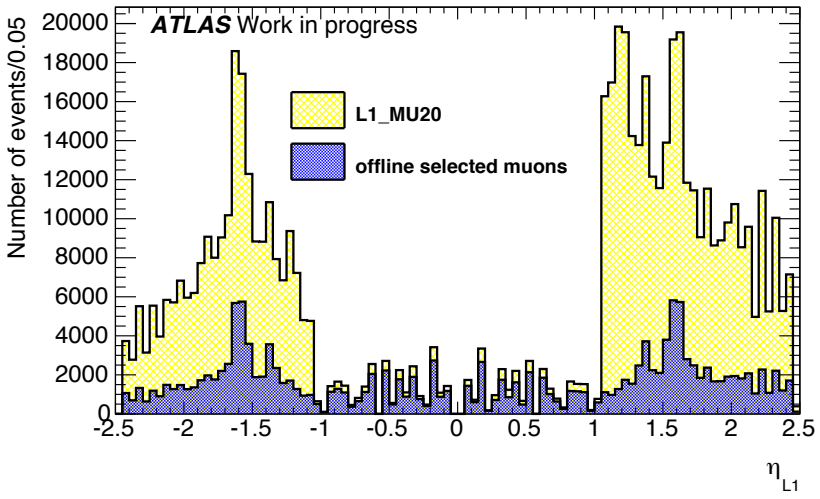


図 2.16　Run-1 における閾値 20 GeV 以上の L1 ミューオントリガー と、実際にオフライン解析で再構成されたミューオンの η 分布 [7]。

デンスも Run-2 中に導入される予定である。

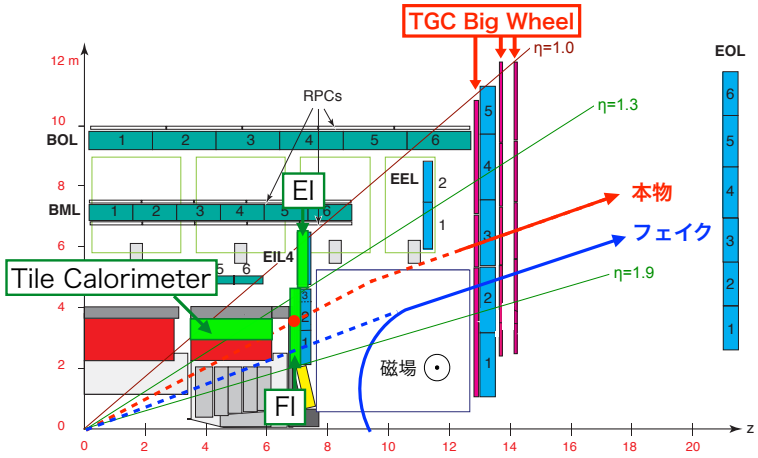


図 2.17　Inner Coincidence の概念図。$1.0 < |\eta| < 1.3$ では EI および Tile Calorimeter と、$1.3 < \eta < 1.9$ では FI とコインシデンスを取ることで、フェイクを排除することができる。

2.1.6　リードアウトライン

リードアウトラインはトリガー判定が行われるまでデータを保持し、トリガーを受け取った時のみデータを圧縮・整形し、読み出しボードに送信するデータ通信の流れである。図 2.18 にリードアウトラインの概要を示す。データは ASD → PP → SLB → SSW → ROD という順に受け渡され、最終的に Read Out System （ROS）に送信される。

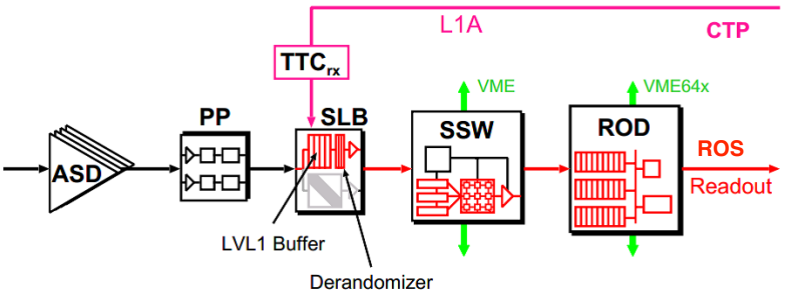


図 2.18　TGC エレクトロニクスのリードアウトラインの概念図 [9]。信号は SLB ASIC で一時的にバッファーに保存され、L1A（Level-1 Trigger）信号を受信した場合のみ後段に送信され、読み出される。SSW 及び ROD ではデータの圧縮・整形を行い、ROS に送信する。

以下、ASD と PP ASIC まではトリガーラインと共通であるので説明を省略する。また、SL ボード上にも SLB ASIC が設置されており、受信した HPT ボードのトリガー判定結果、および SL のトリガー判定結果を読み出し

ている。SLB ASIC 以降のリードアウトの流れは図 2.18 に示したものと全く同様であるので、図 2.18 では省略している。

SLB (リードアウトライン)

SLB ASIC のリードアウト部分は、主にトリガーが発行されるまでデータを保持するバッファーの機能を有する。これを Level-1 Buffer と呼ぶ。Level-1 Buffer は L1 トリガーが発行されるまで最大 128 BC 分の情報を保持することができる。L1 トリガーが発行された場合、そのイベントおよび前後 1 BC の情報を読み出し、読み出しラインの同期に必要なイベント ID 情報を付加した後、シリアル化の処理を行い後段の SSW に送信する。SSW へは 4 対の LVDS 信号線を用いてシリアル通信でデータが送信される。

Star Switch (SSW)

SSW では複数の SLB ASIC からの情報を受信し、一度シリアル信号をパラレル信号に変換する。その後、受信したデータを圧縮した後に、複数の SLB ASIC の情報をまとめて、G-Link で ROD に送信する。

Readout Driver (ROD)

ROD は複数の SSW からの光信号を受信し、まとめて後段の ROS に送るモジュールである。ここでは同じ BC の情報をまとめて、ID 情報およびヘッダー・フッターを付加する （Event Building)。Event Building の後、データを整形してから S-Link と呼ばれる通信規格で ROS（Read Out System）に送信する。

2.2　ミューオントリガーのアップグレード

1.5.1 節で説明した通り、Phase-1 アップグレードでは特にトリガーシステムの改良が重要である。この節ではミューオントリガーアップグレードの物理モチベーションについて詳しく言及したのちに、その基本的な方針について説明する。

2.2.1　物理モチベーション

1.5.1 節で示したように、物理アクセプタンスを維持するためにミューオントリガーの果たす役割は大きい。ここでは、Higgs 粒子の精密測定におけるミューオントリガー改良の必要性について述べる。

Higgs 粒子の結合定数の精密測定においては、様々な生成・崩壊モードを網羅的に調べて、標準理論からのズレ及び新物理への示唆を得ることが目的である。数ある生成・崩壊過程の内、レプトンを終状態に含む過程は QCD ジェットによる背景事象から区別しやすいため系統誤差を抑えた精密な測定ができる重要なチャンネルとなる。以下では、ミューオントリガーが特に重要な役割を果たすチャンネルの例として、$WH \to \mu\nu bb$ チェンネルを例に説明する。

WH 生成過程は、W の崩壊によるレプトンを用いてトリガーをかけることで、Higgs の崩壊過程によらずデータ取得ができる点が強みである。Higgs の崩壊した先の粒子によるトリガーがかけづらいチャンネルにおいては、この生成過程を用いることで感度を大きく上げることができる。特に、Higgs の bb への崩壊は、他の生成モードでは QCD による背景事象と区別することが難しく、系統誤差が大きくなってしまう。Higgs と quark との結合定数の精密な測定には $WH \to \mu\nu b\bar{b}$ チャンネルが重要になる。

図 2.19 に、WH 生成過程によって生じた W が崩壊したミューオンの p_{T} 分布を示す。シングルミューオントリガーの閾値を現在の Level-1 トリガーと同じ 20 GeV に保った場合は信号収集効率は 93% であるのに対し、仮に閾値が 40 GeV まで上がった場合は収集効率が 61% になることがわかる。p_{T} 閾値 40 GeV とは、ミューオントリガーのアップグレードを行わなかった場合に Run-3 でレートを保つために設定しなければならなくなる p_{T} 閾値である。$WH \to \mu\nu bb$ チャンネルのアクセプタンスを落とさないためには、ミューオントリガーの改良が必須である。

ここでは詳細は述べないが、この他にも、$H \to WW^* \to \ell\bar{\ell}\nu\nu$ や、$H \to \tau^+\tau^-$ のチャンネルにおいて低いレプトントリガー閾値を保つことは重要である [3]。

2.2.2　Upgrade Strategy

Run-3 におけるトリガーメニューの一例を表 2.2 の 3 列目に示す。ルミノシティ 3 $\times 10^{34}$ cm^{-2}s^{-1} において、シングルレプトントリガーの最も閾値が低いトリガーである L1MU20 で、トリガーレートを 15 kHz に抑えることが要求される。仮に Run-2 のトリガーシステム のままこのルミノシティに達したとする場合のトリガーメニューが 2 列目に示してあり、この場合は L1MU20 に 28 kHz のトリガーレートを割くことになることがわかる。このとき他のトリガーは大きく帯域を失うこととなり、例えば Topological Trigger には 5 kHz しかレートを割り当てることができず、低い p_{T} のオブジェクトを用いる様々な物理に対するアクセプタンスを失う。他のトリガーに帯

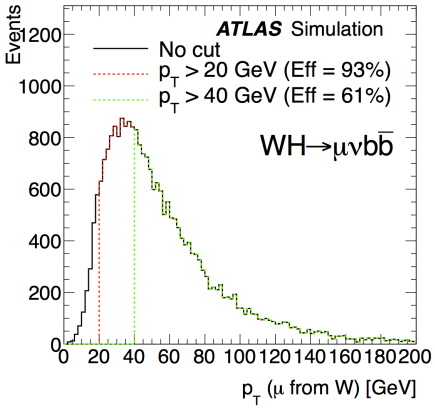


図 2.19　WH による生成において W → $\mu\nu$ に崩壊した場合のミューオンの p_{T} の分布 [4]。p_{T} 閾値が 20 GeV の場合、信号収集効率は 93% であるのに対し、40 GeV の場合は 61% ほどになってしまう。

域を残すために、ミューオントリガーの閾値を上げることでトリガーレートを下げる場合は、これもまた大きく物理アクセプタンスを失うことになる。

Run-2 において、ミューオントリガーの帯域のほとんどをフェイクトリガーが占めていることが分かっている。これを落とすトリガーロジックを実現できれば、p_{T} 閾値を現在と同じ 20 GeV に維持したまま、トリガーレートを抑えることが可能となる。Phase-1 Upgrade での基本戦略は図 2.20 に示したように、Run-3 で新たに導入される検出器 New Small Wheel とのコインシデンスを取ることで、この目標を達成することである。

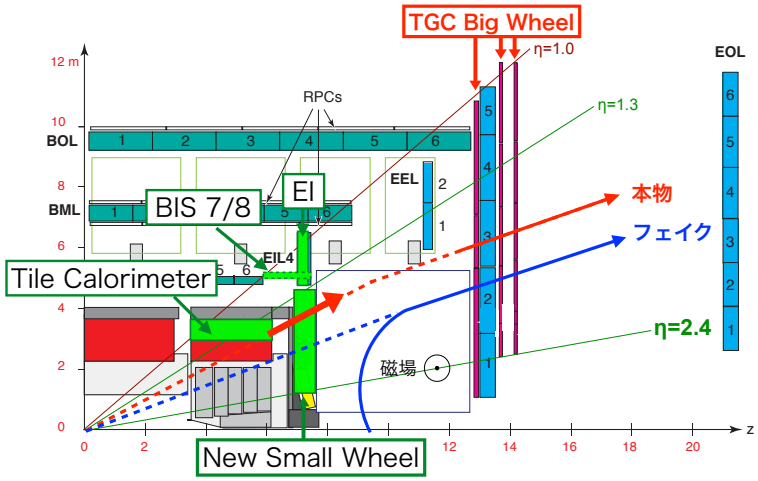


図 2.20　Run-3 における Inner Coincidence の概念図。FI が NSW で置き換えられている。1.0 < $|\eta|$ < 1.3 では EI および Tile Calorimeter に加え BIS 7/8 と、1.3 < η < 2.4 では New Small Wheel とのコインシデンスを取る。

図 2.21 に、ミューオントリガーが発行された RoI の η 分布を示す。ここでは、Run-1 の 14.0pb^{-1} のデータを

Run 1			Run 2			Run 3		
	Offline p_{T} Threshold [GeV]	Rate [kHz]		Offline p_{T} Threshold [GeV]	Rate [kHz]		Offline p_{T} Threshold [GeV]	Rate [kHz]
EM18VH	25	130	EM30VHI	38	14	EM25VHR	32	14
EM30	37	61	EM80	100	2.5	EM80	100	2.5
2EM10	2x17	168	2EM15VHI	2x22	2.9	2EM12VHR	2x19	5.0
EM total		**270**			**18**			**20**
MU15	25	150	MU20	25	28	MU20	25	15
2MU10	2x12	14	2MU11	2x12	4.0	2MU11	2x12	4.0
Muon total		**164**			**32**			**19**
EM10VH_MU6	17,6	22	EM15VH_MU10	22,12	3.0	EM10VHR_MU10	17,12	3.0
			EM10H_2MU6	17,2x6	2.5	EM10HR_2MU6	17,2x6	1.0
TAU40	100	52	TAU80V	180	4.7	TAU80VR	180	3.2
			2TAU50V	2x110	3.8	2TAU40VR	2x100	3.9
2TAU11I_TAU15	30,40	147	2TAU20VI_3J20	2x50,60	5.2	2TAU15VR_3J15	2x40,50	8.1
2TAU11I_EM14VH	30,21	60	2TAU20VI_ EM18VHI_3J18	50,25,60	2.8	2TAU15VR_ EM13HR_3J13	40,20,50	3.3
			TAU15VI_MU15	40,20	3.8	TAU11VR_MU11	35,12	6.4
TAU15_XE35	40,80	63	TAU20VI_ XE40_3J20	50,90,60	4.4	TAU15VR_ XE40_3J15	40,90,50	5.0
Tau total		**238**			**20**			**25**
J75	200	34	J100	200	7.0	J100	200	7.0
4J15	4x55	87	4J25	4x60	3.3	4J25	4x60	3.3
			J75_XE40	150,150	8.3	J75_XE40	150,150	8.3
XE40	120	157	XE90	250	10	XE70	200	13
Jet/$E_{\mathrm{T}}^{\mathrm{miss}}$ total[a]		**306**			**25**			**25**
Topological triggers		**-**			**∼5**			**∼20**
Total		**∼800**			**∼100**			**∼100**

表 2.2　Run-3 でのルミノシティ3.0 $\times 10^{34}$ cm^{-2}s^{-1} における L1 トリガーメニューの一例 [3]。1 列目は仮に Run-1 の終盤のトリガーメニューをそのまま適用すると仮定した場合のトリガーレートを示している。2 列目は Run-2 終了時点でのトリガーシステムのまま Run-3 を迎えた場合に想定されるトリガーメニューである。3 列目は Run-3 のトリガーシステムにおいて予想されるトリガーメニューの一例である。

用いている。ヒストグラムの外枠が Run-1 における分布で、白の部分が Run-2 におけるコインシデンス（EI/FI および Tile Calorimeter）で排除される部分である。EI/FI および Tile Calorimeter の Inner Coincidence を要求する範囲は $1.0 < |\eta| < 1.9$ であり、コインシデンスを取る事によりこの範囲では Run-2 でも大きくトリガーレートが抑える事ができている事がわかる。

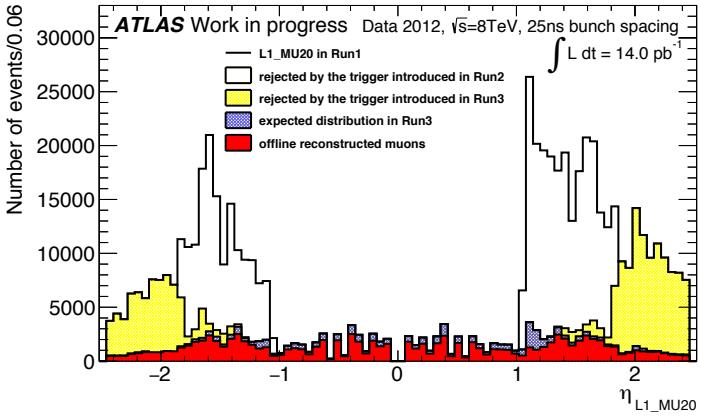


図 2.21　閾値 20 GeV のミューオントリガーを発行した RoI の η 分布。EI/FI 及び Tile Calorimeter が存在する $1.0 < |\eta| < 1.9$ ではトリガーレートが大きく削減されている。一方、$1.9 < |\eta|$ では L1 でコインシデンスを取ることができる検出器が無いため、トリガーレートが他の領域と比べて高い。

$1.9 < |\eta|$ では Run-2 ではコインシデンスを取る事ができないため、トリガーレートが高いままである。New Small Wheel は $|\eta| < 2.4$ までカバーするので、コインシデンスを取る事でこの領域のトリガーレートも大きく削減できると見込まれる。第 5 章で詳しく述べるが、BW と NSW の η・ϕ の差、及び NSW で得られた角度情報 $d\theta$ を用いてトリガー効率を落とさないような大まかなカットをかけるだけで、大きくトリガーレートを削減できることが分かっている [3]（図中の黄色部分）。この NSW コインシデンスを用いた場合に予想されるトリガーレートを図 2.22 に示す。p_{T} 閾値 20 GeV において、Run-2 トリガーシステムのままでは 30 kHz 以上になるのに対し、Phase-1 Upgrade 後では 13 kHz にまで抑えられる事が分かった。これは Run-3 に対する要求を満たしている。

BW と NSW とのコインシデンスロジックを導入した場合の、Run-3 におけるミューオントリガーシステムの概念図を図 2.23 に示す。現行の TGC のエレクトロニクスは共通であるので、 2.1 節で説明した通り HPT による ΔR、$\Delta\phi$ の計算が行われる。New Small Wheel に関しては、検出器近くに設置されたフロントエンド回路で飛跡の位置及び角度情報が計算される。Run-3 ではこれらの情報を組み合わせたトリガー判定を行うために NSW からの高速通信で送信される情報を受信するためのインターフェイスが必要となる。上記ロジックを実装しトリガーレートを抑えるためには、この 2 つの情報を組み合わせて最終的なトリガー判定をする、現行の Sector Logic に代わる新たなトリガー判定基盤を作成する必要がある。

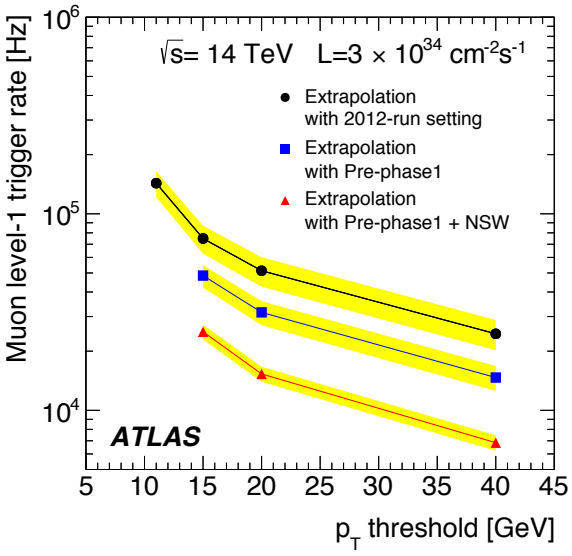


図 2.22　Run-3 における Upgrade 前後のトリガーレート。黒い点は Run-1 終了直前のトリガーシステムの場合のトリガーレートを示す。青い四角の点は Run-2 終了時点（Phase-1 Upgrade 直前）の場合の、赤い三角の点は NSW のコインシデンスを導入した場合のトリガーレートである。

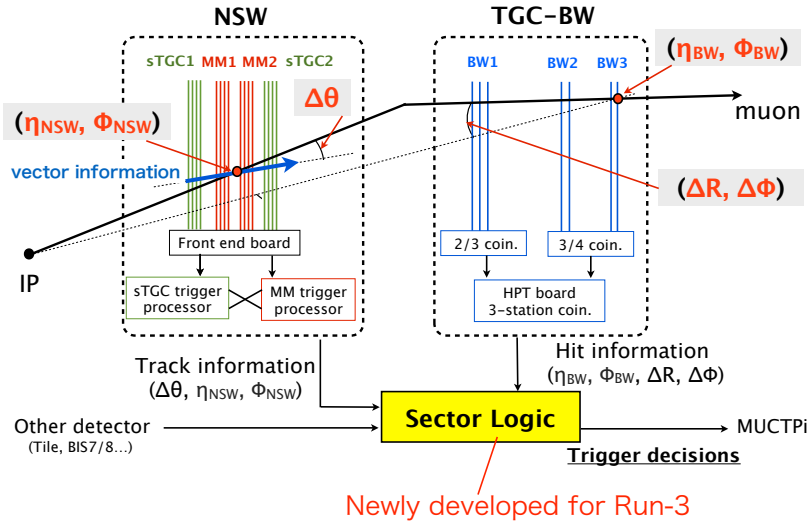


図 2.23　Phase-1 Upgrade 後のエンドキャップミューオントリガーの概要。BW と NSW の情報を合わせてトリガー判定を行う、新たな Sector Logic ボードの開発が必要である。

2.2.3　読み出しラインの改良

現在、SL のデータの読み出しは SL ボード上に設置された SLB ASIC を用いて行っている。SLB ASIC は SSW へ 4 対の LVDS シリアル通信でデータを送信し、SSW は G-Link で ROD へ、ROD は S-Link で ROS へと順にデータを送信している。SSW や ROD は専用のハードウェアで実装されているため、機能のアップグレードが難しい。特に、NSW の情報も含む大きなデータサイズを扱うには機能を拡張する必要があるが、そのためには専用のボードを新たに設計しなければいけなくなる。そこで、Phase-1 Upgrade では今後の拡張性も考慮して図 2.24 のようなアップグレードを行う。まず、SLB ASIC の行っていたデータのバッファリングを新 SL の FPGA 上で実装することで、SLB ASIC を省くことができる。次に、SSW の行っていたデータ圧縮も FPGA 上で実装することにして、さらにデータを Ethernet で送ることができれば、SSW を市販のネットワークスイッチ変更することができる。加えて、これまで専用ハードウェアで実装されていた ROD を、PC 上にソフトウェアとして実装することにする。これにより、PC の入れ替えまたはソフトウェアの更新で、容易に機能をアップグレードすることができる。

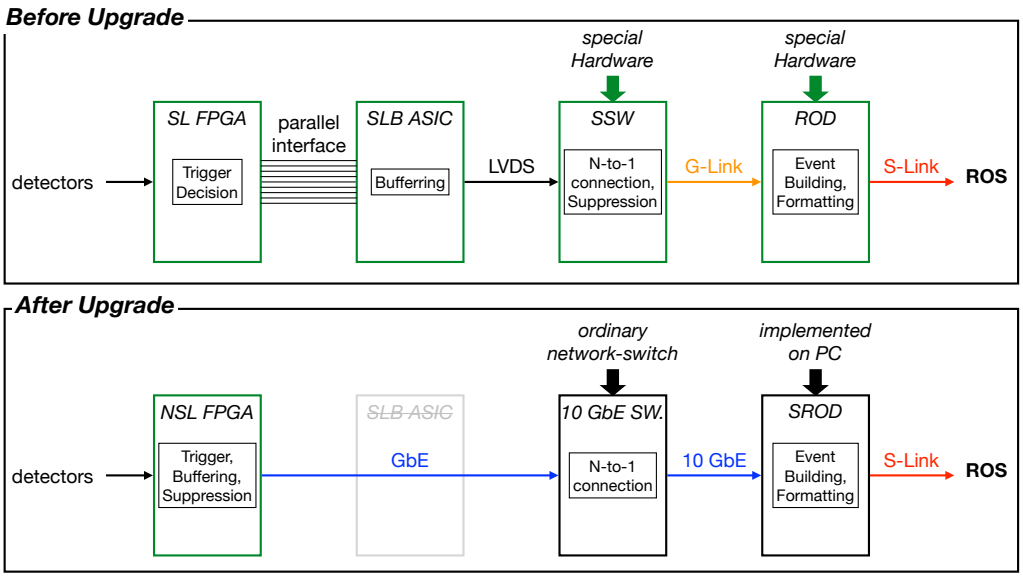


図 2.24　Phase-1 Upgrade 前後のトリガー情報読み出しライン。Upgrade 前では Sector Logic の情報の読み出しに SLB ASIC を含め 3 つの専用ボードを経由している。Upgrade 後では、読み出し機能を PC 上の ROD（SROD）に実装することで、専用ボードを省くことができる。

第3章　新トリガー判定ボードNew Sector Logicの開発

ATLAS実験Level-1ミューオントリガーの改良のために、新しいトリガー判定ボードNew Secor Logic（NewSL）の開発を行った。本章ではまず3.1節で、New Small WheelからNewSLが受け取るデータの内容を確認した後、3.2節でNewSLに要求される性能について説明する。3.3節ではそれらの要求を満たすように開発したNewSLボードのデザインを説明する。3.4節に、新たに導入された2つの新技術「GTX高速トランシーバ」と「SiTCP」について言及した後、3.5節でボードのテストとその結果を報告する。

3.1　New Small Wheelからの受信データ

前章で述べた通り、NewSLの主な役割はNew Small Wheelからの信号とBig Wheelからの信号を合わせたトリガー判定を行うことである。NewSLのデザインの説明に入る前に、New Small Wheelからの受信データの内容を確認しておく。

1枚のNewSLボードが扱う範囲は、現在のSector Logic Boardと同じくTGCの2トリガーセクター分である。TGCのトリガーセクターは図3.1の緑の線で分けられた範囲で、全方位角を48等分（Forwardトリガーセクターでは24等分）にした範囲である。NSW側のセクターの分け方はTGCとは異なり、Smallセクターと Largeセクターの2種類が存在する。NSWセクターとTGCトリガーセクターとの関係は、図3.1のようになっている。TGCではトリガーセクター間のヒット情報の共有は行わないため、粒子が通りうるすべてのNSWセクターからの情報を受け取る必要がある。図3.1に示すように、NewSLは最大3つのNSWトリガーセクターから情報を受け取る。

表3.1　NSWにおける1トラック分の情報。1トラックあたり24-bitでヒット情報がエンコードされ、NewSLに送られる。

情報	sTGC type	MM type	$\Delta\theta$	ϕ position	η position	spare	合計
ビット幅	2	2	5	6	8	1	24

NSWからの1トラック分の情報を表3.1に示す。

sTGC typeおよびMM typeは、それぞれsTGCとMMにおけるコインシデンス情報である。sTGCおよびMM 8層のうち、コインシデンスが取れた層が多いほど質の高いトラックであるとして、2-bitにエンコードされて送信される。

$\Delta\theta$は図3.2のように定義される。衝突点の中心からまっすぐに飛来した粒子は$\Delta\theta = 0$となるが、実際は衝突

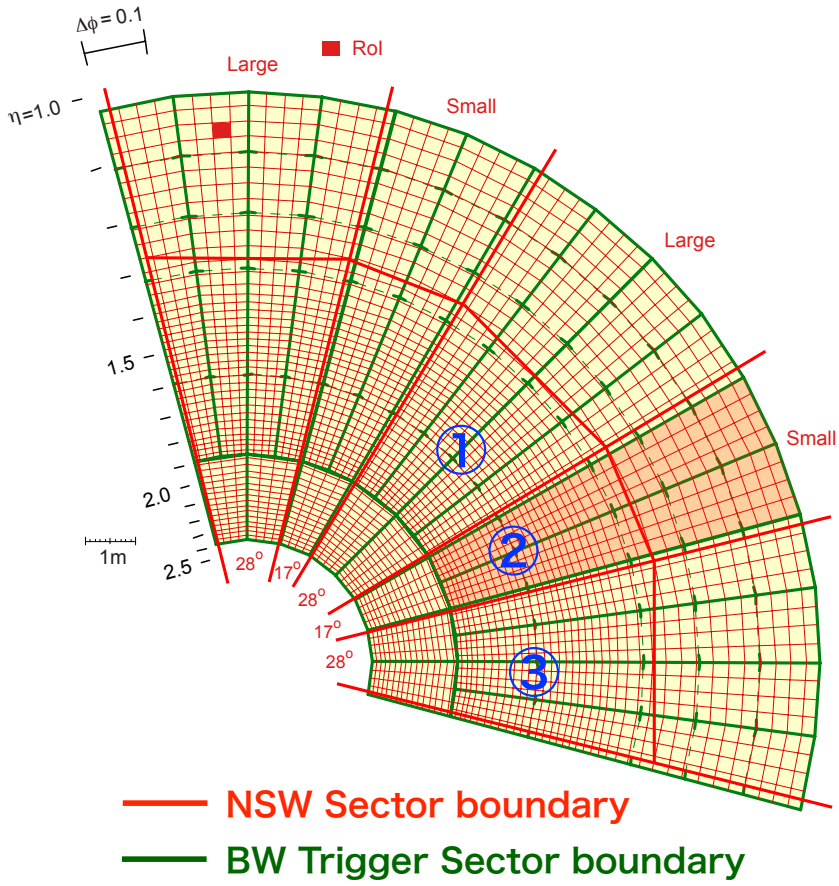


図 3.1　TGC トリガーセクター と NSW セクターとの関係。1 つの NewSL Board は 2 つの TGC トリガーセクターを担当する。赤く塗られた TGC の 2 トリガーセクターは、磁場による曲がりも考慮してすぐ隣の NSW セクターの飛跡情報も利用する必要があるため、この NewSL は 3 つの NSW セクターから信号を受信する。

点が中心から $\sigma \sim 5$ cm ほどの広がりを持つことと、NSW の位置に飛来するまでにカロリーメータ等の物質との相互作用による多重散乱をすることに起因して、$\Delta\theta$ は 0 を中心とする分布を持つ。図 3.3 に $\Delta\theta$ の分布を示した。このように、衝突点由来のミューオンは 99%以上、$|\Delta\theta| < 15$mrad の中に収まる。NSW における $\Delta\theta$ の分解能は約 1 mrad なので、$\Delta\theta$ を -15 mrad $\sim +15$ mrad を 1 mrad 刻みで 31 段階、5-bit で表現して NewSL へ送信する。

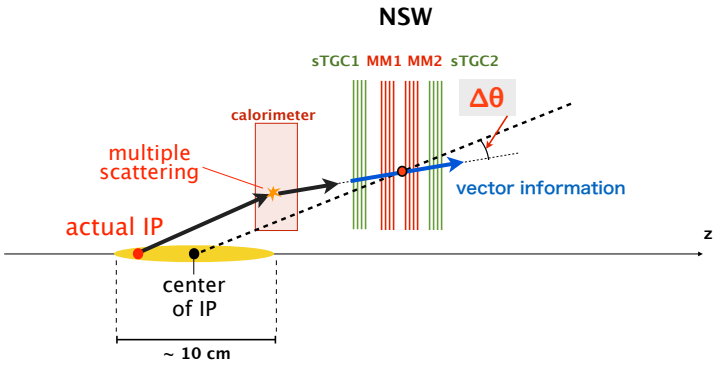


図 3.2　$\Delta\theta$ の定義。衝突点の中心と、ヒット位置を結んだ直線から計算した θ と、実際のトラックの θ の差で定義される。

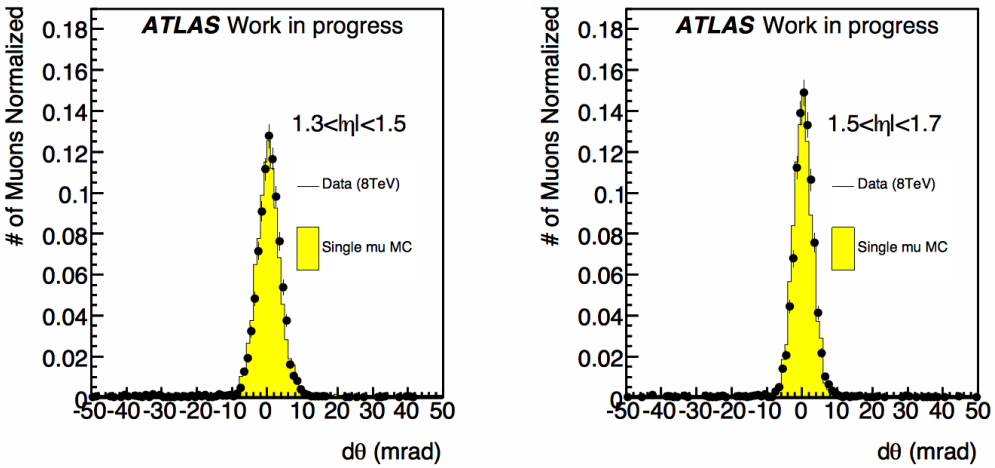


図 3.3　$\sqrt{s} = 8$TeV のデータと MC から得られた $\Delta\theta$ 分布。左は $1.3 < |\eta| < 1.5$、右は $1.5 < |\eta| < 1.7$ のもので、ほとんどのエントリーが $|\eta| < 15$mrad 以内にあることがわかる。

ϕ/η position は、それぞれ ϕ 方向・η 方向の位置を表していて、それぞれ 5-bit/ 6-bit で表現される。これは、NSW で L1 段階での分解能の目標が ϕ 方向で 10 mrad、η 方向で 0.005 であることから定めた。NSW 1 枚が担当する ϕ の範囲は Large Sector で $2\pi/12$ [rad] であり、10 mrad の分解能でこの範囲を分けると 2π [rad]/12/10 [mrad] = 52.3 分割できる。これを 2 進数で表現するために、6-bit （64 段階）にエンコードして NewSL に送ることとなる。η 方向に関しても同じ議論で、NSW の担当する $\eta = 1.3 \sim 2.4$ を分解能の 0.005 で分けると 220 分割となる。従っ

て、NewSL への送信には 8-bit（256 段階）必要である。

最後に、1 枚の NewSL ボードが NSW から受け取るトラック情報の数を考える。NewSL はトリガー判定を通過したトラックの情報を 1 トリガーセクター当たり最大 3 候補に絞り後段の MuCTPi に送信する。1 枚の NewSL は 2 トリガーセクターを担当するため、最大 6 トラックを選ぶ必要がある。NSW からは少なくともこの 6 トラックよりも多い候補を受け取る必要があるが、NSW におけるノイズ等による誤ったトラックを送ってしまう可能性を考慮すると、トラック候補を多く送っておく方が望ましい。現在のところ、NSW 1 トリガーセクターから 8 トラック分の情報が NewSL に送られることになっている。

以上の議論により、NewSL は NSW から 1 BC 当たり 24[bit/track] × 8 [track] × 3 [NSW triger sector] = 576-bit を受信することになる。これは、イベント ID 情報等を含んでいない、純粋にトラック情報部分のみのデータ量である。この時の通信速度を見積もると、576-bit /BC (= 25 ns) ∼ 約 23 Gbps となる。

3.2　New Sector Logic に対する要求

Phase-1 Upgrade 後の新しいトリガー判定ボード、New Sector Logic に求められる性能を以下にまとめる。図 3.4 にその概念図を示す。

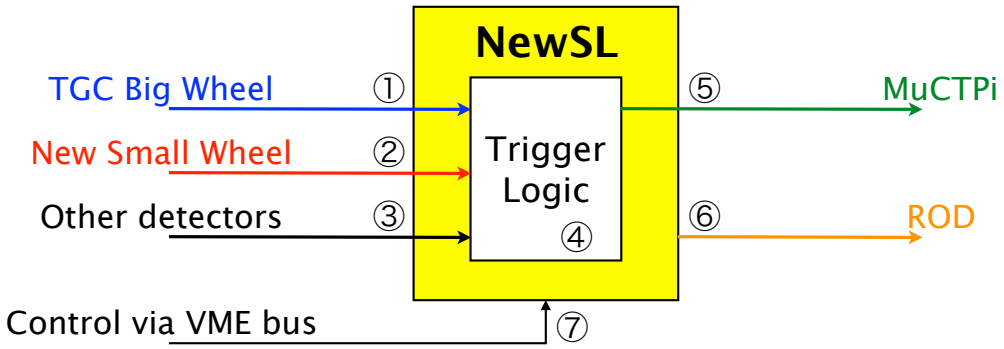


図 3.4　NewSL に求められる主な役割の概念図。3 種類の入力及び 2 種類の出力、トリガーロジックの実装とコントロール信号が必要である。

① TGC Big Wheel の信号の受信

TGC Big Wheel 検出器及びフロントエンド回路は、Phase-1 Upgrade での変更予定はない。従って、現在と同じプロトコルで HPT ボードから TGC BW の情報を受け取る入力方法を実装する必要がある。

② New Small Wheel の信号の受信

新たなコインシデンスを導入するため、New Small Wheel からの情報を受けられる入力方法を実装する必要がある。特に、L1 トリガーは Fixed Latency システムであるので、データ受信の際には常に Latency が一定となるようなデザインとすることが求められる。さらに、NSW からの信号は上で述べたように最低でも 576-bit /BC であるため、高い転送速度の通信の確立が必要である。

③ その他の検出器の信号の受信

NSW の無い $1.05 < |\eta| < 1.3$ の部分では、現状使われている EI チェンバーに加え Tile Calorimeter 及び BIS 7/8 と呼ばれる RPC チェンバーからの情報を用いたコインシデンスを行う。それらの信号を受信するための入力方法を用意しておく必要がある。

④ トリガー判定ロジックの実装

NewSL は各検出器から受信した情報を用いてトリガー判定を行うボードであるが、このトリガーロジックは、入力情報が多ければ多いほど複雑なものとなるため、それを実装する論理回路の規模も大きくなる。Run-3 におけるトリガー判定をハードウェアで実装するためには大規模な回路を実装する必要がある。検出器からの受信情報を用いた大規模なトリガーロジックが実装できるチップを用いる必要がある。

⑤ トリガー判定結果の送信

トリガー判定した結果を、MuCTPi ボードへと送信する通信方法を実装する必要がある。この通信も Latency を一定にするようなデザインが求められる。

⑥ 入力信号の読み出し

2.2.3 節で説明した通り、NewSL ボードは受信したデータを保持・圧縮し、後段の SROD へ Ethernet で送信する機能を有する必要がある。この機能を果たすために、Ethernet Cable による I/O インターフェイスを搭載していることが要求される。

⑦ VME interface による制御

現行の ATLAS のミューオンシステムは VME クレートを用いてコントロールされており、Phase-1 Upgrade 後でも VME による制御は継続する予定である。従って、NewSL も VME バスを用いてコントロールできるインターフェイスを持つことが望ましい。

3.3　New Sector Logic のデザイン

Phase-1 Upgrade に向け、New Sector Logic ボードを新しく開発した。New Sector Logic の第 1 プロトタイプボードは 2015 年 10 月に完成し、簡単な I/O 試験による性能評価を行った。プロトタイプボードで発見されたいくつかの項目に修正を加えた第 2 プロトタイプボード（NewSL ver.2、以下単に NewSL と呼ぶ）5 枚が、2016 年 3 月に完成した。この節では、開発した NewSL ver.2 のハードウェアデザインについて説明する。また、節 3.3.4 では、ver.1 から ver.2 における主な変更点について説明する。

3.3.1　ボードデザインの概要

図 3.5 の左図に開発した NewSL ボードの写真を、右図に主なチップと配線の概略図を示す。メインのトリガーロジックは FPGA 上に実装する。要求④を満たすように大規模ロジックが実装可能なものを選択した。入出力ポート（I/O ポート）としては、要求① を満たすように G-Link 入力ポートを搭載した。さらに、要求②を満たすために、FPGA に搭載されている高速トランシーバ GTX を用いた I/O を実装した。ここで要求③ のために、GTX I/O を NSW の受信に用いる 6 本よりも余分に設置した。要求⑤の MuCTPi への情報送信には、GTX I/O

の送信ポートを用いることとした。要求⑥の SROD への情報送信には、TCP/IP を用いた通信を採用し、PHY チップを搭載した。最後に、要求⑦を満たすために、VME 9U 規格でボードをデザインした。

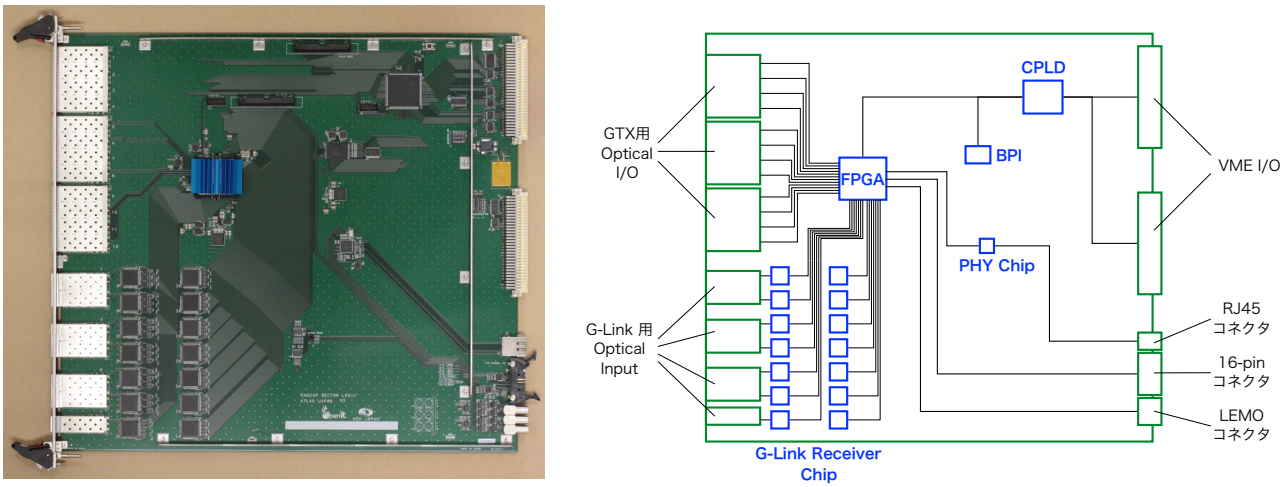


図 3.5　(左) NewSL ver.2 の写真。(右) 主なチップと配線の概略図。緑色のブロックで I/O ポートを、青色のブロックで主なチップを示している。

3.3.2　主な IC チップ

FPGA　トリガー判定及び読み出しを行うためのチップとして、Xilinx 社の Kintex-7 Series FPGA、XC7K325T を採用した。XC7K325T は高速トランシーバー GTX を搭載しており、1 本の信号線で 10 Gbps 以上の速度での通信が可能である。表 3.2 に、同じランクの FPGA 性能をまとめた。ロジックセルとは小規模な演算処理（和・積、数 bit のビット演算等）に用いられるブロックの単位で、一般的な論理回路の規模に対応する。BRAM は Block RAM の略で、FPGA 上に搭載された大容量のメモリであり、大量のデータの保存もしくは大規模な LUT の実装に用いられる。GTX は、Xilinx 社 Kintex-7 Series FPGA に搭載された高速トランシーバの名称である。ユーザー I/O pin は、ユーザーが自由に配線できる FPGA と外部チップとの接続用 I/O である。

NewSL FPGA に要求される I/O pin の数は非常に多く、G-Link にはコントロール信号を含め 21 pin × 14 レーン分、VME バスのアドレスおよびデータで 40 pin、PHY チップとの通信線で約 35 pin、その他約 30 pin を合わせて、400 pin 以上必要である。そのため、500 pin の I/O を持つパッケージを選択した。500 pin の I/O を持つ Kintex-7 Series のパッケージは XC7K325T と XC7K410T の 2 種類ある。GTX トランシーバの数は共通の 16 個で、主な違いはロジックセルおよび BRAM の個数である。GTX の個数については、後述するが 16 レーンあれば十分である。リソースについては、BRAM の個数が重要である。これは、現行の ΔR - $\Delta\phi$ コインシデンスのような大規模な LUT を FPGA 上で実装する際には大容量メモリである BRAM を用いる必要があるからである。また、NewSL の FPGA は読み出し回路も兼ねるため、データを保存するバッファーも持つ必要がある。このよ

表 3.2　Kintex-7 Series の主な仕様

デバイス	ロジックセル	BRAM 個数 (36 Kb)	GTX	ユーザー I/O pin 数
XC7K70T	65,600	135	8	300
XC7K160T	162,240	325	8	400
XC7K325T	326,080	445	16	500
XC7K355T	356,160	715	24	300
XC7K410T	406,720	795	16	500
XC7K420T	416,960	835	32	400
XC7K480T	477,760	955	32	400

うな大きなバッファーにもやはり BRAM リソースを用いる。一方でロジックセルを利用するような小規模ロジックを多数実装することはないため、ロジックセルの数は重視する必要はない。実際、現行の SL において BRAM リソースは特に RoI の数の多い Endcap SL では 96% 以上（Forward では約 80%）を使用しているのに対し、ロジックセルのリソースは 30%∼40% しか使用されていない。

今回、ボードを作成する段階では NSW とのコインシデンスにどれくらいの BRAM リソースが必要か試算されていなかった。安価なパッケージである XC7K325T でも現行の SL の約 10 倍の BRAM リソースを有しているため、NSW コインシデンスを実装するのに十分であると判断し、今回は XC7K325T を採用した。

CPLD　VME バスを用いた制御（以下 VME 通信と呼ぶ）のために、Xilinx 社製の CPLD、CoolRunner-II Family の XC2C256-7PQ208C を用いた。CPLD は不揮発性メモリを有しているため、電源投入後にデザインの書き込みをする必要がなく、即時に VME 通信を開始することができる。CPLD を用いることで、電源投入後の FPGA への回路デザインの書き込み制御、及びその後の FPGA への VME 通信を行うことができる。また、後述の BPI への制御も CPLD を介して行う。

BPI　FPGA の回路情報を格納しておくための、不揮発性のメモリである。Micron 社製 JS28F256P30T を設置した。このメモリに FPGA 回路デザインを保存しておき、CPLD からコントロールすることで、電源投入後に高速で回路情報を FPGA に書き込むことができる。

G-Link Receiver Chip　HPT Board からの情報は G-Link で受信する。NewSL では G-Link 受信チップとして現行の SL と同じく HDMP-1034A を搭載している。G-Link 受信チップは、シリアルで受信した信号を、ユーザーが供給したクロックに同期したパラレル信号に変換し出力するチップである。NewSL ではこの G-Link チップに 40 MHz の LHC クロックを供給することで、16-bit または 17-bit の TGC Big Wheel のデータを LHC クロックに同期したパラレル信号として受け取り、FPGA でのトリガーロジックに用いる。

Ethernet PHY　Ethernet での通信のために Ethernet PHY チップが必要である。NewSL では Microchip Technology 社製の LAN8810i-AKZE を搭載した。PHY チップと FPGA を適切に繋ぎ、後述の SiTCP 技術を用いることにより、TCP/IP でのデータ通信が可能である。

水晶発振器　NewSL ボード単体での運用ができるよう、水晶発振器 MXO3-7050C を搭載した。LHC のクロック 40 MHz (正確には約 40.079 MHz) に合わせるため、特別に周波数を 40.079 MHz にカスタムしたもの (Si5334C-B05812-GM) を用いた。このクロックは、仮に何らかの問題で外部からのクロックが停止した場合においても動かしたいロジック、即ち、主に TCP/IP 通信等 LHC との同期の必要のない部分のロジックの実装に用いる。また、NewSL 単体でのボードテストにも用いることができる。

Clock Jitter Cleaner　GTX の駆動に必要な基準クロックを生成するために、Clock Jitter Cleaner Si5334 を搭載した。Si5334 は入力したクロックの周波数を 4 倍にした後に、クロックジッターが少なくなるように波形を整形して送信する素子である。このチップも、LHC の 40 MHz のクロックを受け 160 MHz を出力するために特別にカスタムしたものを用いた。この Jitter Cleaner に入力するクロックのソースは水晶発振器からのクロック（Internal Clock）、16-pin コネクタで受信するクロック（TTC Clock）、LEMO で受信するクロック（External Clock）の 3 種類から、ジャンパーピンで選択することができる。

3.3.3　入力・出力ポート

ここでは、NewSL に実装された入出力（I/O）ポートについての解説を行う。

GTX 用 SFP+ モジュール　FPGA に搭載されている高速トランシーバ GTX による通信のインターフェイスとして、SFP+モジュールを搭載した。SFP+は、光信号を電気信号に、またはその逆の変換をしてデータを送受信するためのモジュールで、NewSL では Avago Technologies 社の AFBR-709SMZ というモデルを用いた。GTX I/O は入力・出力共に 12 本 搭載されており、受信側は最大 6 本を NSW の信号に用いる。残り 6 本を必要に応じてその他の検出器からの信号に用いることができる。送信側は 12 本のうち 2 本を MuCTPi への送信に用いる。

G-Link 用 SFP モジュール　G-Link 通信のための光変換モジュールとしては、W-Optics 社の SAB-1AC1-111 というモジュールを採用した。G-Link は受信ポートのみ 14 ポート搭載しており、そのうち 12 本（Forward では 6 本）を Run-2 と同じく Big Wheel の情報を受信するために用いる。Endcap 用の NewSL は、EI の情報を受け取るために残りの 2 本を用いる。

16-pin コネクタ　16-pin を用いて、LVDS レベルで LHC クロックおよびイベント ID 情報を受け取る。ID 情報の意味およびその使用方法については次章で解説する。

LEMO コネクタ　デバッグ用に設置した I/O。NIM レベルでの信号の入出力ができ、入力 2 本、出力 4 本が設置されている。入力のうち 1 本にはクロック信号を入力することができ、Clock Generator 等で生成した外部からのクロックでボード全体を駆動することができる。

RJ45 コネクタ　Ethernet を接続するポートとして RJ45 コネクタを設置した。

テストピン　試験用に 16-pin のテストピンを設置した。FPGA 内の信号を出力することで、内部ロジックのデバッグに用いることができる。

3.3.4　NewSL ver.1 から ver.2 での主な変更点

Clock Jitter Cleaner Si5334 の周波数カスタマイズ

作成した 2 枚の ver.1 ボードで GTX の試験を行った際に、片方のボードではうまくいき、もう片方のボードではうまくいかない、という事象が発生した。この原因を調べた結果、図 3.6 にあるように Si5334 チップから出力信号が出ていないことがわかった。

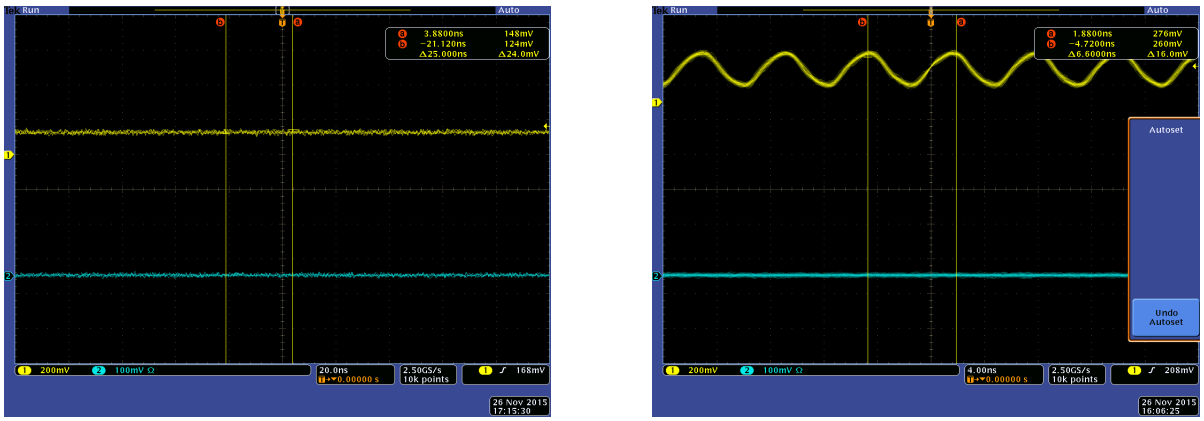


図 3.6　(左) GTX が機能しないボードの Si5334 チップの出力をオシロスコープで見たもの。クロック信号は出力されていないことがわかる。(右) GTX が正しく動くボードの Si5334 チップの出力をオシロスコープで見たもの。160 MHz のクロック信号が出力されている。

Si5334 はもともと、とある周波数にカスタマイズされたものが市販で売っており、当初 ver.1 ではその中で最も周波数の中央値が 40.079 MHz に近い「Si5334C-B00103-GM」を用いていた。このチップは周波数 38.88 MHz のクロックに最適化されたチップである。我々が使用したい 40.079 MHz クロックはこの市販のチップの周波数域のちょうど境界に対応すると考えられ、その結果、チップの個体差により動作するもの・しないものが出てしまっていた。

NewSL ver.2 ではオーダーメイドでカスタマイズした Si5334C-B05812-GM という型番のチップを用いることにした。Si5334C-B05812-GM は 40.0786 MHz に最適化されており、これは十分に我々の用いる 40.079 MHz に近い。新しいチップに交換後テストを行った結果、後で述べるように全てのボードで GTX は正しく動作することが確認できた。Jitter Cleaner の対応周波数領域は思っていたよりも狭いことがあるため、今後の回路開発においては特に注意すべき点である。

PHY チップからの受信クロックの配線

一般的に FPGA に入力するクロック信号は、専用のピンに配線する必要がある。これは、同期回路においてクロックの役割は特別で、そのため FPGA 上で特殊な配線をする必要があるためである。クロック用のピンから入力された信号は、クロックバッファーと呼ばれる回路を経由して位相の調整を行うことができる。

PHY チップと FPGA を接続する信号線には、PHY - FPGA 間のデータ送受信に用いるためのクロック線がある。このクロック線も、FPGA のクロック用のピンに入力する必要がある。PHY チップからのクロック信号はシングルエンド信号で出力されるが、FPGA 上のクロック用のピンは作動信号でクロックを受け取る仕様になって

いる。NewSL ver.1 では、「N」側にシングルエンド・クロック信号を接続し、作動ピンの「P」側をグラウンドに接続していた。しかし、テストの際に用いる Firmware（Firmware については次章で詳しく説明する）を作成する段階で、クロック配線をうまく行えないということがわかった。調査の結果、クロック信号を「N」側に入力していたことがこの問題の原因であり、正しくは「P」側に入力しなければいけなかったことがわかった。これは、FPGA 上は P 側の入力信号と N 側の入力信号を対称に扱わず、P 側をマスター、N 側をスレーブとして別々に取り扱っていることに起因する。こうすることで、マスター側の入力をシングルエンド・クロックの入力に使用する際、スレーブ側をグラウンドに繋がず、一般の I/O として利用できる。ver.2 ではこの配線を修正し、クロックの配線を正しく行うことができるようになった。このように、FPGA の仕様から、ユーザーの想像しないような制約がかかることがあるので、ボードを実際に作成する前に簡単な Firmware をあらかじめ作成して、FPGA 内での配線が正しく行えることを確かめておくことが推奨される。

3.4　NewSL における新技術

3.4.1　GTX 通信

GTX トランシーバは Xilinx 社 Kintex-7 FPGA に搭載されている高速トランシーバである。GTX を用いた通信では、1 本のシリアル通信で最大 12.5 Gbps の通信速度を実現することができる [11]。ここでは、下で説明する 8B/10B 変換を用いる場合のトランスミッター（TX）およびレシーバー（RX）における主な信号処理について説明する。

GTX TX ブロック

図 3.7 に GTX TX のブロック図を示した。GTX TX では FPGA 上のロジックで作成されたパラレルデータをシリアルデータに変換し 1 対の差動信号で送信する。

- FPGA TX Interface
  ユーザーは送信したいデータをパラレル信号として、TX Interface に受け渡す。8B/10B 変換を用いる場合、選択できるパラレルデータの幅は 16-bit、32-bit、または 64-bit である。このパラレルデータを、ユーザーは TXUSRCLK2 と呼ばれるクロックに同期して送信する。TXUSRCLK2 は GTX TX が基準クロックから生成するクロックであり、転送レートおよびパラレルデータ幅により適切な周波数に分周されている。

- 8B/10B Encoder
  8B/10B 変換は IBM 社が開発したシリアル通信の際に用いられるエンコード・デコードの方式である。8B/10B Encoder では、入力されたパラレル信号を 8-bit 単位に分解し、あらかじめ定められたパターンに従って 10-bit のデータに変換する。8B/10B により受信側ではデータからのクロックの復元が可能となる。また、「Comma」と呼ばれる特別なパターンを TX 側が送ることにより、シリアル信号をパラレル信号に変換する際の境界の情報を得ることができ、GTX RX の項で説明するような「Word Alignment」を行うことができる。8B/10B エンコードロジックは、PCS Parallel Clock（TXUSRCLK）と呼ばれるクロックによって実装される。TXUSRCLK も GTX TX が基準クロックから生成するクロックで、ユーザーが用いる TXUSRCLK2 と

位相が完全に同じになるように調整してある。周波数は転送レート・パラレルデータ幅により TXUSRCLK2 と同じまたは 2 倍となっている。位相・周波数の関係が完全に決まっているので、次に示すような FIFO 構造により位相差を吸収する必要はない。

- Phase Adjust FIFO
  8B/10B 変換されたデータは XCLK というクロックに載せ替えられ、幾つかのロジックを経て PISO ブロックに入力される。XCLK は PISO ブロックで用いられる Serial Clock に対して、位相・周波数の調整が行われているクロックである。XCLK もユーザーが与える基準クロックから生成されるため、TXUSRCLK と周波数の関係は確定していて、基本的には全く同じ周波数である。一方で、XCLK と TXUSRCLK との間の位相の関係はデフォルトでは規定されていない。通常、2 つの位相または周波数の関係が一定でないクロック間でデータの受け渡しを行う際には間に FIFO 構造を入れる等の工夫が必要である。従って、ここでは通常 Phase Adjust FIFO を挟んでデータを受け渡す。

  ここで、TXUSRCLK に XCLK と全く同じクロックを用いた場合、Serial Clock・XCLK・TXUSRCLK・TXUSRCLK2 の全クロックの位相および周波数の関係が確定することがわかる。このように設定した場合、クロック間をまたぐ信号に FIFO 構造を挟むことなくデータ送信を行うことが可能であるため、Phase Adjust Buffer を省略（バイパス）することができる。Phase Adjust Buffer をバイパスした場合のみ、FPGA TX Interface から TX Driver までの処理にかかる時間が常に一定となる。NewSL では Fixed Latency のために Phase Adjust FIFO はバイパスする。

- PISO
  Parallel In Serial Out の略で、パラレルで入力された信号をシリアル化して送信するブロックである。XCLK に同期して入力されたパラレル信号は、TX Serial Clock に同期したシリアル差動信号として出力される。

GTX RX ブロック

図 3.8 に、GTX RX のブロック図を示した。GTX RX ではシリアル差動信号として受信したデータをパラレル信号に復元して FPGA RX Interface に出力する。

- RX CDR
  CDR は Clock Data Recovery の略称で、受信したデータから、ユーザーが与えた基準クロックを元にしてシリアルクロックを抽出する。TX 側で 8B/10B 変換等のデータ変換が行われていることにより、データの High/Low の切り替わりの周期から送信の際に用いられた TX Serial クロックを復元できる。復元されたクロックを Recovery Clock、または RX Serial Clock と呼び、この周波数は受信データの周波数と厳密に一致している。

- SIPO
  Serial In Parallel Out の略称で、受信したシリアルデータをパラレルデータに変換して出力する。受信するシリアルデータは RX Serial Clock に同期しており、出力するデータは XCLK に同期している（TX 側の XCLK とは別のものである）。この XCLK は RX Serial Clock から生成されており、位相・周波数の関係は確定しているため、ここで FIFO 構造を入れる必要はない。

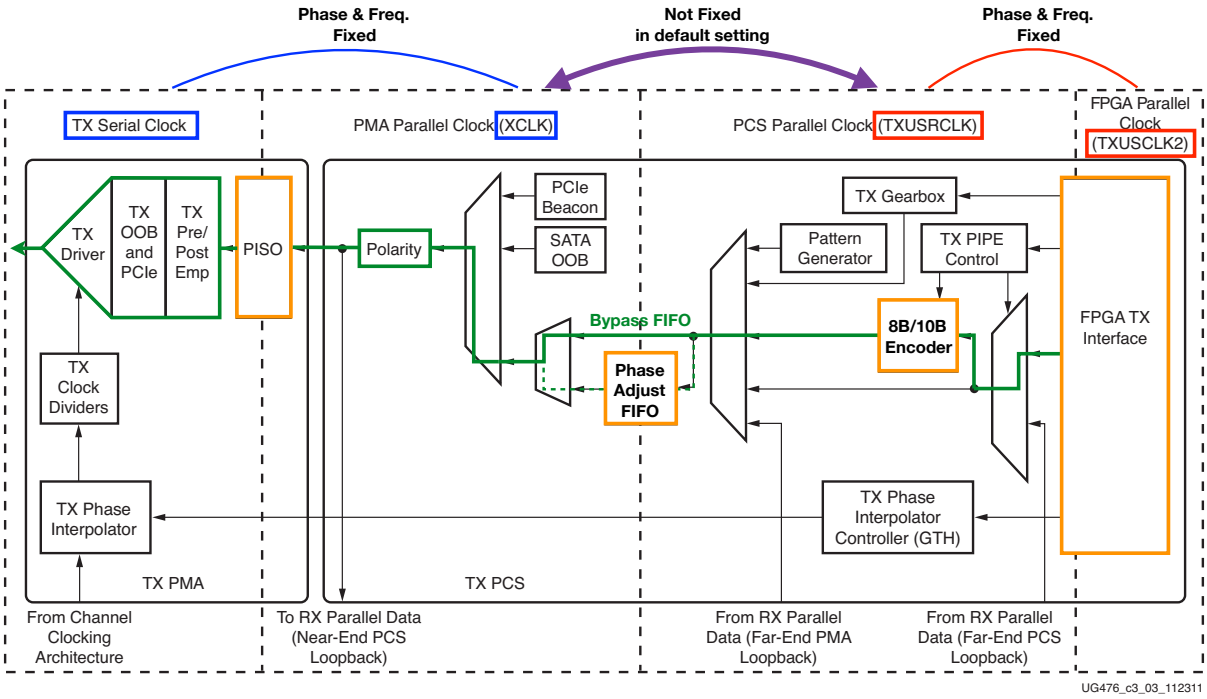


図 3.7　GTX Transceiver TX 部分のブロック図 [11]。各点線で囲まれた部分は 1 つのクロックドメインに属しており、同じクロックドメイン内の信号処理はすべて同じクロックにより行われている。データは FPGA TX Interface から順に左に送られ、各ブロックで処理されて TX Driver からシリアル信号で出力される。緑色の実線で Phase Adjust FIFO をバイパスする場合（NewSL で用いられる）のデータの流れを示した。オレンジの枠で囲まれたブロックは特に重要な処理を担当するため、本文で詳しく説明している。

- Comma Detect and Align

  Comma Detect ロジックでは、TX 側から送られてくる Comma と呼ばれる特別なパターンを識別する。この時、どのようなパターンをコンマとして用いるかは、受信側と送信側であらかじめ合わせておく必要が有る。Comma を発見すると、それ以降のデータは Comma を起点として規定ビットずつ読まれ、1 つの Word を復元することができる。これを「Word Alignment」と呼ぶ。特に、8B/10B 変換を用いている場合は、Comma を起点として 10-bit ずつ読み取り、次の 8B/10B Decoder ロジックに出力する。

- 8B/10B Decoder

  8B/10B 変換パターンに従って、10-bit のデータを 8-bit に復元する。ここで、Comma が検出された Word に対しては、「Comma を検出した」というフラグを合わせて後段に送信するため、ユーザー側のロジックでも Comma が正しく検出されているか確認することができる。

- RX Elastic Buffer

  8B/10B 変換までは XCLK に同期した処理であるが、FPGA RX Interface へは RXUSRCLK および RXUSRCLK2 に同期して送信する必要がある。ここでクロックドメインをまたぐため、RX Elastic Buffer で位相の差を吸収する。RXUSRCLK として CLK と位相・周期ともに全く同じクロックを用いることで、この RX Elastic Buffer をバイパスし、Serial 受信から FPGA RX Interface までの処理にかかる時間を一定にすることができる。NewSL では Fixed Latency を実現するために、RX Elastic Buffer はバイパスする。

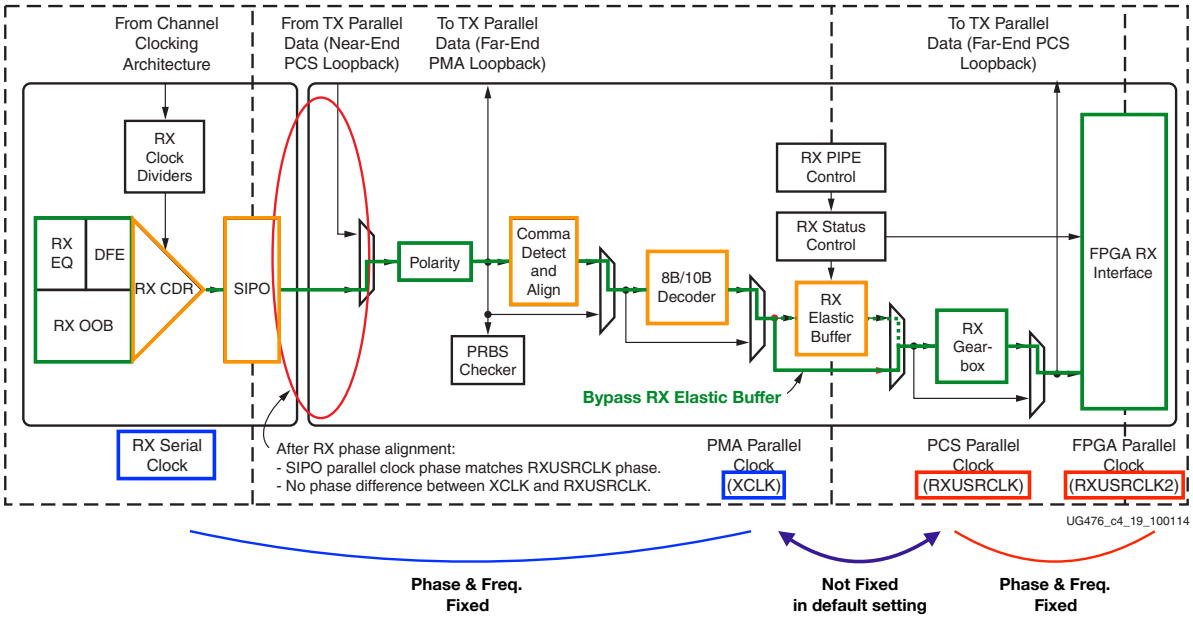


図 3.8　GTX Transceiver RX 部分のブロック図 [11]。各点線で囲まれた部分は 1 つのクロックドメインに属している。データは RX CDR から順に右に送られ、FPGA RX Interface にパラレルデータとして出力される。緑色の実線は Phase Adjust FIFO をバイパスする場合（NewSL で用いられる）のデータの流れを示している。オレンジの枠で囲まれたブロックは特に重要な処理を担当するため、本文で詳しく説明している。

3.4.2　SiTCP

SiTCP とは、FPGA を Ethernet と接続する技術で、これを用いることで TCP/IP を用いたデータ転送をハードウェアで実装することができる [12], [13]。SiTCP は Xilinx FPGA 上に実装できるライブラリとして提供されており、この特徴は比較的小さな回路規模（最大約 3000 slice = 約 20000 ロジックセル）で実装でき、イーサネット上限値の 1 Gbps での安定通信を実装できる点である。また、必要な外付け部品が Ethernet PHY チップと MAC アドレスを格納する EEPROM のみと少ない点も利点である。

図 3.9 に、SiTCP を用いたデータの読み出しの流れを示した。検出器からの情報に対してなんらかの処理をするボードに、ユーザー回路と合わせて SiTCP を実装することで、Ethernet 通信を通して標準的な PC でデータ読み出しが可能である。PC からのコントロールでボードにアクセスすることも可能である。また、図 3.10 にツイストケーブル（UTP）を用いる場合のハードウェア上での実装例を示した。このように、FPGA 上に SiTCP を実装し、FPGA 外部には PHY チップを置くだけで容易にデータ通信が実装できることがわかる。

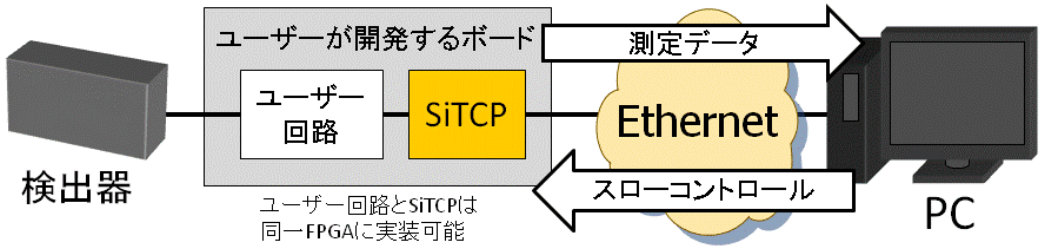


図 3.9　SiTCP の物理実験での使用例。ユーザー回路の情報を SiTCP 技術を用いて直接 PC で読み出すことが可能である [15]。

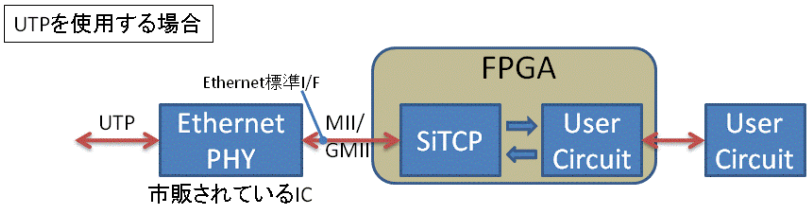


図 3.10　標準的なツイストケーブル（UTP）を用いる際の、SiTCP のハードウェアでの実装方法 [15]。FPGA 上に SiTCP を実装し、外付けチップとして市販の PHY チップを置くだけでデータ通信を実装できる。

NewSL ではこの技術を用いて TCP/IP でのデータ読み出しを行う。以下、SiTCP 技術を用いた通信のことを SiTCP 通信と呼ぶ。

3.5　基礎性能評価

開発した NewSL ボードの基礎的な性能評価を行った。主に、新機能である GTX 通信と SiTCP 通信に焦点をあてて試験を行った。

3.5.1　GTX データ送受信テスト

GTX のデータが正しく送受信できることを確かめる試験を行った。この試験では GTX TXUSRCLK に同期して数を増加するようなの値を送信し、受信側でその値が正しく得られているかを確認した。さらに、テストピンへその結果を出力することで GTX 送受信にかかる時間を測定した。これらのテストを 2 通りの設定で行い、それぞれにおいて送受信にかかる Latency を測定した。表 3.3 に、その 2 通りの設定の詳細を示す。

表 3.3　テスト時の主な設定。GTX 基準クロックは、GTX の TX 及び RX の基準クロックとして用いられるクロック。基準クロックのソースとは、その基準クロック（160 MHz）を作る際の元の 40 MHz クロックのことを表す。転送レートはともに 6.4 Gbps で、FIFO/Buffer は Bypass してある。

	テスト 1	テスト 2
GTX 基準クロック	Clock Jitter Cleaner の出力 160 MHz のクロック	
基準クロックのソース	水晶発振器（Internal Clock）の 40 MHz クロック	
TX パラレルデータの bit 幅	32-bit	16-bit
TX USRCLK2 の周波数	160 MHz	320 MHz
RX パラレルデータの bit 幅	32-bit	16-bit
RX USRCLK2 の周波数	160 MHz	320 MHz
転送レート	6.4 Gbps	
TX Phase Adjust FIFO	バイパス	
RX Elastic Buffer	バイパス	

図 3.11 と図 3.12 に測定したときのロジックアナライザーの出力画面を示した。図の 2 本の信号のうち、上側が送信データ、下側が受信データである。ここでは、送信側が “1” を送ってから受信側が “1” を受け取るまでの時間を計測している。これらを見ると、160 MHz の USRCLK を用いた場合では約 128 ns、320 MHz の USRCLK を用いた場合では約 70 ns の時間差があることがわかる。なお、ここで表示しているのは 160 MHz クロックの場合は 4 回に一回、320 MHz の場合は 8 回に 1 回カウントアップするカウンターの値である。これは、単にロジックアナライザ上での見易さのために信号の入れ替わる周期を遅くするためである。

3.5.2　GTX Bit Error Ratio テスト

次に、GTX でのデータ送受信に失敗する頻度を調べる試験を行った。Bit Error Ratio（BER）とは、送信したデータと受信したデータが一致しないような事象（ビットエラー）がどのくらいの割合で起こるかを示すもので、

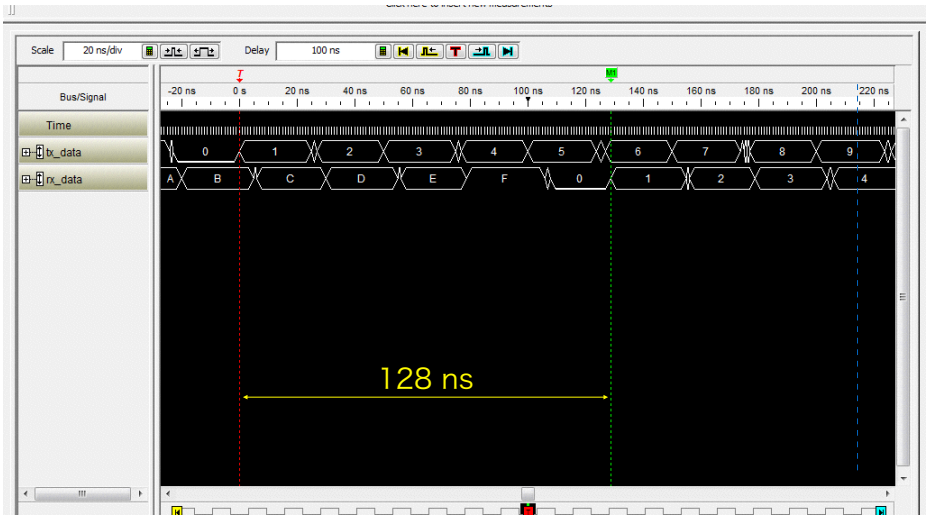


図 3.11　GTX の USRCLK の周期をを 160 MHz としたときのテスト結果。GTX の送信データ・受信データをテストピンで出力し、ロジックアナライザ上での表示させた画面のスクリーンショットである。上の行が送信データ、下の行が受信データに対応し、Latency は 128 ns であった。

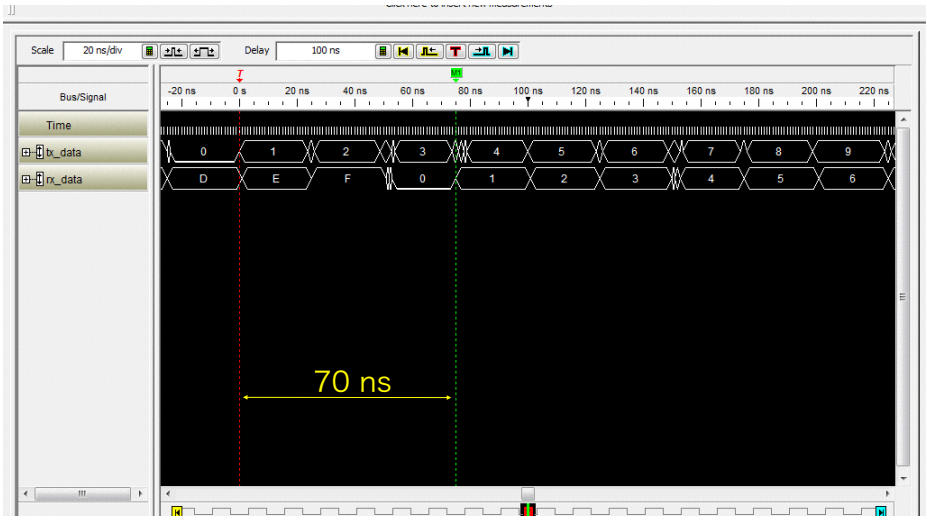


図 3.12　GTX の USRCLK の周期をを 320 MHz としたときのテスト結果。GTX の送信データ・受信データをテストピンで出力し、ロジックアナライザ上での表示させた画面のスクリーンショットである。上の行が送信データ、下の行が受信データに対応し、Latency は 70 ns であった。

（ビットエラー）/（送信されたビット数）で定義される。この Bit Error Ratio を測定するために、IBERT [14]（Integrated Bit Error Ratio Tester）IP Core を利用した。IP Core とはある機能を持った部分回路のブロックのことを指し、これを用いることでその機能の細かいデザインをする必要がなくなるため、回路開発をより迅速に行うことができる。IBERT は Xilinx 社の提供している IP Core で、GTX での BER 測定、及び後述の Eye Pattern テストを、GUI を用いて行うことができるようにするものである。IBERT では擬似ランダムパターンを生成し GTX TX を用いて出力する。出力したデータを自ら受信（ループバック）し、送信したデータと一致しているかどうかを調べ、BER を算出する。

図 3.4 に、作成した 5 枚の New Sector Logic ボード（NewSL ver.2-1 ～ ver.2-5 と表記する）について、転送レート 8.0 Gbps での BER 測定の結果を示す。作成した全 5 枚のボードに対してビットエラーは検出されず、転送したビット数から BER には上限が付けられている。BER は $O(10^{-15})$ 以下であり、安定したデータ通信が確立されていると言える。他の転送レートでも短時間での試験を行っており、特に実際に用いる 6.4 Gbps でもエラーは検出されなかった。転送速度により BER の上限値は $O(10^{-13} \sim 10^{-14})$ 程度が得られている。これらの転送速度での長時間の BER 測定を今後行う必要がある。

表 3.4　IBERT を用いた GTX 8.0 Gbps での通信における Bit Error Ratio テストの結果。ビットエラーは検出されなかったため、BER には上限値が設定されている。送信パターンは擬似ランダム信号（PRBS-31）を用いた。

ボード番号	試験時間 [s]	転送ビット [Gbit]	ビットエラー数	Bit Error Ratio [s^{-1}]
NewSL ver.2-1	41500	335000	0	$< 2.98 \times 10^{-15}$
NewSL ver.2-2	51800	414000	0	$< 2.42 \times 10^{-15}$
NewSL ver.2-3	44600	357000	0	$< 2.80 \times 10^{-15}$
NewSL ver.2-4	44700	357000	0	$< 2.80 \times 10^{-15}$
NewSL ver.2-5	46600	373000	0	$< 2.68 \times 10^{-15}$

次に、IBERT を用いた Eye Pattern の検証を行った。Eye Pattern とは図 3.13 の左側 (a) ～ (c) にあるように、シリアル通信で受信される波形の形を重ねて書いたもののことである。Eye Pattern を書くときは縦軸に電圧を取り、横軸には時間を取る。横軸の範囲としては 1-bit データを転送するために要する時間を取る。このとき、電圧の高低の差が大きく、信号の立ち上がりが早いほど線で囲まれた面積が大きくなることがわかる。転送レートが高いほど立ち上がりの時間はデータの周波数に対して長くなるので、線で囲まれた面積は (c) のように潰れていくこととなる。この形が「目」に似ていることから、これを Eye Pattern と呼び、この「目」が大きいく開いていることを安定した通信を確立するための指標として用いる。

Kintex-7 FPGA 上ではアナログ情報を用いることができないため、代わりに電圧及びタイミングのオフセットをずらしながら BER をスキャンする。横軸に時間のオフセット、縦軸に電圧のオフセットを取り、z 軸に BER を取ることで、Eye Pattern と同じような図を得ることができる。実際の例が図 3.13 の右側である。濃い青色で表示されている部分は測定時間中にビットエラーが検出されなかった領域に対応し、この領域の大きさ・形を見ることで目の開き方を確認することができる。

図 3.14 に、全 5 枚の NewSL ボードの転送速度 8.0 Gbps における Eye Pattern を示す。ボード間で大きな差

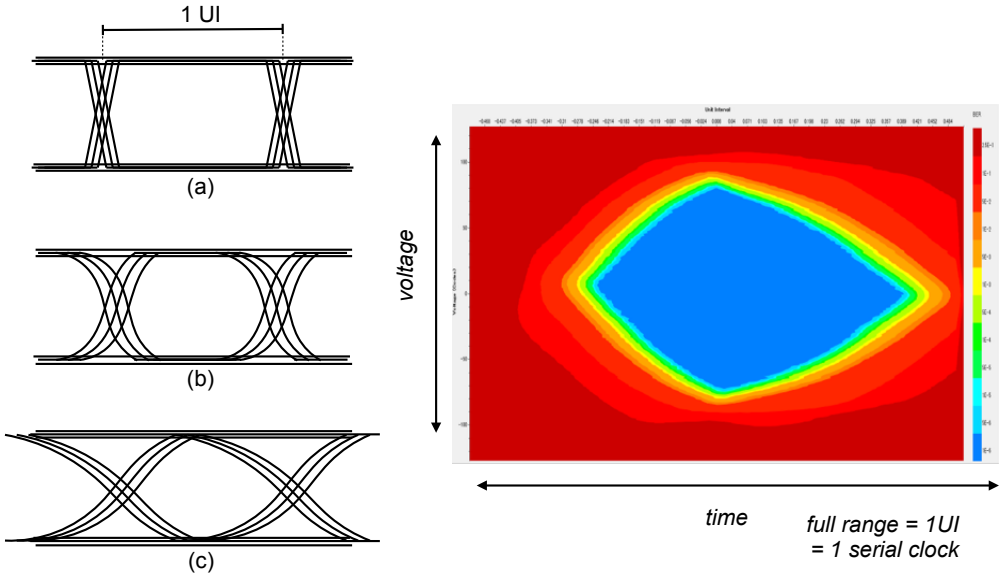


図 3.13　Eye Pattern の概念図と、IBERT を用いて得られた Eye Pattern。左の図では $(a) \rightarrow (b) \rightarrow (c)$ と順に転送速度が高くなっていく様子を示している。

異は見えず、Eye Pattern の目は十分に開いている。図 3.15 は、NewSL ver.2-5 における様々な転送速度に対する Eye Pattern である。転送速度が速くなるにつれ立ち上がりの影響が大きくなり、目の開き方が小さくなっている。付録 A に、NewSL ver.2-5 の全レーンの、試験を行った全転送速度に対する Eye Pattern を示しておく。

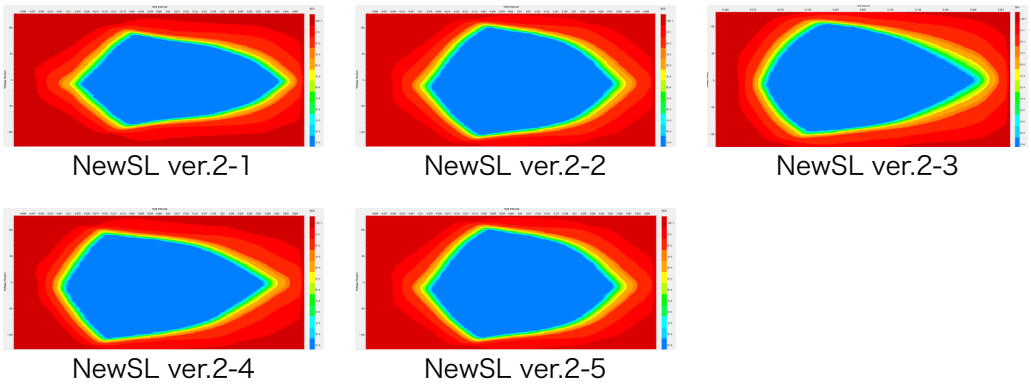


図 3.14　NewSL ver.2 の全 5 ボードに対する 8.0 Gbps における Eye Pattern。ボード間に大きな差異はなく、目が十分に開いていることがわかる。

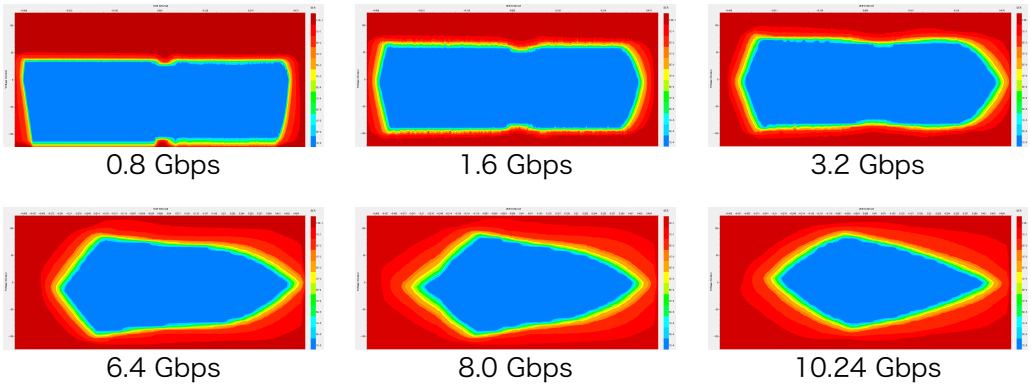


図 3.15　NewSL ver.2-5 の各転送速度における Eye Pattern。転送速度が高くなるにつれ、目が小さく横長になっていく様子が見える。

3.5.3　SiTCP 送信テスト

FPGA に SiTCP を実装し、実際に Ethernet を用いた通信ができるかどうかの試験を行った。NewSL FPGA 上に SiTCP のシステムクロックの 125 MHz 同期してカウントアップする 8-bit のカウンターを実装し、その値を SiTCP を用いて Ethernet で送信した。データの受信には、株式会社 Bee Beans Technologies が開発したソフトウェア、SiTCP Utility を用いた。SiTCP Utility は SiTCP を用いた TCP 通信を行うことができるソフトウェアで、受信データを画面に表示、またはテキストファイルに保存することができる。図 3.16 が、データを画面に

表示させたときのスクリーンショットである。このように、受信データが想定通りにカウントアップしていることが確認できた。

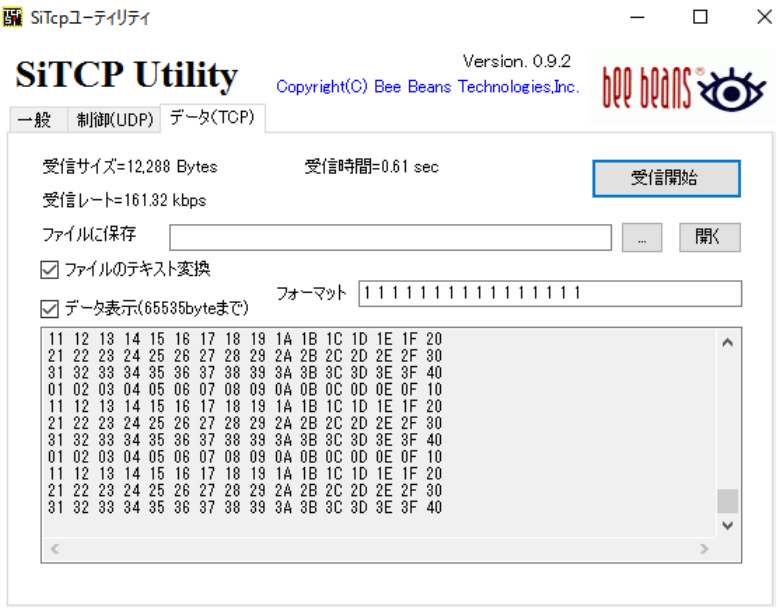


図 3.16　SiTCP Utility の表示画面。受信したデータが画面上に 8-bit 単位で表示されている。データが正しくカウントアップしていることが確認できる。

第4章　New Sector Logic Firmwareの開発

本章では、New Sector Logic のトリガー及び読み出し処理を実装するための Firmware 開発について述べる。まず 4.2 節で開発した Firmware の概要を説明する。4.3 節、4.4 節、4.5 節では NewSL Firmware をインターフェイス部分・読み出し部分・トリガー部分の 3 つに分けて順に説明する。最後に開発した Firmware の性能評価の結果を報告する。

4.1　Firmware とは

FPGA は、回路情報を与えることで様々な回路を実装することができる IC である。FPGA に与える回路情報は、“Soft”ware 上で記述された “Hard”ware 情報であるため、“Firm”ware と呼ばれる。適切に Firmware を開発し、それを FPGA にダウンロードすることで様々な機能を持たせることができる。図 4.1 に、Firmware のイメージ図を示した。Firmware は専用の Software を用いてコンパイルされ、FPGA にダウンロードされる。このコンパイル過程で、ユーザーの定めたロジック及び配線・配置等の制約は回路の情報に焼き直される。本研究では、この開発・コンパイル環境として Xilinx 社の提供する “Vivado” と呼ばれる Software を用いた。使用したバージョンは、Vivado 2016.1 である。

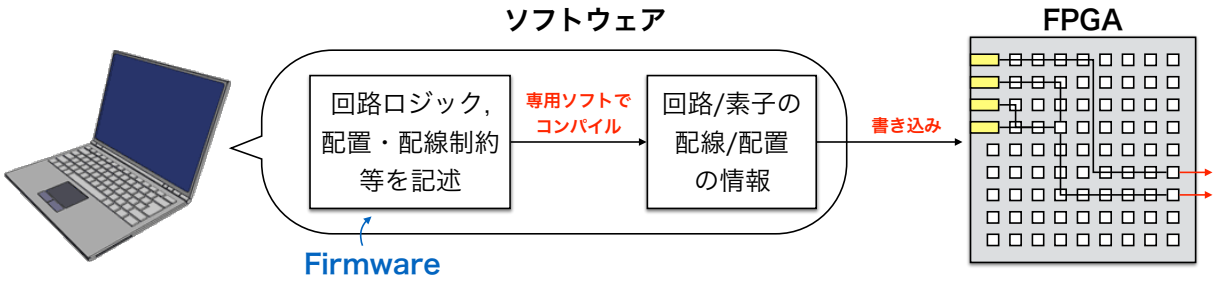


図 4.1　Firmware のイメージ図。Software 上に書かれた回路情報のことを Firmware と呼ぶ。

Firmware は回路を記述するものであるため、実際にどのようにして回路上に実装されるのかを考慮しながら開発を進める必要がある。特に、FPGA では論理回路は同期回路としてクロックに同期した形で実装される。したがって、信号がどのクロックに同期しているか、クロック間をまたぐ場合にどのように処理されるかを考える必要がある。

4.2　Firmware の概要

図 4.2 に、開発した NewSL Firmware のブロック図を示す。Firmware は大きく 3 つの部分に分けられる。1 つめがトリガー部分で、図 4.2 の “Trigger Firmware” と示されたブロック部分である。ここで受信データをもとにトリガー判定を行い、その結果を出力する。2 つめが読み出し部分で、データの保持・圧縮・整形を行い、SiTCP を用いて外部に送信する機能を担当する。図 4.2 では”Readout Firmware” というブロックで示されている。3 つめが、データの送受信を担当する部分で、図 4.2 の外につながる矢印と接続されている “delay” “GTX RX Interface” “GTX TX Interface” の部分である。主にシリアルデータを受信・または送信するための機能を担当する。この章では、それぞれに求められる機能を満たすために、どのように Firmware のデザインしたかを説明する。

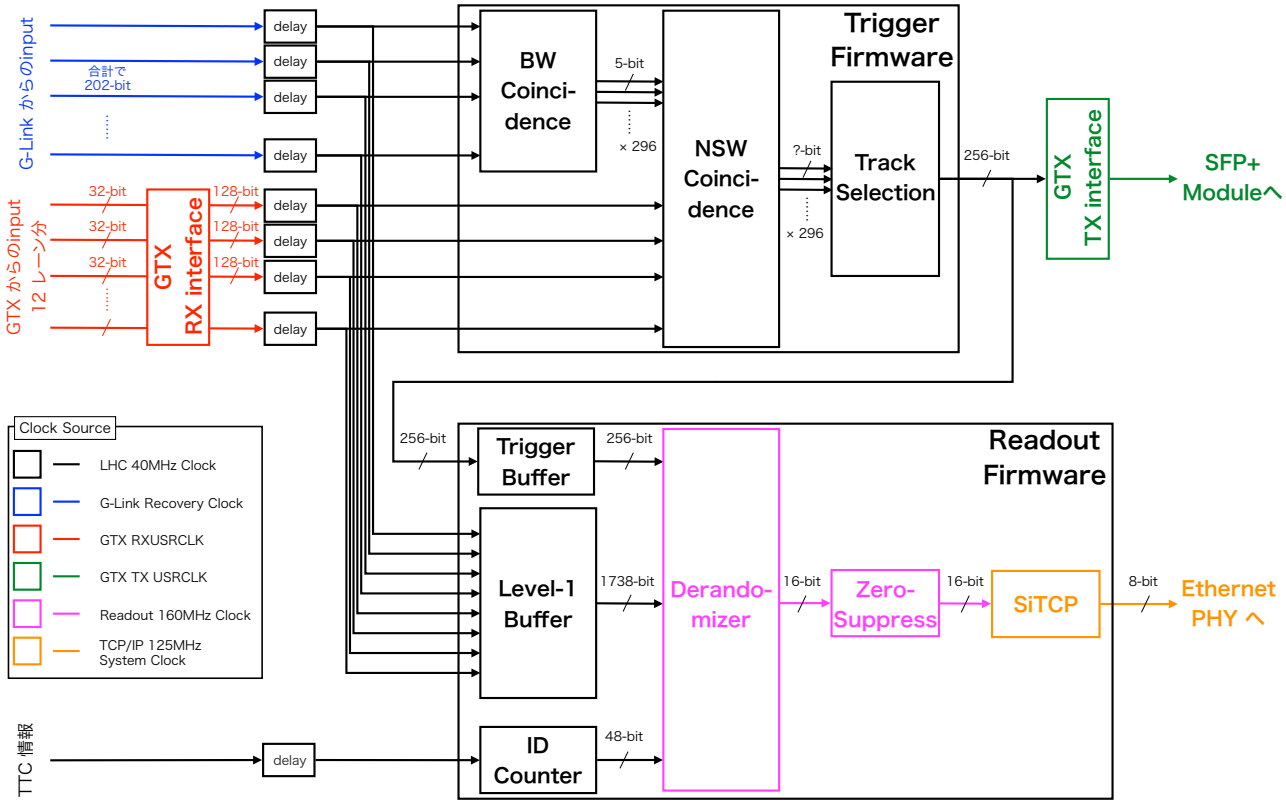


図 4.2　New Sector Logic の FPGA に実装される Firmware の概要。

4.3　GTX/G-Link 送受信 Firmware

4.3.1　GTX 受信フォーマット

NewSL では GTX を用いた高速通信を通じて NSW、Tile Calorimeter、BIS7/8 のデータを受信し、トリガー判定に用いる（3 章を参照）。このうち、NSW のトラック情報は図 4.3 のように定められたデータフォーマットに従って受信される。このフォーマットでは、データのはじめに Comma Word を埋め込むことで常に Word Alignment

が実行できるようにしている。さらに、データの最後に ID 情報をつけてタイミング調整やエラー検知をしやすくしている。このフォーマットに従うと、1 本の GTX レーンで NSW から毎 BC、最大 4 トラック分の情報を受け取ることができる。3 章で述べたように、1 NSW セクターからは 8 本のトラック分の情報を受け取る。従って、NSW 1 セクターには 2 本の GTX レーンを割り当てる。1 枚の NewSL ボードは最大 3 NSW セクターから情報を受け取るので、6 本の GTX レーンを用いることになる。

Words	first byte	second byte
Word-0	comma	comma
Word-1	track-0	
Word-2		
Word-3	track-1	
Word-4	track-2	
Word-5		
Word-6	track-3	
Word-7	ID (4-bit)	BCID (12-bit)

図 4.3　New Sector Logic が 1 BC あたりに New Small Wheel から受信するデータのフォーマット。

この時の GTX 1 本あたりの転送レートを計算しておく。1 BC あたりに図 4.3 で示した 8 words（16 Bytes = 128-bit）の送受信が必要である。この 128-bit のデータは GTX transceiver の処理内で、3 章で説明した通り 8B/10B 変換が行われる。従って、128-bit × 10/8 = 160-bit が転送される。160-bit が 1 BC で、つまり 40 MHz 毎に受信されるので、転送レートは 160-bit × 40 MHz = 6.4 Gbps となる。

の GTX でデータを受信する予定のその他の検出器、Tile Calorimeter および BIS7/8 からのデータ受信のフォーマットはまだ定まっていない。しかし、検出器の分解能およびカバーする範囲を考慮するとこれらの検出器からの情報を合わせても NSW よりも大きいデータ量になることは無いと考えられる。ここでは、他の検出器のために最大でも NSW と同じ帯域を用意しておけば十分であるという仮定のもと、それらからの情報を GTX 6 レーン、1 レーンあたり 6.4 Gbps でデータを受信すると考えることにする。従って、NewSL が GTX で受け取る情報は 6.4 Gbps × 12 レーンで、76.8 Gbps となる。

4.3.2　GTX 受信部分のデザイン

GTX 受信 Firmware の役割は、GTX で受けた信号を一時的に保存し、それを 40 MHz の LHC Clock に同期した形で後段の Trigger Firmware および読み出し Firmware に送信することである。GTX RXUSRCLK に同期して受信した信号を、LHC クロックに同期した信号として出力するので、この部分でクロックドメインの変更が行われる。通常クロックドメインを変更する際は FIFO を挟むことで、2 つのクロック間で信号のやりとりを直接行わないようにする。しかし、FIFO を用いると一定の Latency であるという保証がなくなってしまうため、NewSL ではこれを採用することはできない。従って、FIFO を用いない 2 クロック間のデータの受け渡しを確実に行うため、以下に説明するような実装をした。

GTX 受信 Firmware は 2 つの部分に分けられる。1 つ目が GTX の RXUSRCLK に同期して受信されるデータを 128-bit の形に変更して出力する GTX RX Interface 部分である。もう一つが GTX RX Interface から出力

される 128-bit のデータを 40 MHz に同期した信号に変更し、適切な遅延をかけることで全 12 レーンからの信号のタイミングを調整する、delay 部分である。

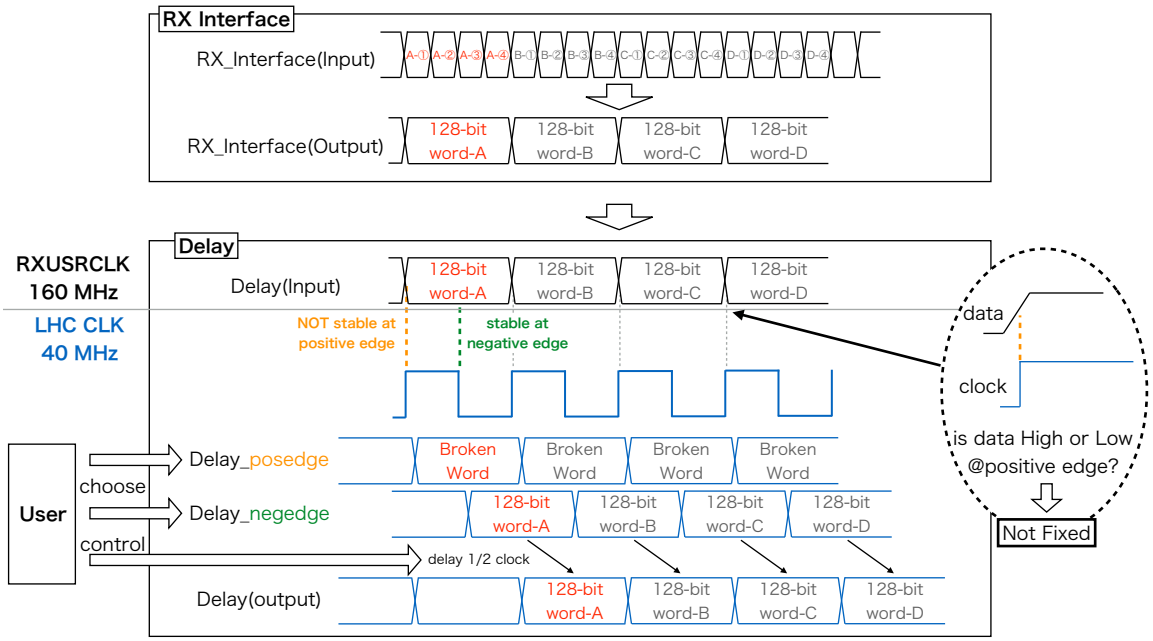


図 4.4　GTX RX Firmware による処理。データおよびクロックの線の色は同期しているクロックの違いを表している。

図 4.4 に、GTX 受信部分の信号処理のイメージ図を示した。GTX からのデータは GTX RXUSRCLK に同期して受信される。私の開発した Firmware では 160 MHz の RXUSRCLK を用いて、32-bit のデータを受け取るデザインとした。このとき受信されるデータは図 4.4 の一番上にあるように、128-bit を 1 word とすると 1/4 word ずつ 4 回に分けて 160 MHz のクロックに同期して受信される。図では word-A を A-①、A-②、A-③、A-④ の 4 回に分けて受信している。RX Interface 部分ではどの部分は A-① に対応するか、つまりどこに Comma Word があるかを識別して、Comma Word を見つけたらそこを起点に GTX から受信した 4 クロック分のデータを 128-bit にまとめて出力する、という機能を実装した。ここで、RX Interface の出力データは RXUSRCLK で数えて 4 クロックに 1 回値が更新されるようなデータであるので、この段階ではまだ RXUSCLK に同期したデータであることに注意する。

RX Interface から出力された 128-bit データはそのまま Delay モジュールに入力される。この Delay モジュールではデータを 40 MHz の LHC クロックに同期した信号に変更する。クロックドメインを変更するには、LHC クロックの立ち上がりで Delay(Input) のデータを読み、それを出力する、という機構を入れる必要がある。これを「ラッチ」と呼び、例えばこの場合では Delay(Input) の信号を LHC クロックでラッチしている。ここで注意しなければいけないのが、LHC クロックと USRCLK 位相の関係は分からないということである。これは、ボード上の LHC 40MHz クロックと RXUSRCLK の位相差は、ファイバー内の信号伝搬による遅延に依存するためである。位相の関係が図 4.4 の Delay(Input) とそのすぐ下の LHC クロックのような関係になっていた場合、LHC クロックの立ち上がりでラッチすると Delay(Input) のデータは安定していない。これは、前のデータから新しい

データへの更新がなされている最中に LHC クロックでその値をラッチしようとするからである。このような位相の関係の場合クロックの立ち上がりでデータをラッチするとデータ転送に失敗してしまう。一方、立ち上がりでラッチが失敗する場合は、立ち下りでは必ずラッチは成功する。クロックの立ち下りは立ち上がりとちょうど位相が半クロック分違うため、Delay(Input) 信号はデータ更新とデータ更新のちょうど真ん中でラッチすることができるからである。Delay モジュールではユーザーが外部から立ち上がりか立ち下り、どちらでラッチしたデータを用いるかをあらかじめ決めておくことができるようにしている。各々の GTX レーンに対して正しいエッジを選択しておくことでクロックドメインをまたぐデータの受け渡しを実現した。最後に、立ち上がりまたは立ち下りでラッチされた信号に対して 1/2 クロック単位で遅延をかけ、出力する。これにより、レーンごとのケーブル長等による遅延を吸収し、同じ BCID に属するデータを一斉に後段に送ることができる。

以上をまとめると、RX Interface と Delay 回路で適切に処理された信号は、LHC クロックの立ち上がりに同期した、BCID のそろった信号となる。この信号が後に説明する読み出し Firmware および Trigger Firmware へと入力される。

4.3.3　GTX 送信フォーマット

GTX TX はトリガー判定結果をシリアル化し、MuCTPi へ送信する。図 4.5 に、そのフォーマットを示した。送信フォーマットも受信フォーマットと同じく、Comma Word から送信する。これにより MuCTPi 側で Word の始まりを定義できる。Comma の後にはトリガー判定結果として、4 本のトラック候補の情報が送信される。トラック情報には RoI 8-bit、p_{T}4-bit、フラグ信号 4-bit 分が含まれる。RoI は Endcap 部で 148 個であったので、8-bit（$2^8 = 256$ パターン）で送信する。p_{T} 情報は、4-bit 割り当てているため最大 16 段階で p_{T} ラベルをつけて送信することができる。フラグ情報は 1-bit をミューオンの電荷情報に、残り 3-bit を NSW からのフラグ（トラックの質を表す情報）に割り当てている。

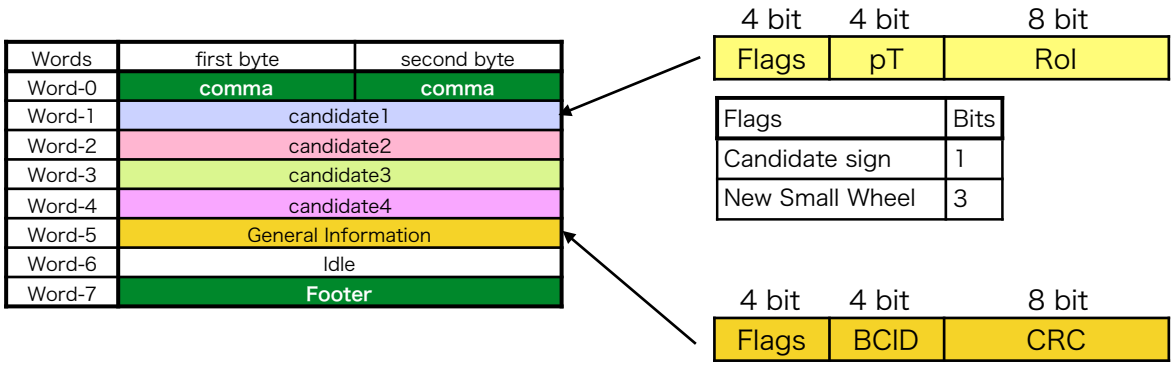


図 4.5　New Sector Logic が MuCTPi へ送信するデータのフォーマット。

4.3.4　GTX 送信 Firmware

GTX 送信 Firmware のデザインについて説明する。GTX で送信するトリガー判定情報は LHC の 40 MHz クロックに同期してトリガー Firmware 部分から出力される。一方、GTX TX へはデータを TX USRCLK に同期させて入力させる必要があるため、クロックドメインの変更を行う必要がある。GTX 受信 Firmware と同様に、FIFO を用いると Latency が一定とならないため、注意してデザインする必要がある。

GTX TXUSRCLK の周波数は 160 MHz を選択した。この 160 MHz の TXUSRCLK は NewSL ボード上で Clock Jitter Cleaner から作られた 160 MHz の基準クロックを元に、GTX トランシーバ内で生成される（周波数は同じなので、位相調整が行われるのみである）。Jitter Cleaner に入力されるクロックは LHC 40 MHz クロックであるので、結局 TXUSRCLK は LHC 40 MHz クロックから生成されていることになる。クロックから新たなクロックを生成する処理は完全に Latency が一定なので、LHC クロックと TXUSRCLK の位相の関係は常に一定である。この点は受信 Firmware と異なる。

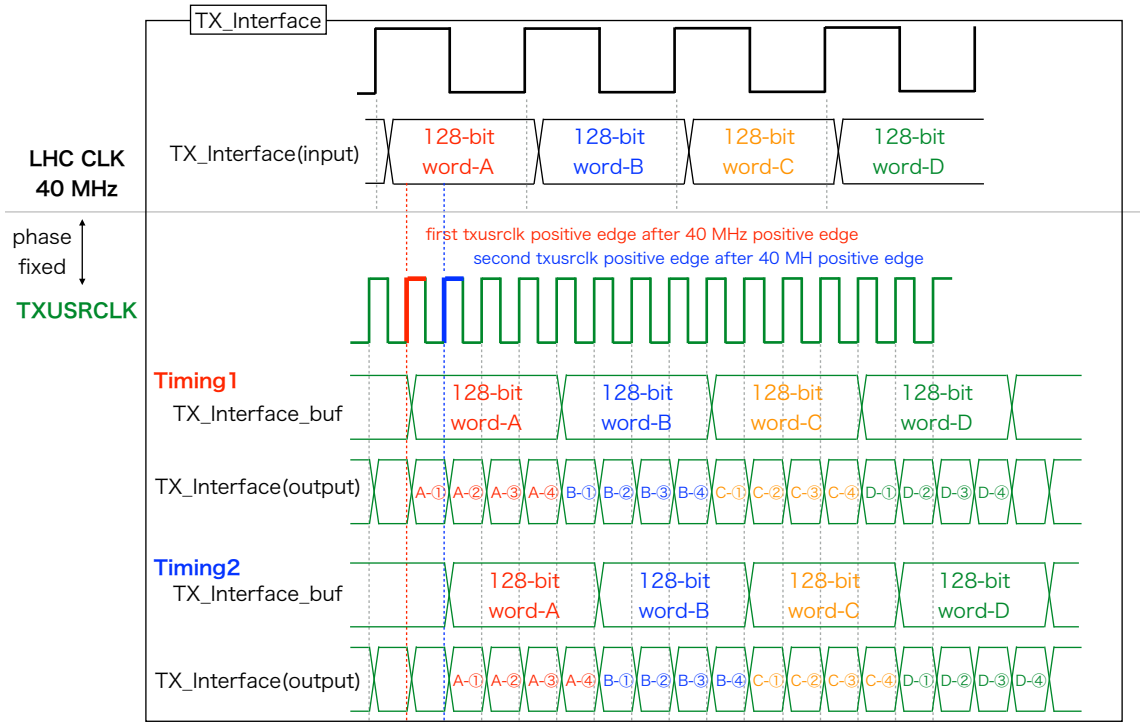


図 4.6　GTX TX Firmware による処理。

図 4.6 に、GTX TX Interface モジュールの信号処理の概念図を示した。TX Interface へは, トリガー判定結果を図 4.5 のようにフォーマットした 128-bit の情報が 40 MHz LHC クロックに同期して入力される。これを、160 MHz の TXUSRCLK に同期した 32-bit word 4 つに分割して送信するのが、このモジュールの役割である。TX Interface では、まず 40 MHz に同期した 128-bit の情報を、160 MHz に同期した 128-bit の情報に変更する。ここではこれを TX_interface_buf と呼ぶことにする。図 4.6 の Timing 1 と書かれた部分に注目すると、TX_interface(input) は赤色の 160 MHz の立ち上がりでラッチされ、TX_interface_buf に保存されていることがわかる。その後はこの TX_interface_buf を 4 回に分けて TX_interface(output) に出力するが、ここはクロックドメインの変更がないの

で Vivado が最適に配線することができる。

上記のロジックを導入するときに注意するべきことがある。それは、はじめに TX_interface(input) の情報をラッチときのクロックのエッジの選択である。仮に LHC クロックの立ち上がり後、1 つ目の TXUSRCLK 立ち上がり（図 4.6 の赤色のクロック立ち上がり）で情報をラッチした場合、Timing 1 に示されているようなタイミングで送信することとなる。一方、LHC クロックの立ち上がり後、2 つ目の TXUSRCLK 立ち上がり（図 4.6 の青色のクロック立ち上がり）で情報をラッチした場合は、Timing 2 に示されているようなタイミングとなる。Timing 1 と Timing 2 を見比べると、送信される情報の Latency が異なることがわかる。実際は 3 つ目、4 つ目の TXUSRCLK でも異なるタイミングで送信されるので、計 4 パターンの Latency の異なるパターンが存在する。どのようにして Timing が決まるかが重要であるが、これは TX interface の回路ロジックが動き出した瞬間に決まる。ロジックが動き始めたときの初めのクロック立ち上がりが上記 4 Timing のいずれに当たるかによって、送信 Latency が変わってしまう。これはつまり、Firmware の FPGA へのダウンロードの度に、もしくはロジックをリセットする度に Latency が変わってしまうということを意味する。これを避けるために、TXUSRCLK のどの立ち上がりで TX_interface_buf に情報を保存するかを決めておく必要がある。私は、ロジックリセット完了後に TXUSRCLK の立ち上がりを常に一意のタイミングにするシークエンスを導入した。実際には TX_interface(input) のデータ更新とデータ更新の中心付近の、最も安定している部分でラッチする Timing 2 を採用した。これにより、常に一定の Latency でのデータ送信が可能となった。

4.3.5　G-Link 受信 Firmware

G-Link 受信 Firmware は、GTX の場合と異なり Delay モジュールのみである。これは、GTX Interface で行っているような、信号の並列化の処理がすでに G-Link チップでなされているからである。しかし、G-Link 信号の位相と LHC クロックの位相の関係は一定でないため、GTX の場合と同様に立ち上がり・立ち下りを選択する仕組みは必要である。Delay モジュールの実装方法は、データ幅が 16-bit または 17-bit であることを除けば GTX の delay モジュールと同様である。

4.4　読み出し Firmware

開発した読み出し Firmware のブロック図を、図 4.7 に示した。読み出し Firmware は Level-1 Buffer、Trigger Buffer、ID Counter、Derandomizer、Zero-Suppress、SiTCP の 6 つのサブモジュールからなる。

4.4.1　Level-1 Buffer、Trigger Buffer

Level-1 Buffer、Trigger Buffer では、Level-1 Trigger（L1A）が発行するまで全てのデータをバッファーに保存し、トリガーが発行された場合にその BC と前後 1 BC 分、合計 3 BC 分のデータを読み出す。図 4.8 に L1Buffer 及び Trigger Buffer の概要を示した。これらの Buffer としては、L1A が到着するまで十分にデータを保持しておけるように、深さ 128 のメモリを用意した。Level-1 Buffer の入力 bit 幅は、1738-bit とした。これは、G-Link で受信される TGC Big Wheel の情報と、GTX 12 レーンで受信される NSW と他の検出器の情報の合計で、202-bit（G-Link）+ 128-bit（GTX）× 12 = 1738-bit となる。Trigger Buffer は NewSL 自身の Trigger 判定情報を保存

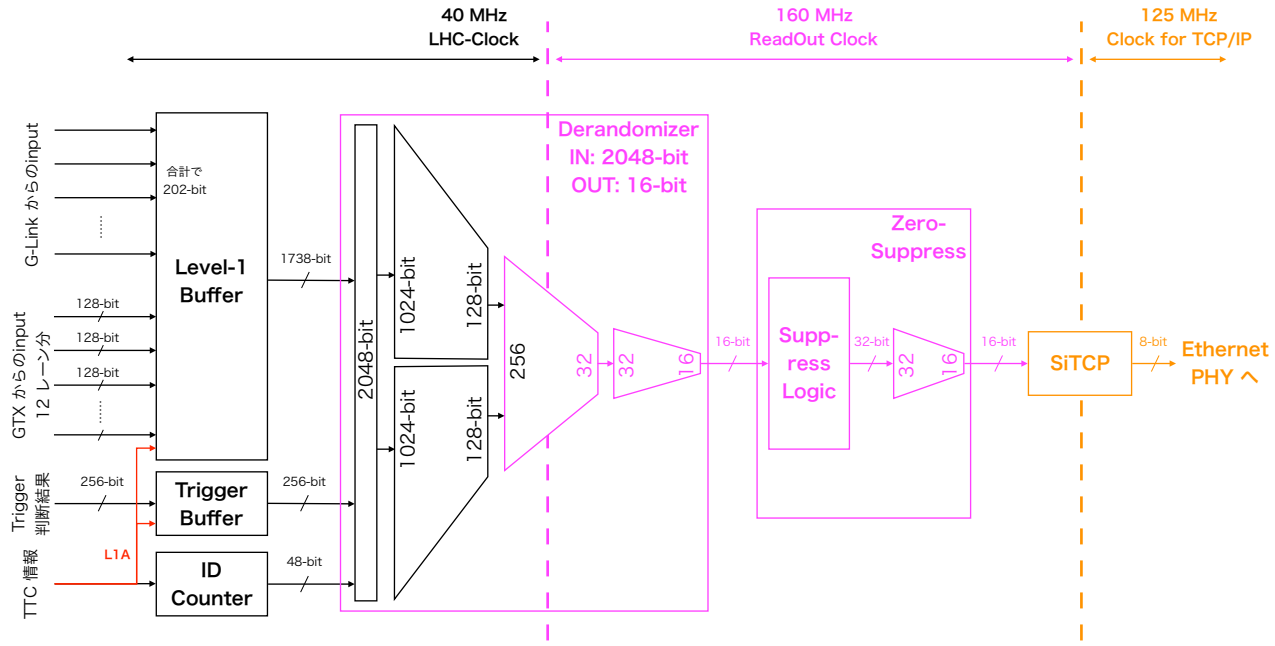


図 4.7　デザインした ReadOut Firmware の概要。

しておくためのバッファーである。Trigger Buffer には、図 4.5 にあるトラック 4 本分、64-bit を保存することとした。NewSL は 2 トリガーセクターを担当するので、128-bit 幅のメモリを用いた。トリガー判定は受信データを用いて行うので、受信データそのものに対してトリガー計算に要した論理遅延分、遅れて出力される。この分の遅延を吸収するために、Level-1 Buffer と Trigger Buffer は別のメモリを用意した。読み出す深さを調節することで同じ Bunch ID に対応する情報を後段の Derandomizer に送信することができる。

図 4.8 には実装の例を示した。この例では、Level-1 Buffer にデータが入力されてから 60 BC 後に L1A が到着すると仮定している。そのため、L1A が到着した瞬間に、Level-1 Buffer の深さ 60 の位置のデータと、その前後 1 BC （つまり深さ 59 と 61 の位置のデータ）を読み出せば良い。また、この例ではトリガー判定に 4 BC かかると仮定している。このとき、Trigger Buffer にデータが書き込まれるのは、Level-1 Buffer にデータが書き込まれるタイミングと比べて 4 BC 遅くなるので、深さ 55 ～ 57 にある 3 BC 分のデータを読み出せば良い。このように、メモリのどの深さのデータを読み出せば良いかは L1A の到着するタイミング及びトリガー判定にかかる時間に依存するため、Firmware を書き込んだ後にユーザーが設定できるようにした。

4.4.2　ID Counter

受信データとともに、そのデータの ID 情報を付与して送る必要がある。この ID 情報を元に High Level Trigger は他の検出器と同じイベントの情報を共有し、より高い精度のトリガーを行うためである。ID には Bunch ID （BCID）及び Event ID の 2 種類がある。

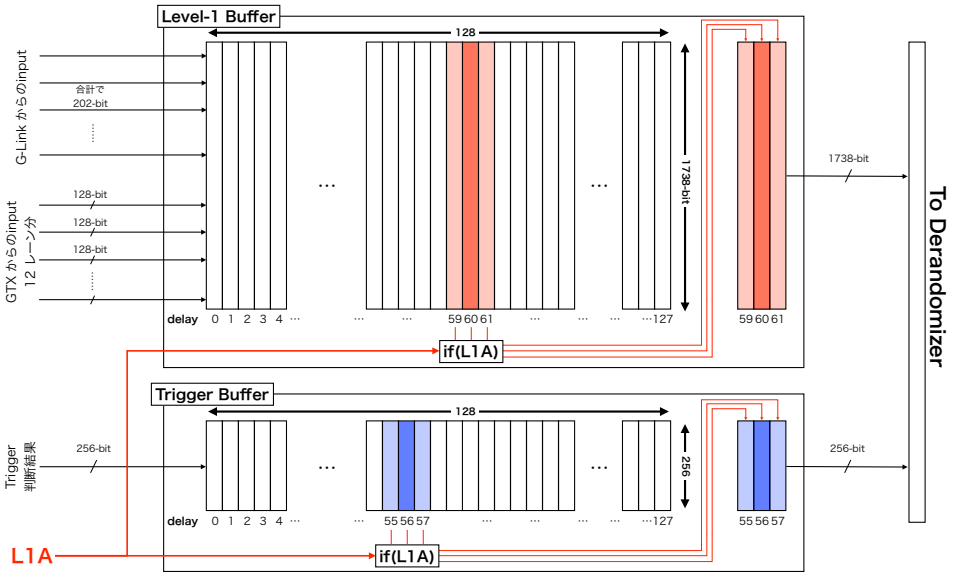


図 4.8　L1Buffer および Trigger Buffer の概要。

BCID は、LHC のバンチ構造を利用した ID 情報で、ある特定のバンチを BCID=0 として、そこから数えて何バンチ目の交差にあたるかを示す情報を持つ。LHC 1 周には 3564 個のバンチ構造が含まれるので、BCID は 0~3563 までで定義されており、16 進数で表示すると 0x0~0xDEB となる。従って、BCID を表すのに必要なビット幅は 12-bit である。LHC のバンチ交差は 40 MHz で行われているため、この BCID は 40 MHz 毎に 1 ずつ数を増やす。

Event ID は、LHC のランが開始してから何回 L1A 信号が発行されたかを示す ID 情報である。L1A の回数を数える ID 情報なので、L1ID とも呼ばれる。L1ID は NewSL では 12-bit まで数えており、L1ID をリセットする信号を受信したら、再び 0 に戻す。後段の読み出し回路では L1ID を常に監視しておいて、L1ID が 0 に戻ったら Extended L1ID と呼ばれるカウンターでその回数を数えて、L1A と合わせて 24-bit の ID 情報で全イベントを識別する。

これらの ID を付与するために NewSL で必要な情報は、BCID を数えるための 40 MHz の LHC クロック、BCID のリセット信号（BCID Reset、BCR）、L1ID を数えるための L1A、L1ID のリセット信号（Event Counter Reset、ECR）の 4 つである。これらの情報は TTC（Trigger、Timing、Control）システムから受信する。TTC システムとは、トリガー信号（L1A）及びタイミング信号（LHC クロック、BCR・ECR）、コントロール信号（Test Pulse、BUSY）を合わせて全 L1 システムに供給するシステムである。

NewSL では 16-pin フラットケーブルで TTC 信号を受信し、FPGA に入力している。ID Counter 部分ではこれらの情報から BCID・L1ID を数え、データを読み出すタイミングと合わせて次の Derandomizer へ入力する。ID Counter の実装方法はシンプルで、BCID Counter は LHC クロックの立ち上がりのたびにカウントアップし、BCR でリセットする 12-bit カウンター、L1ID COunter は LHC クロックの立ち上がり時 L1A 信号が受信されている時のみカウントアップし、ECR でリセットする 12-bit カウンターの出力として実装している。

4.4.3　Derandomizer (FIFO)

L1A が発行されたイベントに対しては、Level-1 Buffer、Trigger Buffer 及び ID Counter からの出力をまとめて 2048-bit としたのちに、FIFO にデータを保存する。この FIFO は 40 MHz で 2048-bit の情報を受信し 160 MHz 読み出しクロックに同期した 16-bit で出力するものであり、 これを Derandomizer と呼ぶ。FIFO の幅については、Vivado の IP Core を用いて作るのが最も資源を効率良く利用できるが、Core で設定できる FIFO の入力幅が 1024-bit までで、かつ出力幅は入力幅 の 1/8 まで、という制限があるので、図 4.9 のように実装した。

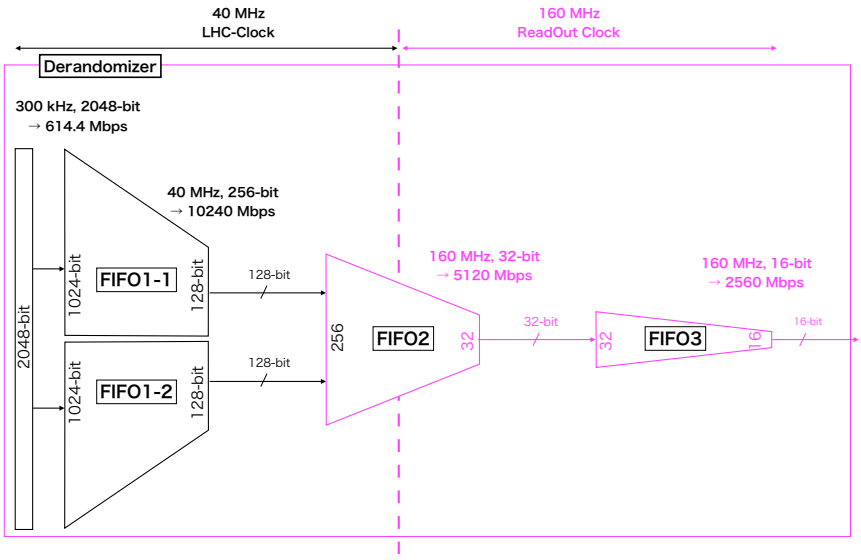


図 4.9　Derandomizer の概要。各 FIFO の出力部分の上には読み出し処理速度を示した。FIFO1-1 に関しては、入力レートも示している。また、各 FIFO の入力・出力のビット幅もブロック内に記した。線の色の違いはクロックドメインの違いを表している。

受信した 2048-bit のデータは初段の 2 つの FIFO に 1024-bit ずつ入力される。初段の FIFO（FIFO1-1 及び FIFO1-2）は入力された 1024-bit を 128-bit 8 回に分けて読みだす。2 段目の FIFO（FIFO-2） には初段の 2 つの FIFO の出力 128-bit を合わせて 256-bit としたものが入力される。FIFO-2 は入力された情報を 32-bit ずつ、8 回に分けて読みだす。この FIFO-2 の前後で 40 MHz LHC クロックから 160 MHz 読み出しクロックへのクロックドメインの変更が行われる。最終段の FIFO（FIFO-3）では入力 32-bit を、2 回に分けて 16-bit ずつ読み出し、次の Zero-Suppress モジュールへと入力する。

FIFO の深さについては、データの入力レート及び読み出しのレートを考慮して決定する必要がある。深さを大きく取ると、FIFO 内に一時的にためておけるデータの量は大きくなるが、その分必要な FPGA 上の資源は大きくなる。一方、深さが小さすぎると短い時間内に偶然トリガーが発行されるイベントが多かった場合に、すぐに FIFO の容量がいっぱいになってしまい、データを失ってしまう。必要な性能を保ちつつ最小の資源で消費量済むように、Queueing 理論に基づいて FIFO の深さを決定した。ここで Queueing 理論について簡単に説明する。入力の平均レート λ、出力の平均レート μ、深さ N の FIFO 構造がある時に、この深さ N が全てデータで満たされる確率 τ を求めたい。入力が完全にランダムで行われると仮定すると、深さ N のうち n が埋まっている確率 P_n

は、

$$P_n = \frac{(1-\rho)\rho^n}{(1-\rho^{N+1})}$$

となる。ここで、$\rho = \frac{\lambda}{\mu}$ である。$n = N$ となる確率が τ であるので、

$$\tau = P_N = \frac{(1-\rho)\rho^N}{(1-\rho^{N+1})}$$

となる。

今考えている FIFO の場合、入力のレートは L1A レート 100 kHz、1 回の L1A で 3 イベント分読みだすので 300 kHz となる。1 イベントあたり 2048-bit 入力されるので、FIFO に入力される情報は 614.4 Mbps である。出力のレートは 160 MHz で 16-bit ずつ読みだすため、2560 Mbps である。従って、$\rho = 0.24$ となる。これを用いて計算すると、τ は $N = 4$ では約 0.0025、$N = 8$ で 0.8×10^{-6}、$N = 16$ で $\sim 1 \times 10^{-10}$ となる。FIFO が全てデータで満たされている時に新たなデータが到来するとデータロスを起こすため、データロスが起こる頻度は、トリガーレートが 100 kHz、$\tau = 1 \times 10^{-10}$ の時 100 kHz $\times 10^{-10} = 10^{-5}$ Hz となる。これは、~ 30 時間に 1 回データロスが起こることに対応し、十分な性能とは言えない。$N = 32$ としておくと、$\tau \sim 1 \times 10^{-20}$ となり、データロスの頻度は $\sim 10^{-15}$ Hz で十分に低くなる。したがって、FIFO の深さとしては 32 以上を確保しておけば十分となる。

以上は Derandomizer を一つの FIFO と見て計算した場合の結果であるが、Derandomizer に含まれる FIFO 一つ一つの処理速度を見ると図 4.9 のようになる。図を見てわかるように、各 FIFO の出力レートは最終段の出力レートより大きく、ボトルネックとなっている部分は無いため、上で述べたように Derandomizer を 1 つの FIFO として扱って良い。

以上の議論から、Derandomizer に含まれる各 FIFO の深さを表 4.1 のように決定した。ここで注意するのは、上の計算に用いた N は「何イベント分データを保持できるか」に対応するものなので、そのまま FIFO の深さとはならないことである。例えば 256-bit の入力幅の FIFO2 を考えると、1 イベント = 1024-bit × 3 BC を保持するには、FIFO の深さ $(1024 \times 3)/256 = 12$ 分の領域を必要とする。このようにして、各 FIFO の深さから N を計算し、それらを合わせたものを Derandomizer 全体としての N として τ を見積もった結果、$\tau \sim 1.8 \times 10^{-120}$ と、十分に低い値が得られた。

表 4.1　Derandomizer の FIFO の幅及び深さ

	入/出力 bit 幅	入/出力クロック周波数	入力レート	出力レート	ρ	N	深さ	τ
FIFO1	1024/ 128	40/ 40 MHz	307.2 Mbps	5120 Mbps	0.06	170	512	-
FIFO2	256/ 32	40/ 160 MHz	614.4 Mbps	5120 Mbps	0.12	21	512	-
FIFO3	32/ 16	160/ 160 MHz	614.4 Mbps	2560 Mbps	0.24	2	512	-
全体	2048/ 16	40/ 160 MHz	614.4 Mbps	2560 Mbps	0.24	193	-	1.8×10^{-120}

4.4.4　Zero-Suppress

Zero-Suppress は、データを圧縮するために実装したロジックである。このロジックは 0 の割合の多いデータの圧縮に特化したロジックで、基本的な原理はデータの 0 でない部分のみを送ることによって転送データ量を減らすものである。特に、NewSL の受信するデータはヒットがあった RoI に関するデータ以外はほとんど 0 なので、このロジックは非常に有効となる。図 4.10 に、Zero-Suppress ロジックの概略図を示した。

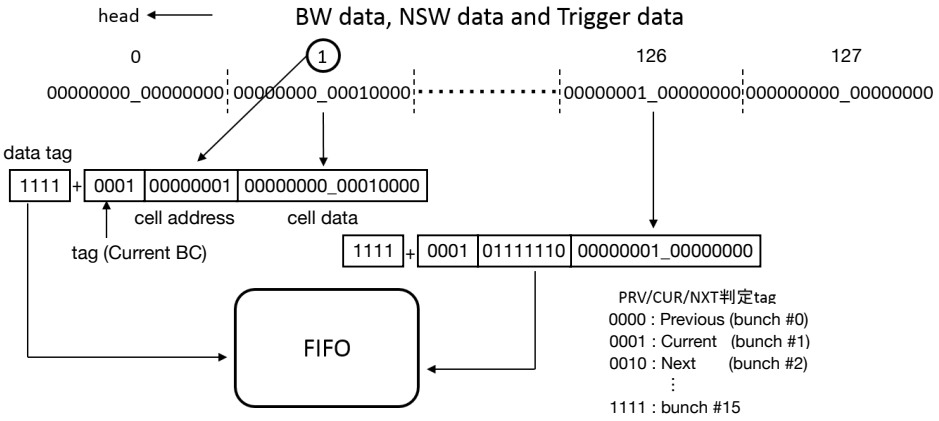


図 4.10　実装した Zero-Suppress ロジック [16]。

Zero-Suppress Logic では、まずデータを 16-bit（これを 1 cell と呼ぶ）ずつ読み出し、その 16-bit 中に 0 ではない bit があるかどうかを判定する。全て 0 だった場合、そのデータを捨てる。1 が 1 つでもあった場合は、それが読み始めてから何 cell 目かを識別する 16-bit のパターンをその cell のデータ（cell data）に付けて、32-bit で送信する。このように、0 でない cell に関してはデータ量が 2 倍になるが、0 の割合が多い場合はこのロジックでデータ送信量を減らすことができる。

データに追加する 16-bit のパターンは、表のようになる。まず、先頭 4-bit は全て 1 にする（Data Tag）。これは後述の SROD でのデータ識別に使用する。次の 4-bit が Bunch Tag と呼ばれるものである。Level-1 1Buffer の項で説明した通り、L1A が発行された時にはそのイベントと前後 1 BC 分のデータを送信する。L1A が発行された BC を Current BC、その前後を Previous/ Next BC と呼ぶ。Bunch Tag は cell がどの BC のデータの cell であるかの情報を与える。Bunch Tag は 4-bit 用意したため、最大 16 BC 読み出しまで対応する。残りの 8-bit でデータの中の何番目の cell であるかの情報、cell address を追加する。Zero-Suppress ではデータを 16-bit 毎に区切るため、1 BC 2048-bit のデータは 128 個の cell に分けられる。128 個の cell の位置は 7-bit で表現できるため、実質は 8-bit 中 7-bit を用いて cell の位置を表すことになる。

	Data Tag	Bunch Tag	cell address	cell data
bit 幅	4-bit	4-bit	8-bit	16-bit

図 4.11　Zero-Suppress 後のデータのフォーマット。

後段の SROD へ送るフォーマットにおいて、データの前後に付けるヘッダー・フッターや ID 情報部分は、Zero-Suppress フォーマットと異なるため、特別な処理が必要である。そのため、Zero-Suppress ロジックは、データの圧縮だけでなくこれらの処理や付加といった、データフォーマット整形処理も行う。図 4.12 が、NewSL から SROD に送信するデータのフォーマットである。NewSL から ROD へ送信するデータの長さは、Zero-Suppress ロジックにより可変長になる。0 でないデータが多ければ多いほど、データ長は長くなるからである。SROD ではイベントごとの情報をまとめて後段に送るため、データの始まりと終わりを判別して 1 イベント分の情報を抜き出す必要がある。データの最初と最後にヘッダー・フッターを付加することにより、それを可能にする。しかし、仮に Zero-Suppress されたデータの中にヘッダー及びフッターと同じパターンがあると、誤ってイベントの始まり・終わりを検知してしまう可能性がある。これを避けるために、Zero-Suppress のデータは 32-bit のはじめの 4-bit を必ず ”1111” （Data Tag）から開始することとした。SROD 側で 2 Word、32-bit 単位でデータの処理を行うことにすれば、データ部分は必ず ”1111” から始まるのに対し、ヘッダー・フッター及び ID は ”1111” 以外のパターンから始まるため、誤った判定を避けることができる。

<table>
<tr><td>Words (16-bit)</td><td colspan="2">Data Tag Bunch Tag</td><td>Bunch Tag</td></tr>
<tr><td>Word 0</td><td colspan="3">Header (0xB0D0)</td></tr>
<tr><td>Word 1</td><td>0000</td><td colspan="2">L1ID (12-bit)</td></tr>
<tr><td>Word 2</td><td>0000</td><td colspan="2">BCID (12-bit)</td></tr>
<tr><td>Word 3</td><td>0000</td><td colspan="2">SLID(12-bit)</td></tr>
<tr><td>Word 4</td><td colspan="3" rowspan="2">Zero-Suppressed data
Every 32-bit word starts with “1111”</td></tr>
<tr><td>Word (2N-2)</td></tr>
<tr><td>Word (2N-1)</td><td colspan="3">0x0000</td></tr>
<tr><td>Word 2N</td><td colspan="3">Footer (0xE0D0)</td></tr>
</table>

図 4.12　SROD へ送信するデータのフォーマット。

Zero-Suppress ロジックにより圧縮･整形された 32-bit のデータは、FIFO 入力されたのちに 16-bit で読み出され、次の SiTCP モジュールに受け渡される。この FIFO の深さを前述の Queueing 理論で定めるには、Zero-Suppress 後のデータのレート及び SROD でのイベント処理速度を知る必要がある。しかし、Zero-Suppress のデータ圧縮率は入力データの 0 の多さに大きく依存するため、NSW のノイズレート等に大きく依存し、評価ができない。また、SROD についても現在ソフトウェア開発中であり、処理速度は今後大きく改善する可能性も有る。従ってここで FIFO の深さは決定せずに、余裕を持って深さ 2048 とすることにした。この場合、ρ（= 入力レート/出力レート）= 0.99 においても FIFO が満たされる確率 $\tau \sim 1 \times 10^{-9}$ となる。

4.4.5　SiTCP

3 章で説明した通り、SiTCP は FPGA を Ethernet に接続する技術である。SiTCP を実装した FPGA と Ethernet PHY Chip を正しく接続することにより、少ないロジックリソースの使用で FPGA からを Ethernet 経由でデー

タを送信することができる。Gigabit Ethernet で用いるクロックの周波数は 125 MHz と決まっており、これは読み出しクロックの 160 MHz と異なるため、FIFO 構造を用いて周波数の差を吸収する。また、SiTCP のデータを PHY チップに受け渡すインターフェイス部分へは、データ幅 8-bit で送信する必要があるので、この FIFO でデータ幅の調整も行う。FIFO の深さは、Zero-Suppress の FIFO と同じく、入力・出力のレートがまだ確定していないため、大きめに 2048 とした。図 4.13 に SiTCP モジュールの概要を示す。

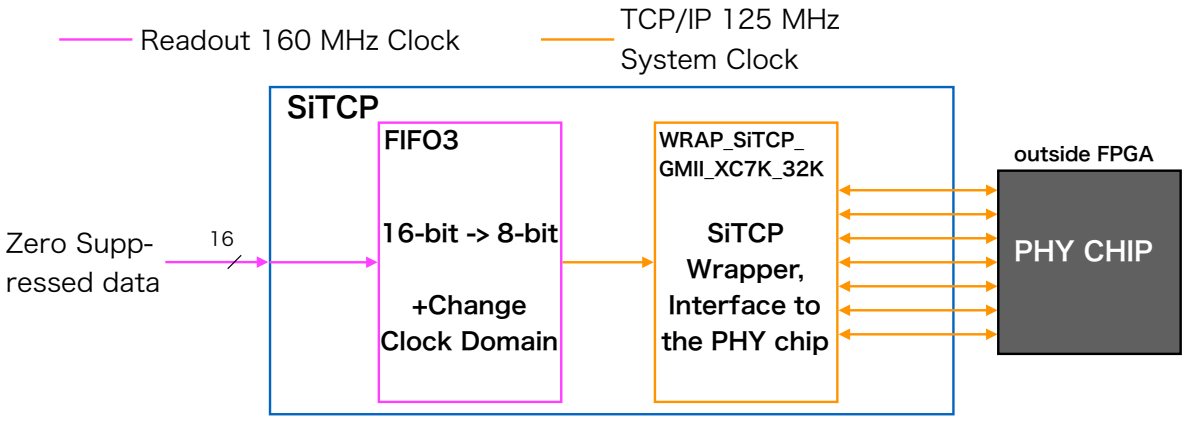


図 4.13　実装した SiTCP モジュールの概要。

4.5　Trigger Firmware

図 4.14 に、トリガー判定部分の Firmware のブロック図を示す。トリガーは 2 段階で判定を行い、第一段階では TGC Big Wheel の情報を用いたコインシデンスを取り、その後 New Small Wheel の情報を用いてトリガー判定を行う。最後に Track Selection ロジックで p_{T} の高い候補を選び、トリガー判定結果として GTX TX で MuCTPi へデータを送信する。

一段階目の Big Wheel コインシデンスは、基本的に現在のトリガーロジックと同じものを採用する。二段階目の Inner Coincidence 部分で NSW の情報をどのように用いてトリガー判定を行うかが、アップグレードにおいてトリガーレートをどのくらい削減できるかを決める。このトリガーロジックの詳細については、次章で述べる。また、二段階目のコインシデンスロジックの Firmware 上での実装方法に関しては、リソースとトリガーロジックとの両面を考えながら調整が必要である。これに関しても次章で考察するので、ここではトリガー Firmware の大枠を述べるに留める。

4.6　性能評価

4.6.1　GTX Fixed Latency の評価

図 4.15 に Latency テストの全体像を示す。

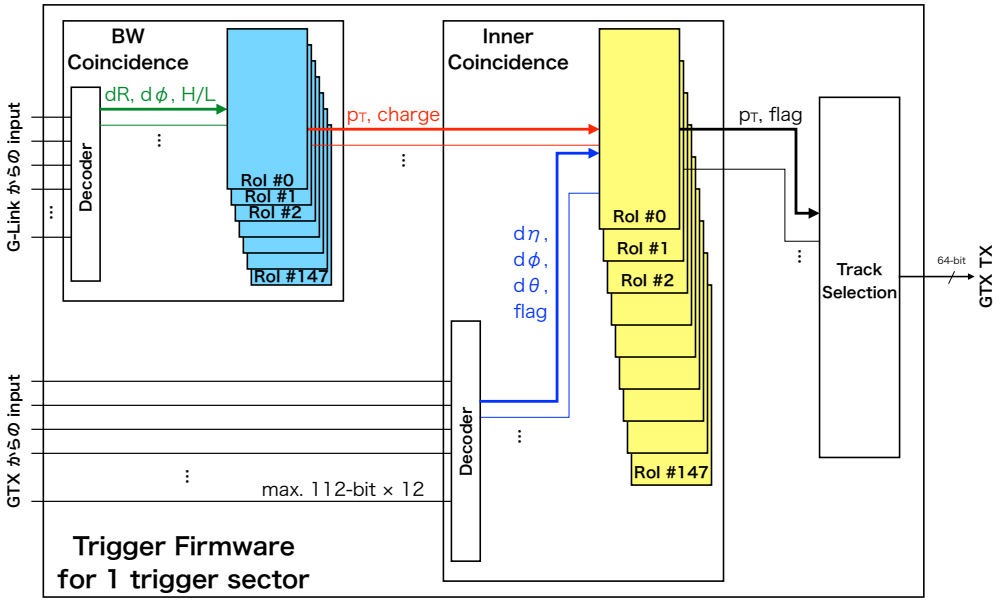


図 4.14　トリガー判定部分の Firmware のブロック図。各 RoI に対して BW 及び NSW のコインシデンス用のブロックを用意している。

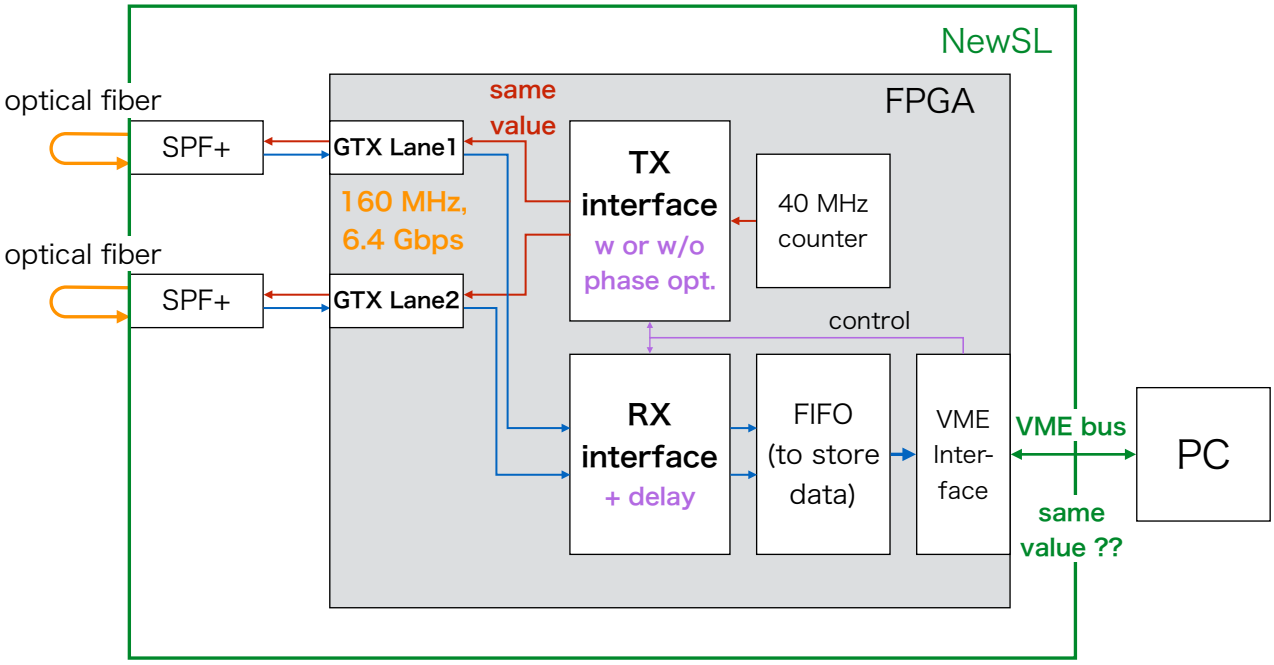


図 4.15　Latency テストの全体像。40 MHz で値が増加するカウンターの値を 2 本の GTX レーンで同時に送信し、それを光ファイバーを通じて再び受信する。受信したデータを 40 MHz に読み直した際に常に同じ値となるかを確認する。

このテストの目的は 40 MHz から 160 MHz へのクロックドメイン変更を常に Fixed Latency で実現できることを確認することである。テストの方法を以下に説明する。はじめに、40 MHz のカウンターを用いてテストパターンを作成する。このテストパターンは TX interface によって 160 MHz の GTX TXUSRCLK に同期した信号に変換され、2 つの GTX トランシーバに入力される。この時、この 2 レーンには同時に信号が入力される。この信号は GTX でシリアル化され出力される。これをループバックして受信したものを、RX interface で受信し、delay 処理を行い 40 MHz に同期した信号に変換する。これを FIFO に保存し、VME バスを経由して PC で値を読み出す。読み出し結果において、2 レーン間でカウンターの値があっているかどうかを確認する。以上が一連のテスト操作である。

確認しなければいけないのは、GTX のリセット後に Latency に変更が無いかどうかである。つまり、TX 及び RX でのクロックドメイン変更が正しい設定で行われた場合では、GTX のリセット前後で常に Latency が一定であることを確認したい。これを試験するために、上の一連の操作を 1000 回繰り返し、カウンターの値の差を調べた。また、対象実験として送信側でリセット後に TXUSRCLK の位相を調整するシークエンスを入れなかった場合と、受信側で立ち上がり・立ち下りの調整を行わなかった場合の試験も行った。図 4.16 及び図 4.17 に、試験結果を示す。

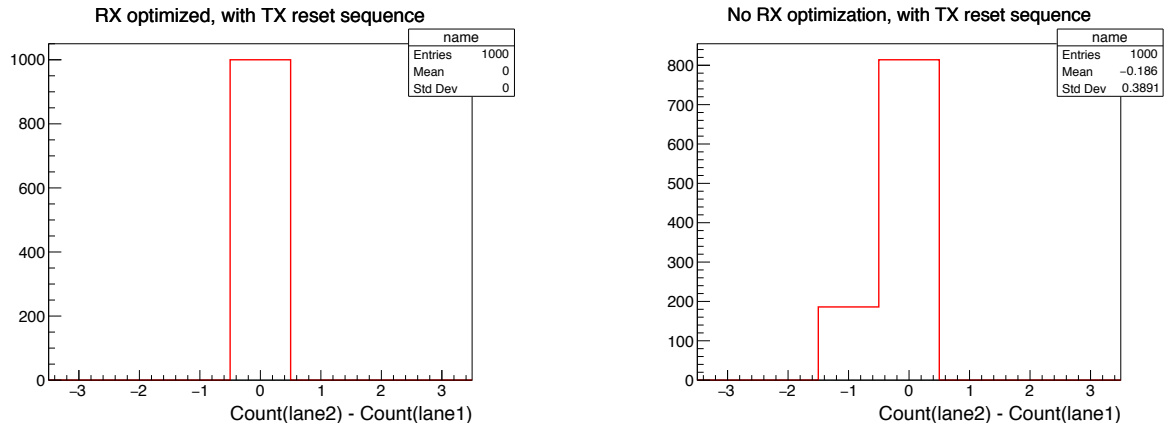


図 4.16　TX の位相合わせのロジックを入れた場合のカウンターの値の差。左が RX の立ち上がり・立ち下りを各レーンで最適なものを設定した場合。右が最適でないものを設定した場合。

図 4.16 は TX 位相合わせを行った場合の結果で、左がそれに加えて RX 側で適切な遅延パラメータを設定した場合、右が RX 側では最適でない遅延パラメータを設定した場合の結果である。図 4.17 は TX 位相合わせを行わなかった場合の結果で、左右は図 4.16 と同様 RX 遅延パラメータの最適化の有無に対応する。TX・RX 側で適切な信号処理をした場合のみ、Latency が一定になっていることが分かる。

以上により、私の開発した Firmware を用いて Fixed Latency でのクロックドメイン変更が実現できることが確認された。

4.6.2　読み出し Firmware の評価

図 4.18 のようなセットアップで、読み出し Firmware の評価を行った。

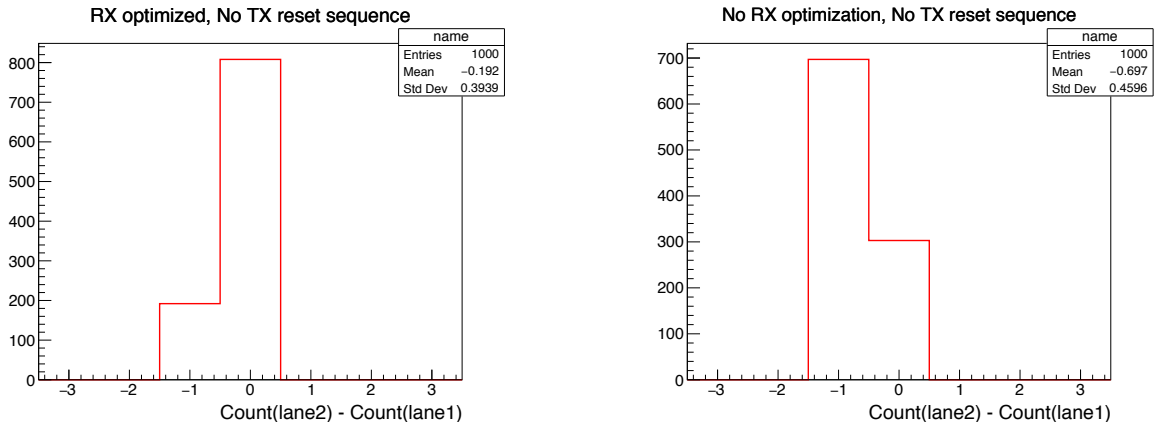


図 4.17　TX の位相合わせのロジックを入れなかった場合の場合のカウンターの値の差。左が RX の立ち上がり・立ち下りを各レーンで最適なものを設定した場合。右が最適でないものを設定した場合。

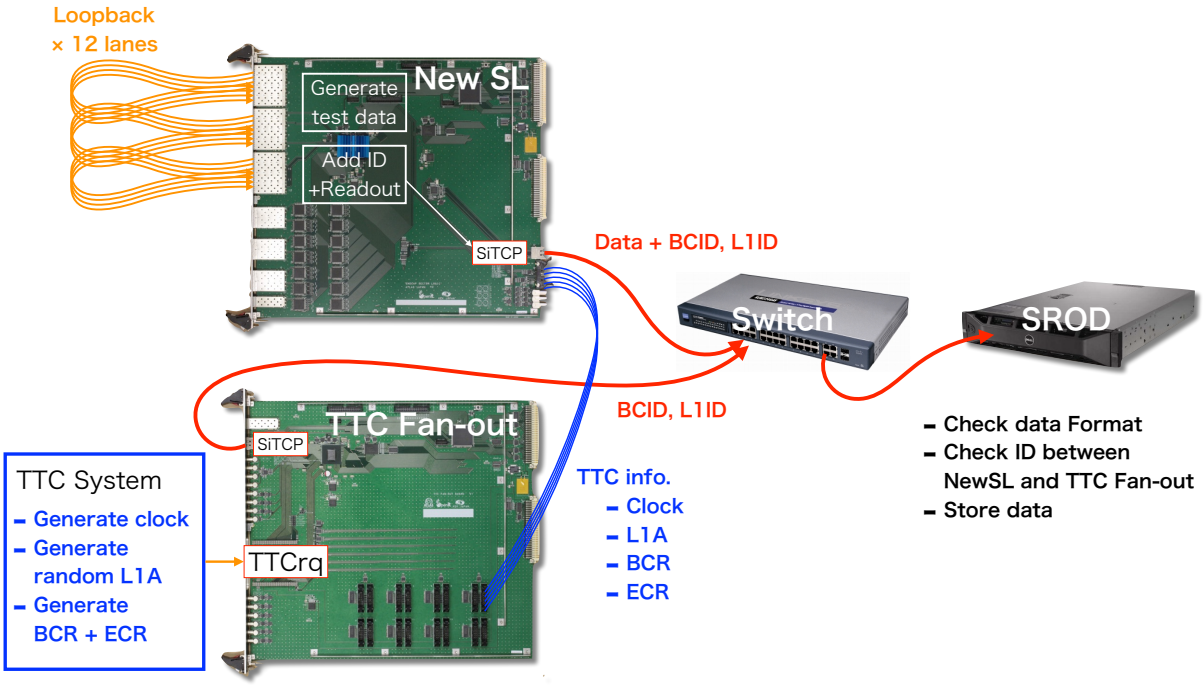


図 4.18　読み出し Firmware テストのセットアップ。

テストで TTC Fan-out というボードを用いた。TTC Fan-out ボードは光信号で供給される TTC 信号を電気信号に変換しフラットケーブルで NewSL に送信するために新たに開発されたボードである。実際の Run-3 での運用時は多数の NewSL に TTC 信号を分配するために使用される。また、同時に受け取った TTC 信号に含まれる ID 情報を SiTCP を用いて SROD に送信することができる。これにより、SROD では NewSL から受信した ID 情報と TTC Fan-out から受信した ID 情報を比較することで正しくデータが送られているかを監視することが可能となる。

今回のテストではこの仕組みを用いて NewSL の読み出し Firmware の評価を行う。NewSL では内部で作成した読み出し用のテストデータを GTX でループバックして受信し、読み出し Firmware で整形・圧縮した後に SiTCP を用いて SROD へ送信する。この時に、データ送信のトリガー及び ID 情報の付加のための TTC 信号を TTC Fan-out から受信する。TTC 信号は前段の TTC システムで生成され、TTC Fan-out を経由して NewSL に届けられる。TTC システムの詳細は省略するが、今回のテストでは L1A 信号は平均 100 Hz でランダムに生成し、BCR・ECR 信号は周期的に生成することにした。TTC Fan-out は TTC システムから受信した ID 情報を SROD に送信する。SROD では NewSL と TTC Fan-out の ID 及びフォーマットを確認する。

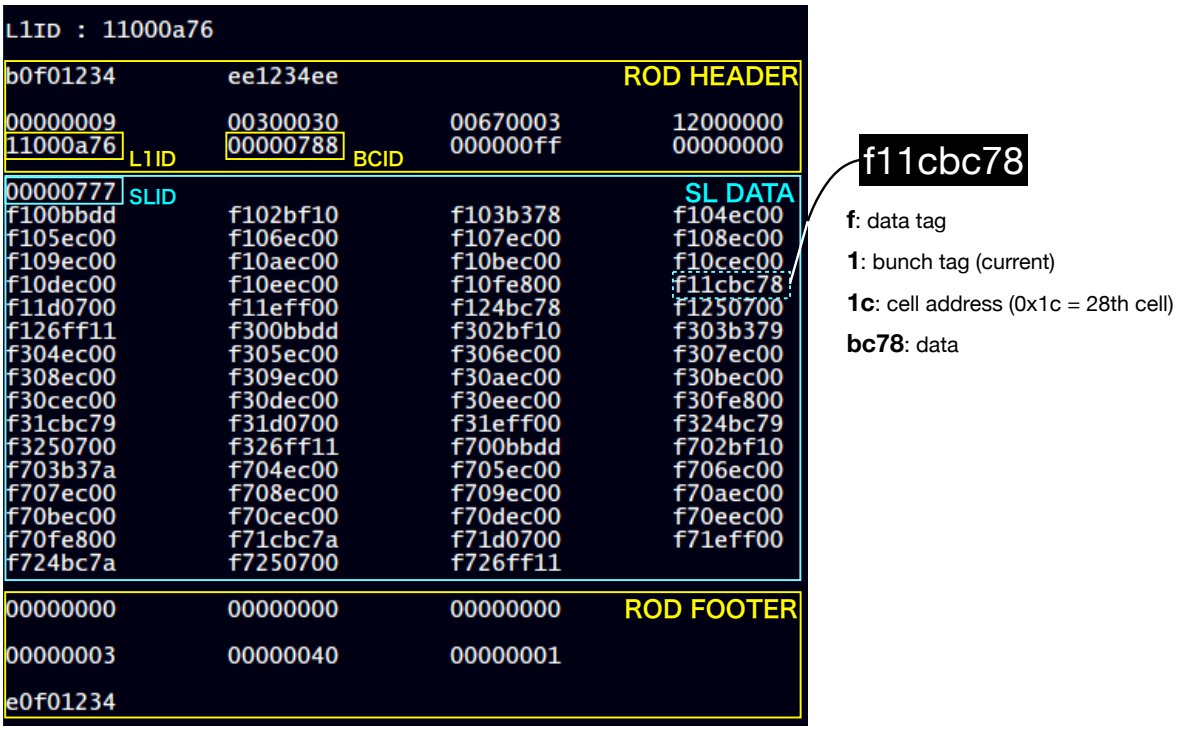


図 4.19　読み出し Firmware テストの結果。

図 4.19 に SROD で保存された 1 イベント分のデータを示す。SROD では両者の ID が一致しており、ヘッダーが正しく検出された場合に、それらの ID 情報から SROD のフォーマットに従った新たなヘッダーを作成しイベントデータの読み込みを開始する。NewSL のデータを順に受信し、データの終わり部分（NewSL データのフッター）を検出したら、自らもフッターをデータの最後に付加し、1 イベント分のデータとして保存する。この過程で ID 情報の不一致やヘッダー・フッターの未検出があった場合はエラーを出力した上でデータ保存を停止するデ

ザインにしてある。今回のテストでは 100 Hz の L1A レートでは数時間連続でデータ取得を行ってもエラーを検知せずに安定して動作することを確認した。また、図 4.19 にあるようにイベントのデータ部分では Zero-Suppress のフォーマット通りのデータが正しく受信されていることがわかる。また、データの内容も想定したものと一致していることを確認した。

第5章　New Sector Logic におけるトリガーロジックの開発

5.1　$d\eta$、$d\phi$、$d\theta$ によるカットロジック

まず考案されたのが、EI/FI コインシデンスと同様の位置のマッチングを用いたロジックである。NSW は EI/FI に比べ $|\eta|$ の大きい範囲までカバーするため、その領域でのフェイクトリガーの大幅な削減が期待される。TGC BW でのヒット位置と NSW のヒット位置から η、ϕ についてそれぞれ $d\eta = \eta_{BW} - \eta_{NSW}$、$d\phi = \phi_{BW} - \phi_{NSW}$ を定義し、これらに η の範囲毎に一定のカット閾値を設けることで Inner Coincidence を取り、フェイクトリガーを抑制する。$d\theta$ では全領域で $d\theta < 0.015$ の条件を設ける。これらの閾値はシミュレーションデータから、トリガー効率を 100% に保つように定めた。表 5.1 にカット閾値を示した。このような閾値を設けた場合に、閾値 20 GeV のミューオントリガーの RoI の η 分布は図 5.1 のようになった（図 2.21 の縦軸の対数を取ったもの）。NSW からの情報を用いてイベントカットを行うロジックを加えることで、特に $1.9 < |\eta| < 2.5$ の部分で大きくトリガーレートを抑制できていることがわかる。これは主にフェイクトリガーを排除できたことによるレート削減で、この結果ミューオントリガー全体のトリガーレートはルミノシティ 3.0×10^{34} cm^{-2}s^{-1} で 13 kHz になる [3]。

表 5.1　カット条件

	L1MU20			L1MU10		
$\|\eta\|$	$d\eta$ cut 値	$d\phi$ cut 値	$d\theta$ cut 値	$d\eta$ cut 値	$d\phi$ cut 値	$d\theta$ cut 値
1.3 ∼ 1.5	0.05	0.06	0.015	0.075	0.08	0.015
1.5 ∼ 1.7	0.05	0.06	0.015	0.075	0.08	0.015
1.7 ∼ 1.9	0.05	0.06	0.015	0.09	0.08	0.015
1.9 ∼ 2.1	0.05	0.06	0.015	0.12	0.08	0.015
2.1 ∼ 2.3	0.07	0.06	0.015	0.12	0.08	0.015
2.3 <	0.07	0.06	0.015	0.12	0.08	0.015

このロジックの実装後に予想されるトリガーの η 分布を図 5.1 の青色のヒストグラムで示した。赤色の部分は実際にデータからミューオンが再構成されたものの η 分布であり、$|\eta| > 1.3$ の領域のみを見ると青色のヒストグラムのうち約 90%を占めている。本物のミューオン以外の粒子によるフェイクトリガーは 10%ほどしか存在せず、現在と比較すると大きく削減することに成功している。一方、閾値である 20 GeV 以上の p_{T} を持つミューオンは図の緑のヒストグラムであり、上のロジックを実装した後でも全体の 10∼20% に留まっている。これはつまり、発行されたトリガーがのうち 80∼90%が、閾値以下の低い p_{T} を持つミューオンによるものであることを示す。

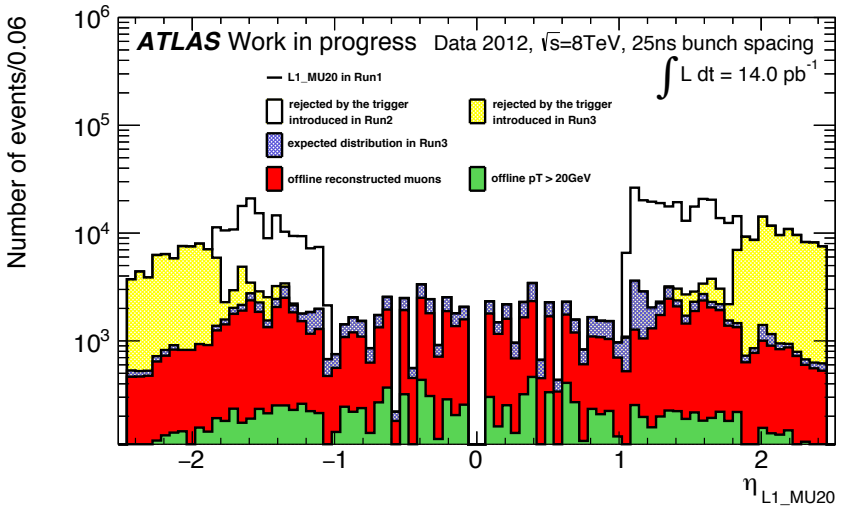


図 5.1　New Small Wheel の位置マッチングを導入した場合の各 p_{T} のトリガー効率

5.2　位置のマッチングを用いたロジック

5.2.1　位置マッチングロジックの概要

New Small Wheel の持つ高い位置分解能を活かして閾値より低い p_{T} のミューオンを削減することで、さらにレートを削減するロジックを考案した。図 5.2 に高い p_{T} 及び低い p_{T} のミューオンの描く飛跡のイメージ図を示す。

TGC BW では、L1 トリガーで用いることができる最小の読み出し単位が RoI であり、η 方向の分解能は約 ~ 0.02 である。一方、NSW での η 方向の分解能は ~ 0.005 と想定されている。BW での分解能が不足していることにより誤ってトリガー判定をしてしまうような低い p_{T} のミューオンのうち、NSW の分解能を用いれば排除できるものは一定数存在すると考えられる。NSW の高い分解能を活かして、M3-M1 間で行っている $dR : d\phi$ ロジックを M3-NSW 間の $d\eta : d\phi$ でも適用すれば、低い p_{T} のミューオンを選択的に削減することが期待できる。

この研究は過去になされている [17] [18] [19] が、この時は p_{T} 閾値 20 GeV のトリガーに対してのみであった。私はこの研究を確認するとともに p_{T} 閾値 10 GeV のトリガーに対しても適用し、その評価を行った。

5.2.2　位置マッチング Coincidence Window の定義

以上のロジックを実装するためには、NSW における $d\eta : d\phi$ の Coincidence Window （CW）を検出器の分解能に基づいて決定する必要がある。しかし、NSW 検出器はまだ ATLAS に導入されていないため、実際の衝突データを用いた評価は行うことができない。従って、モンテカルロシミュレーション（MC）を用いた手法で Coincidence Window を定め、トリガーの削減率を見積もる事にした。シミュレーションで NSW が配置される位置における粒子の飛跡情報を得て、それを NSW の分解能まで落としたものを NSW のヒット情報の代わりに用いた。TGC の情報もシミュレーションで同時に得られるので、これらのデータを用いて Coincidence Window を定義した。

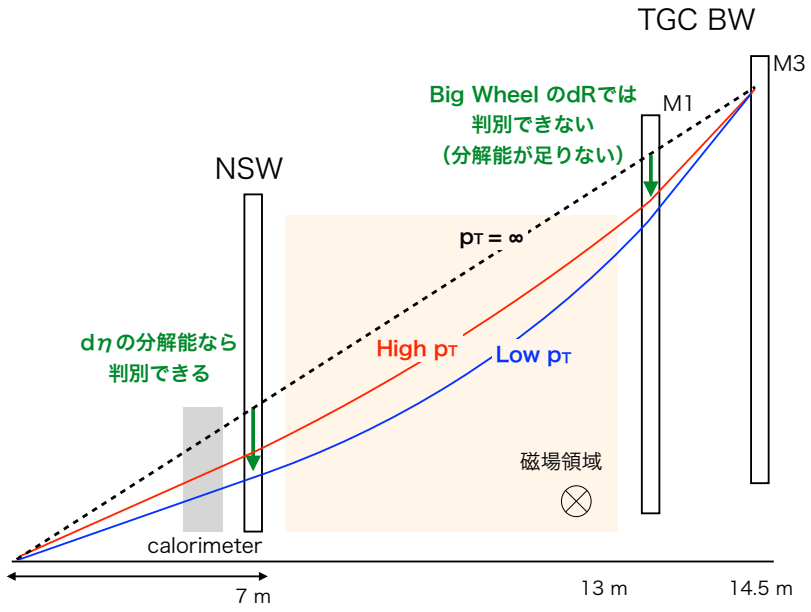


図 5.2　p_{T} が高い場合、低い場合のミューオンの飛跡のイメージ図。

Coincidence Window を定義するための MC のデータサンプルは、1 イベントに 1 本のミューオンのみを発生させる「シングルミューオン」サンプルを用いた。p_{T} が 10 GeV のイベントサンプルを 100 万イベント分、15 GeV を 100 万イベント分、20 GeV を 80 万イベント分、40 GeV を 60 万イベント分用いた。Coincidence Window は TGC BW の RoI 毎に定義するので、まず、ある RoI でミューオンが検出された時の、NSW の位置での $d\eta : d\phi$ 分布を求めた。図 5.3、図 5.4 に各 p_{T} のミューオンに対して得られた分布を示す。$d\eta : d\phi$ 分布は、p_{T} が 40 GeV のサンプルでは原点付近を通るのに対し、低い p_{T} では 2 つの成分に分かれることがわかる。これは、図 5.5 に示すように、ミューオンの電荷によって曲がる方向が違うことに起因する。p_{T} が小さいほど大きく曲がるため、この 2 成分も大きく分離する。また、磁場の分布が違うことによりこの分離の方向は RoI によって大きく異なるため、RoI 毎に最適な Coincidence Wondow を定める必要があることがわかる。

これらの分布を元に、トリガー効率が 99% となるように、すなわち Window に収まらないミューオンの数が 1%以下になるように、Coincidence Window を定めた。p_{T} 閾値 20 GeV のトリガーのための Coincidence Window 作成の際は、20 GeV 及び 40 GeV のサンプルのヒット分布を合わせ、それらのエントリーの 99% 以上を含むように、LUT を定めた。p_{T} 閾値 10 GeV のトリガーのための Coincidence Window を定める際は、10~40 GeV のサンプル全てを用いた。それぞれで定めたい CW はそれぞれ 20/10 GeV 「以上」のミューオンに対するものなので、閾値以上の高い p_{T} のミューオンサンプルを含める必要がある。図 5.6 ~ 図 5.7 に、得られたヒット分布と、そこから定義される Coincidence Window の例を示す。赤く塗られたマス目がトリガーを発行する Coincidence Window であり、$(d\eta, d\phi)$ がこの Coincidence Window に入っている場合にトリガーを発行する。

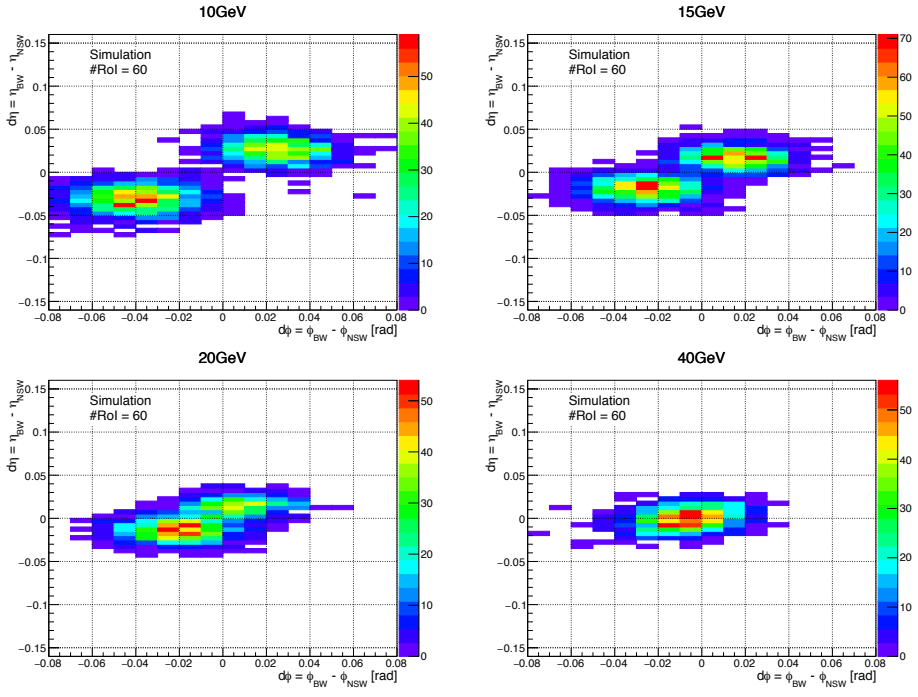


図 5.3　各 p_{T} のサンプルミューオンによるヒット分布。TGC の RoI 番号 60、$\eta \sim 1.46$ の RoI にヒットがあった場合の例。

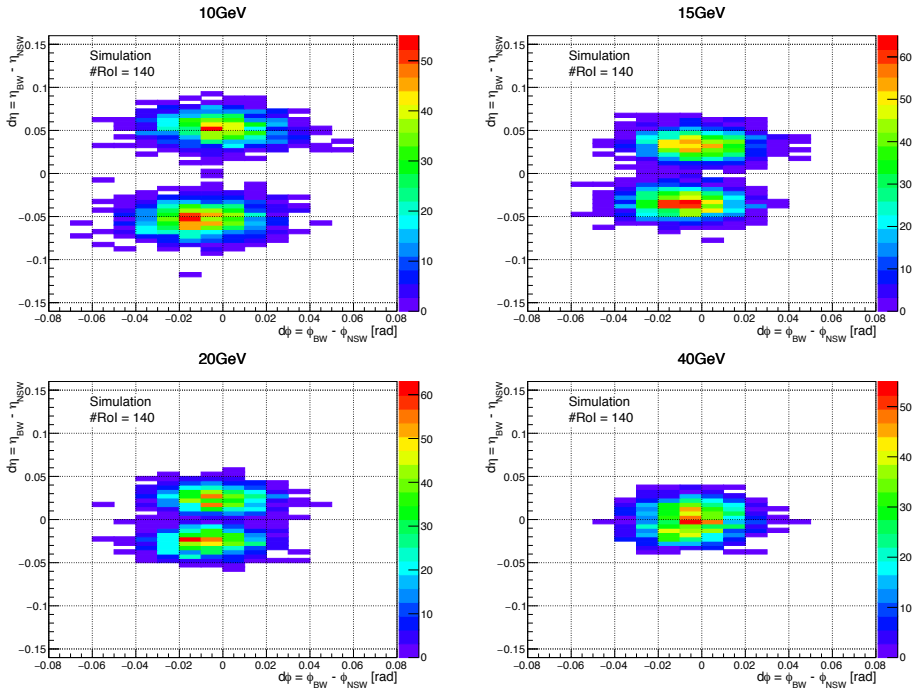


図 5.4　各 p_{T} のサンプルミューオンによるヒット分布。TGC の RoI 番号 140、$\eta \sim 1.93$ の RoI にヒットがあった場合の例。

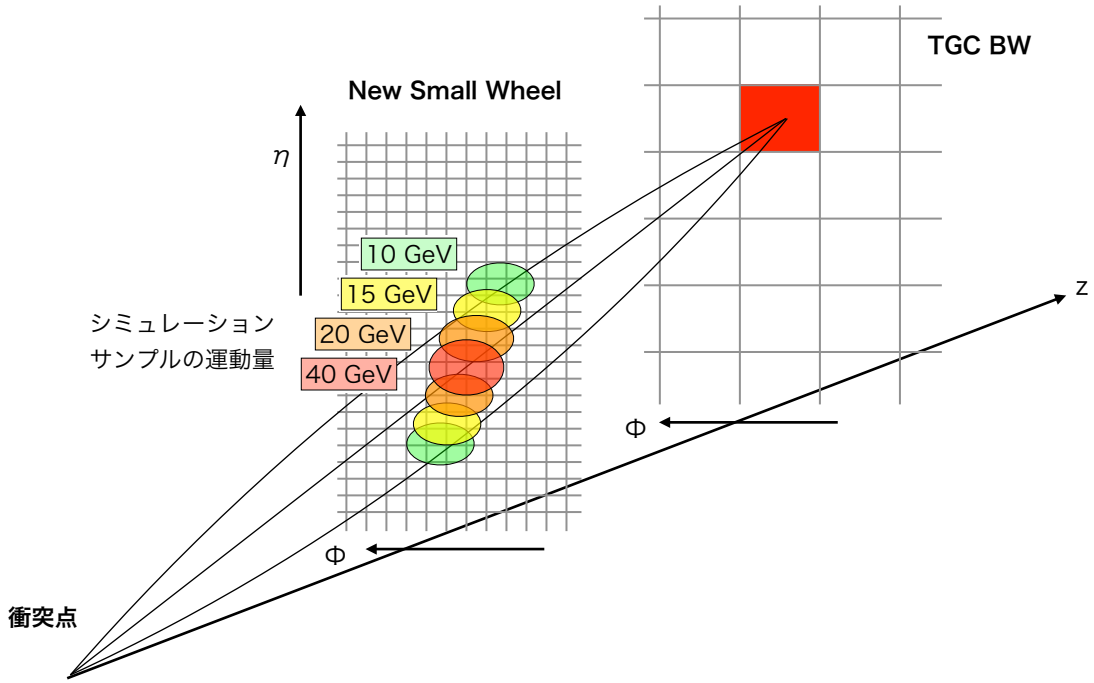


図 5.5　各 p_{T} のミューオンによる Coincidence Window のイメージ図。

5.2.3　位置マッチングの性能評価

新しく定めた Coincidence Window を使用した場合、低い p_{T} のトリガーをどれくらい削減できるかを評価した。評価には、Coincidence Window を定める際に用いたデータと異なる、シングルミューオンイベントを約 100 万イベント含む MC サンプルを用いた。このサンプルは 0 ～ 100 GeV で一様な p_{T} 分布、及び NSW に入射する $1.3 < |\eta| < 2.5$ で一様な η 分布をもつ。

p_{T} 閾値 20 GeV の L1MU20 と、p_{T} 閾値 10 GeV の L1MU10 に対して評価を行った結果をそれぞれ図 5.8、図 5.9、に示す。図中の L1MU20/10 + NSW(cut) は前節で述べた閾値を設定することによるカットロジックを導入した場合、L1MU20/10 + NSW(ϕ) は NSW の位置マッチングを要求した場合のトリガー効率を表す。各図の左上のプロットは全イベントに対するトリガー効率を、右上のプロットは TGC 単体の L1MU20/10 を通過したイベントに対するトリガー効率を示す。また、左下のプロットは NSW(cut) と比較した NSW(ϕ) のトリガー効率である。

NSW のコインシデンスを要求することで、L1MU20 と L1MU10 の両方に対しトリガー閾値より高い p_{T} のミューオンに対する効率は 99% 以上に保ちながら、閾値以下のミューオンを選択的に排除していることが分かる。また、カットベースのロジックを採用した場合に比べて、Coincidence Wondow を用いた場合の方が閾値以下の p_{T} のミューオンをより大きく削減できる。特に、L1MU10 に対しては Coincidence Window を用いたロジックを用いた場合、最も良いところでカットベースに対してさらに約 70% のトリガー候補の削減が可能であることがわかる。

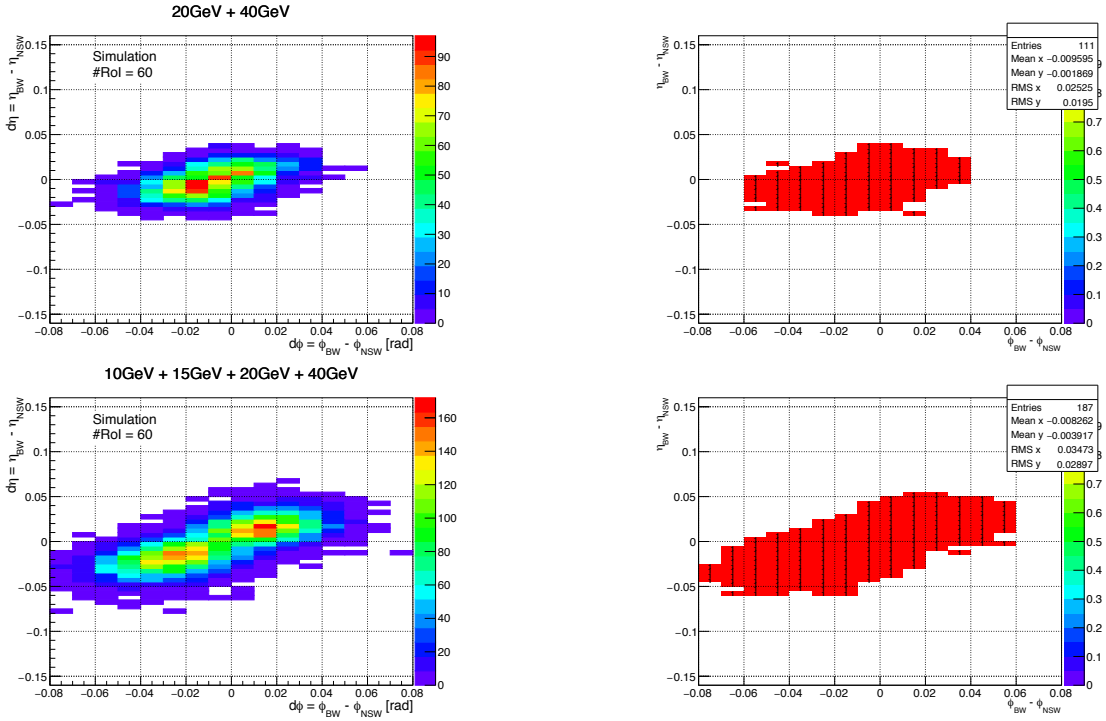


図 5.6　（左）RoI 60 番および 140 番における、p_{T}= 20 及び 40 GeV のミューオンによる $d\eta : d\phi$ のヒット分布。（右）ヒット分布の 99% を含むように定義した $d\eta : d\phi$ の Coincidence Window。

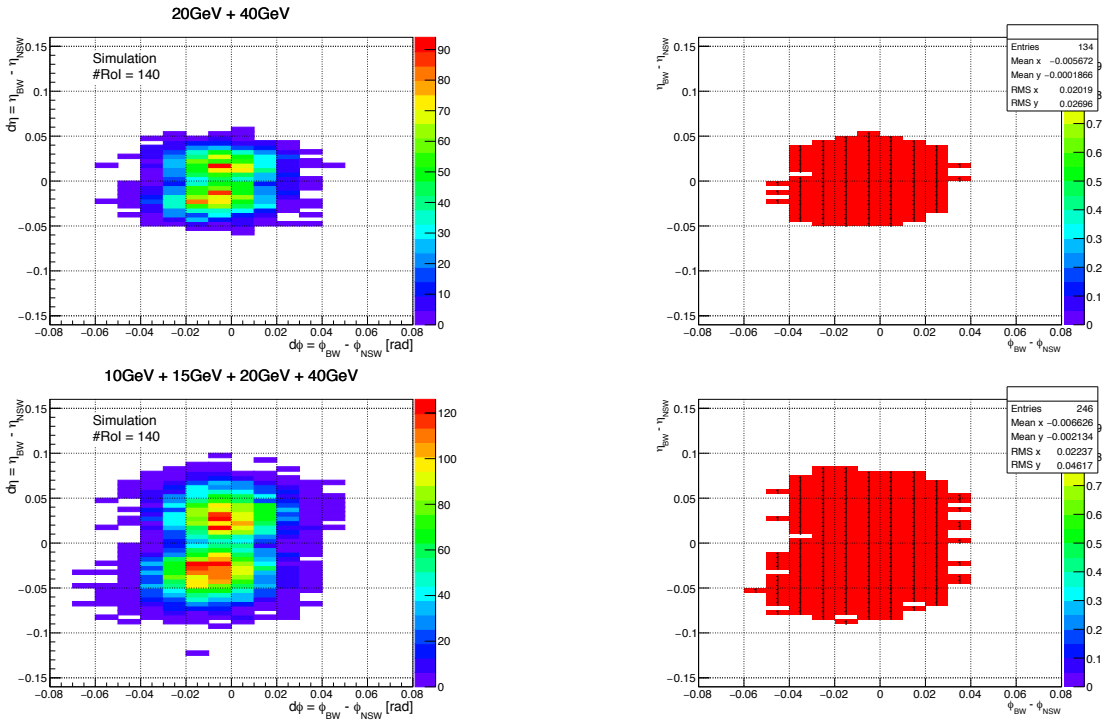


図 5.7　（左）RoI 60 番および 140 番における、p_{T}= 10、15、20 及び 40 GeV のミューオンによる $d\eta : d\phi$ のヒット分布。（右）ヒット分布の 99% を含むように定義した $d\eta : d\phi$ の Coincidence Window。

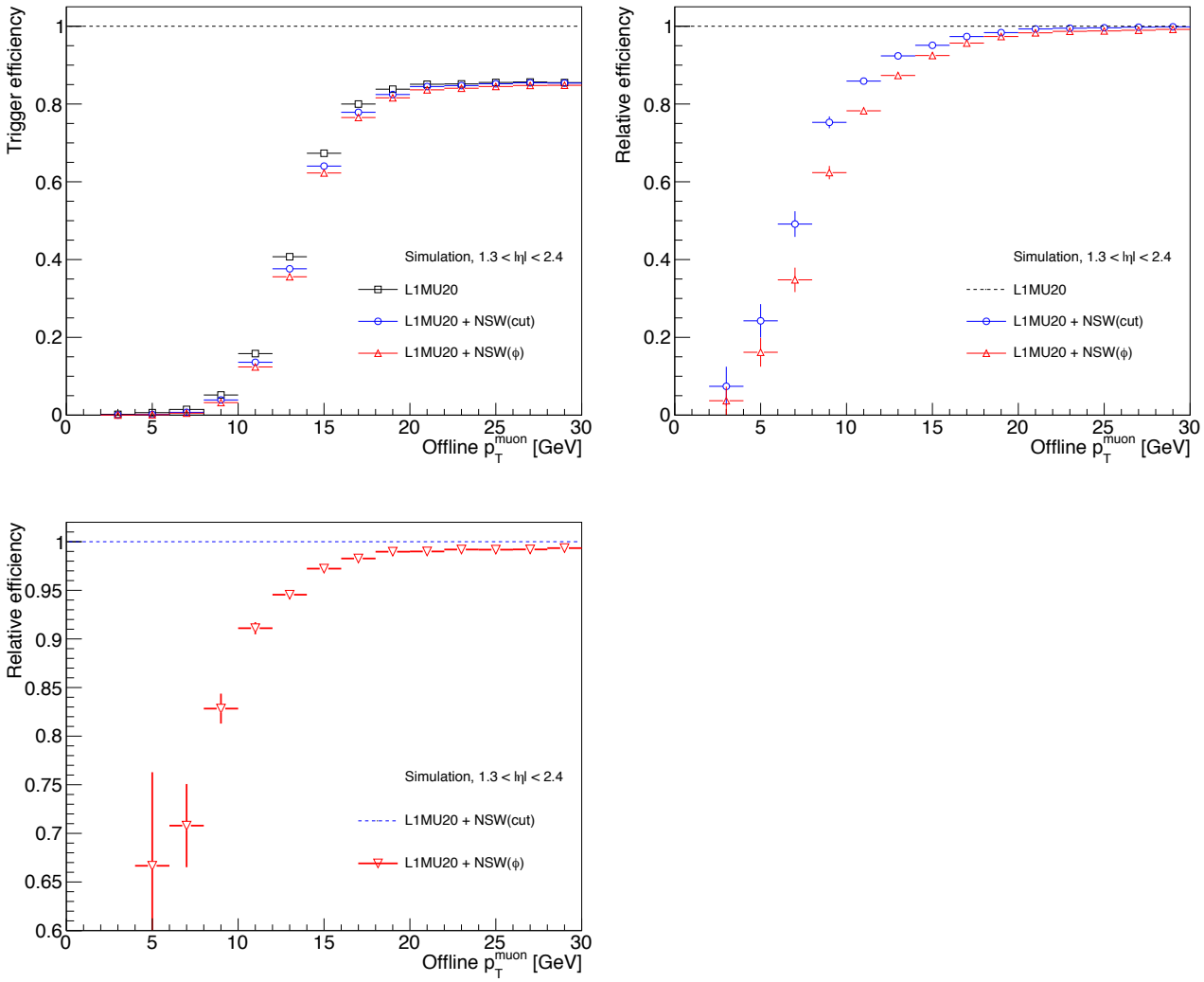


図 5.8　New Small Wheel コインシデンス前後の L1MU20 の各 p_{T} のミューオンに対するトリガー効率。左上が、すべてのミューオンに対するトリガー効率を示す。右上は TGC 単体の L1MU20 を通過したイベントに対するトリガー効率。左下は、NSW(cut) を要求した場合のトリガー効率に対する、NSW(ϕ) を要求した場合のトリガー効率の比を表す。

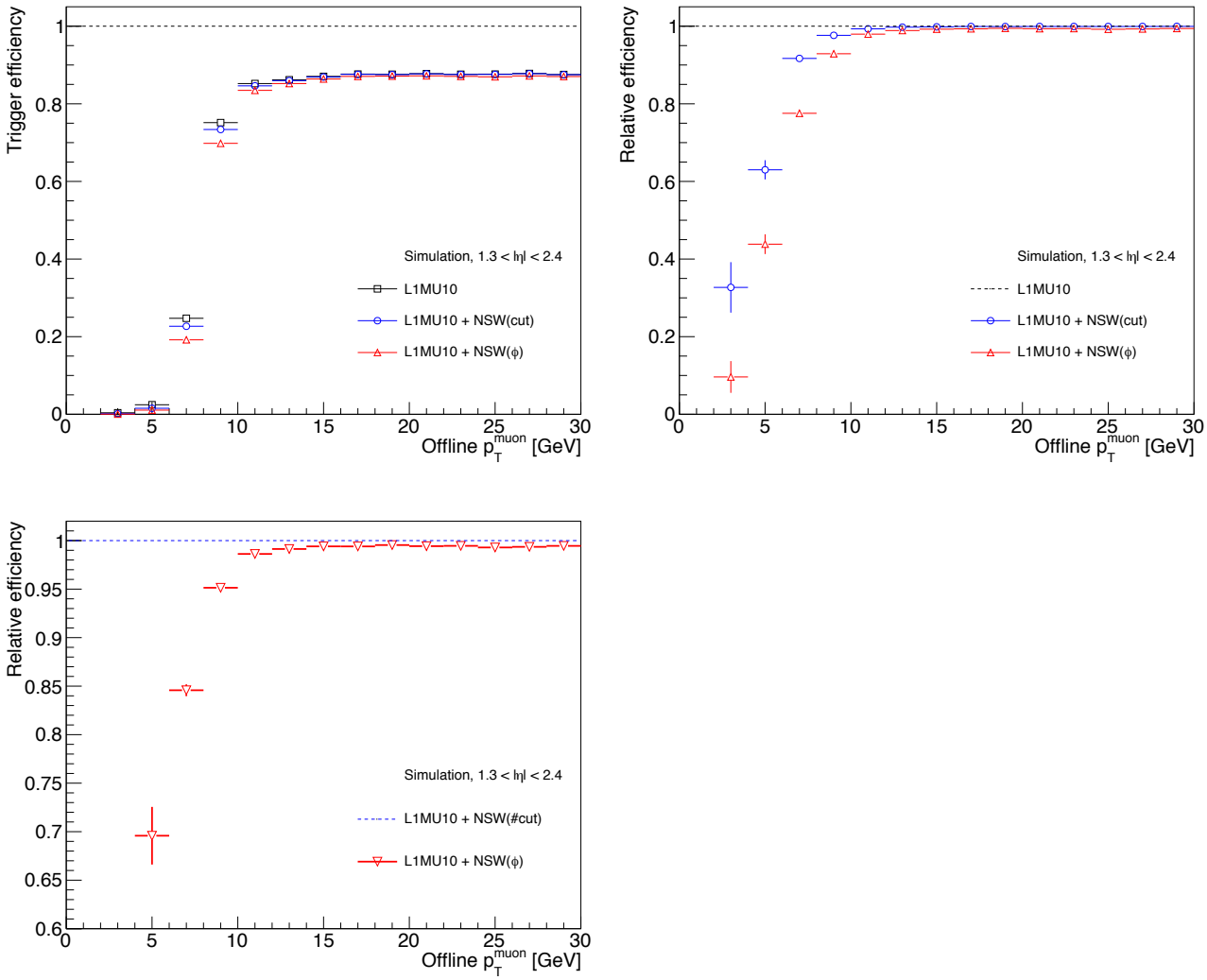


図 5.9　New Small Wheel コインシデンス前後の L1MU10 の各 p_{T} のミューオンに対するトリガー効率。左上が、すべてのミューオンに対するトリガー効率を示す。右上は TGC 単体の L1MU10 を通過したイベントに対するトリガー効率。左下は、NSW(cut) を要求した場合のトリガー効率に対する、NSW(ϕ) を要求した場合のトリガー効率の比を表す。

5.3　角度マッチングを用いた新ロジック

5.3.1　角度マッチングロジックの概要

New Small Wheel は高い位置分解能に加えて ∼ 1 mrad の高い角度分解能を持つことが特徴である。角度情報を用いたトリガーロジックは現在のミューオントリガーシステムでは実装されていないため、新しいロジックを開発する必要がある。

図 5.10 に、$d\theta$ を用いたロジックの概念図を示した。これまでに述べたトリガーロジックは、すべてミューオンが検出器の中央から発生することを仮定している。その上で、そこからまっすぐに TGC BW の最外層の RoI まで引いた直線との位置の差を見ることで p_{T} の判断を行いトリガー判定をしている。一方で、実際には衝突点は z 方向に約 10 cm の広がりを持ち、さらに発生したミューオンは内部の物質（主にカロリーメメータ）と多重散乱をするため、まっすぐに磁場に入射しない。図 5.10 に示すように、高い p_{T} のミューオンが実際に残すヒット情報と、衝突点・多重散乱の影響を受けた低い p_{T} のミューオンが残すヒット情報は、位置の情報を見ているだけでは同じように見えてしまう。ここで私は、ヒット位置に加えて磁場に入射する際の角度情報 $d\theta$ を用いればこの 2 つを分離できることに注目した。以下では、角度情報を利用したロジック実装の詳細について述べる。

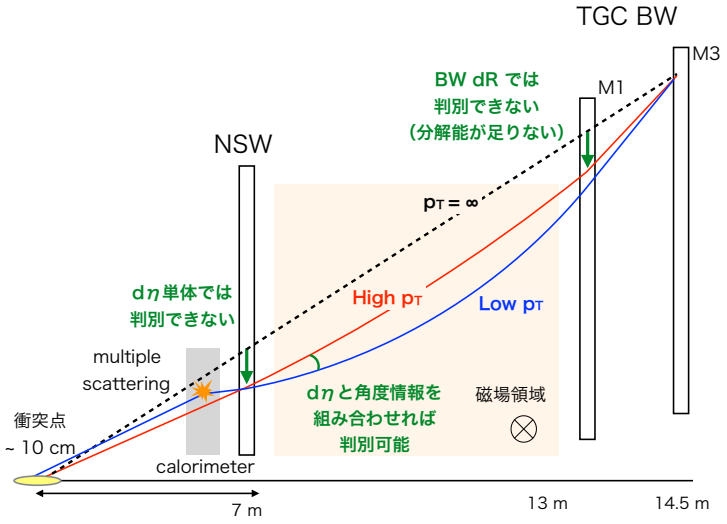


図 5.10　各 p_{T} のミューオンの描く飛跡のイメージ図。

5.3.2　角度マッチング Coincidence Window の定義

前節で述べた通り、角度の情報 $d\theta$ と NSW でのヒット位置 $d\eta$ の情報を組み合わせることで低い p_{T} のミューオンを分離しやすくなると考えられる。図 5.11 及び図 5.12 に、2 つの異なる RoI に対する $d\eta$ と $d\theta$ のヒット分布を示す。用いたサンプルは 5.2.2 節で用いたものと同じである。これらの分布を見ると、$d\theta$ は $d\eta$ と明確な相関を持ち、$d\eta$ と組み合わせることで p_{T} の測定を補助することができると考えられる。

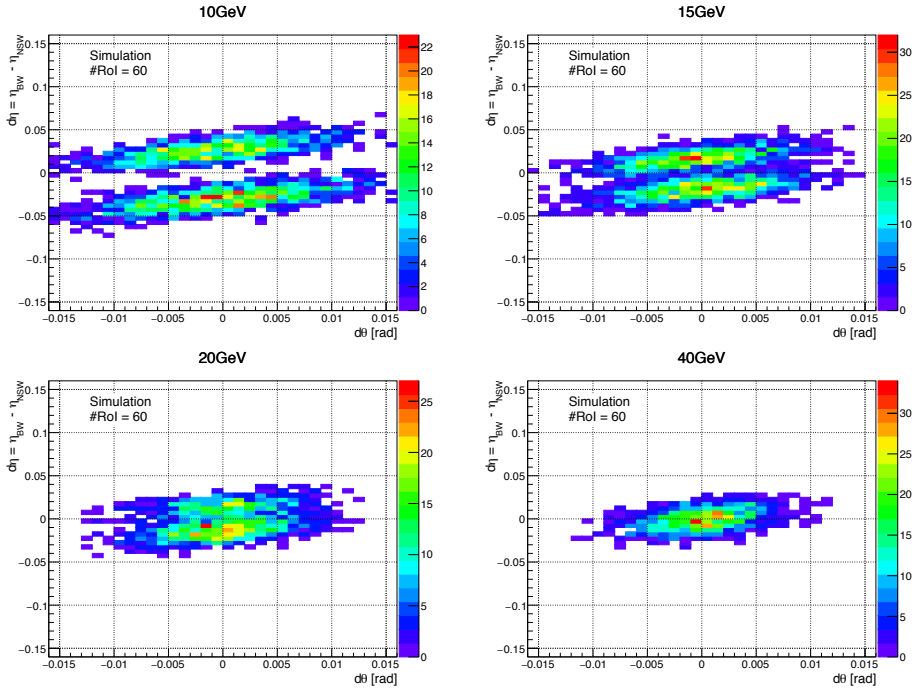


図 5.11　各 p_{T} のサンプルミューオンによるヒット分布。この例は TGC の RoI 番号 60、$\eta \sim 1.46$ の RoI にヒットがあった場合。

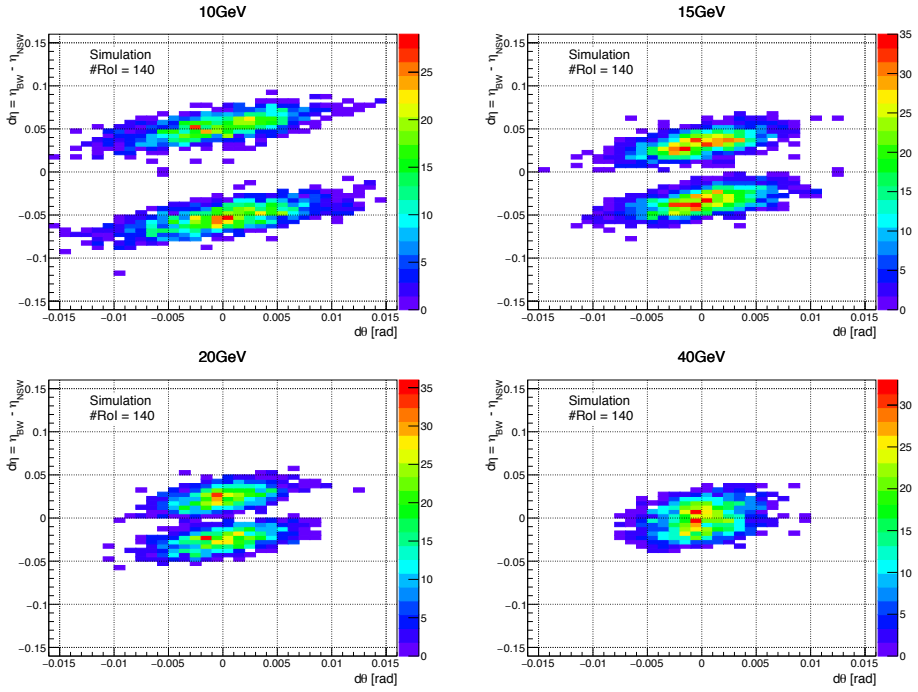


図 5.12　各 p_{T} のサンプルミューオンによるヒット分布。この例は TGC の RoI 番号 140、$\eta \sim 1.93$ の RoI にヒットがあった場合。

このヒット分布から、99% 以上のヒットを含むような Coincidence Window を定めた。図 5.13 ～ 図 5.14 に、20 GeV 及び 10 GeV の各 p_{T} 閾値のトリガー用の Coincidence Window と、それを定める際に用いたヒット分布を示した。

5.3.3　角度マッチングの性能評価

$d\theta$ を用いた新しい Coincidence Window を使用した場合の、低い p_{T} のミューオンに対するトリガーの削減能力を評価した。5.2.3 節と同じく、p_{T} 及び η 分布が一様なシングルミューオンサンプル約 100 万イベントを用いた。p_{T} 閾値 20 GeV の L1MU20 と、p_{T} 閾値 10 GeV の L1MU10 に対して評価を行った結果をそれぞれ図 5.15、図 5.16、に示す。L1MU20/10 + NSW(θ) は NSW の角度マッチングを要求した場合の各 p_{T} に対するトリガー効率である。各図の左上のプロットは全イベントに対するトリガー効率を、右上のプロットは TGC 単体の L1MU20/10 を通過したイベントに対するトリガー効率を示す。左下のプロットは NSW(ϕ) を通過したイベントに対すして NSW(θ) を要求した場合ののトリガー効率である。

閾値以上のミューオンに対する効率は、角度マッチングを要求しても要求前の 99% に保たれている。閾値以下の p_{T} のミューオンに関しては、角度マッチングを要求することでトリガー割合を削減することができた。位置マッチングのみを要求する場合に対して、p_{T} の低い領域でさらに最大 20%の削減を達成した。

5.4　トリガーレートの評価

2016 年に LHC で取得したデータを用いて、考案したトリガーロジックを実装した場合に予想されるトリガーレートの評価を行った。その結果が図 5.17、5.18、である。それぞれ、p_{T} 閾値 20 GeV、10 GeV のトリガーに対応する。横軸は再構成されたミューオンの p_{T} で、図中の黒い点線は現在のトリガーロジックで L1MU20/10 を発行した場合の分布である。現行のトリガーロジックでは特に L1MU20 において、誤って低い p_{T} のミューオンに対しトリガーを発行している場合が多いことが分かる。赤、青、及び緑のヒストグラムはそれぞれ NSW のカット、位置マッチング、角度マッチングを用いた場合の p_{T} 分布である。

この分布から求めたレート削減を表 5.2 にまとめる。p_{T} 閾値 20 GeV のトリガーに関しては、$d\theta$ を要求した場合では、現行の L1MU20 に対して 41%、カットベースのロジックに対して 15%、位置マッチングに対して 7.8% のレート削減が見込まれる。p_{T} 閾値 10 GeV のトリガーではそれぞれ 27%、16%、5.6% の削減となる。また、この時の閾値以上のミューオンに対するトリガー効率は現行の L1MU20/10 に対して 98% 以上であった。$d\eta : d\phi$ 及び $d\eta : d\theta$ の CW 作成の際に、それぞれ 99% のトリガー効率を要求したことから、これら両方のコインシデンスを要求すると約 98% になると予想されるため、得られたトリガー効率は想定通りである。

カットベースのロジックを導入した場合の L1MU20 のトリガーレートは 13 kHz と見込まれているため、そこから $d\theta$ を用いた場合のトリガーレートを計算できる。ここで、このレート削減は NSW がカバーする $1.3 < |\eta| < 2.4$ のみに適用であるという点に注意する。図 5.1 より、この領域の占めるトリガーレートの割合が $|\eta|$ を含む全体のトリガーの約 50 % だとし、さらにフェイクトリガーの寄与を無視すると、トリガーレートの削減は L1MU20 全体のうち 7.5% となる。従って、角度マッチングを採用することで、L1MU20 のトリガーレートは約 12 kHz になると想定される。

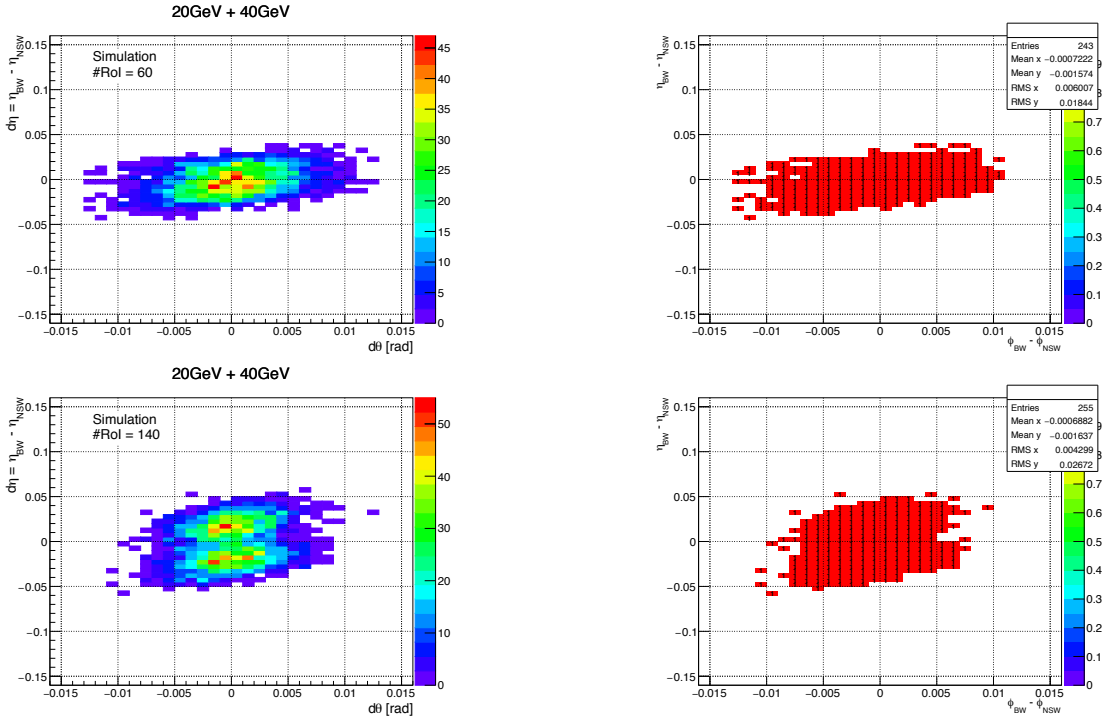


図 5.13　（左）RoI 60 番及び 140 番における、p_{T}= 20 及び 40 GeV のミューオンによる $d\eta : d\theta$ のヒット分布。（右）ヒット分布の 99% を含むように定義した $d\eta : d\theta$ の Coincidence Window。

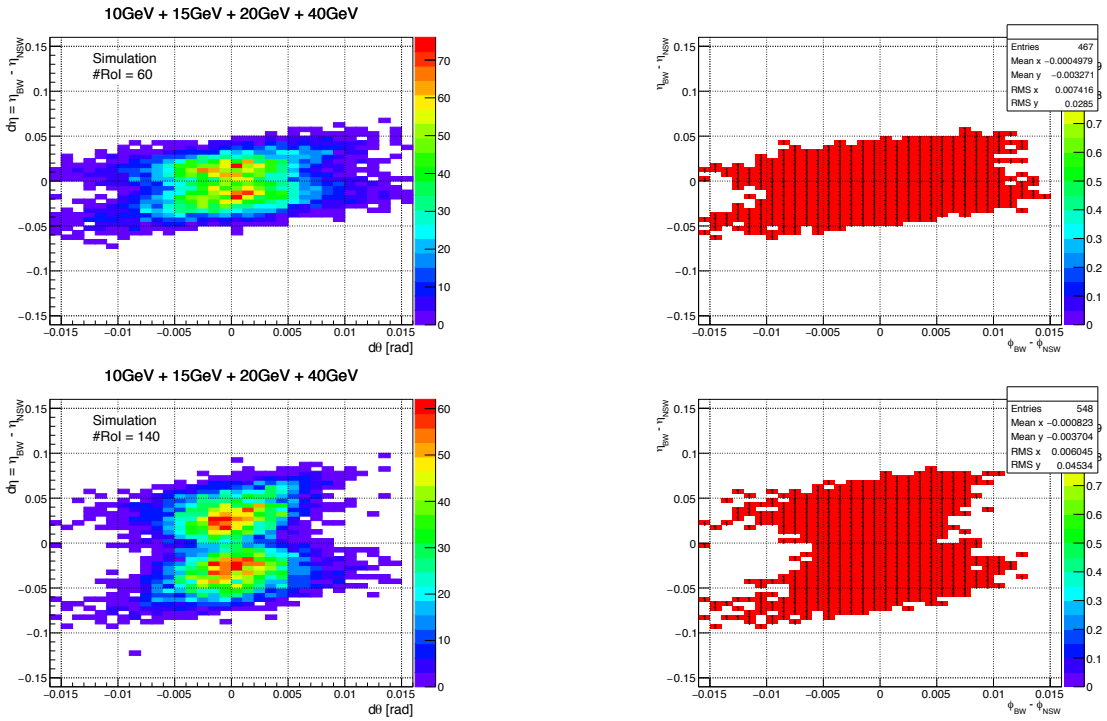


図 5.14　（左）RoI 60 番および 140 番における、p_{T}= 10、15、20 及び 40 GeV のミューオンによる $d\eta : d\theta$ のヒット分布。（右）ヒット分布の 99% を含むように定義した $d\eta : d\theta$ の Coincidence Window。

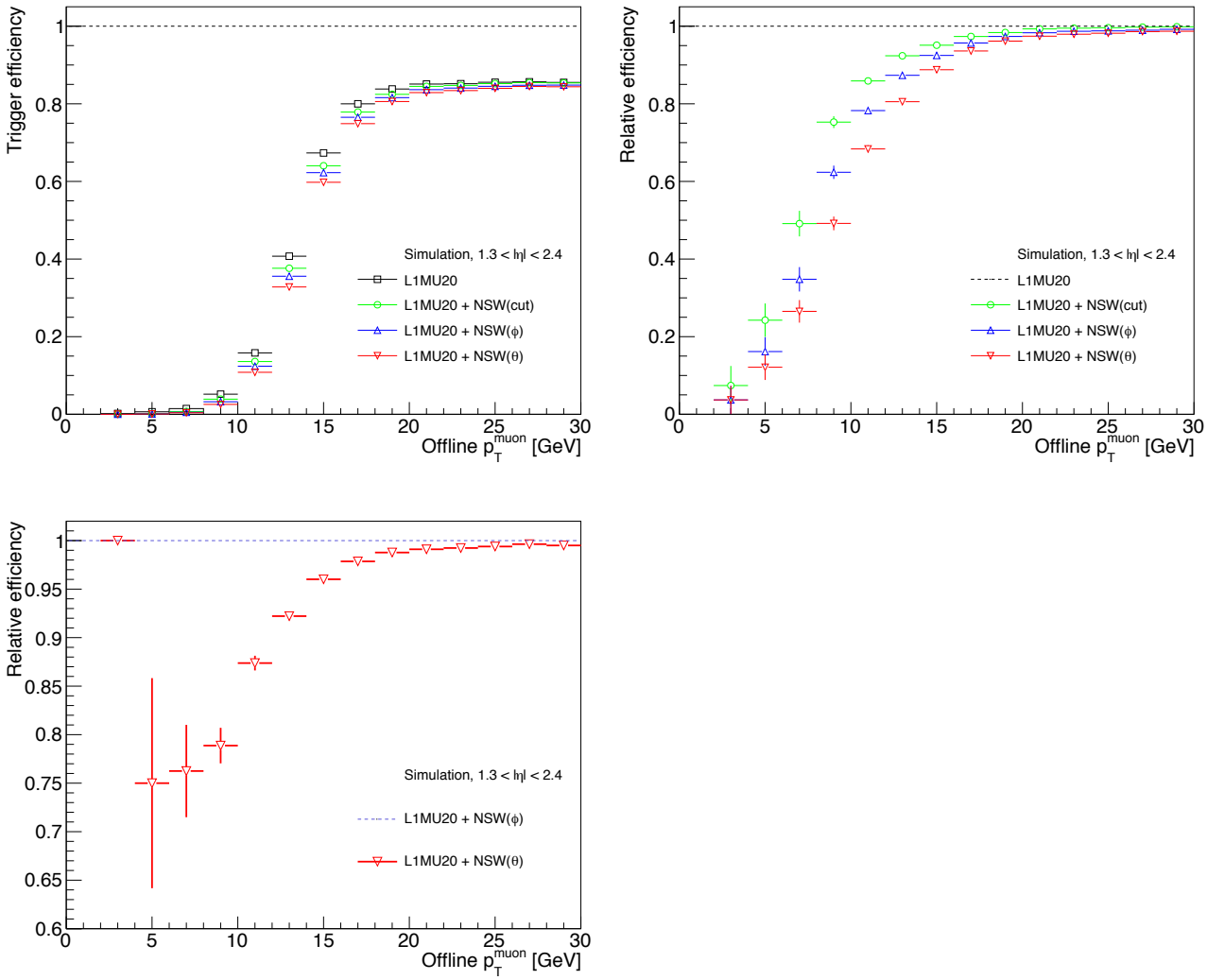


図 5.15　New Small Wheel コインシデンス前後の L1MU20 の各 p_{T} のミューオンに対するトリガー効率。左上が、すべてのミューオンに対するトリガー効率を示す。右上は TGC 単体の L1MU20 を通過したイベントに対するトリガー効率。左下は、NSW(ϕ) を通過したイベントに対して、NSW(θ) を要求した場合のトリガー効率。

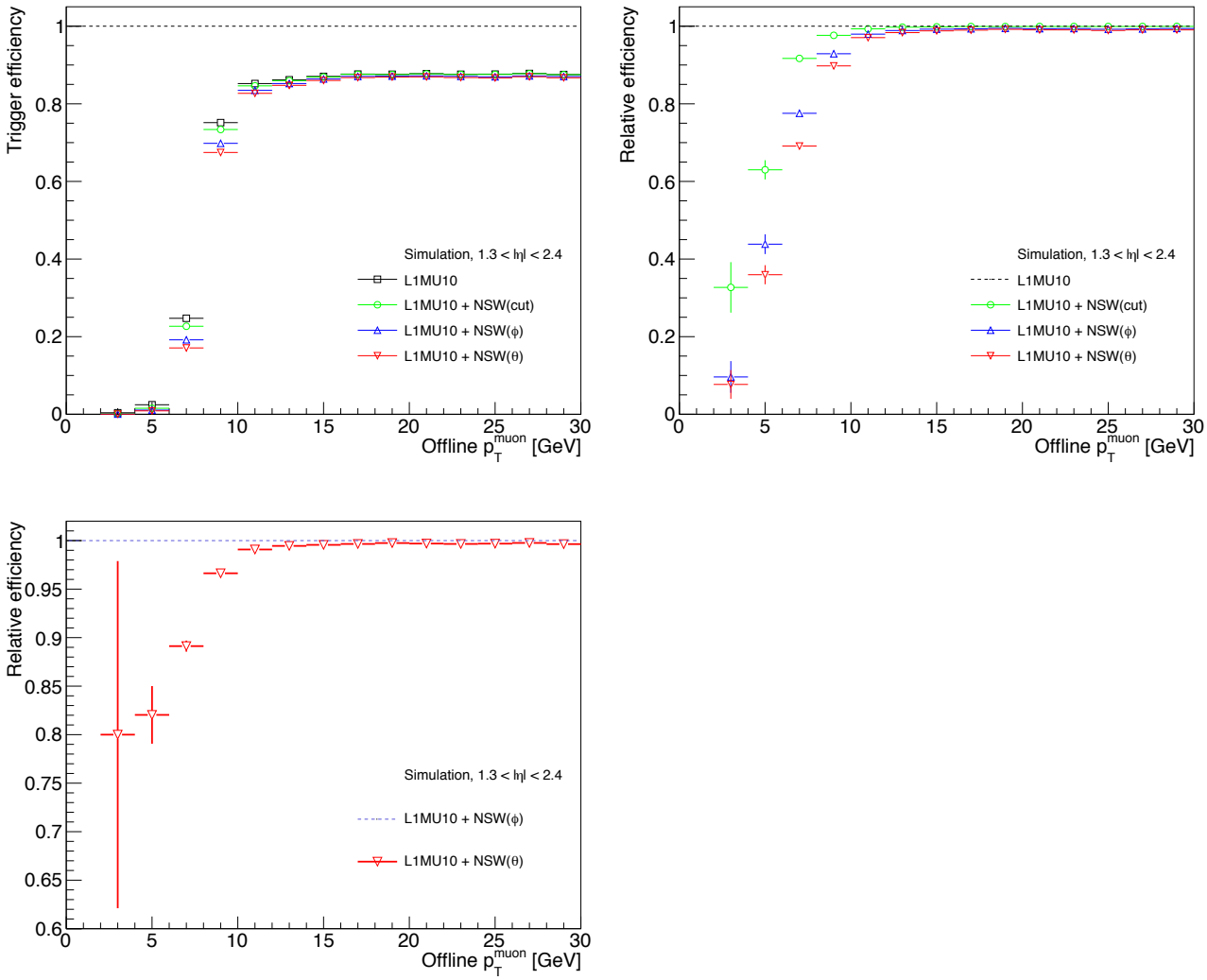


図 5.16　New Small Wheel コインシデンス前後の L1MU10 の各 p_{T} のミューオンに対するトリガー効率。左上が、すべてのミューオンに対するトリガー効率を示す。右上は TGC 単体の L1MU10 を通過したイベントに対するトリガー効率。左下は、NSW(ϕ) を通過したイベントに対して、NSW(θ) を要求した場合のトリガー効率。

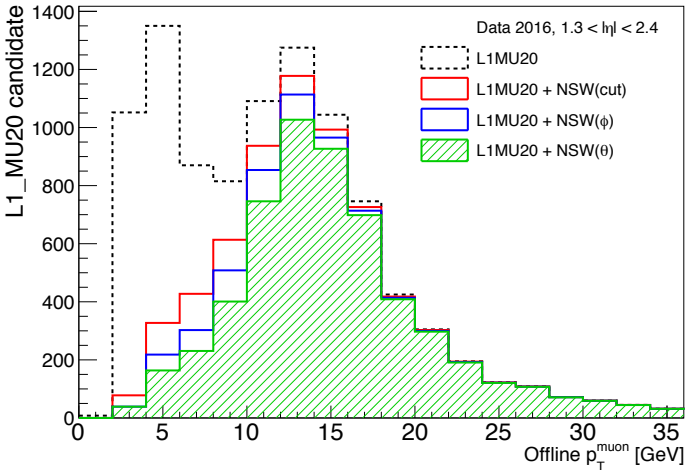


図 5.17　New Small Wheel の角度マッチングを導入した場合に発行される L1MU20 の p_{T} 分布

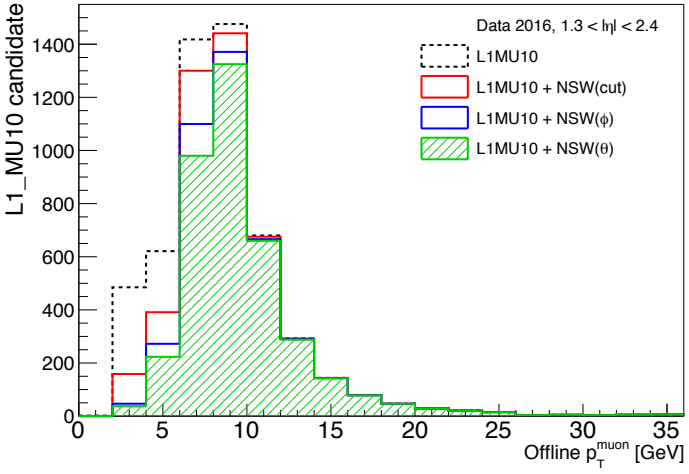


図 5.18　New Small Wheel の角度マッチングを導入した場合に発行される L1MU10 の p_{T} 分布

表 5.2　$d\eta : d\theta$ コインシデンスを導入することで削減されるレートの割合。

	NSW(θ)/(No upgrade)	NSW(θ)/NSW(cut)	NSW(θ)/NSW(ϕ)
L1MU20	−41%	−15%	−7.8%
L1MU10	−27%	−16%	−5.6%

5.5　Firmware への実装

前節までに考案したトリガーロジックが実際に NewSL ボード上の FPGA に実装可能かどうかを検証する必要がある。既に述べたように、Coincidence は BRAM という大規模なメモリを用いて、Look Up Table（LUT）として、あらかじめ FPGA 上に実装しておく必要がある。ロジックを実装するために必要な BRAM 容量を考慮しつつ、トリガー判定の性能を落とさずに最適なロジック構造を最適化することを考える。

まず、BRAM リソースについて説明しておく必要がある。3 章で説明した通り、NewSL で採用している XC7K325T は 445 個の 36 Kb BRAM を有している（表 3.2）。また、1 つ上位モデルの XC7K410T は 795 個の 36 Kb BRAM を有している。36 Kb BRAM とは、名前の通り 36 Kb のデータを保存しておくことができるメモリである。例えば、アドレス 10-bit、データ 6-bit のメモリを BRAM を用いて実装する場合、2^{10} パターンのアドレスに対して、6-bit の出力パターンを保存しておくことになるのでこの時保存されるデータ量は 1024× 6-bit = 6 Kb となる。従って、このメモリは 36 Kb BRAM を 1/6 個分を使って実装することができる。実際は 36 Kb BRAM を 1/6 個だけ用いることはできず、最小単位は 1/2 個（18 Kb BRAM）である。つまり、このメモリを FPGA 上に実装する際に用いる BRAM 資源は 1/2 個となる。

これまで述べたトリガーロジックを自然に実装するなら、図 5.19 のようになる。$d\eta$、$d\phi$ 及び $d\theta$ は、前節でのヒット分布が収まるようにそれぞれ 6-bit、4-bit、5-bit 分用意している。また、RoI ごとに適切にデコードされているとしている。BW のコインシデンスが取れたものに対し、次の LUT で $d\eta : d\phi$ コインシデンスを取り、その後でさらに $d\eta : d\theta$ コインシデンスを取る、というロジック構成である。このとき、アドレス 14-bit と 15-bit の巨大な LUT を 2 つ実装しなければならないため、実はこれは悪い実装例である。実際、例えばアドレス 14-bit、データ 4-bit の LUT を 1 つおくだけで 2^{14}×4-bit = 64 Kb となり、BRAM を 2.5 個使うことになる。RoI は 2 トリガーセクターで最大 296 個なので、この時点で 2 個 × 296 RoI = 740 個となり、FPGA のもつ資源を全て使っても足りない計算となる。仮に XC7K410T にアップグレードした場合でも、他のロジックや読み出し回路に使う分を考慮すると足りなくなってしまう。

そこで私は、BRAM 資源を有効活用するために図 5.20 のように実装することとした。この実装方法では、NSW の 2 つのコインシデンスは BW のコインシデンスが取れたかどうかの結果を受け取らない。代わりに、NSW コインシデンスは「BW が仮にコインシデンスが取れていたら」という前提でコインシデンス処理を行う。そのようにして判断された 3 種類の p_{T}（BW コインシデンス、$d\eta : d\phi$ コインシデンス、$d\eta : d\theta$ コインシデンス）を、最終段の 12-bit の LUT で初めて統合し、「BW のコインシデンスを満たしている」かつ「NSW のコインシデンスを満たしている」ということを要求する実装にした。この時に消費する BRAM 資源を計算する。アドレス 12-bit、データ 4-bit のメモリ は 2^{12}×4-bit = 16 Kb のデータを保持するため、BRAM 0.5 個で実装可能である。アドレス 10-bit 及び 11-bit の場合も BRAM の最小単位が 0.5 Block であることを考えると、使用する BRAM は 0.5

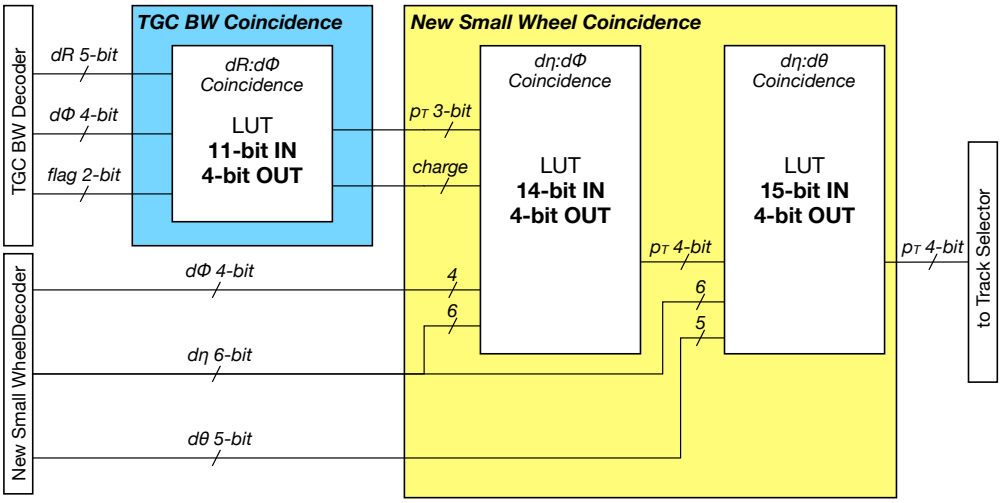


図 5.19　Trigger Firmware 1 RoI 分のブロック図の例。BW のコインシデンスを通過したイベントに対し、NSW の位置のマッチングを取り、さらにそれを通過したものに角度のマッチングを要求する、というロジック。この実装の場合リソースを多く消費してしまう。

個となる。従って、1 RoI に使用する BRAM は 2 個であり、1 枚の NewSL は最大 296 RoI を担当するので全体では最大 592 個の BRAM を使用することとなる。1 つ上のモデルの FPGA なら 795 個の BRAM を有するので、この場合には実装可能である。

さらに私は、最小単位が 0.5 Block であることに着目して図 5.21 のように、2 つの RoI 分の Coincidence Window を 1 つの LUT に実装することとした。10-bit や 11-bit アドレスの LUT は情報量としては 0.5 Block = 18 Kb よりも少ない（それぞれ 4 Kb、8 Kb）が、0.5 Block 分の資源を使って実装されてしまう。図 5.20 とロジックの構造は変えないまま、2 RoI 分の情報を 1 つの LUT に保存することで資源をより有効に活用することができる。このとき、図 5.21 の 4 つの LUT で使う BRAM 資源はどれも最小単位の 0.5 Block なので、2 RoI で 2 個、1 RoI あたり 1 個の BRAM を用いて実装することができる。従って、296 RoI 担当する場合でも使うリソースは 296 個で、読み出し部分での資源利用率にもよるが FPGA をアップグレードしなくても実装可能である。

実際にトリガー Firmware 用の LUT を実装し、BRAM 資源の使用量を確認したのが、図 5.22 と図 5.23 である。図 5.22 は、読み出し Firmware のみを実装した際のリソースの使用率を示すグラフ及び使用されている数を示す表である。BRAM に注目すると、読み出し Firmware で 112.5 個（25%） 使用していることがわかる。図 5.23 は、読み出し Firmware に加えて LUT を 1 トリガーセクター分、つまり 148 RoI 分実装した後のリソースの使用率である。このとき、BRAM は 260.5 個使用しており、使用率は 61% となっている。また、読み出し Firmware のみの場合と比較すると 148 個増えており、RoI あたり 1 BRAM Block と、前述の見積もりと一致している。仮にもう 1 セクター分追加した場合は、さらに 148 個 BRAM が使用されるため、使用率は約 92% となる。

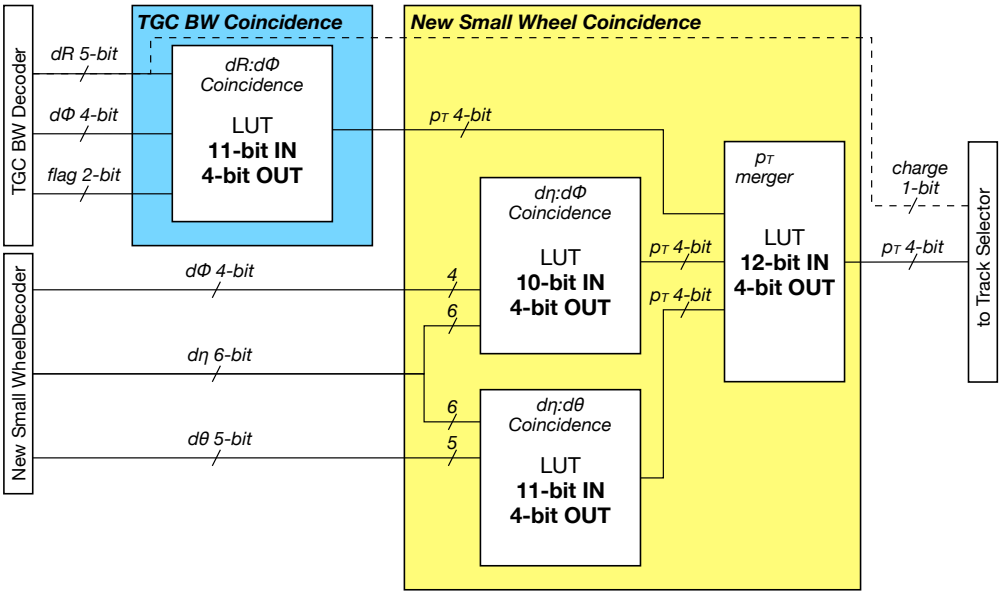


図 5.20　実装した Trigger Firmware 1 RoI 分のブロック図。全てのロジックで並列に p_{T} 計算を行い、最後に 3 種類の p_{T} からトリガー判定としての p_{T} を判断するというロジック構造である。

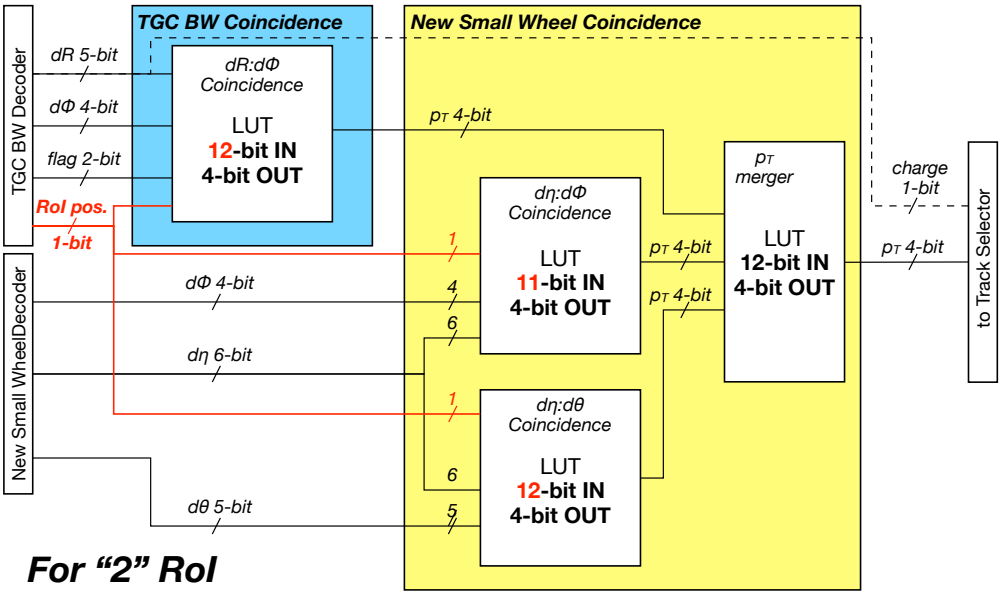
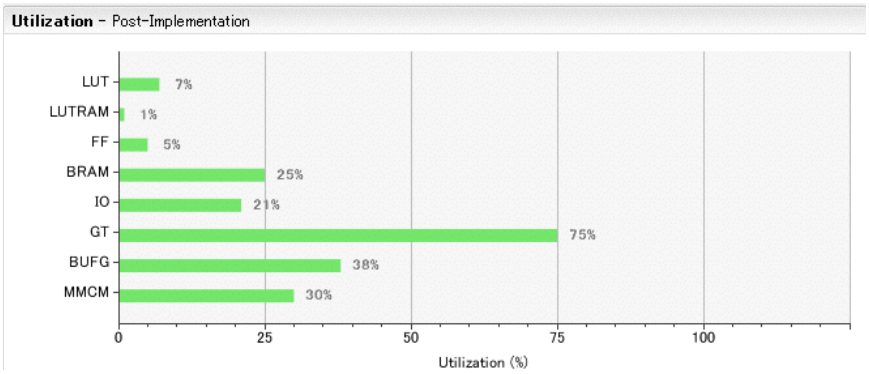


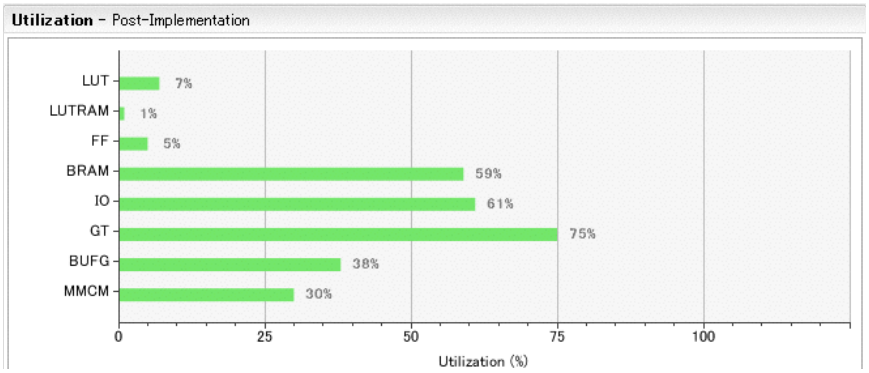
図 5.21　実装した Trigger Firmware 2 RoI 分のブロック図。2 つの RoI 分の Coincidence Window を 1 つの LUT に書き込んであるため、1 つ 1 つの LUT は 2 倍の大きさになる。



Utilization - Post-Implementation

Resource	Utilization	Available	Utilization %
LUT	13831	203800	6.79
LUTRAM	80	64000	0.13
FF	21688	407600	5.32
BRAM	112.50	445	25.28
IO	105	500	21.00
GT	12	16	75.00
BUFG	12	32	37.50
MMCM	3	10	30.00

図 5.22　Readout Firmware 実装段階でのリソースの使用割合。



Utilization - Post-Implementation

Resource	Utilization	Available	Utilization %
LUT	13695	203800	6.72
LUTRAM	80	64000	0.13
FF	21328	407600	5.23
BRAM	260.50	445	58.54
IO	306	500	61.20
GT	12	16	75.00
BUFG	12	32	37.50
MMCM	3	10	30.00

図 5.23　Trigger Firmware 1 トリガーセクター分実装後のリソースの使用割合。

第6章　ビームテストによる性能評価

6.1　目的

開発した New Sector Logic 及びその Firmware の試験として、ミューオンビームによる実際の検出器からの信号を用いたテストを行った。このビームテストは、我々Phase-1 アップグレードチームと Phase-2 アップグレードチームの合同で実施し、検出器上のフロントエンド回路を Phase-2 チームが、読み出し及びトリガーを Phase-1 チームが担当した。Phase-1 チームとしてのビームテストの目的は、主に

- 読み出し Firmware が正しく動作するか、
- GTX 通信で他のボードとエラーなく通信できるか、
- GTX 受信データを用いた Fixed Latency でのトリガー判定を実装できるか、

の 3 点である。本章では、これらテストの内容と結果について述べる。

6.2　セットアップ

ビームテストは CERN にある SPS 加速器を用いたテストを行う SPS beam facility で 2016 年の 10/19 - 11/4 の期間に行われた。ビームテスト全体のセットアップ図を図 6.1 に、検出器の写真を図 6.2 示した。

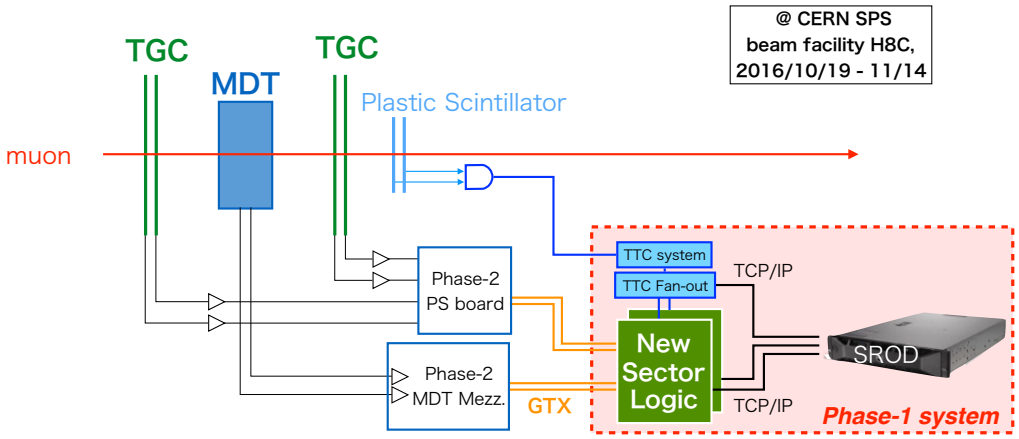


図 6.1　ビームテストの全体のセットアップ。

検出器はビームの上流から TGC、MDT、TGC、Plastic Scintillator（P.S.）の順に配置した。TGC は 2 枚とも実際に ATLAS で使用されているフォワード用の doublet チェンバーを用いた。MDT は実際に使用しているもの

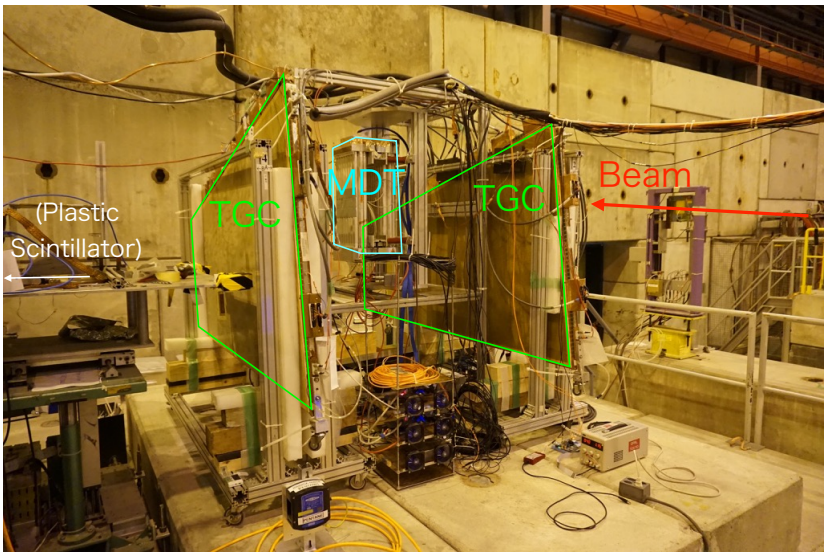


図 6.2　ビームテストの検出器の配置。上流側から TGC、MDT、TGC の順に配置した。写真の外の下流側に Plastic Scintillator が設置してある。

よりも短い、長さ 650 mm のチューブを 8 つ × 4 層からなるテスト用のチェンバーを用いた。Plastic Scintillator は約 10 cm × 10 cm の大きさのものを 2 枚使用した。Phase-2 チームが開発している TGC 用と MDT 用のフロントエンドボードは、検出器から信号を受信し、整形・処理・デジタル化した後に、GTX 通信で送信する。TGC の Phase-2 アップグレード用のフロントエンド回路は新 PS ボードと呼ばれ、TGC からのヒット情報を分解能を落とさずにそのまま GTX を用いた高速通信で送信することができる。MDT のフロントエンドボードは Phase-2 MDT Mezzanine と呼ばれ、FPGA 上に実装された高分解能 TDC を用いてドリフト時間計測を行う。Phase-1 チームはこれらの回路からの信号を New Sector Logic を用いて受信し、Run-3 用の読み出しシステム及びソフトウェアを用いて読みだす。また、Phase-1 チームはトリガーも担当するため、トリガー用に設置した P.S. の信号を受信し、トリガー発行も行う。2 枚の P.S. はビーム下流側に数十 cm の距離でビーム軸に並べて設置される。これらの信号のコインシデンスを取ったものをトリガー信号として用いる。

図 6.3 に、Phase-1 チームの担当する読み出し部分のセットアップを示す。NewSL は TGC 用に 1 枚、MDT 用に 1 枚用意し、それぞれ 8.0 Gbps、6.4 Gbps でフロントエンド検出器からデータを受信する。TTC システムのセットアップは詳細を省略するが、トリガー信号を受信し ID 情報と同期させて New SL に受け渡す。TTC システムへのトリガーの入力は NIM レベルで行えるようにした。入力として Plastic Scintillator のコインシデンスによる信号を与えるか New SL によるトリガー判定結果を与えるかによって、データ取得のトリガーソースを変更することができるようなセットアップになっている。TTC Fan-out モジュールはこの TTC 信号を NewSL に分配するためのモジュールである。NewSL で受信された情報は、トリガーが発行されたイベントのみ SROD に送信される。SROD ではイベントフォーマット及び ID 情報を確認し、1 イベント分の情報をまとめて保存する。TTC Fan-out は、TTC システムから受信した ID 情報を読み出し PC の SROD に送信しているため、この情報と NewSL の ID 情報を比べることで正しくデータ整形が行われているかを SROD でモニターすることができる。

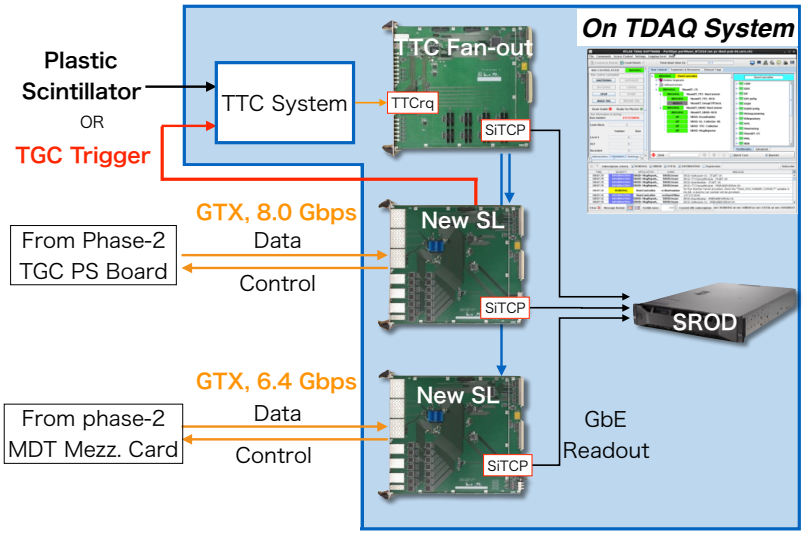


図 6.3　Phase-1 アップグレードチームが担当した読み出し・トリガー部分のセットアップ。

本ビームテストの目的は読み出し部分の性能の評価であるため、ビーム及び検出器の設定等については値を記すに留める。ビームは陽子を標的に衝突させて発生する π 中間子の崩壊によるミューオンビームである。標的からコンクリートの遮蔽体を挟んだ下流に検出器を設置したため、検出器に入射するビーム由来の粒子はほぼ全てミューオンであると考えることができる。ビームのサイズは検出器地点で ~10 cm であり、粒子の飛来レートは最大で数 100 Hz である。本章で主にデータ解析の対象となる TGC は、基本的はデータ取得期間は HV を 2850 V、ワイヤー及びストリップの閾値電圧はそれぞれ+50 mV、-50 mV の設定で運転した。ガスは CO_2 と n-Pentane の混合気体を用い、理想的な 55:45 の比率になるよう、ガスの流入量を管理した。これらは全て、現在 ATLAS 検出器で用いられている TGC の設定とほぼ同じである。MDT に関しては、共同で実験を行った MPI（マックス・プランク研究所）と Michigan 大学の研究チーム、及び日本の Phase-2 アップグレードチームの管理の下、運転が行われた。本章では MDT のデータについての解析は行わないため、細かい設定は割愛する。

6.3　NewSL に実装した Firmware

MDT Mezzanine と TGC PS ボードから受信するデータフォーマットを図 6.4 と図 6.5 に示した。MDT からは BCID 情報を持つデータを 1 BC あたりに 6 × 16-bit 分受信する。TGC からは 2 本のファイバーを使って 1 BC あたりに 16 × 16-bit = 256-bit の情報を受信する。また、TGC へはこのフォーマットに従って NewSL からコマンドを送信できるようにした。これにより、TGC フロントエンドボードを操作してストリップ・ワイヤに対する ASD の閾値の設定及び確認が可能となる。このフォーマットから、GTX の転送レートは MDT 側では 6.4 Gbps、TGC 側では 8.0 Gbps となる。

New Small Wheel 用に開発した Firmware の GTX 転送速度は 6.4 Gbps である。そのため、TGC PS ボード用の Firmware にはいくつか変更を加えた。GTX の設定を 8.0 Gbps 用に改めた上で、さらに、受信するデータの

量が大きいことを考慮してバッファーのサイズの変更も行った。MDT に関しては、検出器でのドリフト時間が最大で 700 ns と大きいため、ヒット情報が数 BC 以内に入らない。MDT の情報を全て読みだすためには、1 回のトリガーに対して 700 ns 分以上、つまり約 30 BC ほどのデータを読み出す必要がある。私が開発した Firmware では 16 BC 読み出しまでしか対応できないため、ここでは MDT 用に 64 BC 読み出し（∼1600 ns）まで対応できるように変更した。

MDT-mezzanine(Tx) → NSL(Rx)

Words (16-bit)	second byte	first byte
Word-0	BCID (last 8-bit)	comma(=bc)
Word-1	16-bit BCID data	
Word-2	16-bit BCID data	
Word-3	16-bit BCID data	
Word-4	16-bit BCID data	
Word-5	16-bit BCID data	
Word-6	16-bit BCID data	
Word-7	send back (status, etc.)	

図 6.4　MDT Mezzanine から受信するデータフォーマット。1BC あたり 16-bit の word を 8 つ、合計 128-bit 受信する。16-bit BCID data と書かれた部分に検出器からの情報が含まれる。

TGC-PS (Tx) → New SL (Rx) [2 fibers]

Words (16-bit)	second byte	first byte
Word-0	BCID (last 8-bit)	comma
Word-1	16-bit data	
Word-2	16-bit data	
Word-3	16-bit data	
Word-4	16-bit data	
Word-5	16-bit data	
Word-6	16-bit data	
Word-7	16-bit data	
Word-8	16-bit data	
Word-9	send back (status etc.)	

x2

New SL (Tx) → TGC-PS (Rx)

Words (16-bit)	second byte	first byte
Word-0	comma	comma
Word-1	(0x0) RESET ECR BCR L1A	(0x00)
Word-2	JTAG (4-bit) x 4	
Word-3	I2C (2-bit) x 2	
Word-4	Command, strobe	
Word-5	address	
Word-6	data	
Word-7	(0x0000)	
Word-8	full board ID	
Word-9	footer (0x0e0d)	

図 6.5　TGC PS ボードと NewSL の間のデータフォーマット。8.0 Gbps の通信で 1BC あたり 160-bit を送受信する。このうち、128-bit 分が検出器からの読み出し情報に当てられている。

トリガー部分に関しては、TGC から受信するデータを用いたトリガーロジックを実装した。TGC と NewSL の接続の模式図を図 6.6 に示した。ストリップ及びワイヤー情報はそれぞれ 1 層あたり 32 チャンネル、4 層分合わせて 128 チャンネルあり、新 PS ボードでは全チャンネルのヒットの有無をそのまま 256-bit 分、GTX で送信する。NewSL はこの情報を用いてトリガーロジックを作成することができる。私が実装したロジックは現在のワイヤー用の SLB ASIC に実装されたロジックを参考にして作成したもので、基本的には doublet のワイヤー 2 層 × 2 チェンバーの計 4 層による 3-out-of-4 コインシデンスロジックでトリガーを発行する。

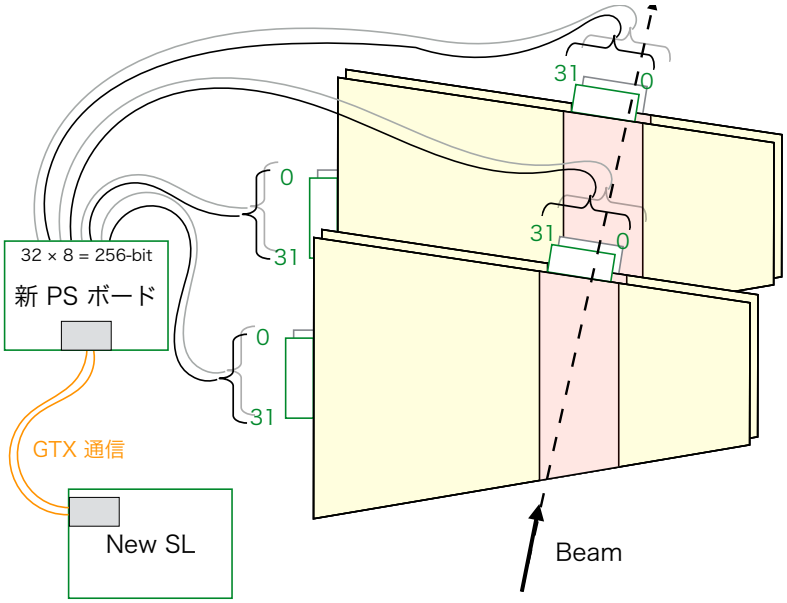


図 6.6　TGC から NSL へ送られる情報と、その配線。使用した TGC は doublet であるため、それぞれ 2 層のワイヤー・ストリップの層を持つ。今回のセットアップでは各層 32 チャンネルずつ読み出し、新 PS ボードへ入力した。ワイヤー・ストリップ共に 2 層 ×2 チェンバー ×32 チャンネル = 128 チャンネルあるため、新 PS ボードは合計 256 チャンネル分の情報を受け取る。

6.4　ビームテスト結果

6.4.1　GTX 通信、及び読み出し Firmware の評価

GTX の受信及び読み出し Firmware の性能を、取得したデータを用いて評価する。用いるデータは、最終日に取得された以下の 7 つのランである。Run 100833 は NewSL トリガーで取得されたデータで、イベント数は約 29 万イベントである。Run 100842 - 100847 は、P.S. のトリガーで取得されたイベントで、TGC 側の設定は変えずに連続して取得されたデータセットである。この 6 ランのデータを合わせて Plastic Scintillator トリガーで取得されたデータとして、NewSL トリガーデータと比較する。Plastic Scintillator で取られたデータセットのイベント数は合計で約 28 万イベントである。

表 6.1　データ解析に用いたラン

Run Number	期間（CERN time）	イベント数	トリガーソース
100833	2016-11-14 04:00:10 — 04:13:46	287798	TGC (NewSL)
100842	2016-11-14 05:18:28 — 05:24:50	51679	Plastic Scintillator
100843	2016-11-14 05:26:54 — 05:33:14	52242	Plastic Scintillator
100844	2016-11-14 05:35:06 — 05:41:37	55010	Plastic Scintillator
100845	2016-11-14 05:43:10 — 05:49:13	50497	Plastic Scintillator
100846	2016-11-14 05:50:33 — 05:56:42	50208	Plastic Scintillator
100847	2016-11-14 05:58:06 — 06:01:29	17601	Plastic Scintillator

まず、信号のタイミング合わせについて評価する。図 6.7 にあるように、TGC、MDT、トリガー信号は全て別々の経路を通じて NewSL に入力される。これらの差を FPGA 内の Delay 部分及び Buffer の読み出し位置によって調節する。Delay 部分は GTX のファイバー 1 本毎に適切な遅延を設定することで、ファイバーごとのタイミングのばらつきを吸収する。また、TTC 信号に対して遅延をかけることでタイミングの調整も行うことができる。Buffer はトリガーよりもデータを早く受信した場合にそれらを保持しておくために用いる。これらのタイミング調整を正しく行うことができているかを、取得したデータで評価した。

特に、トリガーソースを変更した際は大きな Latency の変更が想定される。Plastic Scintillator のトリガーを採用している場合は、2 枚の P.S. の出力のコインシデンス処理の結果が直接 TTC システムに入力される。一方で NewSL によるトリガーでは、TGC PS ボードによる信号処理の及び光信号への変換を経て NewSL で受信された情報をもとにトリガー判定を行うために、TTC へトリガー信号が到着するまでの Latency が大きくなる。

図 6.8 に、TGC のヒットの時間分布を示す。左が NewSL のトリガーで取得されたラン、右が P.S. のトリガーで取得されたランのものである。P.S. のトリガーで取得する場合はトリガー信号がデータよりも早く到着するため、Delay モジュールでトリガー信号を遅延させることによりタイミング調整を行った。この際のトリガー信号に対する Delay モジュールの遅延パラメータは、25 BC である。データに対しては一切遅延は行わず、バッファーにデータを貯める必要もないため、読み出し深さは 1 としている。

NewSL トリガーの場合はデータを受信した後にトリガー判定を行うので、必然的にトリガーの方が遅くなる。この時は、トリガー信号には遅延をかけずに、読み出しバッファーの深さで調節した。具体的には読み出しバッ

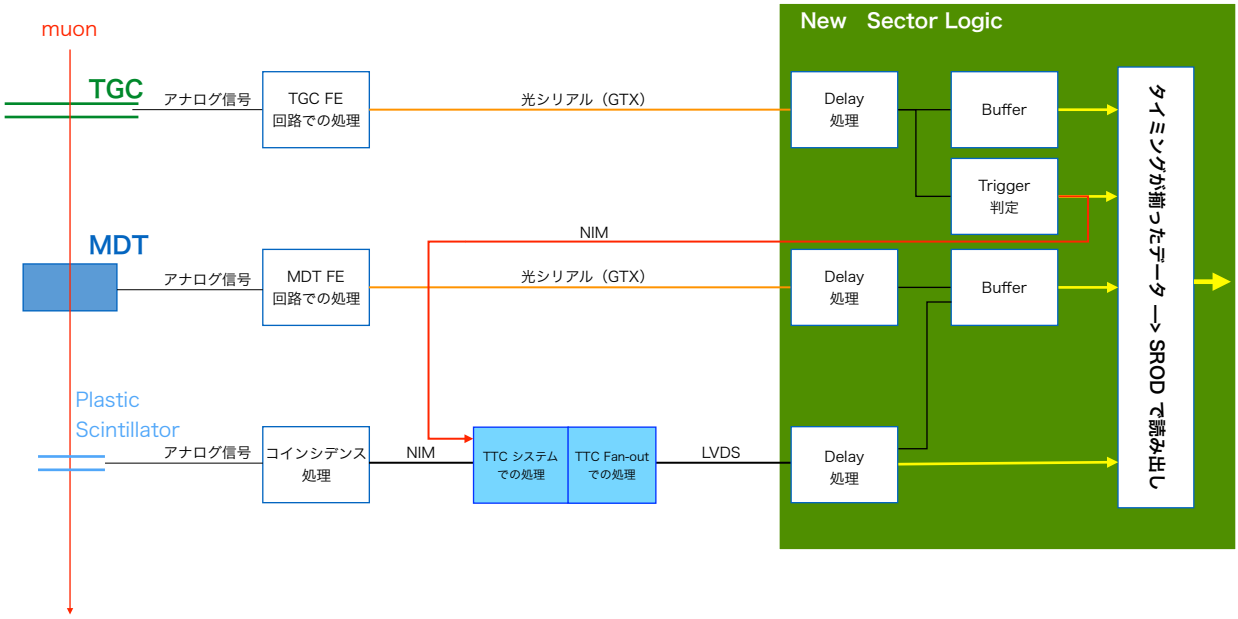


図 6.7　ビームテストにおけるタイミング図。TGC・MDT・トリガーの 3 種類の信号は全て違った経路を通じて New SL に入力される。各ボックスに入力された信号は一定の処理時間後に出力される。

ファーの深さを 16 とすることで、トリガー信号とのタイミングを合わせた。図 6.8 の横軸は、トリガー信号を受信したタイミングからの BC 数に対応する。今回 TGC データの読み出し幅は 15、つまりトリガー信号が来てから 15 BC 分を保存することとしたため、横軸は 1~15 である。これを見ると、NewSL トリガーの際はトリガー受信から 4 BC 付近、P.S. トリガーの際はトリガー受信から 8 BC 付近にヒットが多く見られることがわかる。なお、これらが厳密に一致していないのは、15 BC の幅の中にヒットが見られれば十分であるため、細かい調整の必要がなかったためである。（もちろん、1BC 単位での調整を行うことも可能である。）

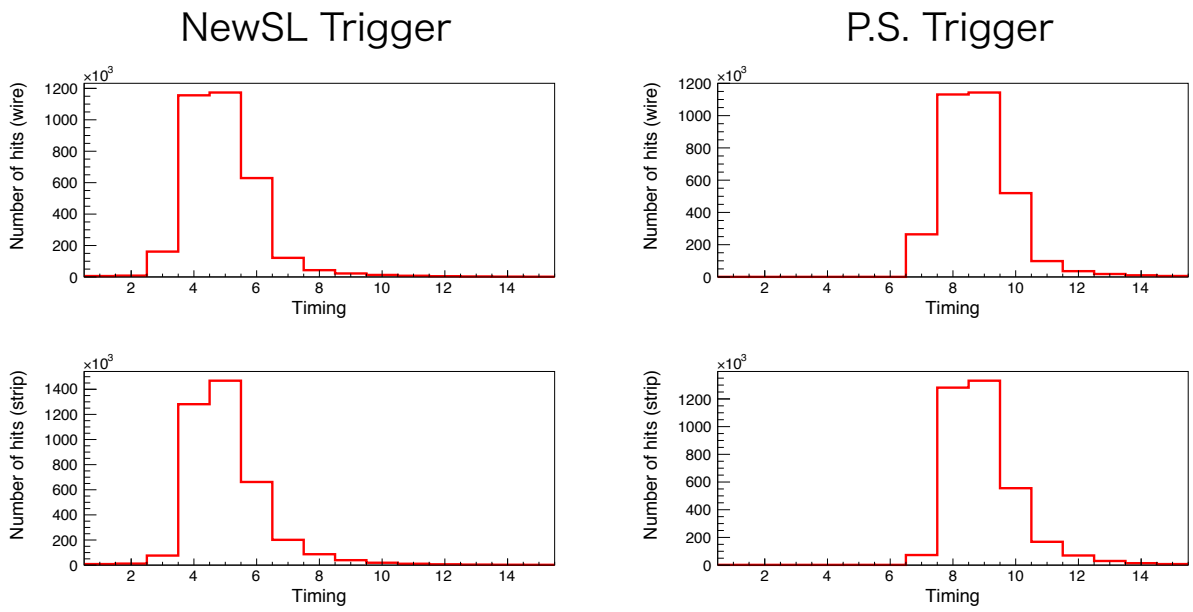


図 6.8　（左）NewSL のトリガーで取得されたイベントのヒットタイミングの分布。（右）Plastic Scintillator のトリガーで取得されたイベントのヒットタイミングの分布。横軸はトリガー信号が来てからの BC 数を表す。上下はそれぞれワイヤー、ストリップのヒットタイミング分布である。

次に、読み出し Firmware の検証を行う。図 6.8 で正しくヒットタイミングを調整できているため、基本的に読み出し Firmware は機能していると考えられるが、ここではデータの内容がもっともらしいものかの確認を行った。図 6.4.1 に、記録された TGC 上でのヒット位置の分布を示す。左の 2 つが NewSL トリガー、右の 2 つが P.S. トリガーで記録されたデータによる分布であり、それぞれワイヤー・ストリップのヒット位置分布に対応する。横軸は検出器のレイヤーで、doublet チェンバーが 2 枚の計 4 層分を 1~4 で示している。縦軸はヒットチャンネルを表し、それぞれの層の 32 チャンネルに対応する。

NewSL トリガーで取得されたデータを見ると、ワイヤー・ストリップ共にヒットが集中している部分があるのがわかる。これがビームの中心付近であり、広がりはワイヤーで約 10 チャンネル、ストリップで約 4 チャンネルほどである。これはそれぞれ約 70 mm、100 mm に対応するため、ビームサイズがおおよその想定通りであるこ

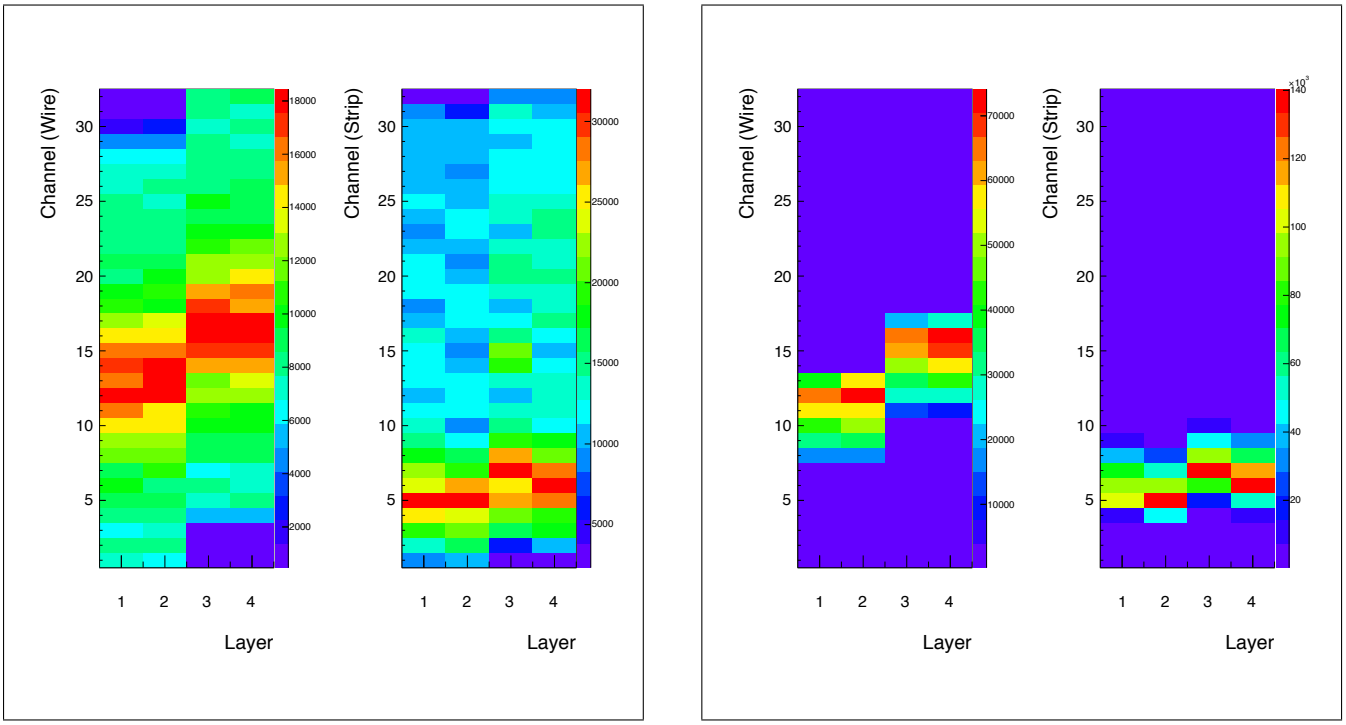


図 6.9　（左の 2 つ）NewSL のトリガーで取得されたイベントのヒット位置の分布。（右の 2 つ）Plastic Scintillator のトリガーで取得されたイベントのヒット位置の分布。縦軸は各レイヤーのチャンネル番号を表し、横軸は上流から数えて何レイヤー目かということに対応する。またそれぞれの左側はワイヤー、右側はストリップのヒット分布である。

とを確認した。また、P.S. トリガーで取得したデータでは、ヒットが見えるチャンネルが限定されている。これは P.S. が設置されている部分に対応すると考えられ、実際にこの範囲は P.S. がカバーしている約 10 cm × 10 cm の範囲と一致していた。以上のように、データはビーム特性及び検出器の配置とよく一致しており、正しいデータ読み出しが実現できていると言える。なお、ワイヤー及びストリップの 1・2 レイヤー目と 3・4 レイヤー目でヒット位置の中心が数チャンネル違うのは、検出器の配置のずれによるものである。

さらに、これらのデータ取得の際には常に GTX のエラーをモニターしている。GTX のエラーとは、Comma を用いたワードアラインメントに失敗した際に検知されるもので、その数を数えることで通信エラーが起きていないかを確認することができる。セットアップが確立したビームテストの後半のランでこのような GTX エラーは検知されておらず、データの損失無く通信ができることを確かめた。

また、データへの ID 情報付与及びフォーマットの整合性についても、SROD でランの最中常にモニターを行っている。SROD は ID 情報の不一致やフォーマットの崩れを検知した際はデータ取得を止めるようにデザインされている。ID・フォーマットのエラーついては今回挙げた 7 つのラン（∼ 50 万イベント）では全く発生しなかった。従って、Firmware の機能として ID 付加及びデータフォーマットは数時間のラン中では安定して正しく処理できていることを確認した。

6.4.2　Fixed Latency でのトリガー判定の評価

図 6.11 は NewSL でのコインシデンスの結果である。横軸は 1 枚目の TGC（TGC1）でのヒット位置、縦軸が 2 枚目の TGC（TGC2）でのヒット位置で、図 6.10 に示すように doublet の 2 層分の情報を合わせて 0∼63 で位置を表現している。NewSL では TGC1 での位置と TGC2 での位置のチャンネル番号の差が一定数以下の場合にトリガー判定をするようなロジックを実装した。図 6.11 の左は NewSL のトリガーで取得したデータ、右は P.S. のトリガーで取得したデータである。Firmware で実装した通り、チャンネル番号の差が小さい時のみトリガー判定が行われていることがわかる。なお、TGC1 と 2 のチャンネル番号が完全に一致しないのは、上でも述べたように検出器の配置がのずれによるものである。P.S. トリガーでデータを取得したランでも、NewSL はトリガー判断、つまり TGC1 と TGC2 で同じ位置にヒットがあるかどうかの判定を行っているが、その結果はイベントの取捨の判断には用いられず、結果が保存されているだけであることに注意する。

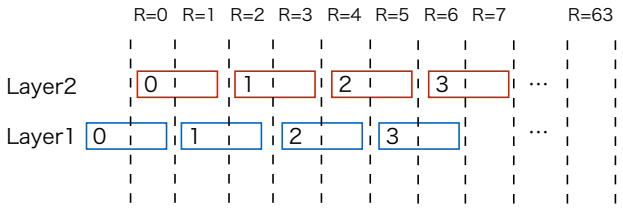


図 6.10　TGC の doublet の層ごとのチャンネルの構造。このように互い違いにチャンネルを読み出すことで分解能を高くしている。

最後に、Fixed Latency が実現されているかを確認した。図 6.12 は、トリガー判定が行われたタイミングの分布である。これは、NewSL でトリガー判定をした時間と、トリガー信号が NewSL に到達した時間との差に対応する。NewSL トリガーで取得したデータでは、すべてのイベントでタイミングが一定であることがわかる。

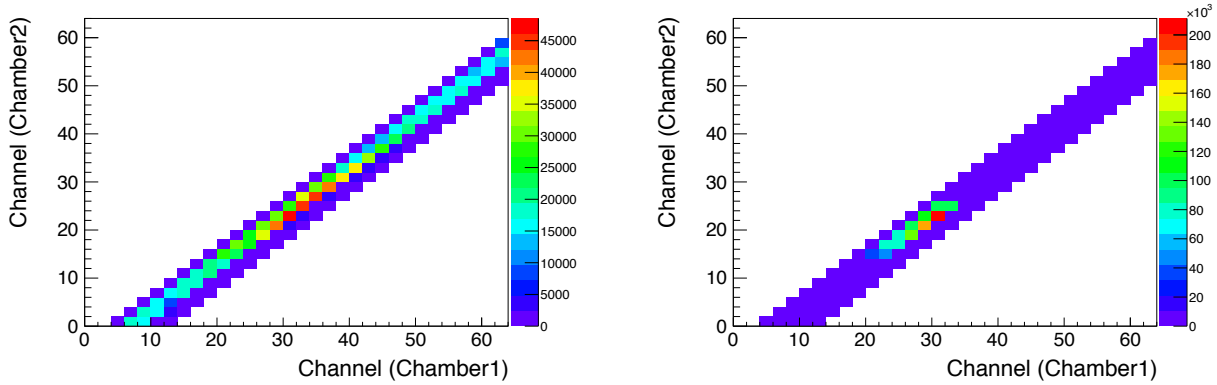


図 6.11　（左）NewSL のトリガーで取得されたイベントのトリガーを発行した RoI の位置分布。（右）Plastic Scintillator のトリガーで取得されたイベントのトリガーを発行した RoI の位置分布。横軸は TGC1（Chamber1）の、縦軸は TGC2（Chamber2）のヒットチャンネルを表す。これらのチャンネルの差が小さい時にトリガーを出力する。

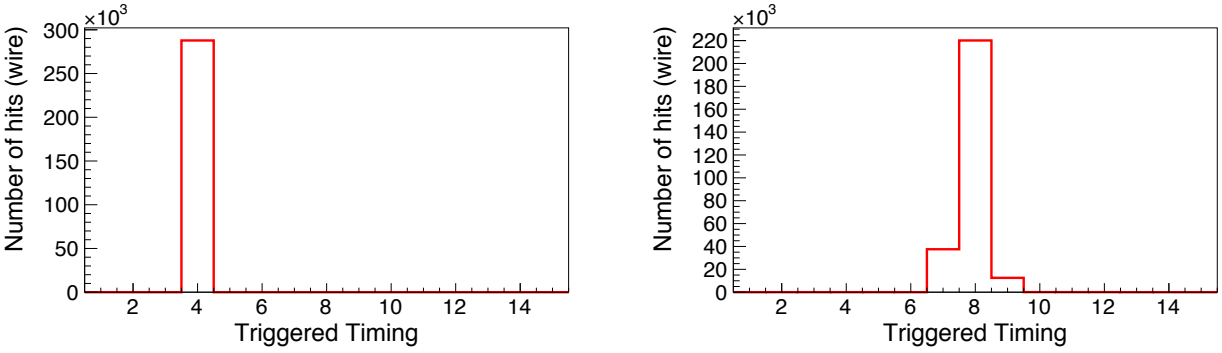


図 6.12　（左）NewSL のトリガーで取得されたイベントの、トリガータイミングの分布。（右）Plastic Scintillator のトリガーで取得されたイベントの、トリガータイミングの分布。

Fixed Latency のトリガーロジックであることを確認するには、検出器でヒット情報が残されてから NewSL がトリガー判定を行うまでの時間も一定であることも確認する必要がある。TGC PS ボードはヒット情報に BCID を付加しており、これと NewSL で付加した BCID の差が常に一定であれば、PS ボードから NewSL 間での Latency は一定であることが確認できる。実際に保存されたデータで BCID の差を見たところ、すべてのイベントで一定であることが確認された。以上より、NewSL において、GTX で受信したデータを用いた Fixed Latency ロジックの実装に成功したことを確認した。

第7章　結論と展望

LHC Run-3 において、ルミノシティは現在の約 2 倍の 3×10^{34} $\mathrm{cm^{-2}s^{-1}}$ まで上昇する。ATLAS 実験ではトリガーのアップグレードが緊急な課題である。本研究では Run-3 で新たに導入されるミューオン検出器 NSW の情報を用いたトリガーの改良を行った。

現在のミューオントリガー検出器 TGC と、新たに導入される New Small Wheel の情報を組み合わせた高性能ミューオントリガーをハードウェアで実装するために、新しいトリガー判定ボード New Sector Logic を開発した。New Sector Logic には新検出器からのデータ受信のために、10 Gbps の高速トランシーバ GTX を実装した最新の FPGA を搭載した。また、TCP/IP によるデータ読み出しのための I/O インターフェイスを設置した。作成した New Sector Logic ボードの性能試験を行い、その結果 Run-3 で要求される性能を満たしていることを確認した。

さらに、Run-3 において New Sector Logic の FPGA に実装する Firmware の開発を行い、GTX データ受信部、及びトリガー判定結果送信部の双方において Fixed Latency のための仕組みを導入することで、Latency を常に一定にできることを確かめた。開発した Firmware をビームテストで使用して、GTX で受信した情報を用いた Fixed Latency トリガー判定に成功した。また、受信したデータの保持・成型・圧縮、及び TCP/IP での読み出しを行う読み出し Firmware の開発も行い、ビームテストにおけるデータ取得及び安定運用に成功した。

ハードウェアの開発と並行して、New Sector Logic に実装されるトリガーロジックの研究を行った。NSW の角度情報を用いた新しいトリガーロジックを考案しシミュレーションにより得られた NSW と TGC のヒット情報を用いてその性能評価を行った。その結果、角度情報を用いた新たなトリガーロジックを適用することで、閾値以上のミューオンのトリガー効率を維持しつつ低い横運動量のミューオンを選択的に排除できることを示した。これを Run-2 でのデータに適用しトリガーレートを計算することで、これまでに見積もられていたトリガーレートに対し NSW のカバーする範囲では約 15%、ミューオントリガー全体で約 7.5% の削減が可能であることを示した。さらに、考案したトリガーロジックを実際にハードウェア上に実装するために、FPGA 上のメモリ資源の有効利用の観点からロジック構造の見直しを行った。新たに提案した構造で実装すれば NewSL に搭載されている FPGA 上に角度ロジックを用いた新トリガーロジックを FPGA 容量の範囲内で実装できることを示した。

本研究で NewSL のハードウェアの設計は完了した。Firmware の設計に関しては必須要素のデザインは固まった一方で、未だ最適化の余地は残されている。例えば、GTX の送受信クロックの周波数は 320 MHz を用いたほうが Latency が短くなることがわかっている。これを用いることができれば良いが、速いクロックを使う場合は FPGA 上の配線の遅延も考慮しながら Firmware のデザインをしなければならない。他にも、読み出し FIFO の深さ、Zero-Suppress ロジックの再検討等、最適化できる部分は残されている。

さらに、本研究で説明したコインシデンスロジックでは、TGC BW の dR の情報は、TGC の $dR:d\phi$ コインシデンスにしか用いていない。この dR と NSW の情報、例えば $d\eta$ や $d\theta$ とのマッチングによってより良いトリガーロジックを組めるかもしれない。考えうるトリガーロジックの中から、BRAM リソースの使用量を考慮に入れながら現実的な範囲での最適なロジックを選択する必要があり、今後の研究課題である。

謝辞

本研究では、様々な方々にお世話になりました。

まず、研究のアドバイスや論文執筆の助言をくださった隅田土詞助教に心より感謝いたします。また、ATLAS 実験を始めるきっかけを与えてくださった石野雅也教授に感謝いたします。

Phase-1 アップグレードチームの皆様にも大変お世話になりました。特に佐々木修教授には、日頃から細かい疑問まで一緒に考えていただきました。大変感謝しております。坂本宏教授にも、ミーティングや発表会の場で様々なアドバイスや鋭い質問をいただき、大変勉強になりました。ありがとうございました。蔵重久弥教授には、特にビームテストにおいて大変なご協力をいただきました。感謝いたします。前田順平講師には、Firmware・Software 開発においてだけでなく、学会等の発表の練習や論文執筆においても細やかなアドバイスをいただきました。心から感謝いたします。岸本巴助教には、特にトリガーロジックの研究の面でお世話になりました。解析に不慣れな私に Software の初歩的な部分から丁寧に教えてくださったおかげで、この修論があります。ありがとうございました。奥村恭幸助教には、Firmware 開発の面で大変お世話になりました。特に、GTX トランシーバ付近のデザインやクロック受け渡し等の細かいデザインの部分は、奥村さんのご協力無しには完成しなかったと感じます。その他、論文や発表資料作成にあたっても細かい指導をいただきまた。ありがとうございました。総研大の鈴木翔太氏には FPGA のイロハを教えていただきました。Firmware 開発や NewSL ver.2 に向けた修正においても、大変ご協力をいただきました。同期の神戸大の谷岡凌氏、水上敦氏、2 年間一緒に実験を進めてくれてありがとう。一緒に議論しながら研究を進めていける同期の仲間がいて、本当に良かったと思います。後輩の岡崎佑太氏、竹田康亮氏にも、特にビームテストで大きく協力してもらいました。特に岡崎氏には、ともに NewSL Firmware を進める中でアイデアを出し合い、良い Firmware を開発するために大きな協力をもらいました。ここで感謝の意を表します。

ビームテストは、初めての経験でわからない中、いろいろな方に指導していただきました。特に名古屋大の戸本誠教授、堀井泰之助教には、大変お世話になりました。検出器の運用から物の準備、ビームタイムの交渉まで、様々な環境を整えていただきました。このお陰で学生は自分の回路の試験に集中できたと思います。心より感謝申し上げます。ビームテストを共に行った名古屋大の水越健太氏、宿谷琴子氏、川口智美氏にも感謝します。

CERN での生活でもいろいろな方にお世話になりました。特に、名古屋大の小野木宏太氏には、日々の生活から週末のアニメ上映会、ワインの嗜みまで、様々な面でお世話になりました。また、東大の森永真央氏、陳詩遠氏にも家に居候させていただき、生活面で大変にお世話になりました。CERN での生活が楽しかったのも名古屋邸に住まわせていただけたからだと思っています。

京大の同期の芦田洋輔氏、田中駿祐氏、中西芳枝氏、野口陽平氏、平本綾美氏に感謝します。同期の皆がいたからこそ、2 年間充実した研究生活を送ることができたのだと思います。特に、コーヒーブレイクメンバーの芦田くん、野口くん。辛い時期もコーヒーを飲んで一息を着く時間があってよかったです。ありがとう、これからもよろしく。

ATLAS 実験の先輩の田代拓也氏、救仁郷拓人氏には、様々な面で大変お世話になりました。毎週のミーティングでは先輩方の鋭い、本質を付くようなアドバイスのおかげでたくさんのことを学びました。また、CERN での手続きの面、生活面でも助言をいただき、謝辞に書くには狭すぎるくらいの感謝の念でいっぱいです。

最後に、自由に人生の道を選ばせてくれた両親に感謝します。本当にありがとう。

参 考 文 献

[1] ATLAS Collaboration, The ATLAS Experiment at the CERN Large Hadron Collider, JINST 3 S08003, 2008 Web

[2] ATLAS Collaboration, 2015 start-up trigger menu and initial performance assessment of the ATLAS trigger using Run-2 data, 12 March, 2016 Web

[3] ATLAS Collaboration, Technical Design Report for the Phase-I Upgrade of the ATLAS TDAQ System. Technical Report CERN-LHCC-2013-018. ATLAS-TDR-023, CERN, Geneva, Sep 2013. Final version presented to December 2013 LHCC. Web

[4] T Kawamoto, S Vlachos, L Pontecorvo, J Dubbert, G Mikenberg, P Iengo, C Dallapiccola, C Amelung, L Levinson, R Richter, and D Lellouch. New Small Wheel Technical Design Report. Technical Report CERN-LHCC-2013-006. ATLAS-TDR-020, CERN, Geneva, Jun 2013. ATLAS New Small Wheel Technical Design Report. Web

[5] ATLAS Collaboration, ATLAS Muon Endcap Level-1 Trigger Update of Technical Design Report, 09, June, 2000.

[6] 木戸将吾, ATLAS 実験 Run2 におけるレベル 1 ミューオントリガーの性能評価及び最適化の研究, 修士論文. Web

[7] 救仁郷拓人, LHC-ATLAS 実験 Run-2 に向けた Level-1 ミューオントリガーアルゴリズムとデータ収集システムの改良, 修士論文. Web

[8] 田代拓也, ATLAS 実験における新しいミューオントリガー回路の開発と実装, 修士論文. Web

[9] 来見田吏恵, ATLAS 実験 RUN2 に向けたレベル 1 ミューオントリガー回路の開発, 修士論文. Web

[10] Osamu Sasaki, Amplifer-Shaper-Discriminator ICs and ASD boards, 1999 Web

[11] XILINX, 7 Series FPGAs GTX/GTH Transceivers User Guide, 2014. Web

[12] http://research.kek.jp/people/uchida/technologies/SiTCP

[13] T.Uchida, Hardware-basedtcpprocessorforgigabitethernet.NuclearScience, IEEETransactions on, Vol. 55, No. 3, pp. 1631-1637, June 2008.

[14] XILINX, Integrated Bit Error Ratio Tester 7 Series GTX Transceivers v3.0, 2016. Web

[15] 内田智久, SiTCP 説明書 第 1.6 版, 2014 Web

[16] 徳永孝之, LHC-ATLAS 実験 RUN3 に向けた新しいミューオントリガー装置の FPGA 読み出し開発とその性能評価, 修士論文. Web

[17] https://twiki.cern.ch/twiki/pub/AtlasPublic/L1MuonTriggerPublicResults/ATL-COM-DAQ-2015-142-fig1a.png

[18] https://twiki.cern.ch/twiki/pub/AtlasPublic/L1MuonTriggerPublicResults/ATL-COM-DAQ-2015-142-fig1b.png

[19] https://twiki.cern.ch/twiki/pub/AtlasPublic/L1MuonTriggerPublicResults/ATL-COM-DAQ-2015-142-fig2.png

付 録A　GTX Eye Patten

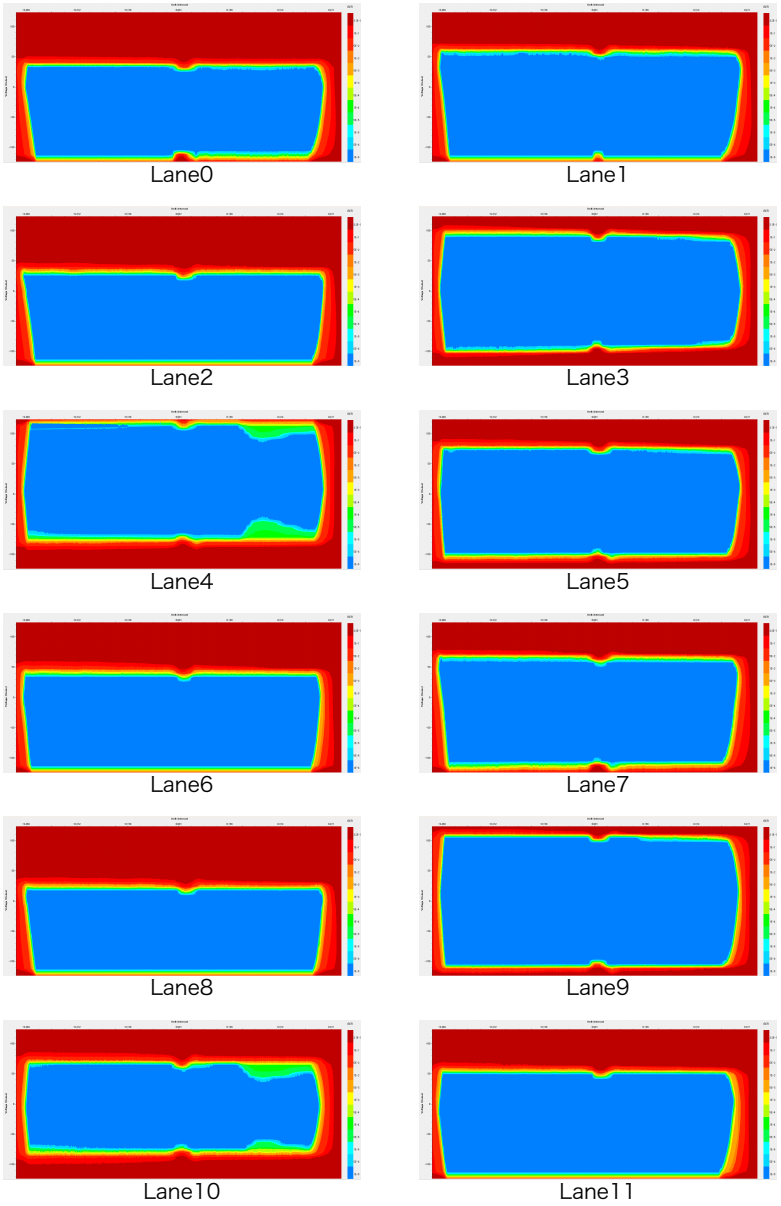


図 A.1　IBERT を用いて得られたアイパターン。転送速度は 0.8 Gbps

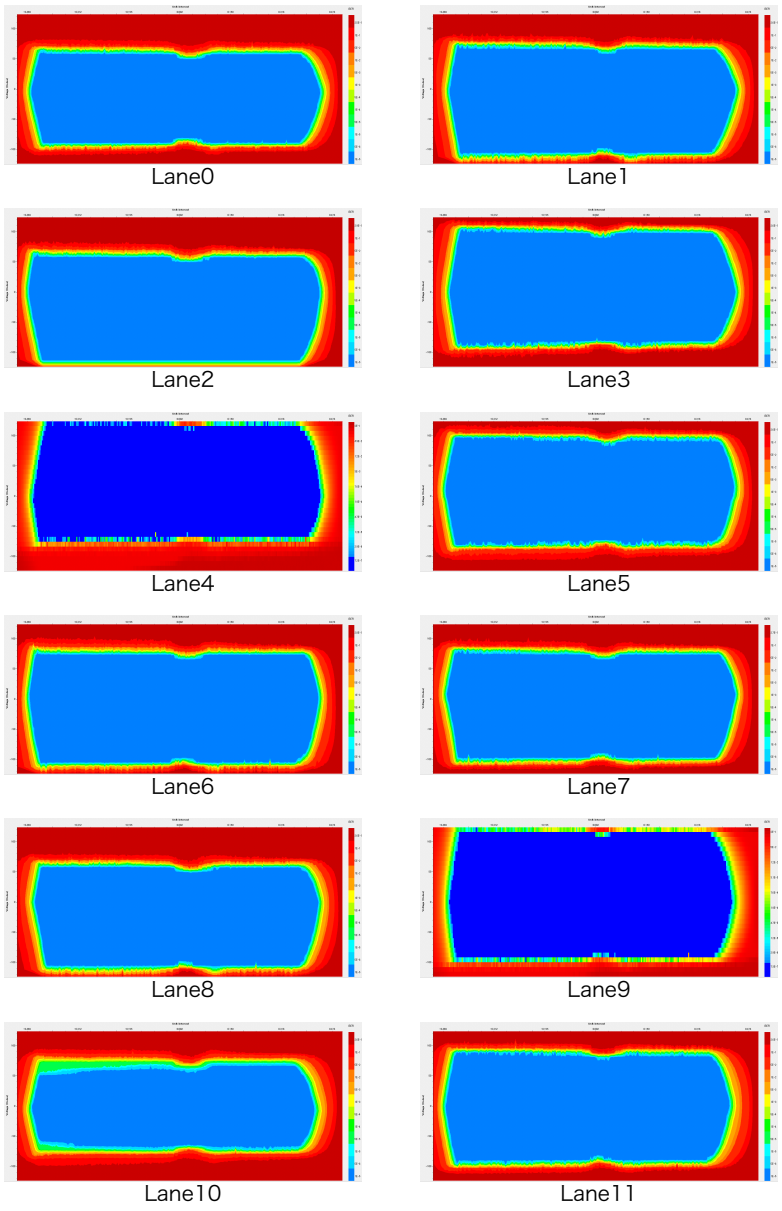


図 A.2　IBERT を用いて得られたアイパターン。転送速度は 1.6 Gbps

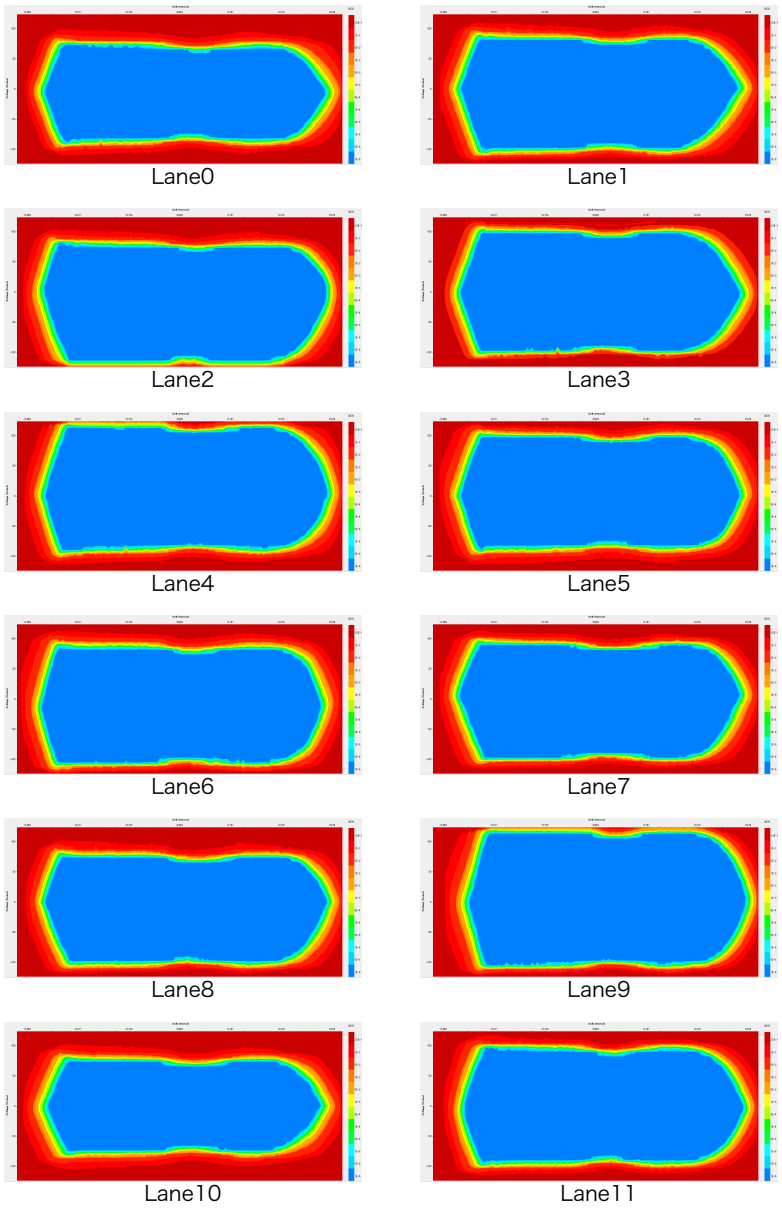


図 A.3　IBERT を用いて得られたアイパターン。転送速度は 3.2 Gbps

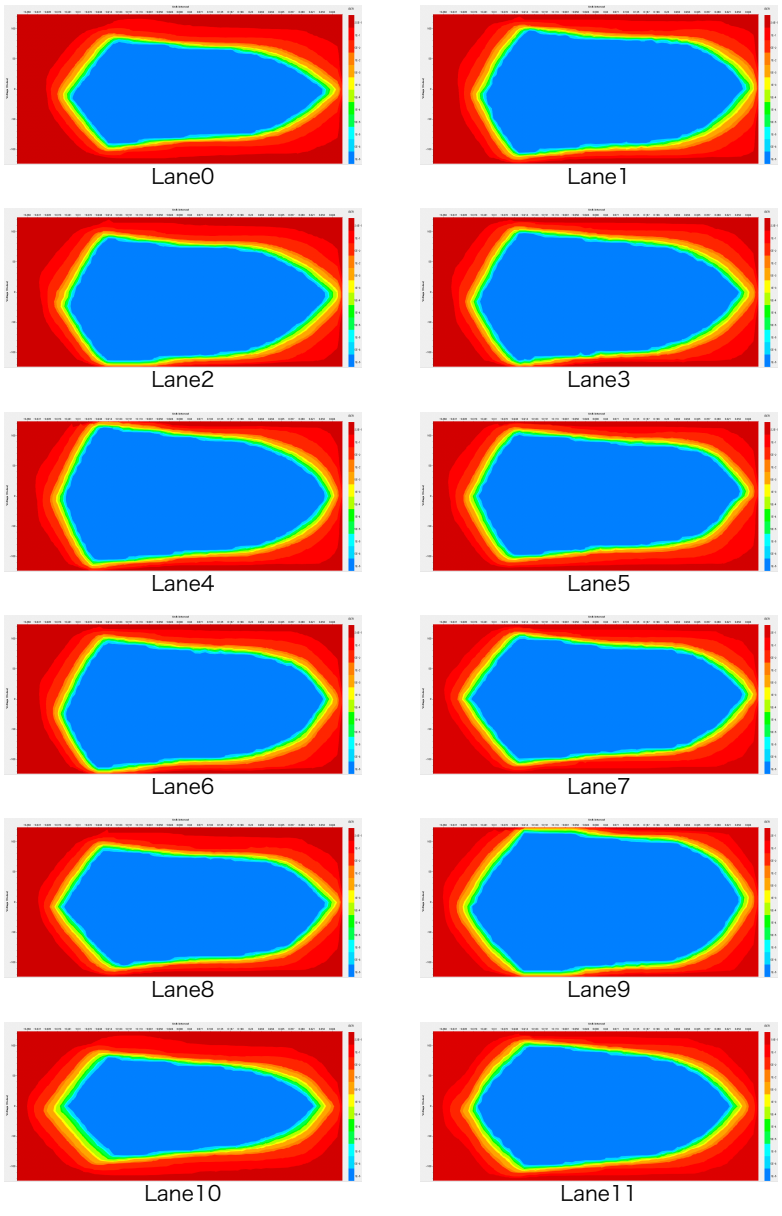


図 A.4　IBERT を用いて得られたアイパターン。転送速度は 6.4 Gbps

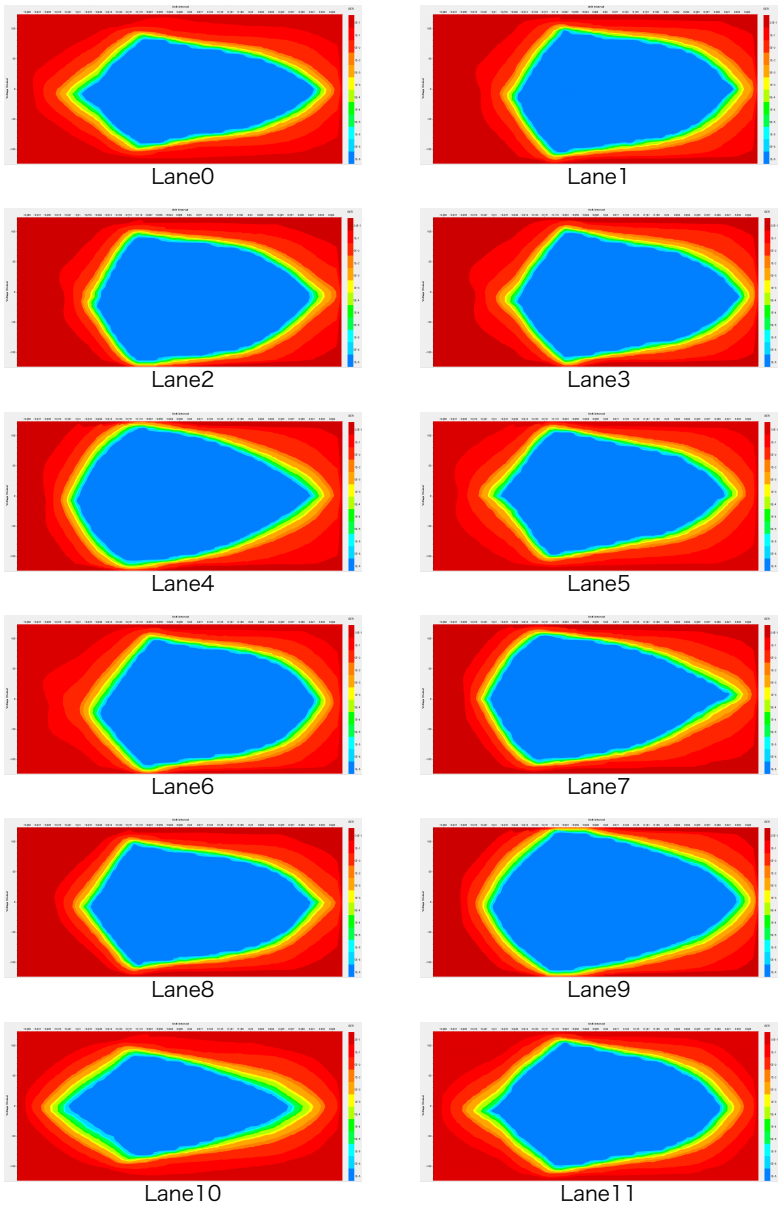


図 A.5　IBERT を用いて得られたアイパターン。転送速度は 8.0 Gbps

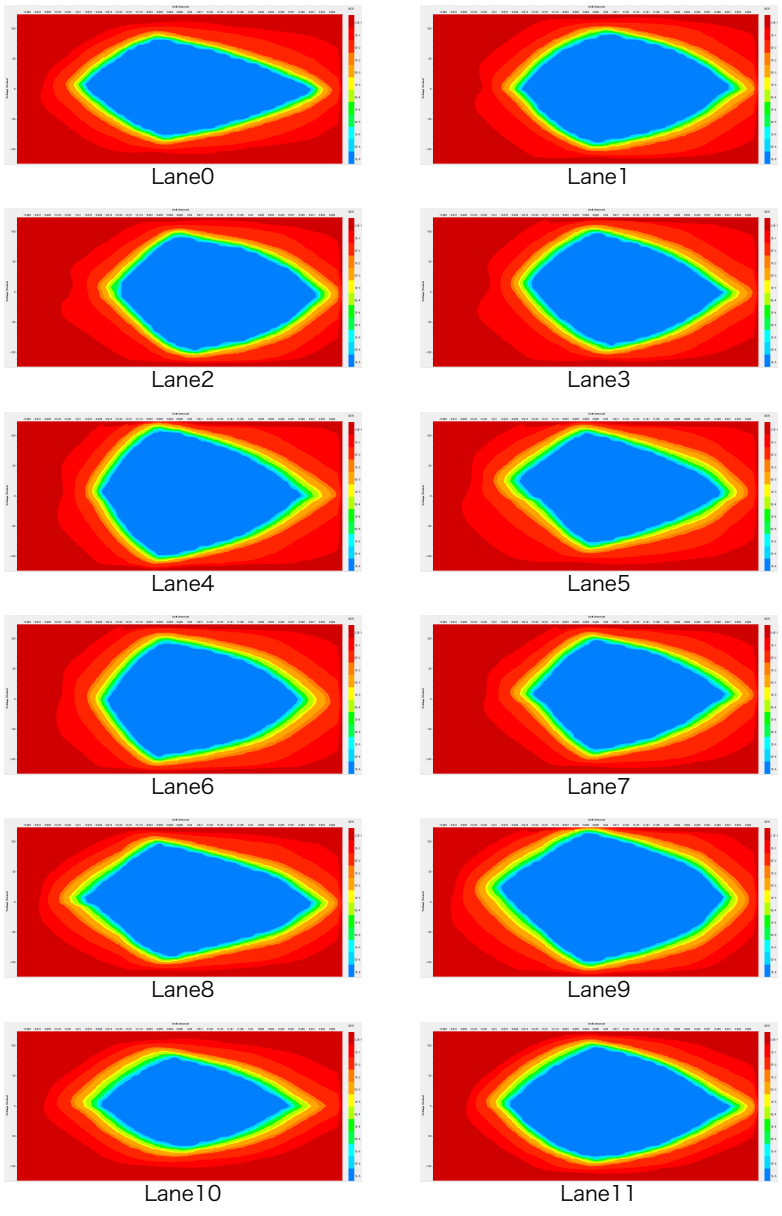


図 A.6　IBERT を用いて得られたアイパターン。転送速度は 10.24 Gbps

付 録 A　GTX Eye Patten　124

†