

修士学位論文

ATLAS 実験ミュオン検出器用トリガーシステムの 現実的シミュレーターの開発研究

信州大学大学院 工学系研究科
物質基礎科学専攻

00SA209G
戸塚 真義

2002 年 1 月 31 日

概要

標準理論の精密検証や超対称性など標準理論を越える超対称性粒子などの探索を目的として、CERN(欧州原子核共同研究機構)において現在、大型陽子・陽子衝突型加速器 LHC を建設中であり、それを使用した ATLAS 実験が 2005 年に開始を予定している。ATLAS 実験では高い衝突頻度で実験を行うため実験データの効率的な収集が要求される。ミュオントリガーシステムはこの要求を満たすべく開発が行われているが、期待通りの性能が出るか等の検証や開発中のエレクトロニクスシステムのテストを目的とし、トリガーシミュレーターの開発を行い、完成したシミュレーターの動作検証を行った。

目次

| | | |
|------------|-----------------------------------|----|
| 1. | ATLAS 実験 | 3 |
| 1.1. | LHC 計画 | 3 |
| 1.2. | LHC の目指す物理 | 3 |
| 1.2.1. | 標準理論の精密検証例 | 3 |
| 1.2.2. | ヒッグス粒子の探索 | 4 |
| 1.2.2.1. | 生成過程 | 4 |
| 1.2.3. | ヒッグス粒子の崩壊過程 | 7 |
| 1.2.4. | ヒッグス粒子の発見能力 | 7 |
| 1.2.5. | 超対称性など標準理論を越える超対称性粒子などの探索 | 9 |
| 1.2.6. | ボトム・クォークの精密研究 | 9 |
| 1.2.7. | QCD の精密研究 | 10 |
| 1.3. | ATLAS 検出器 | 10 |
| 1.3.1. | 内部飛跡検出器 (Inner Detector) | 10 |
| 1.3.2. | カロリメータ | 11 |
| 1.3.3. | ミュオン検出器 | 11 |
| 2. | ATLAS 実験のトリガーとデータ収集システム | 13 |
| 2.1. | 概要 | 13 |
| 2.2. | トリガとデータ収集システムの構成 | 13 |
| 2.2.1. | Level-1 trigger | 13 |
| 2.2.2. | Level-2 trigger | 14 |
| 2.2.3. | Event Filter | 14 |
| 2.3. | LEVEL-1(LVL1) MUON TRIGGER SYSTEM | 14 |
| 2.4. | 前後方ミュオントリガーシステム | 15 |
| 2.4.1. | 前後方ミュオントリガーシステム用 TGC の構成と配置 | 15 |
| 2.4.1.1. | Trigger Sector | 16 |
| 2.4.2. | Trigger の方法 | 19 |
| 2.4.3. | Board Connection | 19 |
| 2.4.4. | PatchPanel | 21 |
| 2.4.4.1. | Connection | 21 |
| 2.4.4.2. | OR 'ing | 26 |
| 2.4.5. | SlaveBoard | 26 |
| 2.4.5.1. | Input | 27 |
| 2.4.5.2. | Coincidence Logic | 28 |
| 2.4.5.2.1. | WireDoublet | 29 |
| 2.4.5.2.2. | StripDoublet | 30 |
| 2.4.5.2.3. | WireTriplet | 31 |
| 2.4.5.2.4. | StripTriplet | 32 |
| 2.4.5.3. | Declustering | 33 |
| 2.4.5.4. | Encode と Output | 34 |
| 2.4.6. | HighPtBoard | 34 |
| 2.4.6.1. | Input | 35 |
| 2.4.6.2. | Decode | 38 |

| | | |
|------------|---|----|
| 2.4.6.3. | MatrixStructure..... | 38 |
| 2.4.6.4. | TrackSelector (H/L - Select) | 39 |
| 2.4.6.5. | Encode と Output | 39 |
| 2.4.7. | SectorLogic..... | 40 |
| 2.4.7.1. | Input..... | 41 |
| 2.4.7.2. | SSC | 42 |
| 2.4.7.2.1. | Selection Rule..... | 43 |
| 2.4.7.2.2. | TGC と SSC の対応..... | 43 |
| 2.4.7.3. | TGC 境界の問題..... | 44 |
| 2.4.7.4. | Decoder | 45 |
| 2.4.7.5. | R- Coincidence Logic..... | 45 |
| 2.4.7.6. | Track Pre Selector..... | 46 |
| 2.4.7.7. | Track Selector..... | 46 |
| 2.4.7.8. | Encoder と Output | 46 |
| 2.4.8. | モジュールの配置 | 46 |
| 3. | TGC トリガーシミュレーション | 47 |
| 3.1. | 目的..... | 47 |
| 3.2. | 構成..... | 47 |
| 3.2.1. | Geant3 DICE | 47 |
| 3.2.2. | Digitize..... | 47 |
| 3.2.3. | Trigger Electronics Modules..... | 47 |
| 3.2.3.1. | PatchPanel 関連クラス | 48 |
| 3.2.3.2. | SlaveBoard 関連クラス | 48 |
| 3.2.3.3. | HighPtBoard 関連クラス..... | 49 |
| 3.2.3.4. | SectorLogic 関連クラス..... | 50 |
| 3.3. | ソフトウェアのテストと評価 | 50 |
| 3.3.1. | Digitize、DICE..... | 50 |
| 3.3.2. | SlaveBoard,HighPtBoard -TestBoard との比較..... | 55 |
| 3.3.2.1. | TestVector..... | 56 |
| 3.3.3. | 運動量 P のトラックでのテスト..... | 57 |
| 3.3.4. | 視覚化..... | 61 |
| 3.4. | R-PHI COINCIDENCEMAP | 65 |
| 3.4.1. | 生成ルール..... | 66 |
| 3.4.1.1. | セルに1つのPtLevelの入力しかないケース..... | 66 |
| 3.4.1.2. | セルに複数のPtLevelの入力(重なる)があるケース..... | 66 |
| 3.4.1.3. | セルに何も入力がないケース | 66 |
| 4. | まとめ..... | 68 |

1. ATLAS 実験

1.1. LHC 計画

現在最も高いエネルギー領域を実現している加速器として LEP (Large Electron Positron Collider) がある。しかし標準理論等の精密検証のためのより精密な測定や、存在が予想されている粒子の探索のために更なる高衝突頻度、高エネルギー領域での実験を行う必要がある。このため現在建設中である LHC (Large Hadron Collider) では、重心系のエネルギーで 14TeV、ルミノシティー $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ を実現することで、ヒッグス粒子の理論上の質量上限 1TeV のカバー、超対称性粒子の探索、トップクォークの精密観測による標準理論の精密検証、ボトムクォークの観測による CP の破れの精密測定等、さまざまな物理の成果が期待されている。

| | | | |
|--------------|--|------------|----------------------|
| 主リング周長 | 26,658,87m | バンチ間隔 | 25ns |
| 入射エネルギー | 450GeV | バンチ長さ(陽子) | 75mm |
| 陽子エネルギー | 7.0TeV | 衝突点でのビーム半径 | 16 μm |
| 鉛イオンビームエネルギー | 574TeV | ビーム衝突角度 | 200 μrad |
| ビーム寿命 | 22 時間 | バンチ内陽子数 | 1×10^{11} 個 |
| ルミノシティー | $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ | | |

表 1 : LHC の主要パラメータ

LHC はスイスのジュネーブにある CERN(欧州原子核研究機構)の LEP のトンネルをそのまま使い、2005 年に稼動開始を予定し現在建設を進めている。

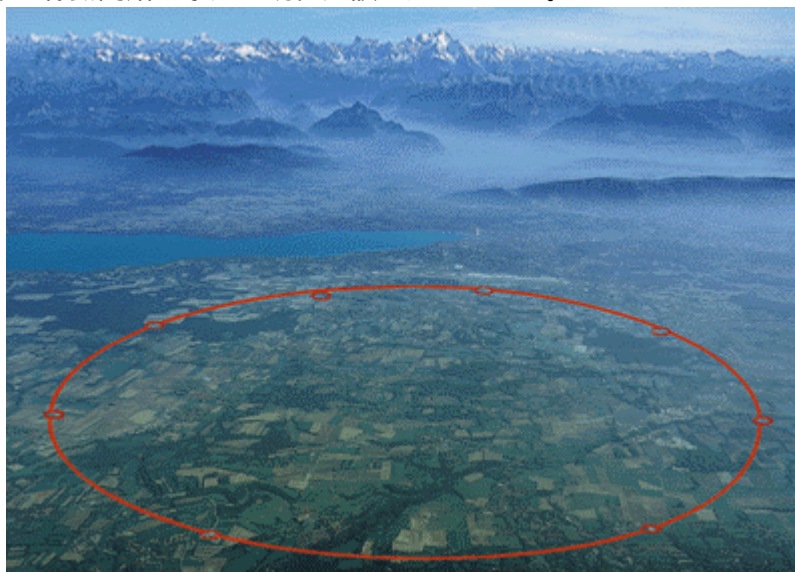


図 1 : LHC

1.2. LHC の目指す物理

1.2.1. 標準理論の精密検証例

Top quark の対生成断面積は、833pb と大きいので、積算ルミノシティー 30 fb^{-1} で、 2.5×10^7 の Top quark の対生成イベントが生成される。この大量のイベントデータを用いて、トップクォークの質量や部分崩壊率の測定によって標準理論の検証を行う。

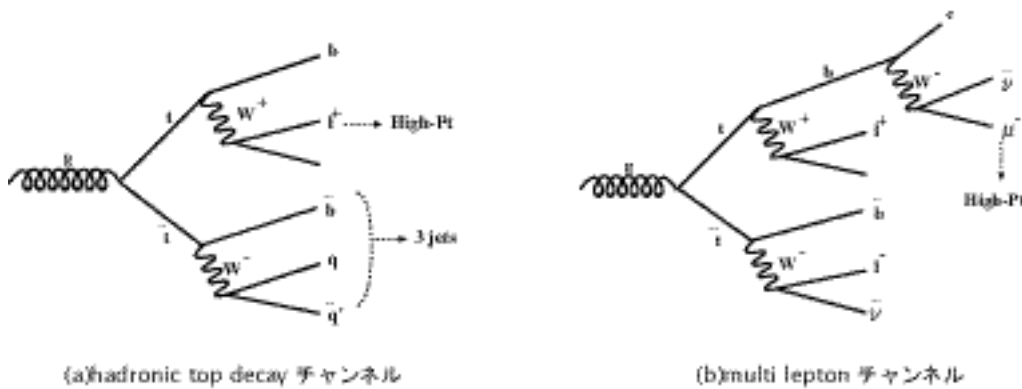


図 2：トップクォークの崩壊モード

質量の測定は、図 2：トップクォークの崩壊モードの $tt \rightarrow bW^+ + bW^+ \rightarrow bb l^+ qq$ において lepton で trigger をかけ測定する。

また、 WW 対生成過程を通して、Strongly Interacting Higgs 粒子などによる電弱対称性の破れの可能性を探る。

1.2.2. ヒッグス粒子の探索

ATLAS¹では標準理論の予言するヒッグス粒子の質量 M_H が 80GeV から 1TeV の範囲で探索することができる。理論的な質量の上限が 1TeV であるため、ヒッグス粒子が存在しているならば ATLAS によって発見することができる。

ヒッグス探索は LHC ATLAS 実験において要であるので詳しく説明する。

1.2.2.1. 生成過程

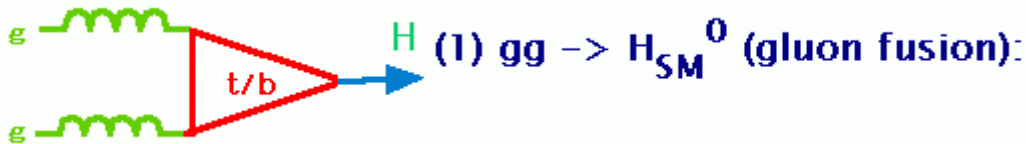


図 3：グルオングルオン融合

この過程の生成断面積は、20pb($M_H=160\text{GeV}$)と最も多い。しかし Higgs 粒子の崩壊から出てきた粒子以外は、大きな横運動量をもつ粒子がなく、バックグラウンドが厳しい過程である

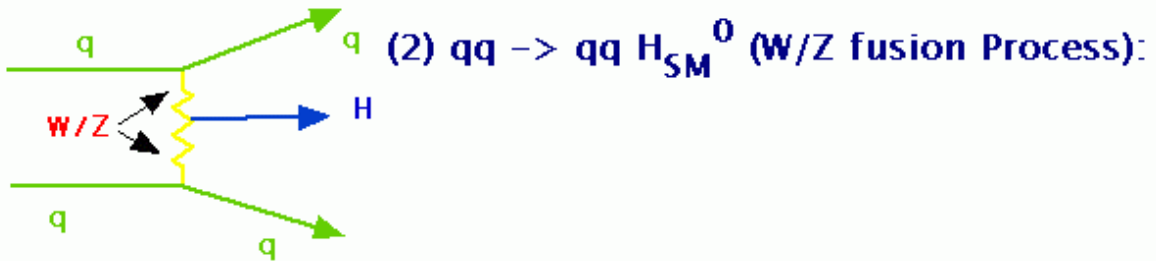


図 4：WW、ZZ 融合

ふたつのクォークから放出された二つの Gauge Boson から H_{SM}^0 が生成されている。反応断

¹ LHC のための汎用測定器 ATLAS (A Toroidal LHC Apparatus) 測定器 1.3 で詳しく説明する。

面積は、3.5pb(MH=160GeV)と 比較的大きく、Gauge Boson を放出して反跳したクォークに起因する 大きな横運動量を持つジェットが2本観測される特徴を持っている。

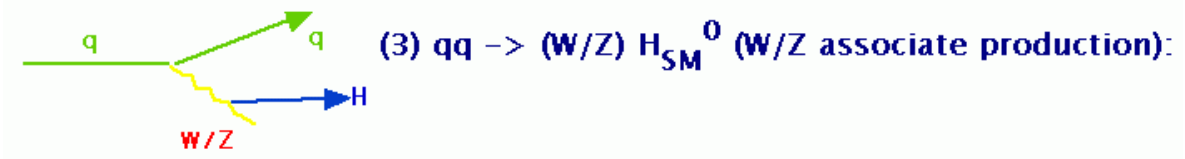


図 5 : W/Z を伴う生成

クォークペアの対消滅で生成された Gauge Boson から、更に H_{SM}^0 が放射される過程であり、1.2pb(MH=160GeV)の反応断面積である。 終状態に、Gauge Boson(W/Z)が観測される特徴を持っている。

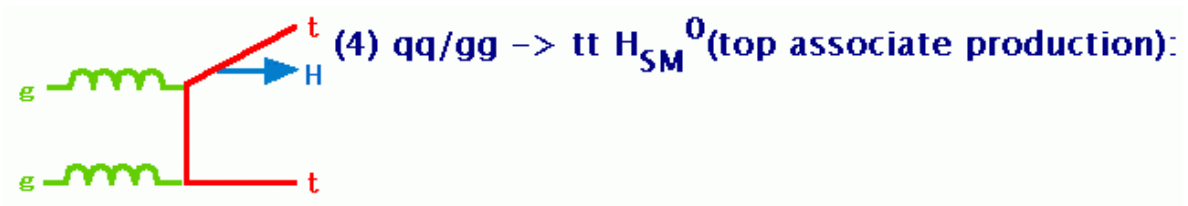


図 6 : tt を伴う生成

対生成されたトップ・クォークから、 H_{SM}^0 が放出される過程である。0.16pb(MH=160GeV)と反応断面積は小さいが、特徴のある Top Quark Pair を終状態に含んでいるので バックグラウンドが少なく、軽い H_{SM}^0 には重要な過程である。

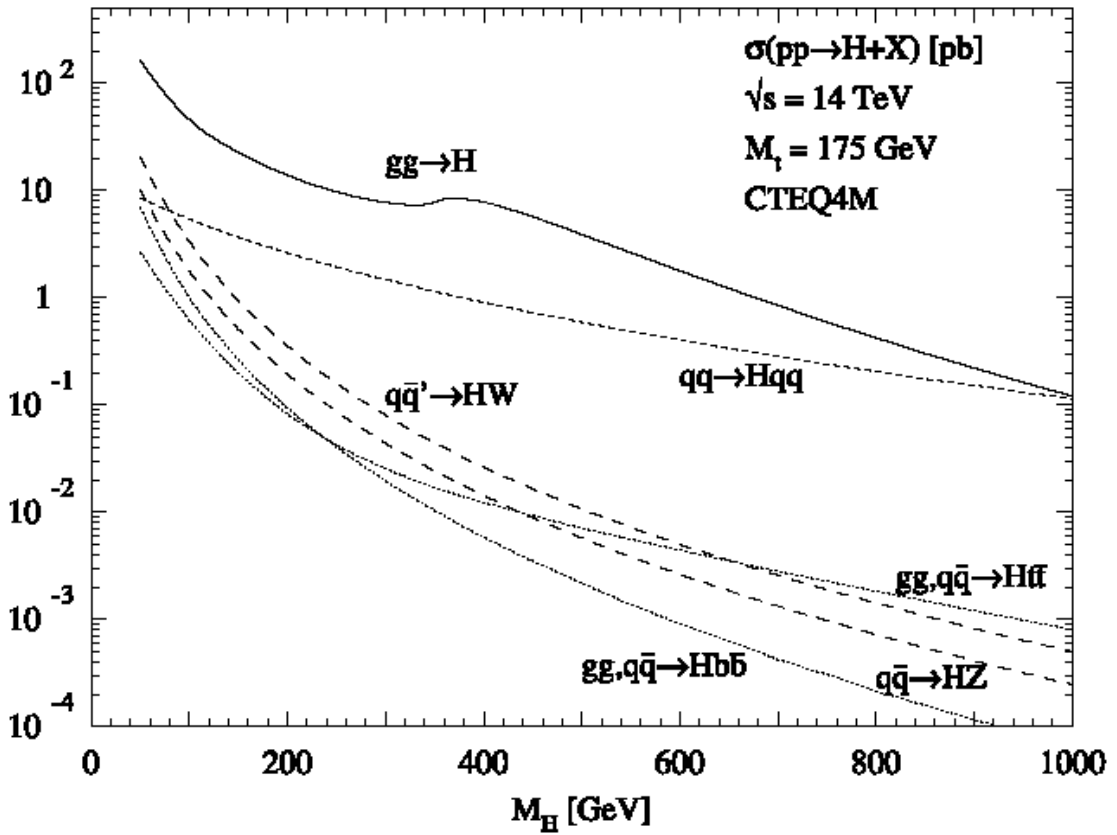


図 7 : SM ヒッグス粒子の生成過程別の散乱断面積

図 7 : SM ヒッグス粒子の生成過程別の散乱断面積にシミュレーションによって得られた陽子陽子衝突でのヒッグス粒子の生成断面積を示した。
 ヒッグス粒子は主にグルオングルオン融合によって生成され、 M_H が大きくなるにつれ WW 融合も同程度になってくることがわかる。
 仮に M_H が 300GeV であった場合、高ルミノシティー運転時 1 年間で $gg \rightarrow H$ によって 10^6 個のヒッグス粒子が生成される。

1.2.3. ヒッグス粒子の崩壊過程

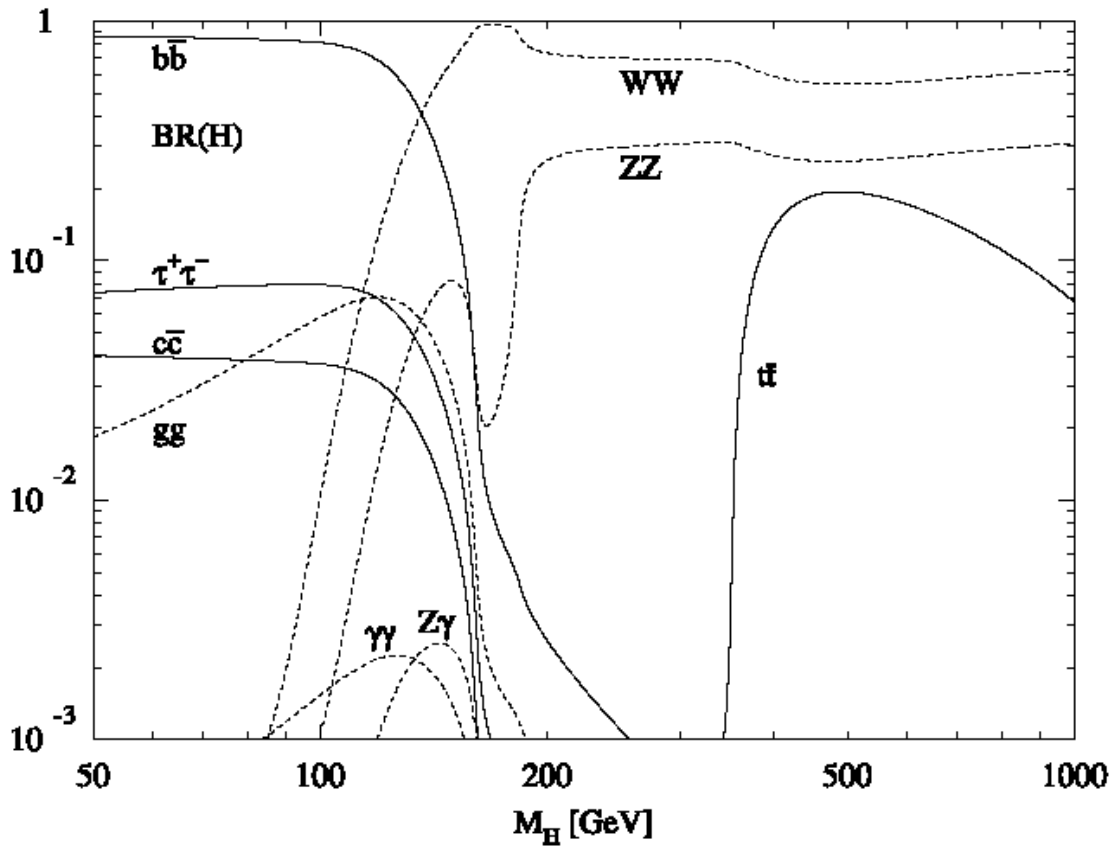


図 8 : SM ヒッグス粒子の主な崩壊過程における分岐比

$90 < m_H < 150 \text{ GeV}$

この領域では $b\bar{b}$, $\tau^+\tau^-$ が多いが、グルオン融合による生成からくる場合 QCD 2jet からのバックグラウンドが大きいのでトリガーをかけるのが困難となる。そこで分岐比は小さいが $H \rightarrow \tau^+\tau^-$ を見る。この領域ではバックグラウンド上の狭いピークを測定する必要がある。そのため電磁カロリメーターには優れた、エネルギーと角度の分解能が求められる。

$130 < m_H < 700 \text{ GeV}$

ここではきれいなピークが期待できる $H \rightarrow ZZ^* \rightarrow 4l^\pm$ を用いる。

$700 < m_H < 1 \text{ TeV}$

この領域では $H \rightarrow WW \rightarrow 2\text{jets}$ と $H \rightarrow ZZ \rightarrow 2l^\pm 2\text{jets}$ を用いる。主なバックグラウンドは $t\bar{t}$, $W+\text{jets}$ と $Z+\text{jets}$ がある。

1.2.4. ヒッグス粒子の発見能力

ATLAS 実験の H_{SM}^0 発見能力を図に示す。この図から 100 fb^{-1} (高ルミノシティー運転時に於いて約 1 年で達する) の積算ルミノシティーで H_{SM}^0 が 5 σ 、 m_H によっては 10 σ 以上の確からしさ²で質量上限の 1TeV の領域で発見することが可能である。

² ATLAS 実験では $S \equiv \frac{\text{Signal}}{\sqrt{\text{Background}}}$ と定義している。

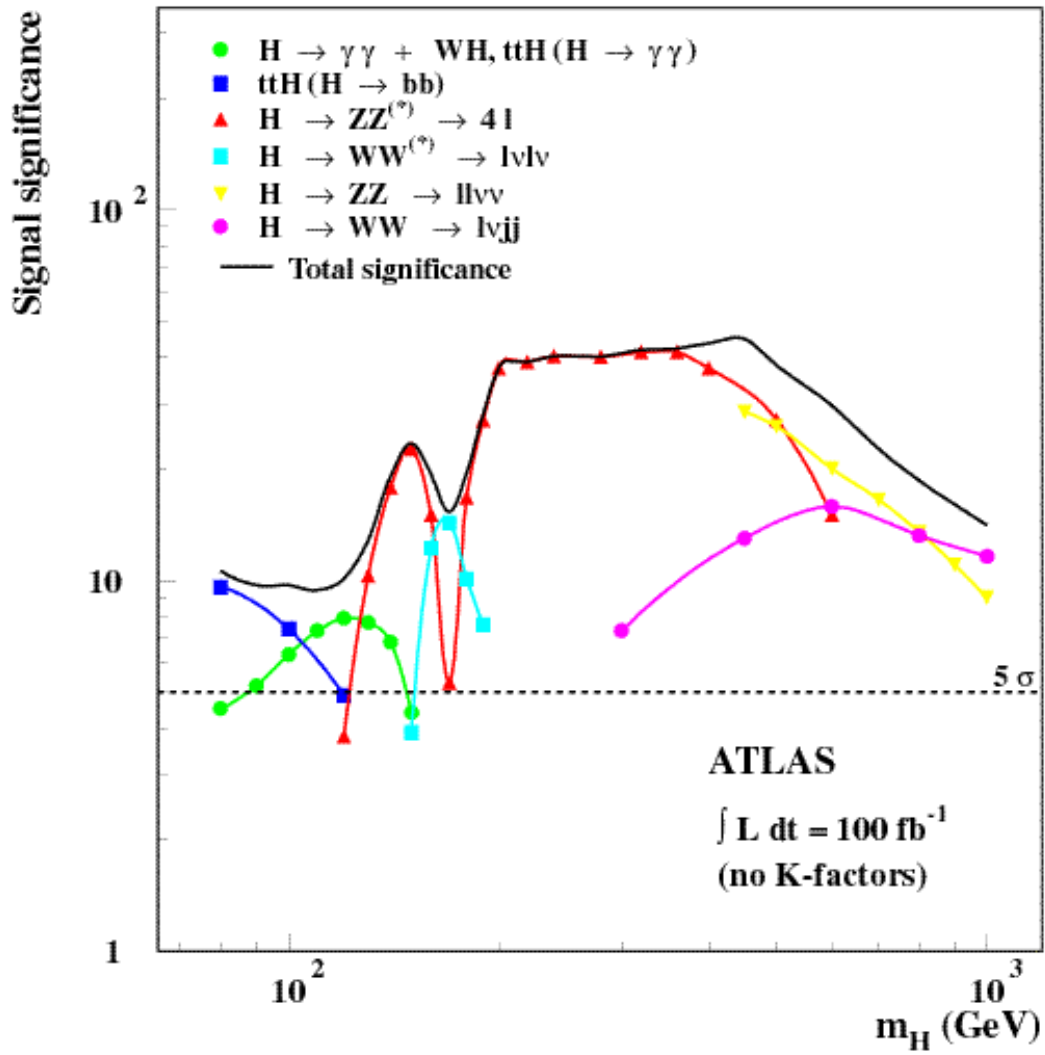


図 9 : ATLAS 実験の H_{SM}^0 発見能力

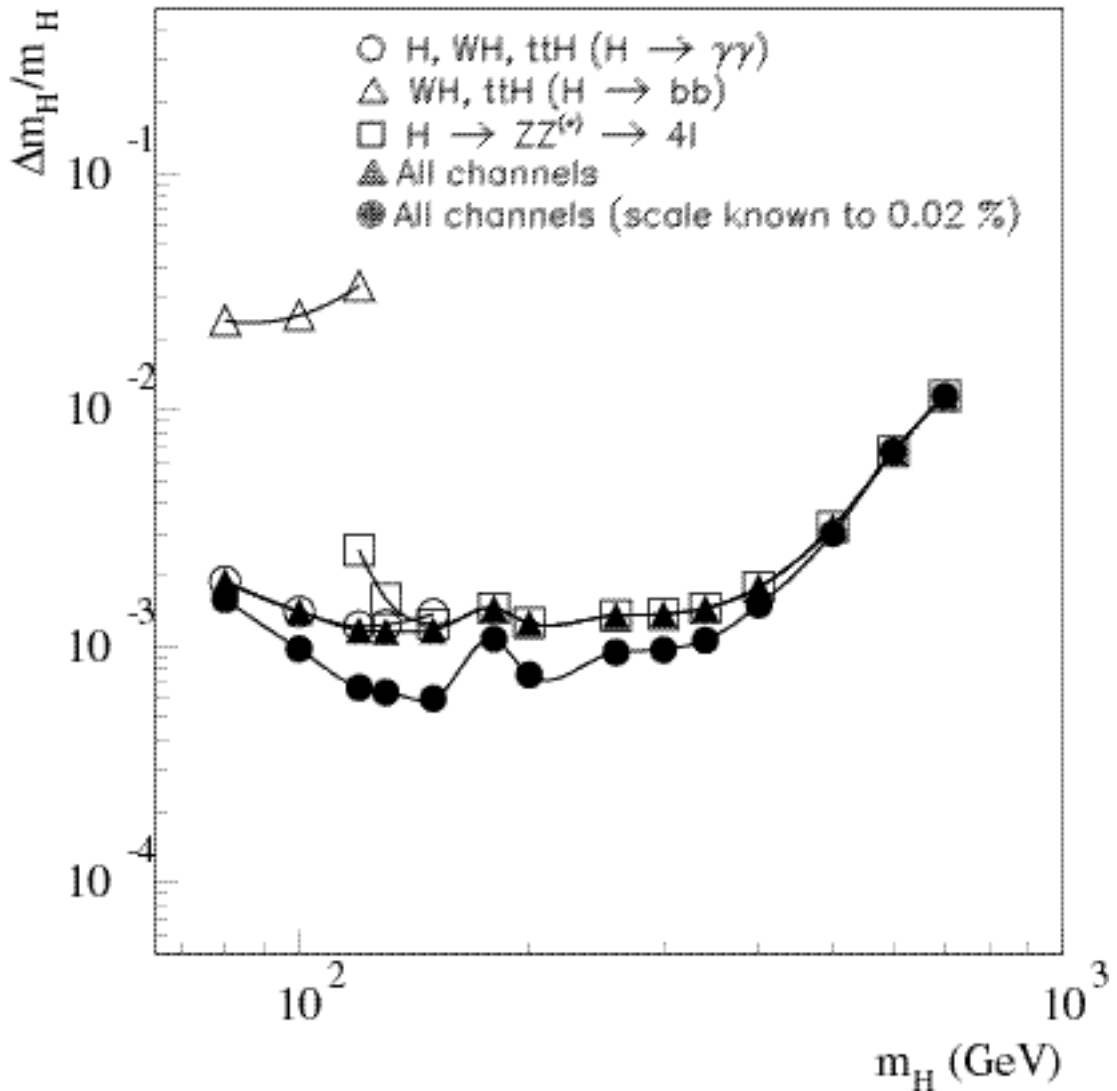


図 10：ヒッグス粒子の質量測定誤差

また図 10：ヒッグス粒子の質量測定誤差のように、発見されたヒッグス粒子の質量を 0.2% の精度で測定することが可能である。

1.2.5. 超対称性など標準理論を超える超対称性粒子などの探索

LEP 実験に於けるゲージ理論の精密検証の結果、超対称性による力の統一の可能性が示唆された。この超対称性は、重力まで統一する理論として最も有望視されているものであり、ATLAS 実験で発見が可能な領域に、数多くの新粒子の存在を予言している。

1.2.6. ボトム・クォークの精密研究

宇宙には、物質と反物質が非対称に存在しており、これが現在の宇宙の豊かな構造の源になっている。この非対称性の謎を解く鍵であると期待されている「CP の破れ」を、ATLAS 実験で生成される B メソンを用いて精密に測定する。また、ボトム・クォークの稀崩壊現象を探索することは、標準理論を超える新しい物理を探る重要な手段でもある。

1.2.7. QCDの精密研究

高い横運動量を持ったジェット・イベントの生成断面積の測定を通して、 10^{-18}cm の極超微細なスケールで、クォークを観察することが可能になり、現在「素粒子」と考えられているクォークに内部構造がないかを探ることが ATLAS 実験では可能である。

1.3. ATLAS 検出器

LHCには4つのバンチ衝突点があり、その1つにLHCのための汎用測定器 ATLAS (A Toroidal LHC ApparatuS) 測定器が設置されている。ATLAS 検出器に要求される性能は $10^{34}\text{cm}^{-2}\text{s}^{-1}$ という高ルミノシティによって電子、光子、ミュオン、ジェットなどが高いレートで検出器に到達するためこれに耐えられ、また確実に測定する必要がある。

ATLAS

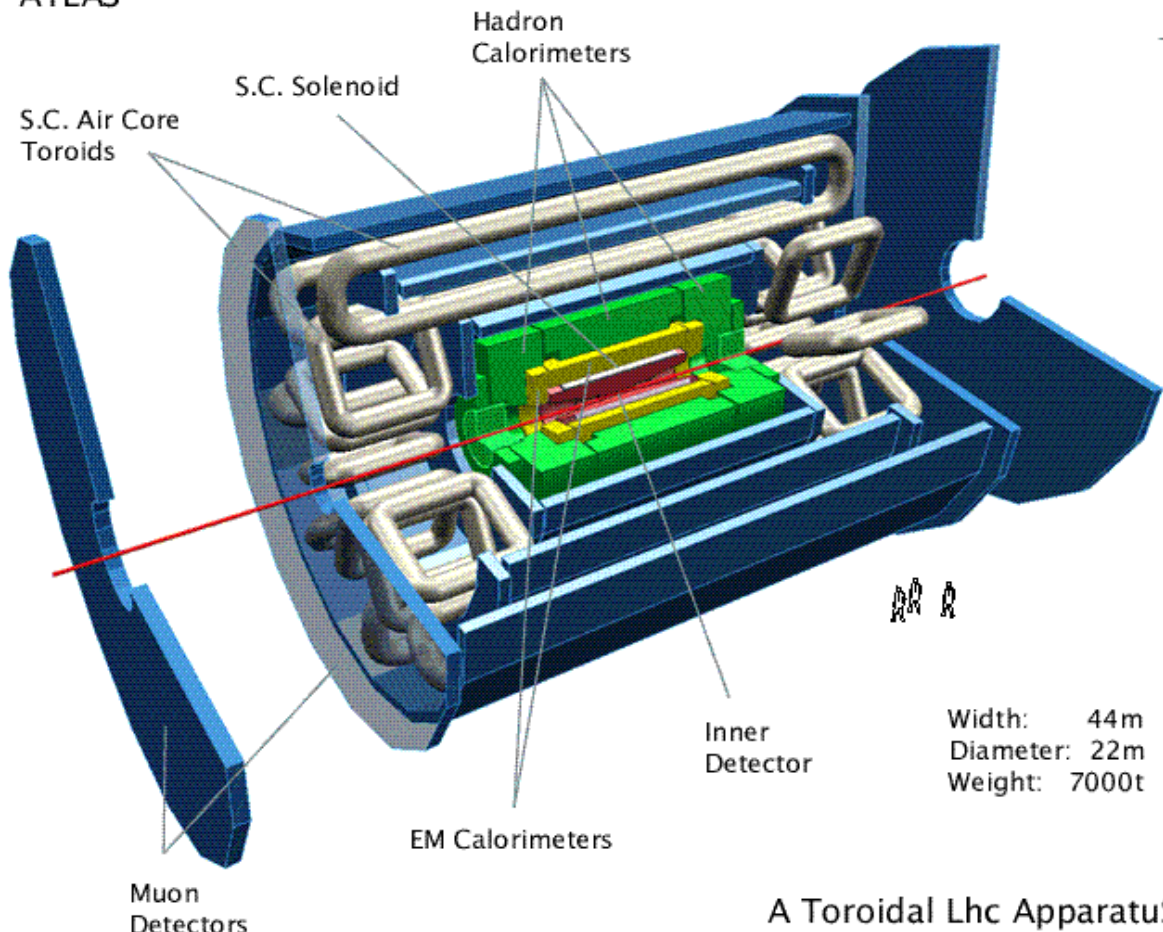


図 11 : ATLAS 検出器

1.3.1. 内部飛跡検出器 (Inner Detector)

Inner Detector は次の検出器から構成される。

- Pixel Detector

最内層に設置され、高精度の位置測定を行う。

- SCT

Pixel Detector と TRT の間に設置され高精度の位置測定を行う。

- TRT

最外層に設置されており、半径 4mm のストローチューブ検出器で、トラッキングの他に遷

移放射を利用した電子識別を行う。

1.3.2. カロリメータ

カロリメータは入射粒子の起こしたカスケードシャワーによって入射粒子のエネルギーを測定する。

1.3.3. ミューオン検出器

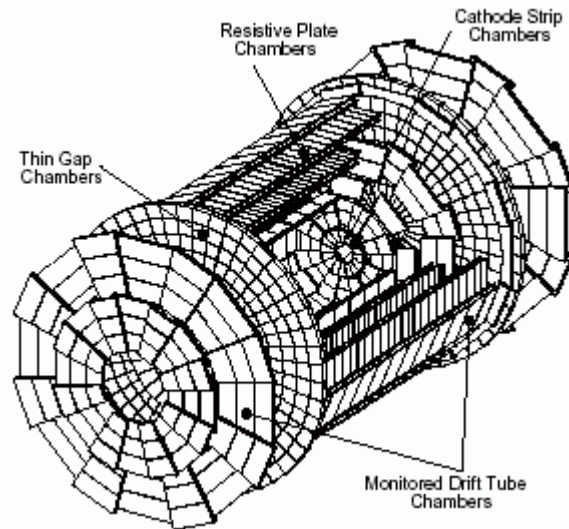
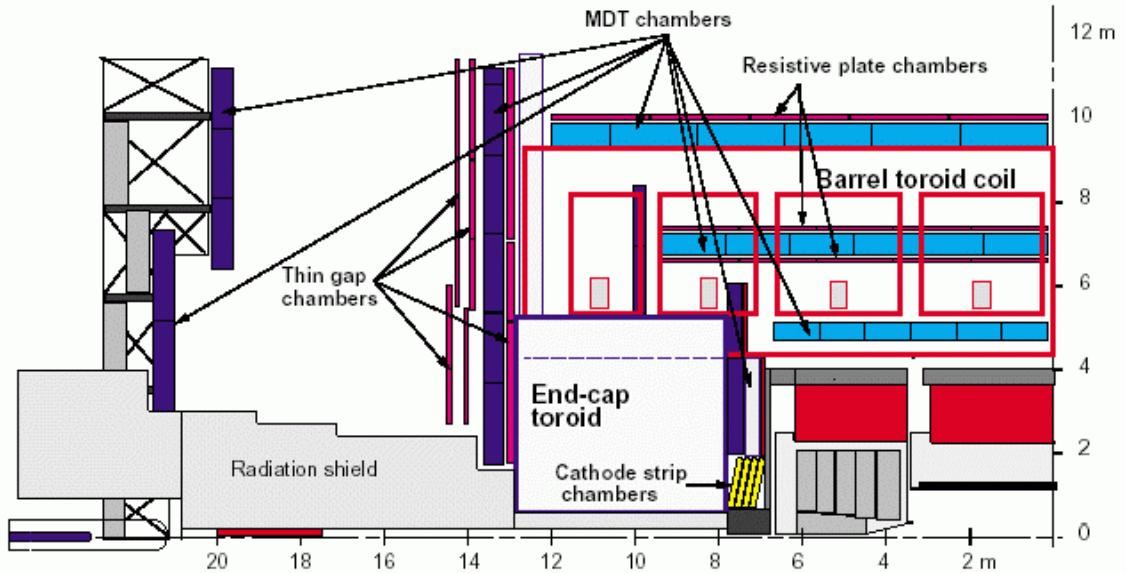


図 12 : ミューオン検出器

ミューオン検出器には次に挙げる検出器がある。

- MDT(Monitored Drift Tube):運動量を高精度で測定をおこなう。
- CSC(Cathode Strip Chamber)
- RPC(Resistive Plate Chamber),TGC(Thin Gap Chamber):RPC はバレル部、TGC はエンドキャップ部のそれぞれミューオントリガーと第二座標の測定を行う

TGCは $1 < |y| < 2.7$ の領域³をカバーし、triplet, middle doublet, pivot doubletと3層から構成される。ガスギャップに張られたワイヤーによってr座標⁴、ワイヤーに対して垂直なカソードストリップによってθ座標を測定する。

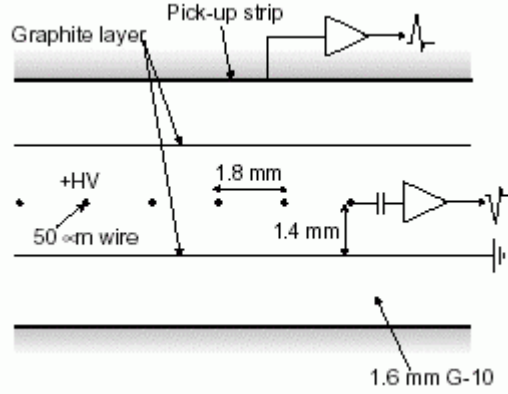


図 13 : TGC の構造

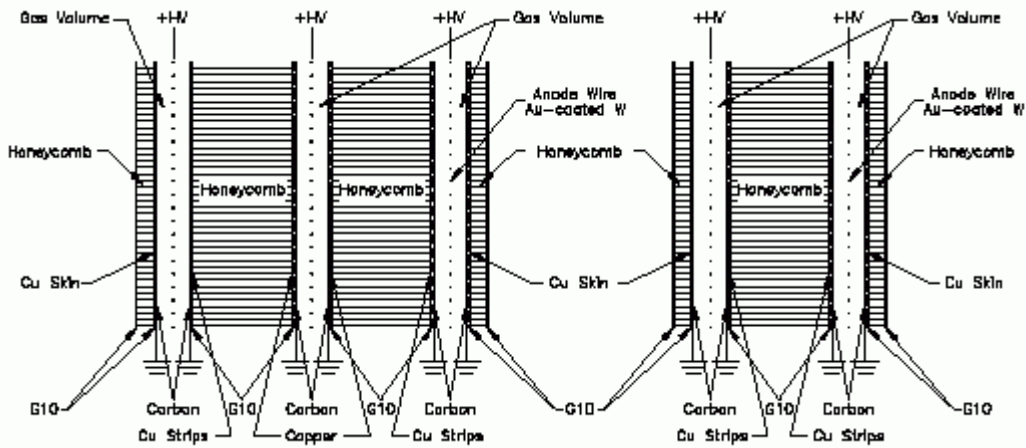


図 14 : TGC のTriplet と Doublet

図 13 : TGC の構造は TGC の断面であり、Graphite にはさまれたガスギャップに 1.8mm 間隔でワイヤーが張られている。Triplet と Doublet の構造を図 14 : TGC のTriplet と Doublet に示した。Triplet はワイヤーの張られているガスギャップ3層とStrip2層、Doublet ではガスギャップ2層Strip2層から構成される。

³ ある位置に飛んでくる粒子数を N とすると $\frac{dN}{d}$ が一定となるよう定義する。ビーム軸と、

InteractionPoint とある位置とをつないだ直線との成す角 θ によって、 $\theta = -\ln \tan\left(\frac{\theta}{2}\right)$ と定義する。

⁴ ATLAS 検出器ではビーム軸方向に z 座標を取った円筒座標系を用いる

2. ATLAS 実験のトリガーとデータ収集システム

LHCでは40MHzでバンチがクロスし1回のクロッシングで平均20個のイベントが発生する。精密な情報全てをデータストレージへ保存することが困難であるので、興味のある重要なイベントのみを選別する必要がある。この選別をTrigger Systemによって行い、最終的に100Hz程度までレートを落とし、ストレージへ保存する。

2.1. 概要

2.2. トリガとデータ収集システムの構成

ATLAS のトリガーとデータ収集 (DAQ) のシステムはその選別の効率を上げるために事象選別を三段階にしている。各段階ではその前の段階の選別によって絞り込まれた領域についての情報を追加することによってより精密な事象選別を行なう。三段階の選別をそれぞれ Level-1 トリガー, Level-2 トリガー, Event filter と呼んでいる。次にそれぞれの選別について説明する。

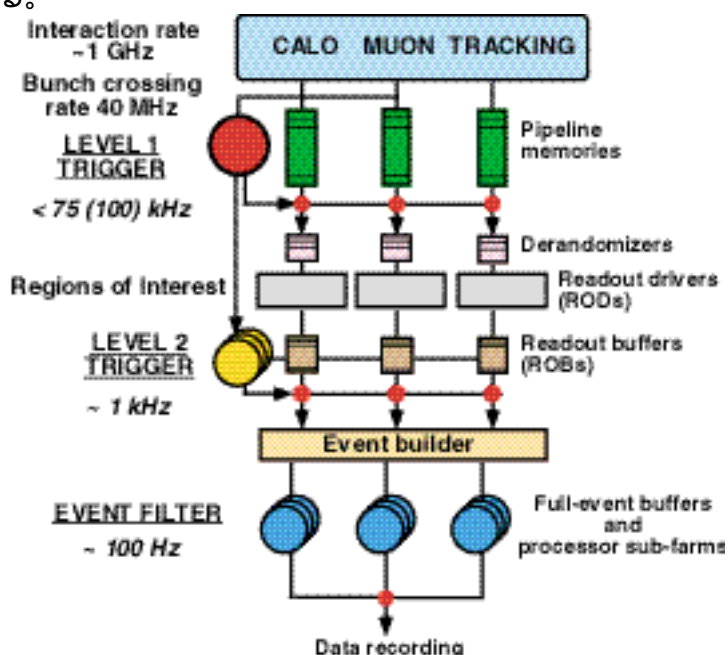


図 15 : トリガーシステムの構成

2.2.1. Level-1 trigger

Level-1(LVL1) トリガーではトリガー用ミュオン検出器から得られる r 座標と θ 座標と横運動量と、全てのカロリメータのエネルギー情報をもとに選別を行なう。Level-1 トリガーの段階ではカロリメータのエネルギー情報は、情報量が非常に多いため位置の精度を落している。この選別により 75kHz までイベント・レートを落す。Level-1 トリガーの Latency⁵ は $2.5 \mu\text{sec}$ である。LVL1 での選別は主にコインシデンス⁶とベターの組み合わせで行なわれる。LVL1 トリガーで選ばれたイベントは Read Out Driver(ROD)によってフロントエンドからデータを転送され Read Out Buffer(ROB) に貯められる。このときデータは時間・空間的に局所的に分布しているので Derandomizer と呼ばれるバッファに一時的にキューとして貯められ転送される。レベル1・トリガーで選択された事象のデータは Level-2

⁵ バンチ衝突後から、LVL1 トリガー判定が LVL2 に送られるまでの、許される時間

⁶ 異なる層の WireGroup(Strip)が、ほぼ同じ の範囲、同じ bunch 内に hit が有った場合そこに粒子の通過があったと判断すること。2 out-of-3 Coincidence とは3層のうち2層以上の hit が有った場合に Coincidence が成立することをいう。

トリガーでの判定が終るまで ROB に保持される。

2.2.2. Level-2 trigger

Level2(LVL2) トリガーでは LVL1 トリガーによって与えられる Region-Of-Interest(RoI) のデータを利用して注目する検出器の領域を絞り込むことで、精度のよい選別を行なう。LVL1 では個々の検出器系で独立して判定が下されたが、LVL2 では内部飛跡検出器を含めて各検出器系からの情報を組み合わせて判定が下される。この RoI のデータに含まれるトリガーの対象の位置情報、横運動量やイベントのエネルギーの和の情報をもとに ROB に保持されている全てのデータの中から必要なデータを選び出して LVL2・トリガーの判定に用いる。LVL2 で必要するイベントデータは LVL1 でのデータよりも精度が高いが、一般に RoI に対応する検出器の全体に対する割合は小さく、データ量の割合は数% 程度である。この RoI の仕組みを使うことでイベントデータの読み出しに必要な時間を抑えることができる。ミュオンについては LVL2 で MDT の情報を追加することで横運動量の精度が高くなるので、閾値付近での選別がより正確になる。またカロリメータの情報を合わせることで、ミュオンの Isolation を判定することが可能になる。LVL2 trigger の Latency は 1msec から 10msec である。

2.2.3. Event Filter

LVL2 トリガー以降は通常オフラインで用いられるアルゴリズムや手法をオンラインに適用したイベント・フィルター(EF) を用いて選別が行なわれる。ここでは最新の検出器の較正や配置、磁場についての情報が用いられる。これは最終的な選別であり、選択されたイベントはオフラインの解析のためにデータ・ストレージに書き込まれる。イベント・フィルターの出力のレートは 100Hz 程度であり、これは平均 2MB/bunch なので 200MB/s に相当する。

2.3. Level-1(LVL1) Muon Trigger System

Muon trigger system の役割はミュオン以外の粒子によるチェンバーの信号や検出器などのノイズを落とし、高い Pt(横運動量)のミュオンを含む事象を選択することである。その際にミュオンの Pt について閾値を設けて、その閾値以上の Pt を持つミュオンだけを選択する。ミュオンの Pt の閾値は Low Pt 用 3 段階、High Pt 用 3 段階の計 6 段階が用意されている。LVL1 ミュオントリガーシステムの機能は

- LVL2 トリガーのための RoI 情報の生成
- CTP(Central Trigger Processor) へ送られる Pt 閾値ごとのミュオンの数の算出
- DAQ システムのためのデータの読み出し
- 検出器またはエレクトロニクスの状態のモニタリング
- 検出器のコントロール(パラメータの設定など)
- トリガーシステム自身のテスト・診断

などがある。Pt 閾値の定義はこの値より大きな Pt のミュオンのトリガー効率が 90% 以上になる Pt の閾値である。また性能に対する要求は

- Pt threshold を 6~35GeV の範囲で設定できる。
- $|\eta| < 2.4$ の範囲においてミュオンに対する平均のアクセプタンスが 90% 以上である。
- 設定した Pt 閾値でトリガー効率が鋭く落ちている。鋭さの定義の例として、実際の Pt が設定した閾値よりも大きいミュオンが 25% 以上あることなどである。しかしこの要求はミュオンがどのような過程(W、Z の崩壊、b クォークの崩壊など) から生じたかを考慮しなければならない。

などがある。LVL1 muon trigger では RPC と TGC の二種類のトリガーのためのミュオン検出器が用いられる。 $|\eta| < 1.05$ の領域(バレル部)のトリガーは RPC が行ない、 $1.05 < |\eta| < 2.4$ の領域(フォワード・エンドキャップ部)のトリガーは TGC が行なう。Muon trigger system の中で Level-1 endcap muon trigger では主に chamber の信号のコインシデンスを要求することでバックグラウンドやノイズなどによる偶発トリガーを防ぐ。こ

のコインシデンスの判定は後述する Slave Board, High-Pt Board, Sector Logic で段階的に行なわれる。今後、コインシデンスの条件を満たした信号をトラックと呼ぶことにする。特に Slave Board のコインシデンスの条件を満たした信号を Low Pt トラックと呼び、High-Pt Board のコインシデンスの条件を満たした信号を High Pt のトラックと呼ぶことにする。Pt の閾値による選別を行なうためには、Level-1 トリガーでもミュオンの Pt を求める必要がある。TGC ではトロイド磁場によるミュオンの変向角から Pt を測定する。低い Pt のミュオンは Slave Board で求めた Middle Doublet と Pivot Doublet の間での r 方向、 θ 方向のミュオンのヒット位置の差 R_{θ} から運動量のそれぞれの方向の成分を測定する。同様に高い Pt のミュオンは High-PT Board で求めた Triplet と Pivot Doublet の間の R_{θ} から運動量のそれぞれの方向の成分を測定する。Sector Logic では High-PT Board からの R_{θ} について Pt の閾値ごとに R- Coincidence を行う。ここで Pt の大きさと R_{θ} 対応づけられ、閾値ごとにトリガーしたミュオンの数が決定する。TGC からの信号は Patch Panel でバンチクロッシング ID、TGC の重なり部分にある WireGroup の OR'ing の処理された後、Slave Board では WireGroup, Strip の信号それぞれについて、triplet と doublet のそれぞれでのコインシデンスがとられる。次に High PT Board では triplet と doublet のコインシデンスの結果を用いて WireGroup, Strip それぞれの信号でのコインシデンスを判定する。最後に Sector Logic では WireGroup と Strip の信号、つまり r 座標と θ 座標についてのコインシデンス (R- Coincidence) を判定する。最終的には Pt の閾値ごとのトリガーしたミュオンの個数を Level-1 trigger の結果として出力する。

2.4. 前後方ミュオントリガーシステム

ここからがトリガーシミュレーションが実装すべきパートとなる。主に機能に絞ってみたいことにする。

2.4.1. 前後方ミュオントリガーシステム用 TGC の構成と配置

Trigger System では複数の TGC を 1 枚の TGC Sector として扱い、TGC Sector 単位で処理を行う。このとき TGC の境目に不感領域を作らないために TGC は重ねられ配置される。この重なりは z の小さいほうを front、大きいほうを back と呼ぶ。また TGC Sector も同様の理由で重ねられ、現在の設定では Even Sector では z が小さく、Odd Sector では z の大きい位置に置かれている。

2.4.1.1. Trigger Sector

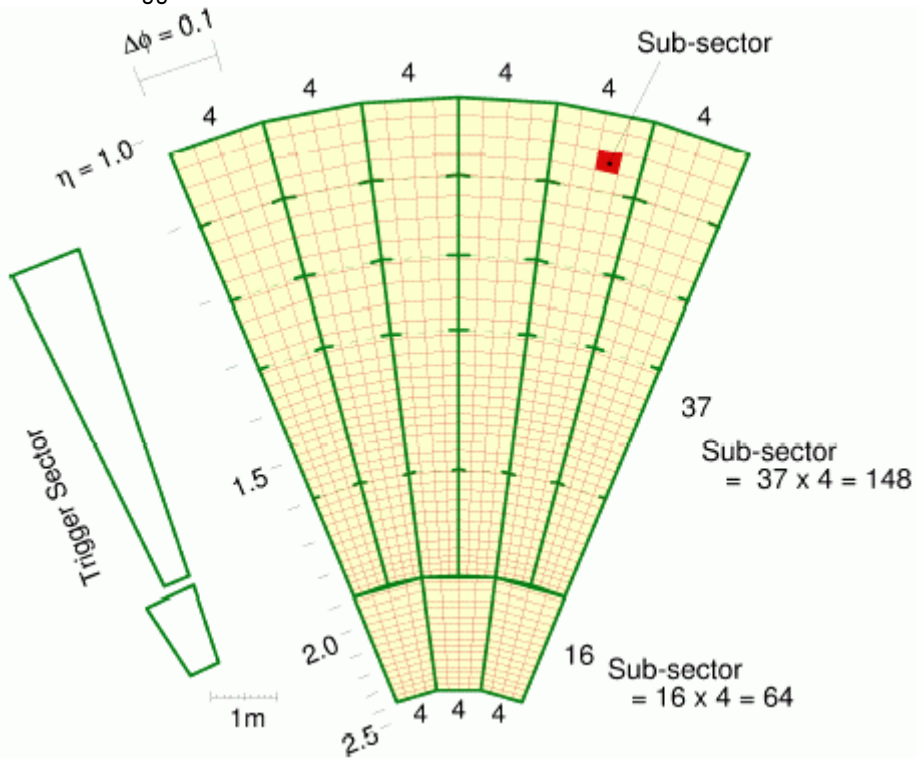


図 16 : Trigger Sector の構成 (x-y 断面)

図 16 : Trigger Sector の構成 (x-y 断面)の Sub-Sector とは SectorLogic が R-Coincidence を行うに R-Coincidence Matrix⁷ がカバーする範囲に対応している。この図に表示されている、Forward⁸, Endcap 含めた合計 9 Sector をまとめて Octant と定義する。

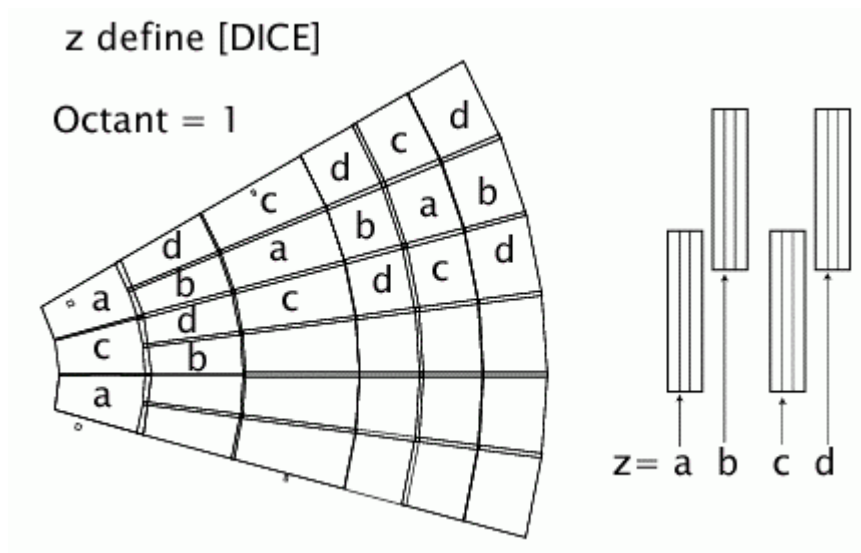


図 17 : TGC の重なり

⁷ 45 で詳しく説明するが、R、を合成し Pt を決定するものである。

⁸ $1.0 < \eta < 1.9$ の領域を Endcap, $1.9 < \eta < 2.4$ の領域を Forward と呼ぶ。

TGCの重なりと、TGC Sectorの重なりによって station⁹内の TGC は4つの z 位置を持つことになる。図 17 : TGC の重なりには TGC を表す箱が4つ示されており、1つの箱には Wire を含む gasgap を表す線が引かれている。この図の場合 Doublet の2本が描かれており Triplet の場合は3本となる。

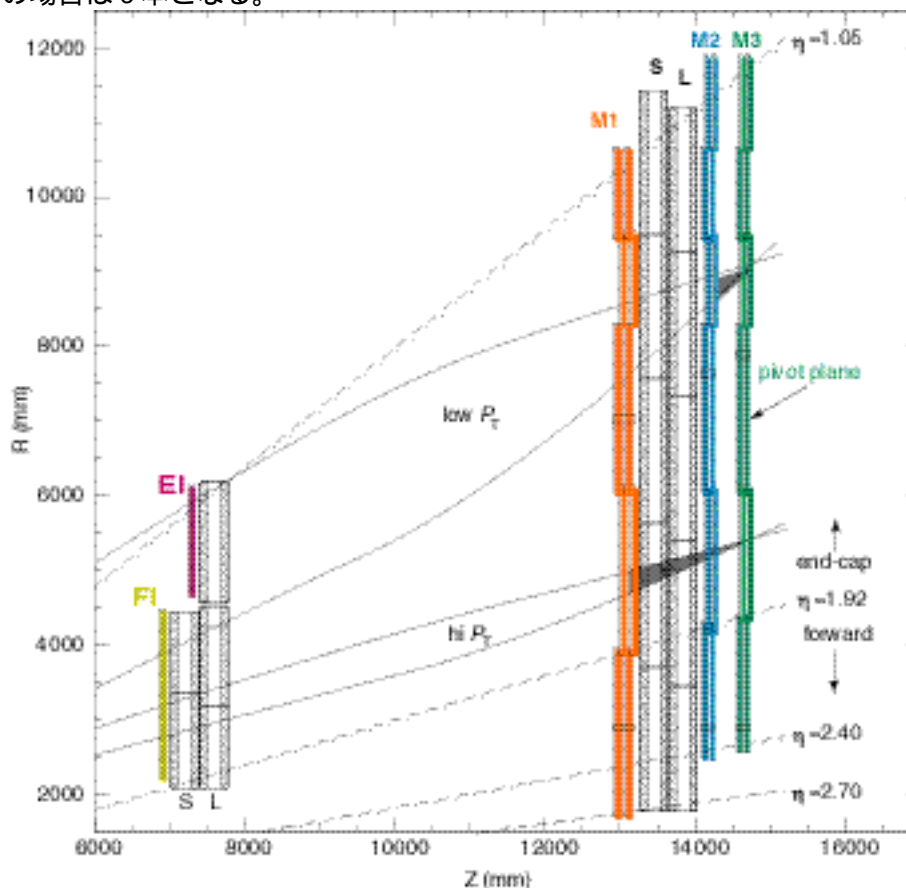


図 18 : TGC の配置 (r-z 断面)

TGC は $z=7\text{m}$ 付近に設置された EI/FI, $z=13\text{m}$ 付近の Triplet (M1, ST1)、 $z=14\text{m}$ 付近の Middle Doublet (M2, ST2)、そして $z=14.5\text{m}$ 付近の Pivot Doublet (M3, ST3) と配置されている。

⁹ Triplet, Middle Double, Pivot Doublet をそれぞれまとめたものを station と呼び ST または M と表す。

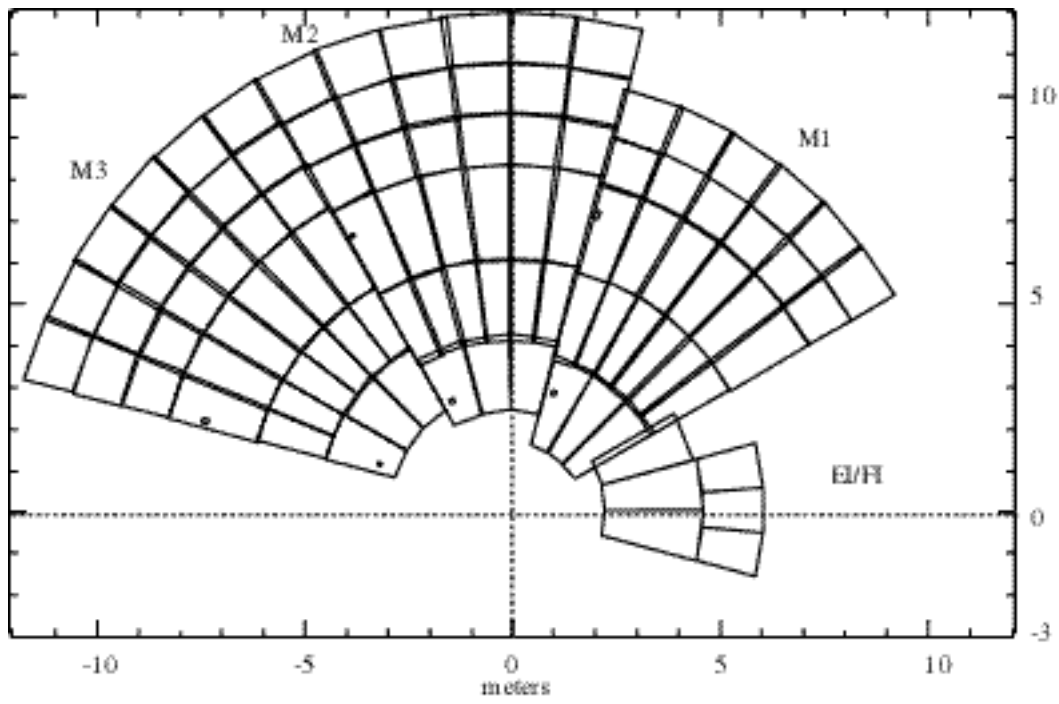


図 19 : TGC の構成 x-y 断面

2.4.2. Trigger の方法

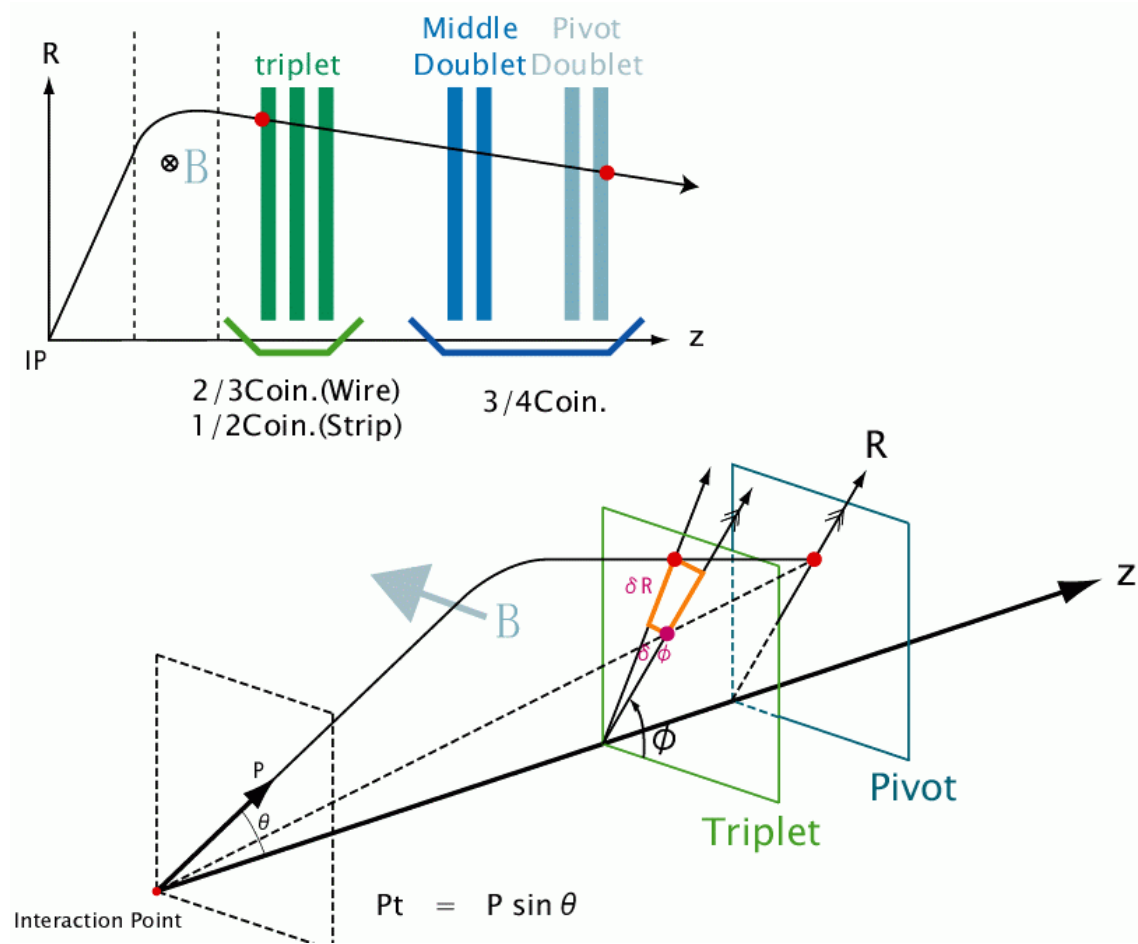


図 20 : Trigger の方法

Interaction Point で生成された粒子は磁場のある領域で曲げられた後、Triplet plane, Middle doublet plane, Pivot doublet plane を通過していく。このとき Pivot plane を通過した場所と Interaction Point とを結んだ直線と Triplet plane との交点と、track が Triplet を通過した点との差 (R , ϕ) によって P_t を求める。この P_t によって Trigger するかどうかを決定する。

2.4.3. Board Connection

Sector 内に設置されるボードと接続を図 21 : ミューオントリガーシステムの構成に示す。WireGroup から送られる信号は WireTripletSlaveBoard, WireDoubletSlaveBoard で処理され WireHighPtBoard へと送られる。一方 Strip の信号は StripTripletSlaveBoard, StripDoubletSlaveBoard を経て StripHighPtBoard へと送られる。

最後に WireHighPtBoard と StripHighPtBoard で決まった R , ϕ 情報と EI/FI の情報が SectorLogic へと送られる。

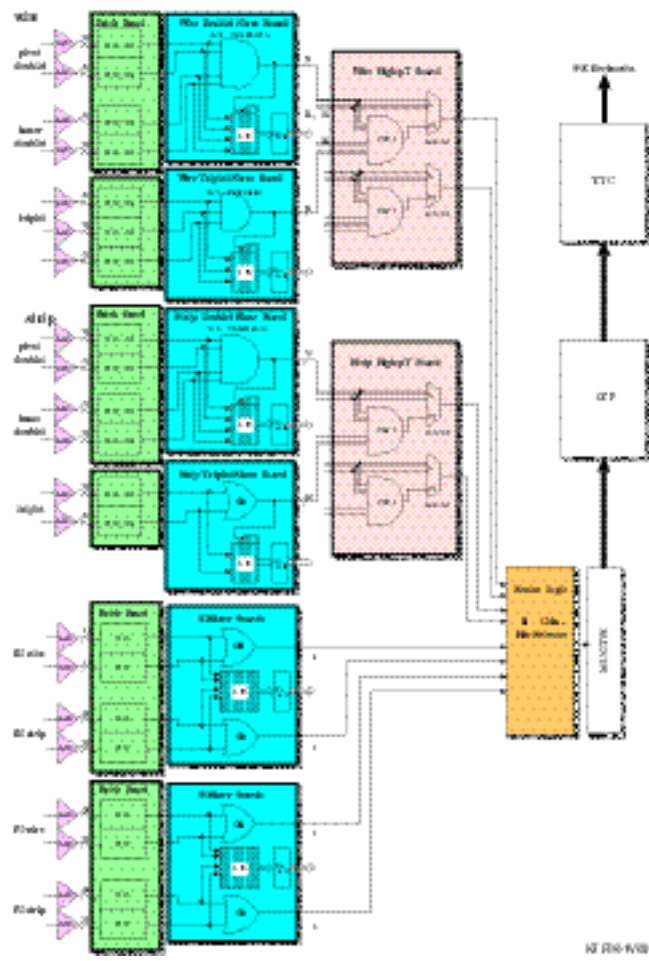


図 21 : ミューオントリガーシステムの構成

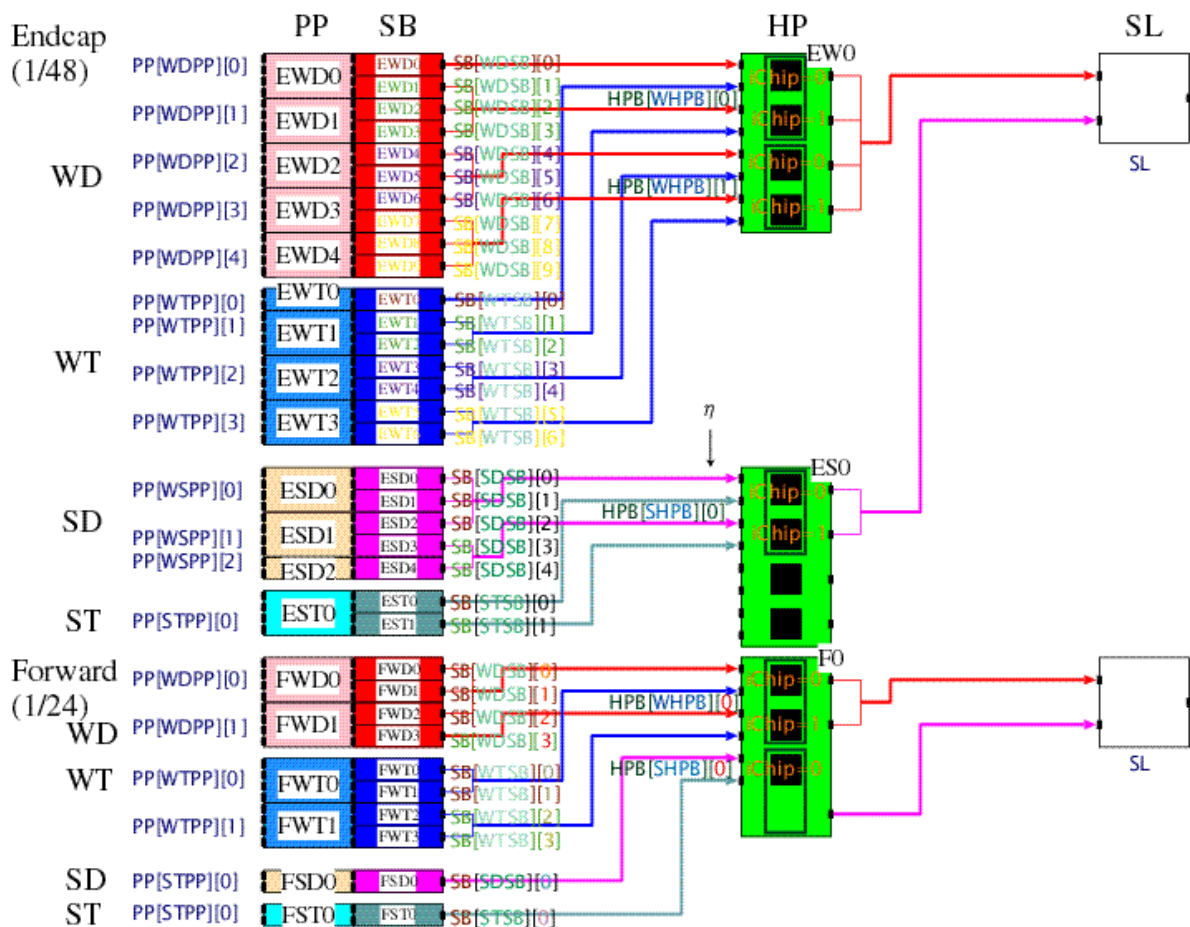


図 22 : ボードの接続とネーミング

図 22 : ボードの接続とネーミングは 1 つの Sector の処理に関わるボードとその名前が記されている。箱の中の名前がボードで名で、NAME[type][number]となっているのはソフトウェアでのインスタンスである。PatchPanel を例として次に挙げる。

PatchPanel : NAME PP

| ボード名 | BoardType | enum |
|----------|-----------|------|
| EWT, FWT | WTPP | 0 |
| EWD, FWD | WDPP | 1 |
| EST, FST | STPP | 2 |
| ESD, FSD | SDPP | 3 |

ボード名の E は Endcap, F は Forward であり、S は Strip、W は Wire としている。

2.4.4. PatchPanel

PatchPanel の機能はバンチクロッシングの Identification(ID)、SlaveBoard への信号の分配、信号の OR'ing であるが、シミュレーションではバンチクロッシング ID 付けを行う必要がないため PatchPanel 内での connection と OR'ing についてのみ説明する。

2.4.4.1. Connection

PatchPanel Id, connector Id, 接続される SlaveBoard Id を図 23 : EndcapWireDoubletPatchPanel から図 27 : StripTripletPatchPanel に示す。

PatchPanel Endcap Wire Doublet

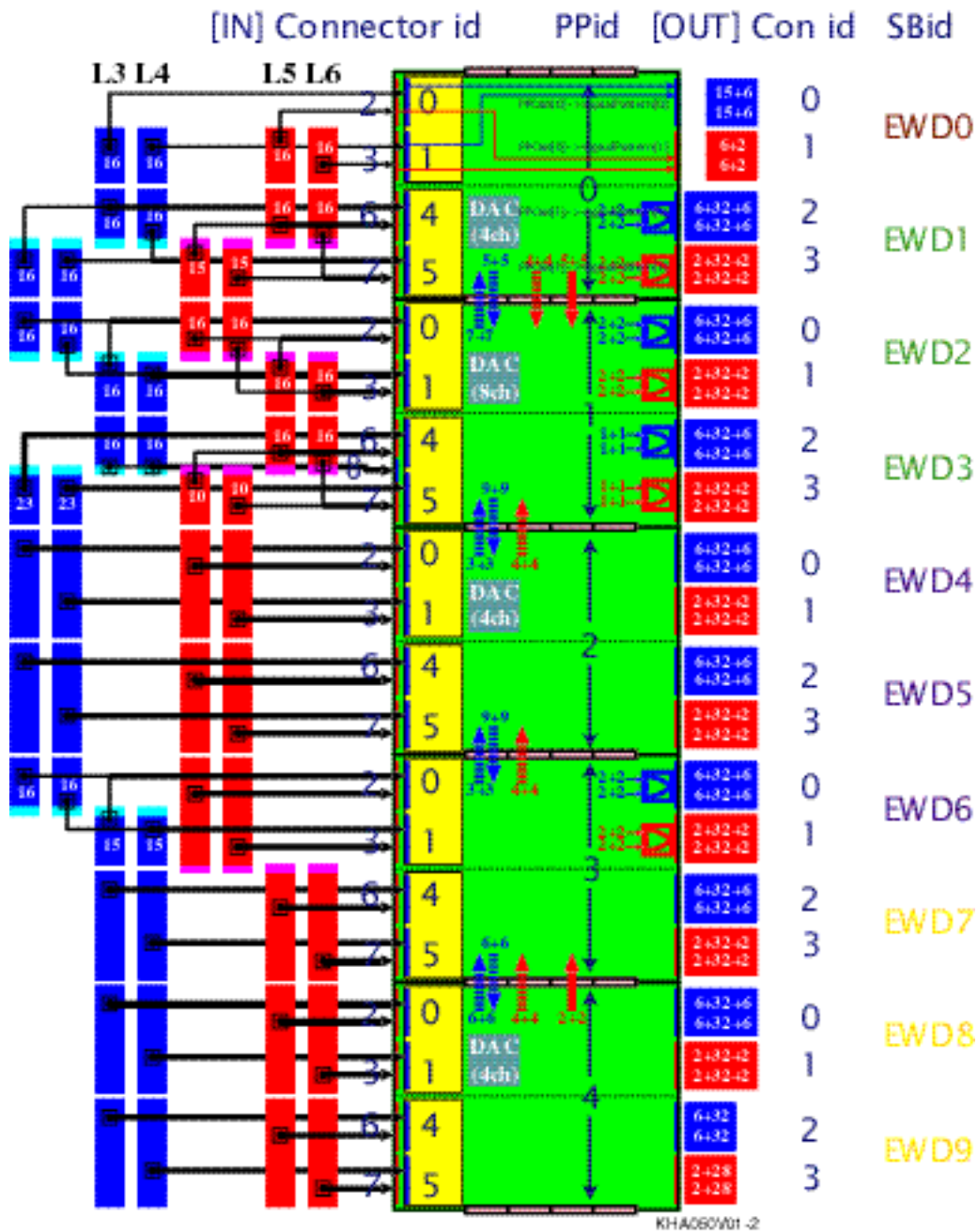


図 23 : EndcapWireDoubletPatchPanel

L3,L4 という表記は

- L0 Triplet Layer0
- L1 Triplet Layer1
- L2 Triplet Layer2
- L3 Middle Doublet Layer0
- L4 Middle Doublet Layer1
- L5 Pivot Doublet Layer0

L6 Pivot Doublet Layer1

としている。図の中央にあるのが PatchPanel のボードで、そこに左側から TGC からの入力
が示されている。L0~6 で描かれている箱がオフセットされているのは TGC Sector 内の
front,back を表しており、左側が front,右側が back となっている。

[IN]Connector id, Ppid, [OUT]Coin id の下にはソフトウェア内での Numbering が記され
ている。

PatchPanel Endcap Wire Triplet

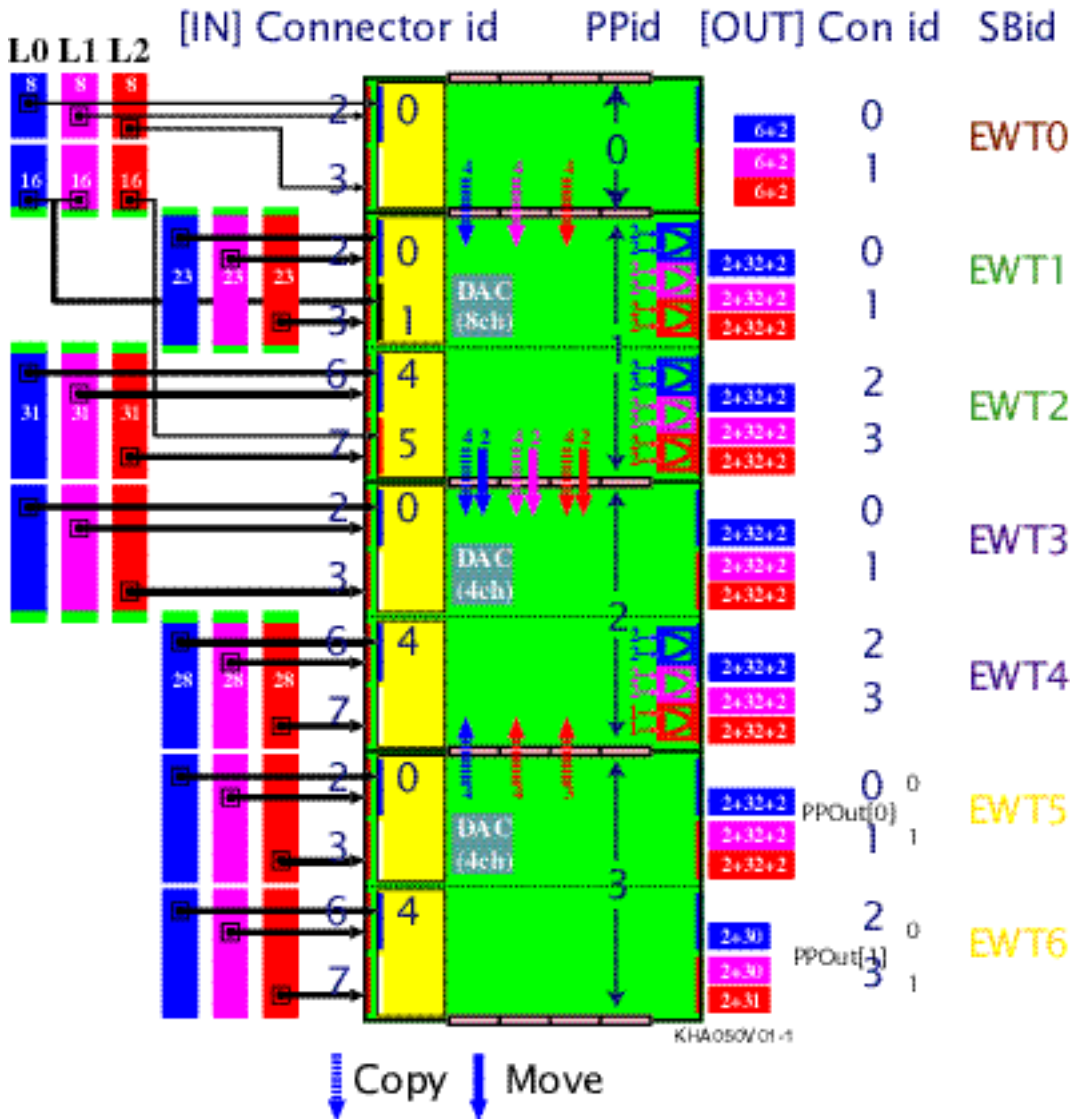
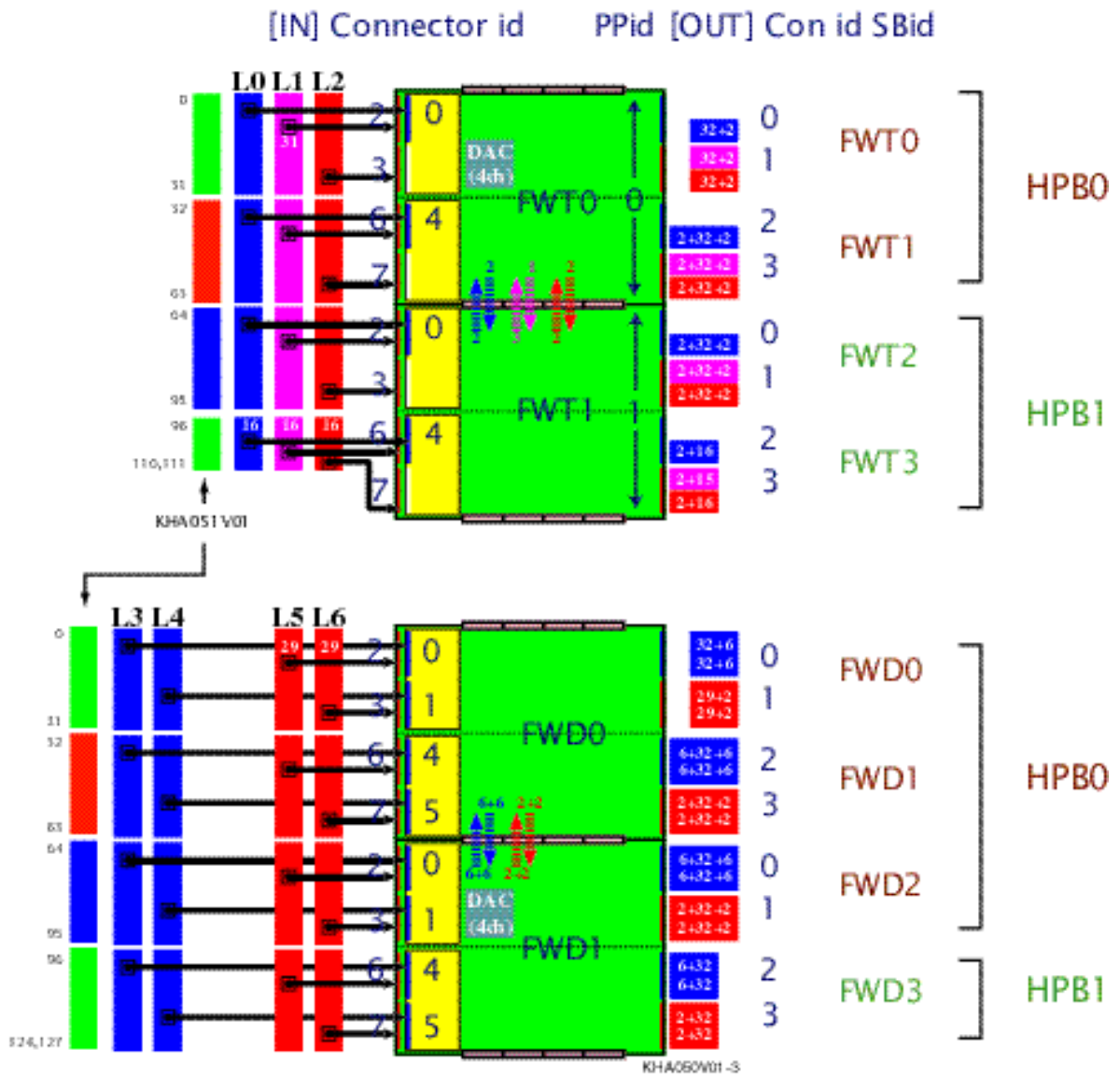


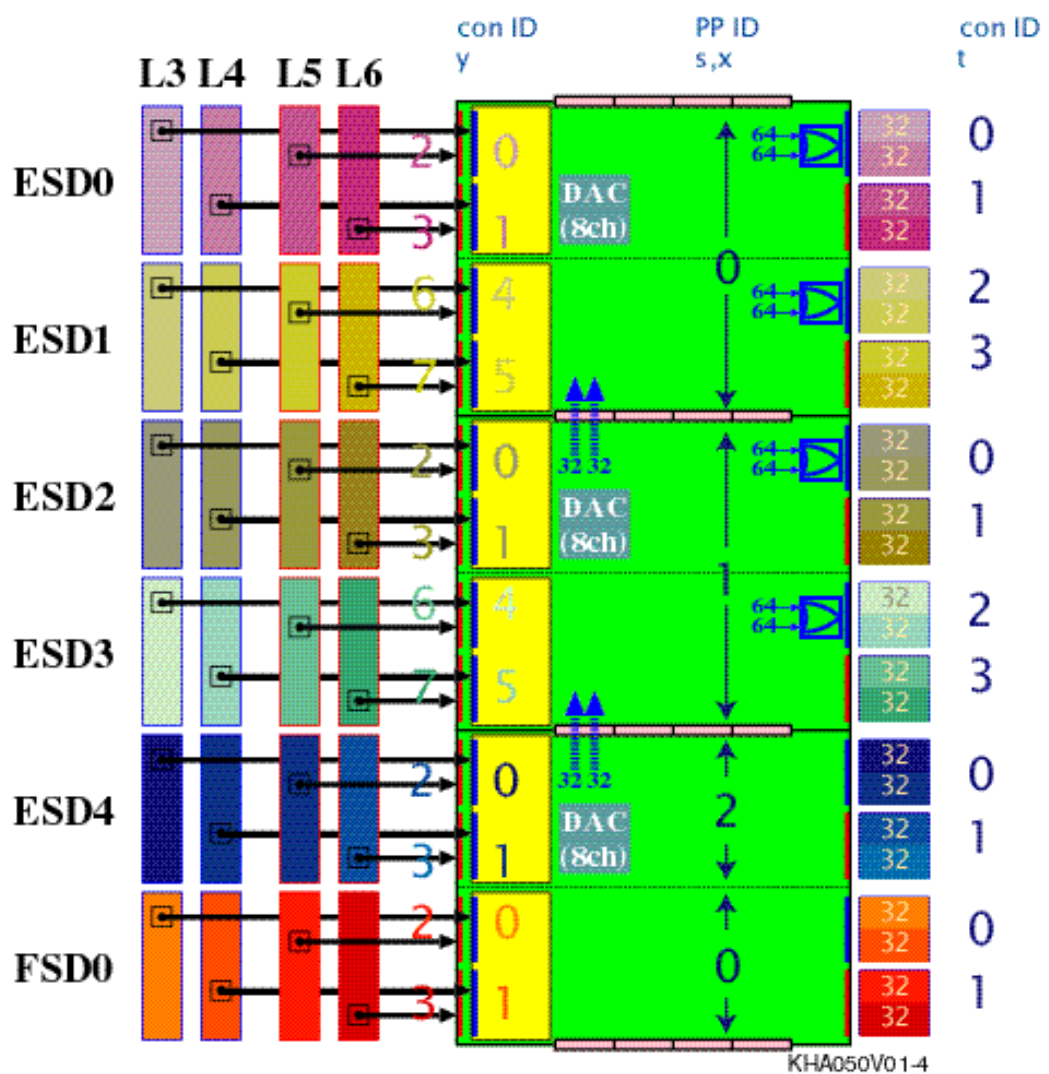
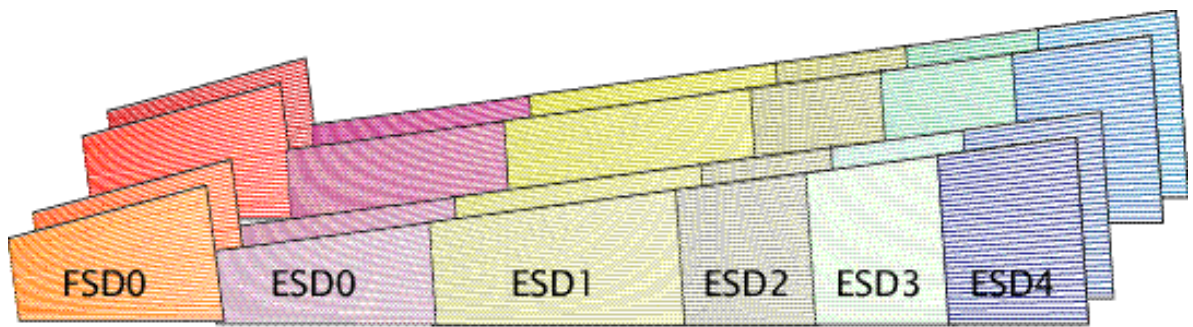
図 24 : EndcapWireTripletPatchPanel

同じ HighPtBoard に接続される SlaveBoard Id は同色で描かれている。

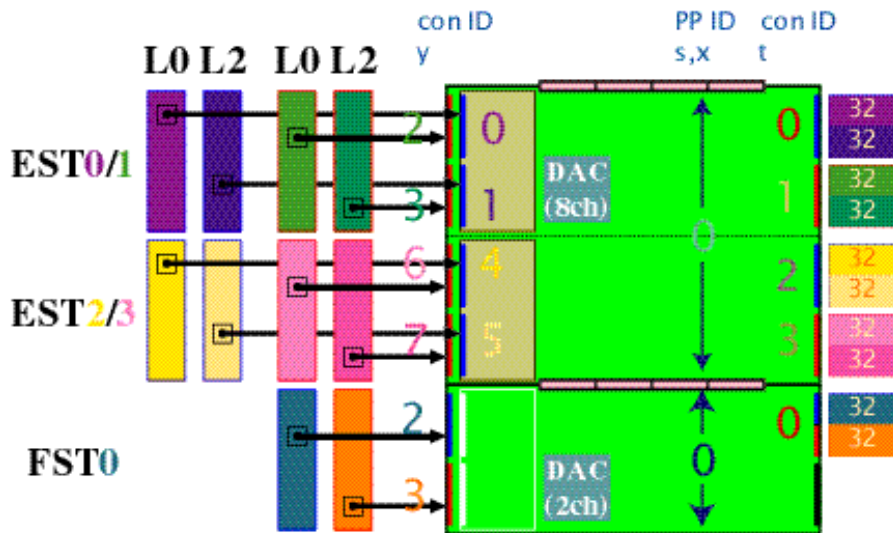
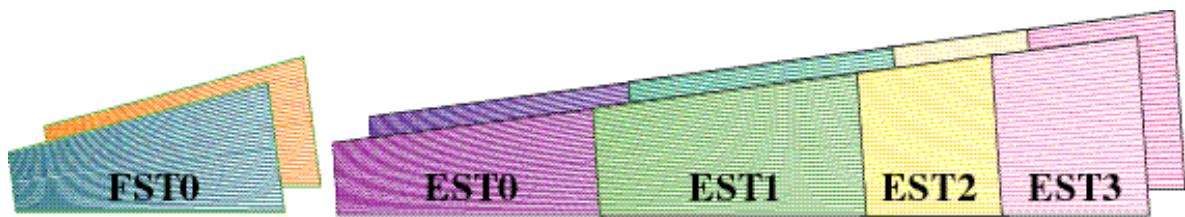
PatchPanel Forward Wire



☒ 25 : ForwardWirePatchPanel



☒ 26 : StripDoubletPatchPanel



| pp.db | EST | 256 | 64 | 64 | 64 | 64 |
|-------|-----|-----|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 2 | 0 | 0 | 0 | 2 |
| 0 | 0 | 3 | 0 | 0 | 0 | 3 |
| 0 | 0 | 4 | 0 | 0 | 0 | 4 |
| ... | | | | | | |

s t uu x y zz (X Y ZZ)
 s = output PP ID
 t = output-connector ID
 uu = output-channel ID of a connector
 x = input PP ID
 y = input-connector ID
 zz = input-channel ID of a connector

図 27 : StripTripletPatchPanel

2.4.4.2. OR'ing

TGC の境界では inefficient な領域を作らないために重なりがあるが、Trigger system では TGC Sector を 1 枚の chamber として処理を行う。このため back と front の TGC の重なり部分にある WireGroup は trigger system 内部では同じ WireGroup として扱う。このための OR'ing を PatchPanel で行う。

TGC 内での WireGroup の Numbering は、台形の短辺側からスタートを 1 として決める。TGC Sector 内で 1 枚の chamber として扱ったときの Numbering を GlobalWireGroupName と呼び、の小さいほうから 0 をスタートとしている。

2.4.5. SlaveBoard

SlaveBoard の主な機能を挙げる。

- WireDoubletSlaveBoard

2 Doublets の 4 層あるチェンバーからの位置を表す WireGroup の信号に対し 3 out-of 4 のコインシデンスをとりヒット位置¹⁰を出力する。コインシデンス条件を満たすトラックに

¹⁰ Coincidence を満たした track の Pivot plane での GlobalWireGroupName をヒット

については MiddleDoublet, PivotDoublet 間の R または を出力する。

- WireTripletSlaveBoard

Triplet の 3 層ある WireGroup からの信号に対し 2 out-of 3 のコインシデンスをとりヒット位置だけを出力する。

- StripTripletSlaveBoard

Triplet の 2 層ある Strip からの信号に対し 1 out-of 2 のコインシデンスをとりヒット位置だけを出力する。

- StripDoubletSlaveBoard

2 Doublets の 4 層ある Strip の信号に対し 3 out-of 4 のコインシデンスをとりヒット位置を出力する。コインシデンス条件を満たすトラックについては MiddleDoublet, PivotDoublet 間の R または を出力する。

2.4.5.1. Input

PatchPanel1 枚に対し SlaveBoard は 2 枚が接続される。そのデータを図 SlaveBoard の入力ビットに示す。WDSB (WireDoubletSlaveBoard) を例に挙げて説明する。WDSB には PivotDoublet から 36channel が入力される。Doublet は 2 枚からなるので 36x2 の入力がある。MiddleDoublet からは 44channel が入力されこれも 2 倍の 44x2 の入力、WDSB には 36x2+44x2channel の入力があることになる。

Layer と書かれている下の箱の中にはソフトウェア内での Numbering が書かれている。

位置と呼ぶことにする。

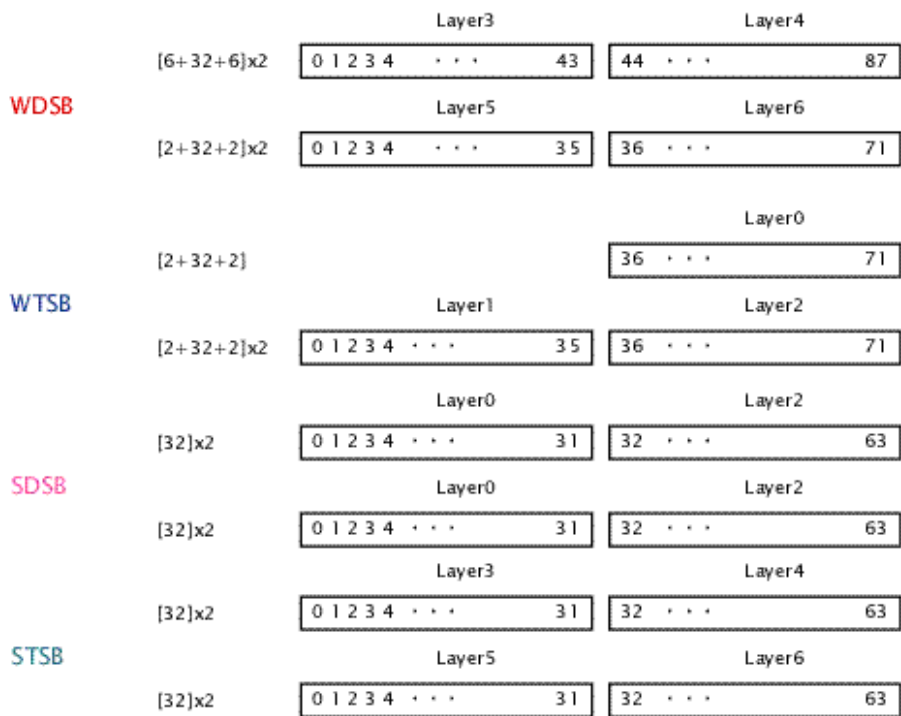
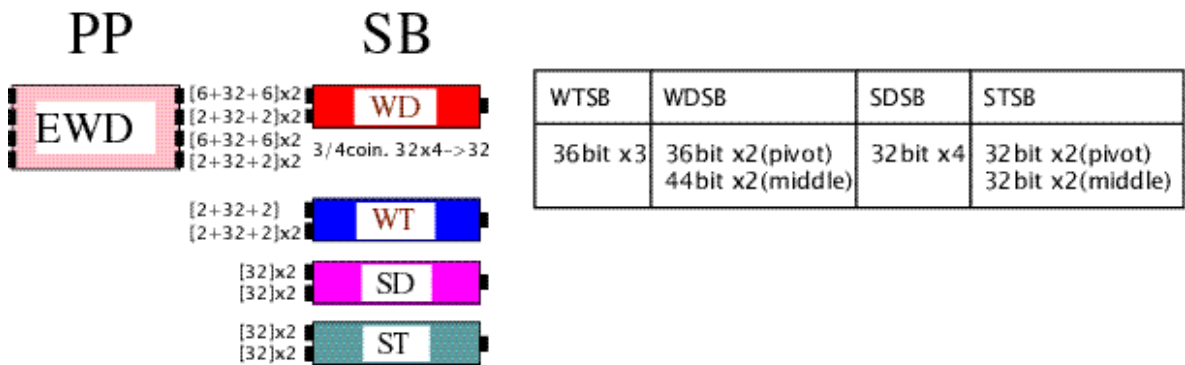


図 28 : SlaveBoard の Input-bit 数

2.4.5.2. Coincidence Logic

入力された信号に対しコインシデンスを取る。WireTriplet では通常 2 out-of 3 でコインシデンスを取るがバックグラウンドの頻度が高い場合のために 3 out-of 3 のロジックも用意する。DSB のコインシデンスロジックは Middle が行、Pivot が列として入力されるマトリックス状になっているためコインシデンスマトリックスと呼ぶ。入力された信号のコインシデンスマトリックスへの接続を示す

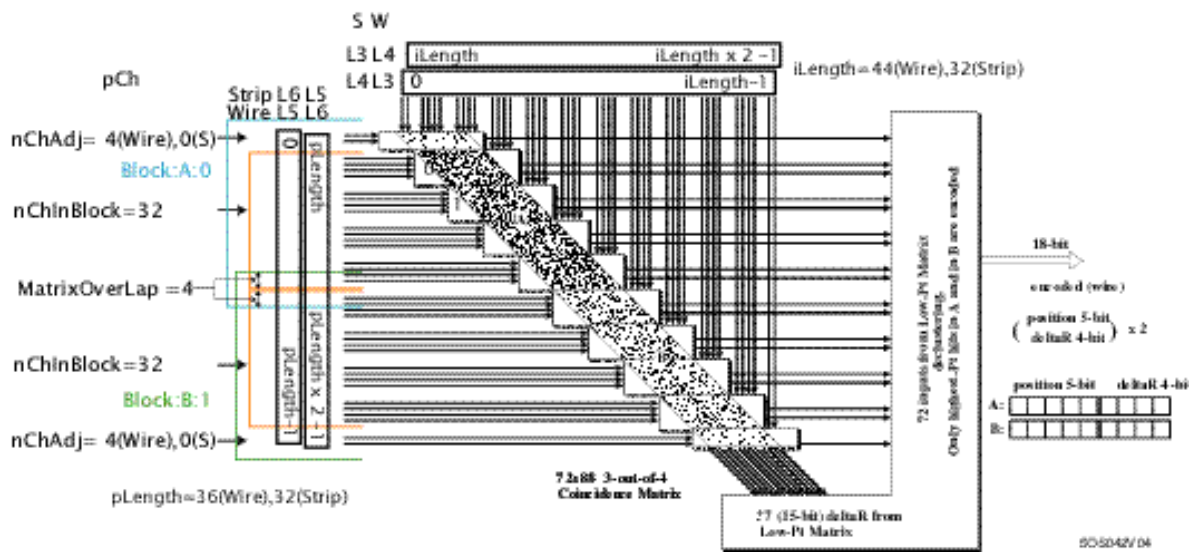


図 29 : DSB での入力から CoincidenceMatrix への接続

コインシデンスマトリックスは図中の pivot からの入力と middle からの入力とが交差した場所にある $0.5+4+4+0.5$ に分けられた長方形の領域にある。次に各ボードごとのコインシデンスロジックを挙げる。

2.4.5.2.1. WireDoublet

コインシデンスマトリックスではまず 1hit, 2hit の判定が行われる。1hit, 2hit とはそれぞれ、Doublet の 2 層のうち 1 層だけに信号があった場合、2 層ともに信号があった場合をいう。図 30 : WDSB の CoincidenceMatrix に示された And 記号が 2hit の判定を行い、長方形の箱で 1hit の判定を行う。これを LocalCoincidence と呼ぶことにし、Coincidence の結果は LCOut[] に収められる。その後 Pivot からの 1hit, Middle からの 2hit またはその逆 Pivot から 2hit, Middle から 1hit があれば 2 out-of-3 が成立したとして Hit 位置 R または \bar{R} を出力する。2 out-of-3 を判定している正方形の箱を element と呼び、ここでは図中の pCh, iCh で示された番号を用いて element(pCh, iCh) とあらわすことにする。pCh はコインシデンスが成立した場合のヒット位置として用いられる。

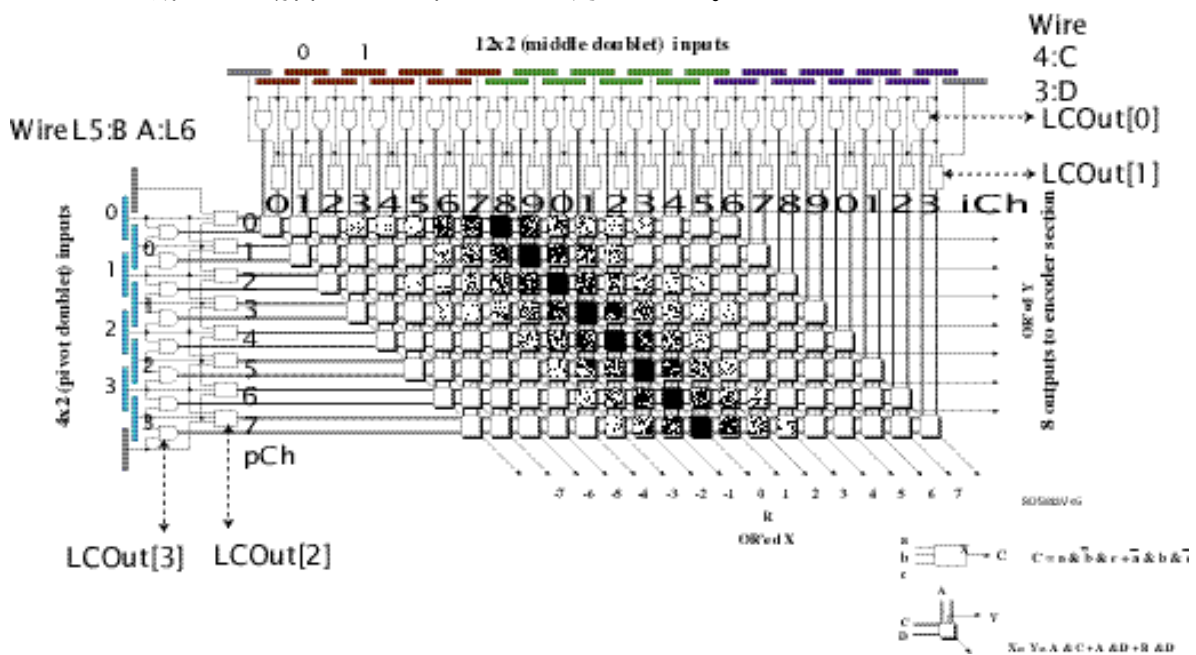


図 30 : WDSB の CoincidenceMatrix

ただし、Doublet では、図 29 : DSB での入力から CoincidenceMatrix への接続で示すように CoincidenceMatrix 4 つ分の領域から出力は 1 つとなるためシミュレーションではそれぞれを分ける必要がないため CoincidenceMatrix 4 つ分を 1 つのオブジェクト¹¹としており、これを CoincidenceSubMatrix と呼んでいる。

2.4.5.2.2. StripDoublet

StripDoublet において WireDoublet と異なる点は入力される信号のチャンネル数のみであると考えていたため WireDoublet と同じ Logic を用いて ASIC の開発をおこなってきた。しかしシミュレーションによって運動量 P が のトラック¹²を入力したところ が 1 となってしまった。これは Middle と Pivot では element が一致していないために起こるものだと考えられる。

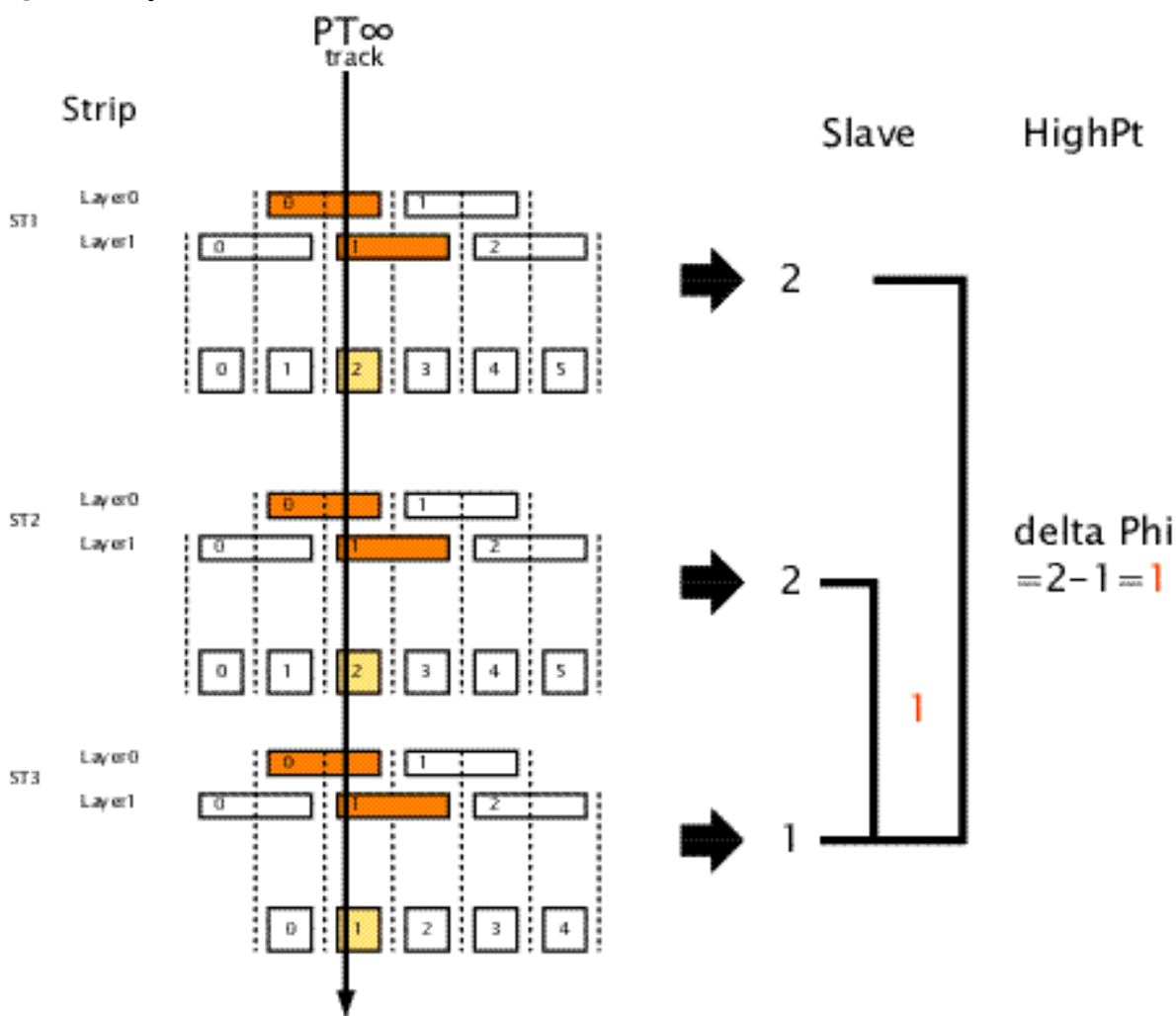


図 31 : が 1 となる問題

この図で、Middle では element 番号が 2 であるのに対し Pivot では 1 となっている。そこ

¹¹ 正確には nChAdj,MatrixOverLap で示された領域を含め 1 つのオブジェクトとなっている。

¹² 飛跡が直線となるトラック

でシミュレーションでは CoincidenceMatrix を図 32: 変更後の SDSB の CoincidenceMatrix のように変更した。これによって Pt の入力に対して $\tau = 0$ とする事が出来た。ASIC に反映するかどうかは現在検討中である。

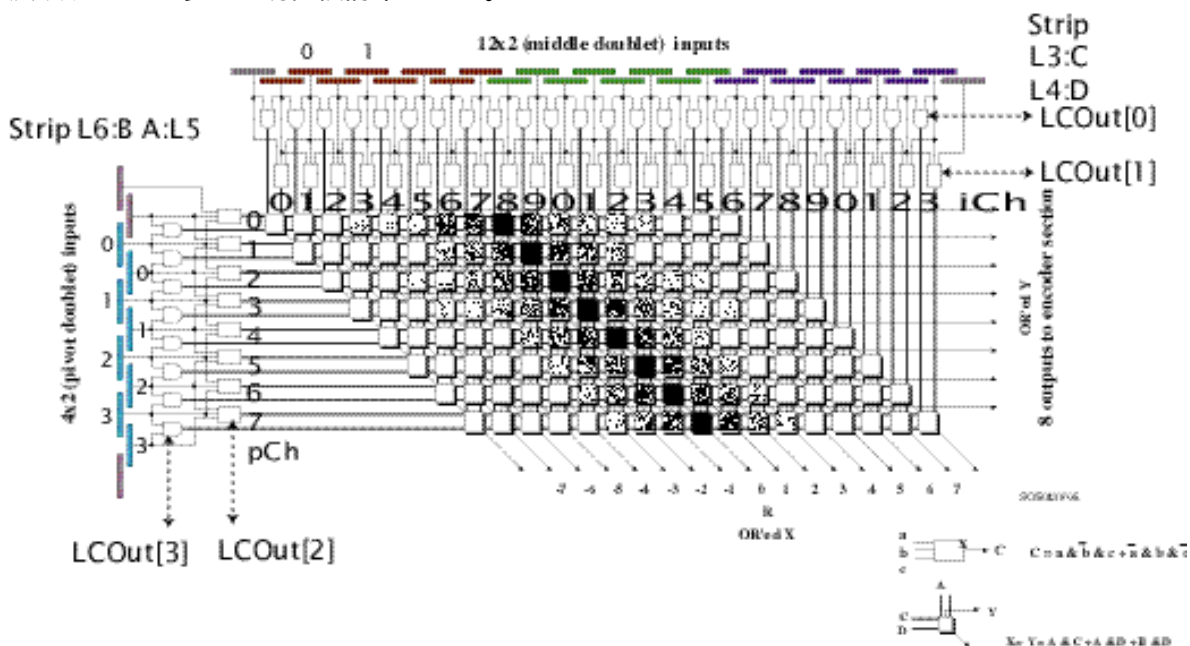
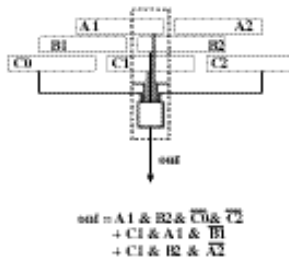
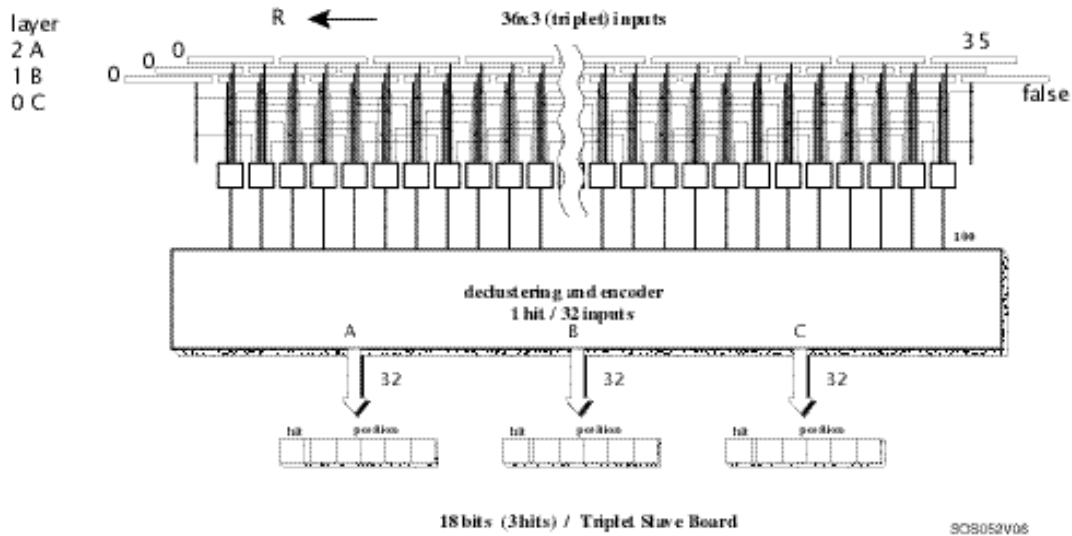


図 32 : 変更後の SDSB の CoincidenceMatrix

2.4.5.2.3. WireTriplet

WireTriplet では Triplet の Layer0,1,2 からの信号に対して 2 out-of-3 のコインシデンスを取る。Logic は図 33: WTSB の CoincidenceMatrix に out として示した。コインシデンスを取った後 32bit ずつ 3つの領域に分けられ、declustering, encode と処理される。



$$\begin{aligned} \text{out} = & A1 \& B2 \& \overline{C0} \& \overline{C2} \\ & + C1 \& A1 \& \overline{B1} \\ & + C1 \& B2 \& \overline{A2} \\ & + A2 \& B1 \& C1 \& \overline{A1} \& \overline{B2} \& C0 \& C2 \end{aligned}$$

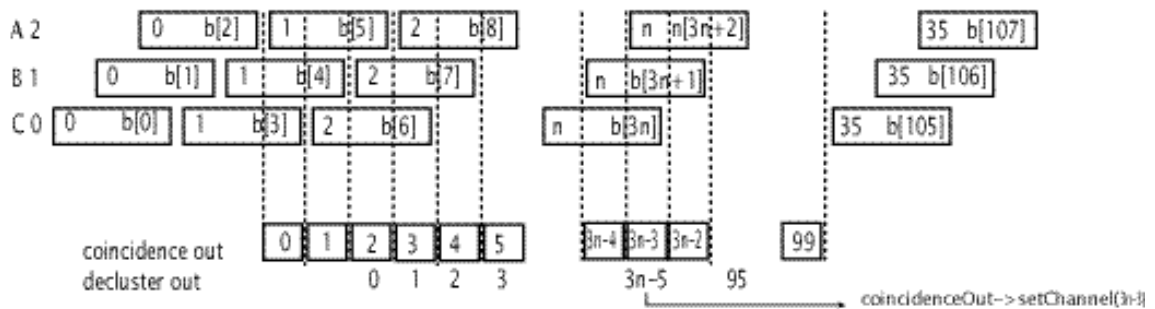


図 33 : WTSB の CoincidenceMatrix

2.4.5.2.4. StripTriplet

StripTriplet の Logic を図 34 : STSB の CoincidenceMatrix に示す。1 out-of-2 の論理に従い出力された後 4 つの領域に 16bit ごとに分けられ、declustering, encode と処理される。

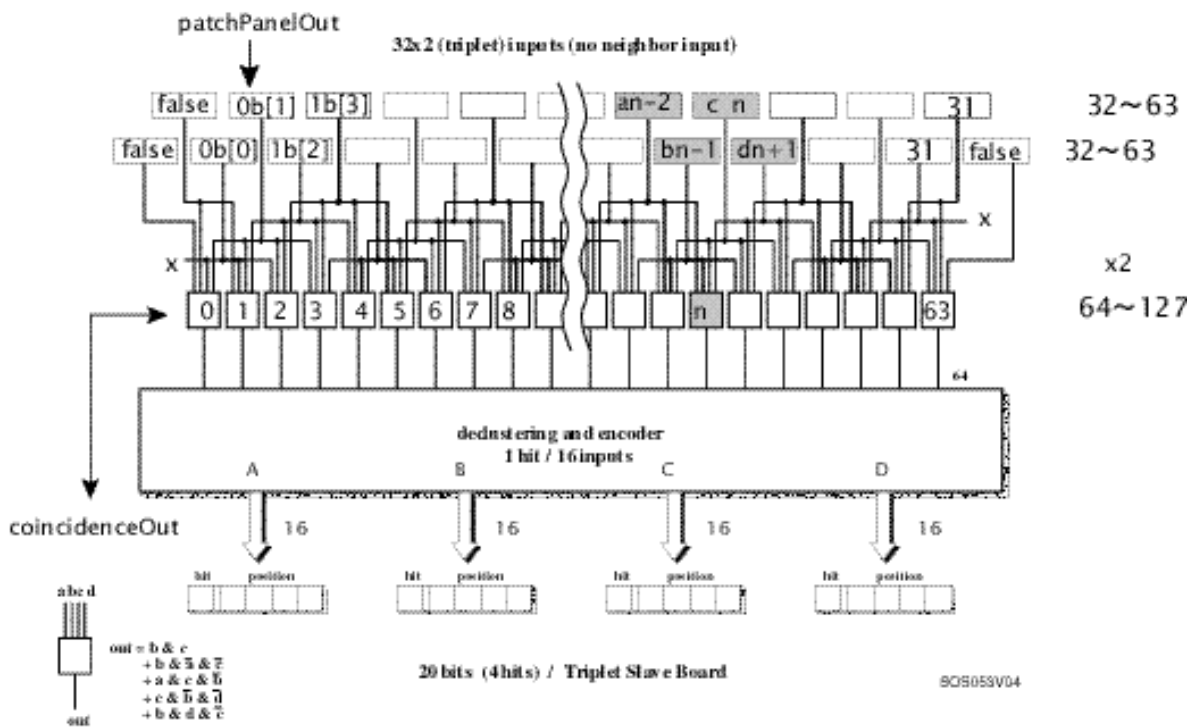


図 34 : STSB の CoincidenceMatrix

なお STSB ではこの CoincidenceMatrix が 2 つ実装される。

2.4.5.3. Declustering

コインシデンスを取られた信号が連続していた場合、その中から 1 の信号だけを選択する。 Declustering logic のように信号番号の小さいほうから 2 番目の channel を選び出力する。

Rule for Declustering

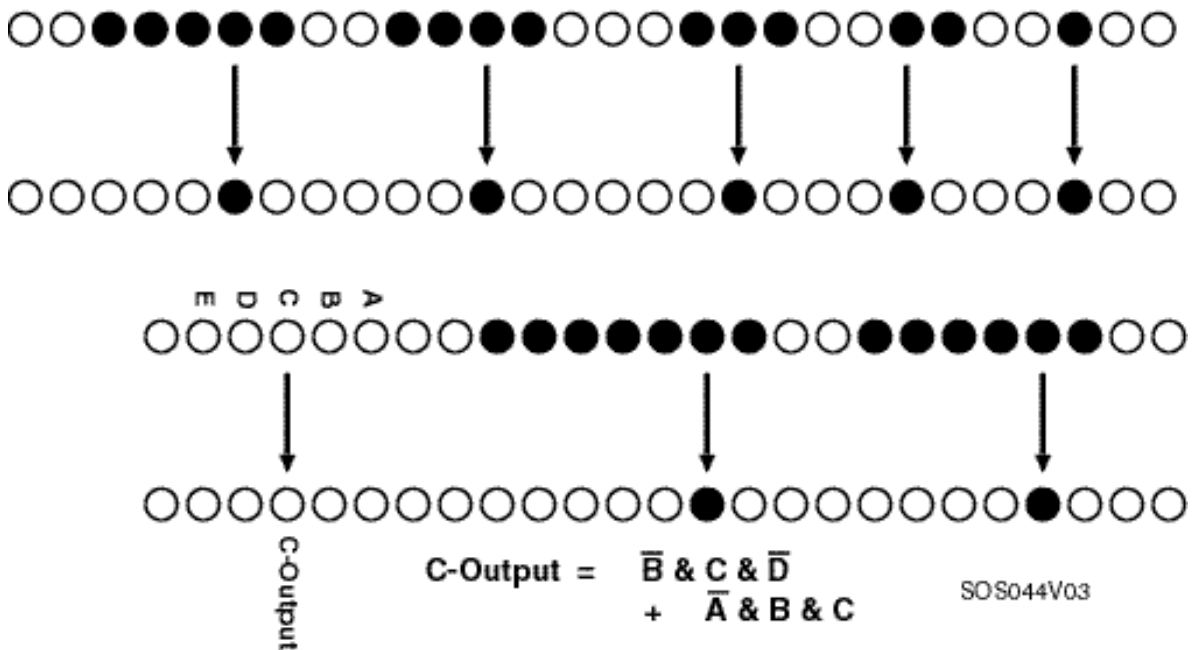


図 35 : Declustering logic

2.4.5.4. Encode と Output

Declustering されたヒット位置は図 SlaveBoard の Output 中の pos で示された bit 数に encode される。WireDoublet と StripDoublet では 32 のチャンネル番号が 5bit に、 R は符号 1bit と r 3bit となる。WireTriplet では 5bit に encode されるがヒットの有無は図中に hit で示した 1bit で表し、ヒットがない場合 hit:0 pos:00000 ヒット位置が 0 の場合 hit:1 pos:00000 と表す。StripTrplet ではルールは WireTriplet と同じで pos を 4bit とする。

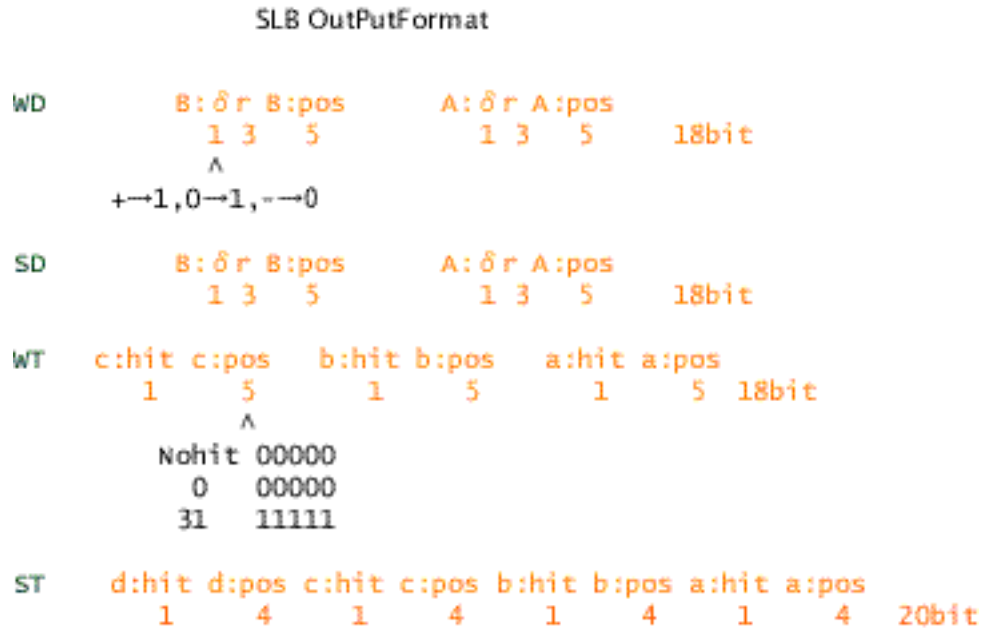


図 36 : SlaveBoard の Output-bit 数

2.4.6. HighPtBoard

HighPtBoard の機能は DoubletSlaveBoard と TripletSlaveBoard から送られてくるヒット位置の信号に対し And を取り、 Doublet、 Triplet 間の 値を求めることである。ヒット位置の差 R または を求めるために DoubletSlaveBoard からの信号は行、 TripletSlaveBoard からの信号は列として And 回路のマトリックスに入力される。

WireHighPtBoard¹³ の 1 つのチップは 6 × 4 つのマトリックスから構成されており個々のマトリックスは Pt の大きい順番 (R の小さい順番) に 1 つのヒットが選ばれる。このときマトリックスの出力はないが、 DoubletSlaveBoard からの信号が R を持っていればその DoubletSlaveBoard からの信号をマトリックスの出力とする。マトリックス 4 つ分をマトリックスブロックと呼ぶことにする。次に TrackSeclector により 6 つのマトリックスブロックから最大 2 つのヒットが選ばれる。このとき HighPt マトリックスからの出力が LowPt の出力より優先される。最終的にヒット位置の情報は捨てられ、 6 つの Block のいずれの出力であったか、 Block の中では上位半分か下位半分かという情報と、 R、 その出力が HighPt が出したものであるか LowPt が出したものであるかという情報が出力される。

¹³ StripHighPtBoard が WireHighPtBoard と異なる点は、 Strip では 2 つのマトリックスが 1 つのチェンバーを受け持つため、 を求める際に別のチェンバーからの信号を区別しそれぞれを別のマトリックスに入力しているところにある。

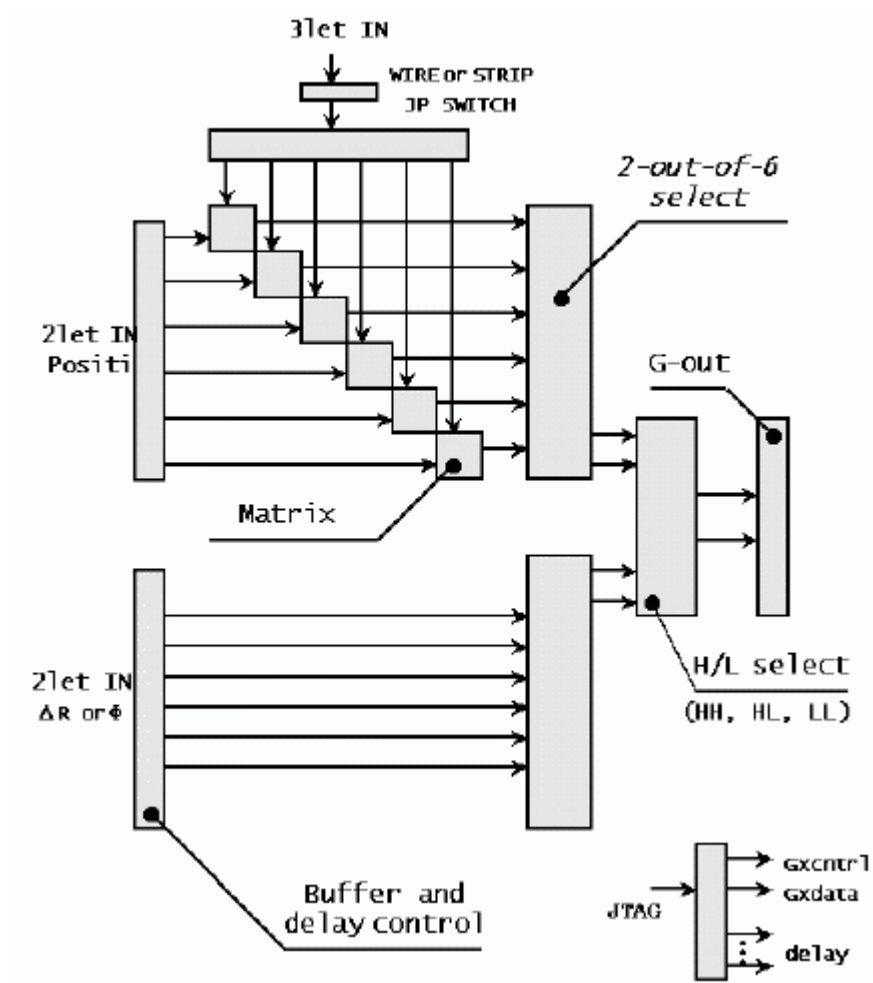


図 37 : WireHighPtBoard の概要

2.4.6.1. Input

WireHighPtBoard への入力を中心となるのは WireDoubletSlaveBoard 3 枚分 (64channel × 3) と WireTripletSlaveBoard 2 枚分 (96channel × 2) である。ただし列として入力される WireTripletSlaveBoard の端の領域についても R を求めるためにその両端に隣から 20channel ずつ受け入れる。

StripHighPtBoard は StripDoubletSlaveBoard 3 枚分 (64channel × 3) と StripTripletSlaveBoard 3 / 2 枚分 (64channel × 2) が入力される。3 × 1 / 2 枚となっているのは StripTripletSlaveBoard 1 枚でチェンバーにして 2 枚分 64 × 2channel の処理を行っているためである。

Bit 数を図 38 : HighPtBoard の Input の bit 数に示す。図 22 : ボードの接続とネーミングに、SlaveBoard で出力され、HighPtBoard へ入力されるデータを書き入れた図が図 39 : SlaveBoard との接続となる。

HPB InPutFormat

WHPB

#BID:banchID HBid:HighPtBoardID TYPE:BoardType[0:wire] REGION:[1:forward 2:endcap]

| | 0A | 0A | 0B | 0B | 1A | 1A | 1B | 1B | 2A | 2A | 2B | 2B | block |
|----|-----|-------|-----|-------|-----|-------|-----|-------|-----|-------|-----|-------|-------|
| WD | pos | delta | pos | delta | pos | delta | pos | delta | pos | delta | pos | delta | type |
| | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | bit |

| WT | 0C | 1A | 1B | 1C | 2A | 2B | 2C | 3A |
|----|-----|-----|-----|-----|-----|-----|-----|-----|
| | pos | pos | pos | pos | pos | pos | pos | pos |
| | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |

SHPB

#BID:banchID HBid:HighPtBoardID TYPE:BoardType[1:Strip] REGION:[1:forward 2:endcap]

| | 0A | 0A | 0B | 0B | 1A | 1A | 1B | 1B | 2A | 2A | 2B | 2B | block |
|----|-----|-------|-----|-------|-----|-------|-----|-------|-----|-------|-----|-------|-------|
| SD | pos | delta | pos | delta | pos | delta | pos | delta | pos | delta | pos | delta | type |
| | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | bit |

| ST | 0A | 0B | 0C | 0D | 1A | 1B | 1C | 1D | 2A | 2B | 2C | 2D |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | pos | pos | pos | pos | pos | pos | pos | pos | pos | pos | pos | pos |
| | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |

図 38 : HighPtBoard の Input の bit 数

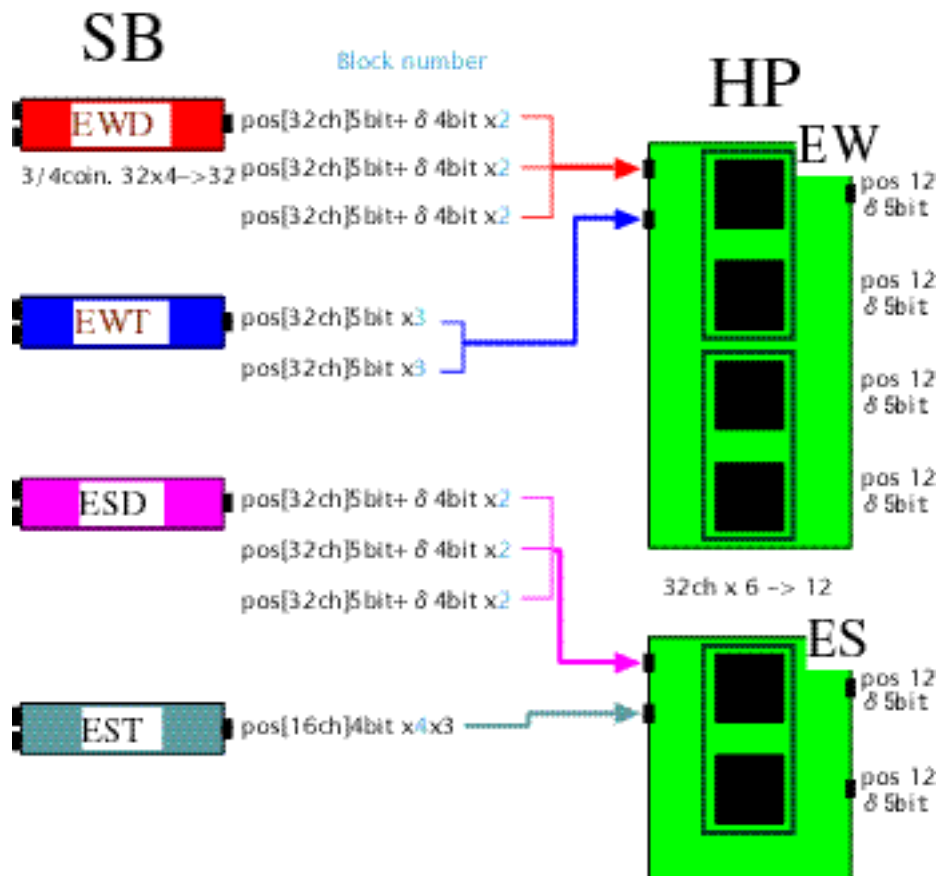


図 39 : SlaveBoard との接続

WireTriplet と StripTriplet の pos にはヒットの有無 1bit が含まれている。
 次に入力されたデータの流れを図 40 : WireHighPtBoard の概要と図 41 : StripHighPtBoard
 の概要に示す。

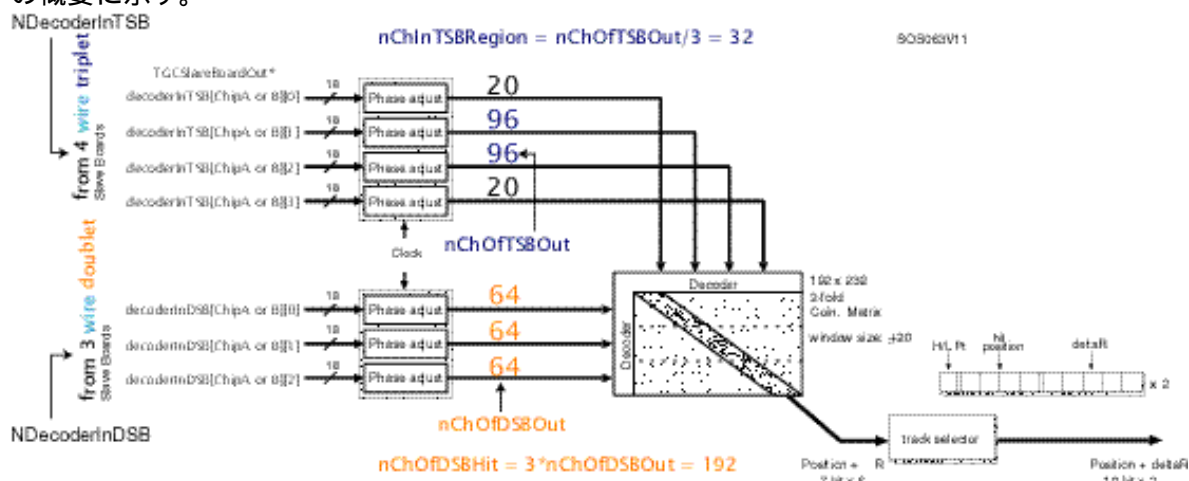


図 40 : WireHighPtBoard の概要

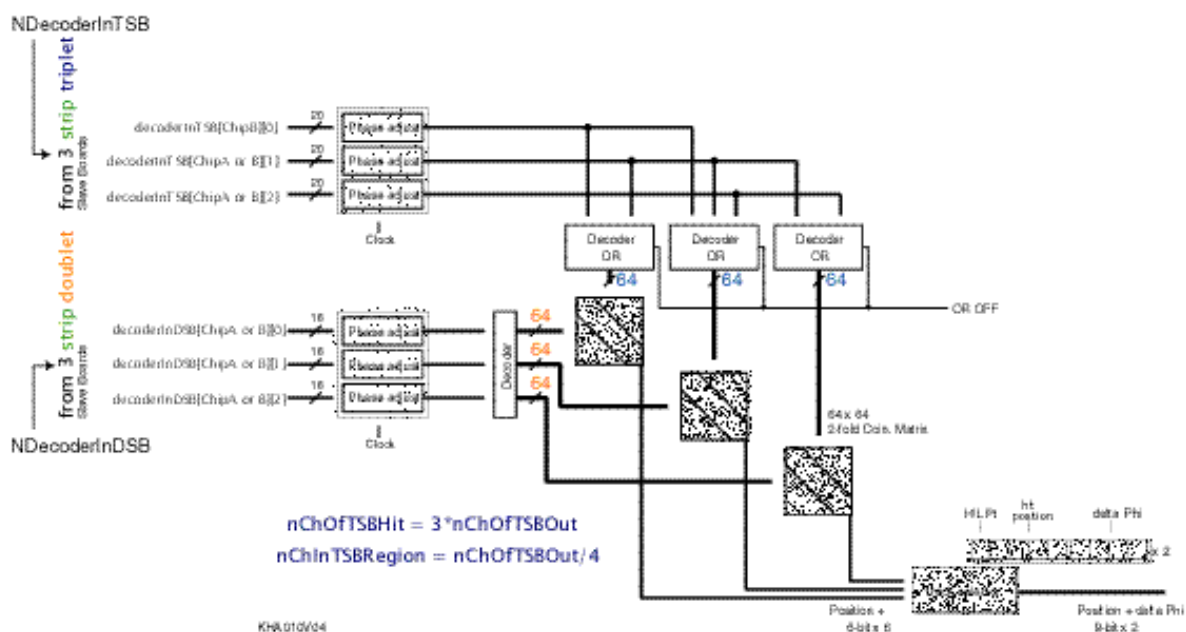


図 41 : StripHighPtBoard の概要

図 40 : WireHighPtBoard の概要のマトリックス部分を詳しく見たものが次の図 42 : WireHighPtBoardのマトリックス部でありマトリックスブロック¹⁴と呼んだものは Matrix 4 つからなり、これが 6 つあることがわかる。

¹⁴ マトリックスブロックは 32channel の領域からなり、R 方向では 1SubSectorCluster:2SubSector、 方向では 0.5SubSectorCluster:2SubSector に該当する。

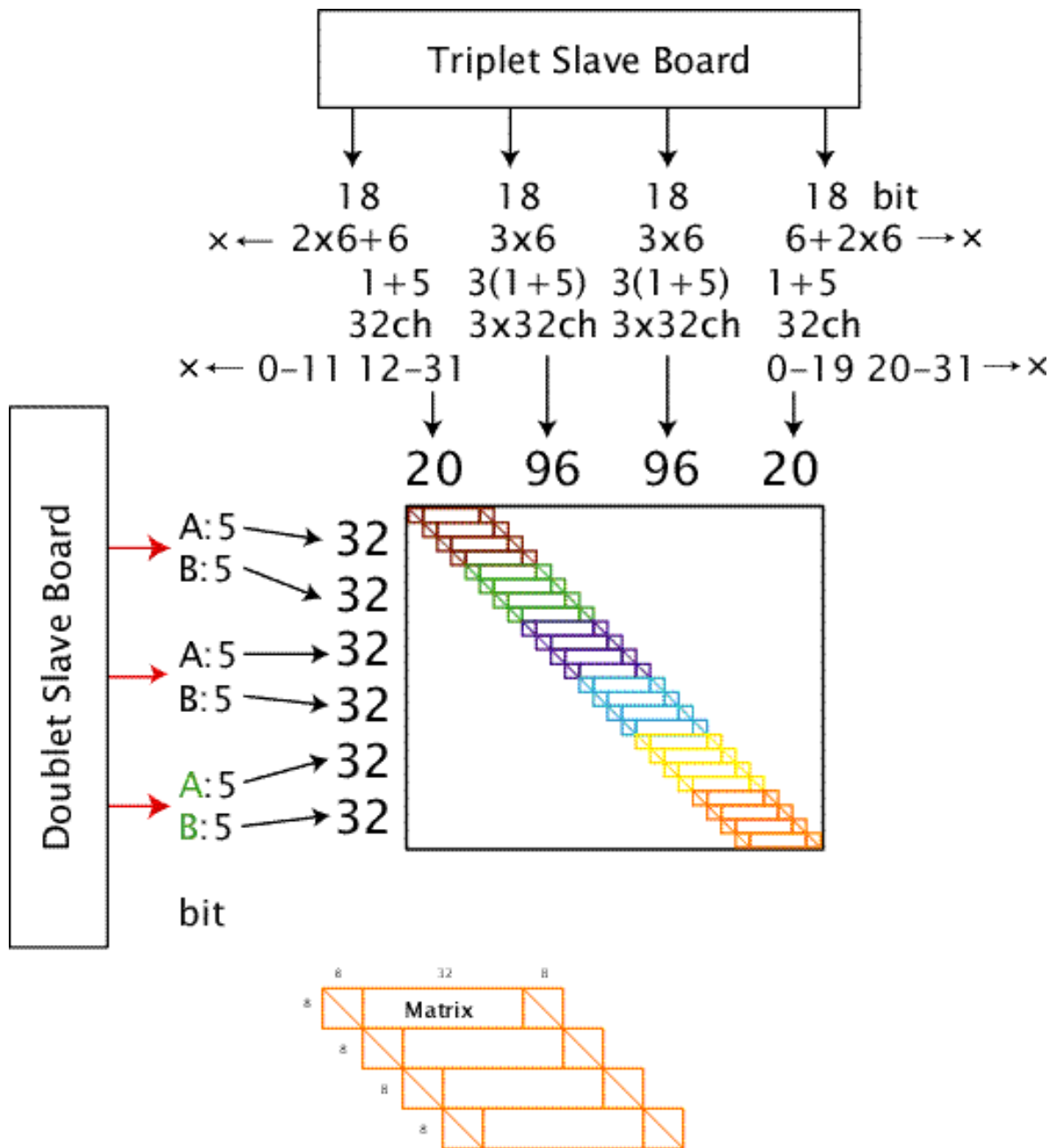


図 42 : WireHighPtBoard のマトリックス部

2.4.6.2. Decode

SlaveBoard のヒット位置の情報は encode されているため decode を行う。pos 5bit は 32channel へ、pos 4bit は 16channel へ decode される。

2.4.6.3. MatrixStructure

マトリックスの行に入力された Doublet、列に入力された Triplet の信号は element で And が取られる。And 論理が成立した element は図に示された 値を出力する。これらの出力のうち の値が最小のものをマトリックスの出力とする。

ただし列の端の領域では

Element(1,0), Element(1,1)は の値として-15、

Element(1,2), Element(1,3)は の値として-14、

Element(1,36), Element(1,37)は の値として 14
 Element(1,38), Element(1,39)は の値として 15
 をそれぞれ出力する。

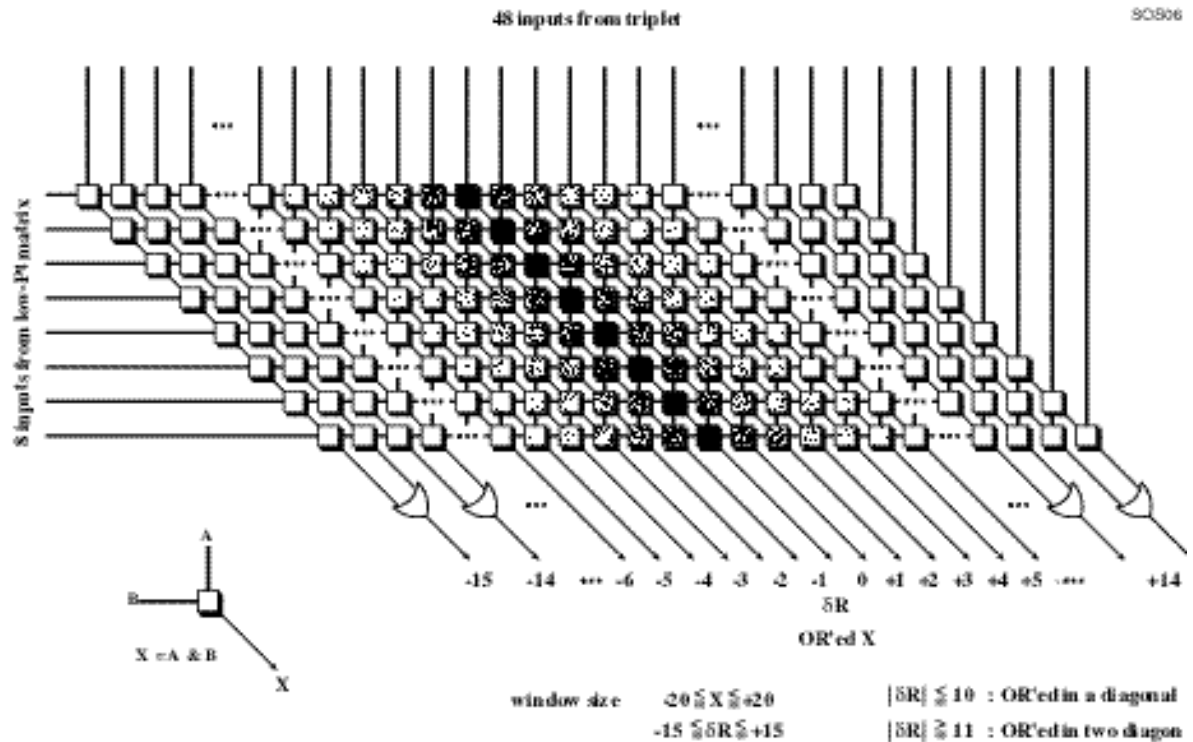


図 43 : HighPtBoard の MatrixStructure

2.4.6.4. TrackSelector (H/L - Select)

6つのマトリックスブロックから最大で6つのヒットが TrackSelector に送られてくる。この中から Pt の大きい順 (値の小さい順) に最大で2つのヒットを選び出力する。出力される2つのヒットには優先順位があり優先順位の高い方を A 低い方を B として出力する。HighPt の出力と LowPt の出力では HighPt が優先され、それぞれ HighPt、LowPt の中ではの値が小さいものが優先される。

2.4.6.5. Encode と Output

マトリックスから出力された の値は図 44 : HighPtBoard の Output の bit 数の dev¹⁵で示された bit 数に Encode される。その後6つの Block のいずれの出力であったか(HitID)、Block の中では上位半分か下位半分かという情報(pos)¹⁶と、 R(dev)、その出力が HighPt が出したものであるか LowPt が出したものであるかという情報(pt)が出力される。

¹⁵ ソフトウェア内では delta を dev とかく

¹⁶ HitID、pos によって示される 12 に分けられた領域は 1SubSector にあたる。

HPB OutPutFormat

WHPB

```
#HPB BID:BanchnID Hbid:HighPtBoardID
      TYPE:BoardType REGION:[1:forward 2:endcap]
```

A

```
Pt HitID pos dev
1 3 1 5 bit total:10
```

B

```
Pt HitID pos dev
1 3 1 5 bit total:10
```

SHPB

```
#HPB BID:BanchnID Hbid:HighPtBoardID
      TYPE:BoardType REGION:[1:forward 2:endcap]
```

A

```
Pt HitID pos dev
1 3 1 5 bit total:10
```

B

```
Pt HitID pos dev
1 3 1 5 bit total:10
```

図 44 : HighPtBoard の Output の bit 数

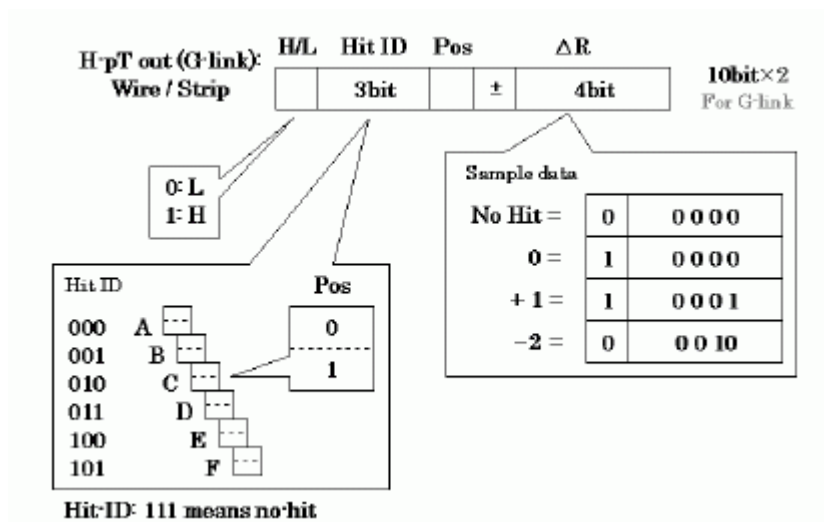


図 45 : HighPtOutPut

2.4.7. SectorLogic

SectorLogicではHighPtBoardの出力する R、 からPtを決定する。 R、 は R-PhiCoincidenceMatrix¹⁷と呼ばれるPtごとに6枚に分けられたMatrixに入力され、 R、

¹⁷ チェンバー上ではSubSecorの領域に対応する

の交差するセルに書かれた値¹⁸がtrueであればそのPtレベルを出力¹⁹する。HighPtBoardまでにR それぞれの方向に対して2つのSubSector²⁰ごとに1つのヒットまで絞られているが、SectorLogicではR方向に2個、方向に4個のSubSectorからなるSubSectorClusterの中で1つのヒットについてR- コインシデンスを判定しこれを出力する。この出力はPreSelectorによってEndcapでは23個、Forwardでは8個のSubSectorClusterの中からPtのレベルごとにRの大きいほうから2番目までのヒットを選択する。さらにTrackSelectorでは各レベルのPreSelectorの出力の中からPtの高い方から2番目までのヒットを選択する。この2ヒットのPtLevel、位置(R、)がMUCTPI (Muon Central Trigger Processor) に送られる。

Level-2 trigger ではSectorLogicから得られるR、の座標からRoIを決定する。

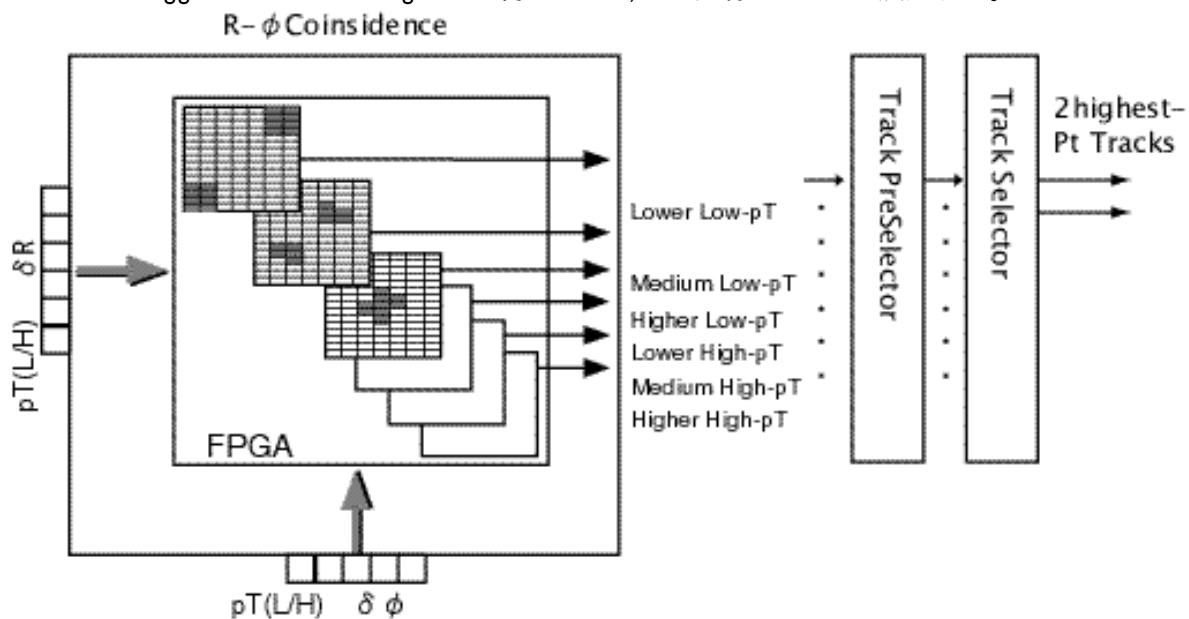


図 46 : SectorLogic の概要

2.4.7.1. Input

HighPtBoardの出力でヒット位置はchannel 単位の情報捨てられておりSubSector 単位の情報になっている。どのSubSector にヒットしたか、値は何れかという情報が入力される。

¹⁸ セルに書かれる値はあらかじめシミュレーションによって作っておくことになる。Mapの作成もトリガーシミュレーションの目的の1つとなっている。

¹⁹ R- コインシデンスと呼ぶ。

²⁰ SubSectorCluster では2つとなっている。

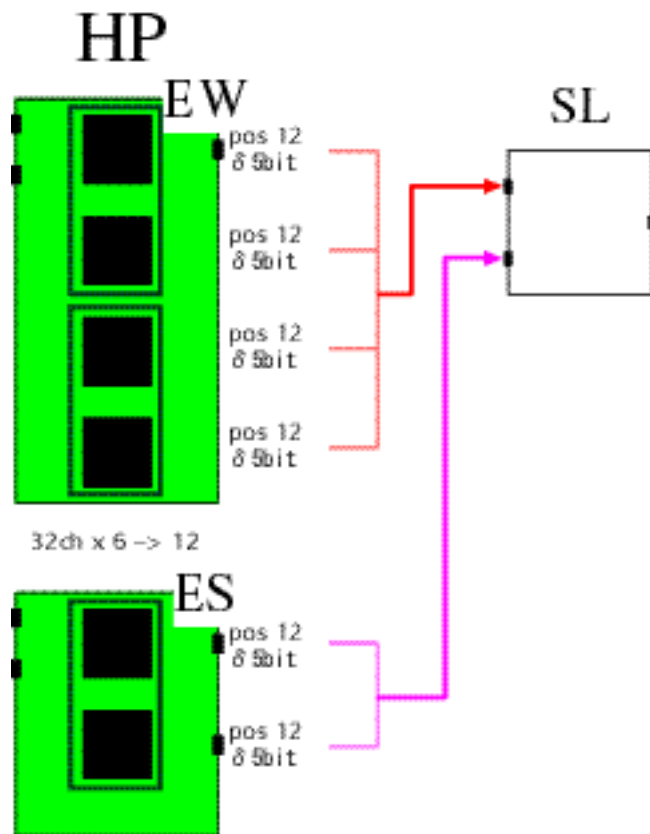


図 47 : HighPtBoard との接続 (Endcap)

SectorLogicInputData

EndcapWire

[pos 2 δR 5bit[SSC0]] × 2 [pos 12 δR 5bit[SSC1-6]] × 2
 [pos 12 δR 5bit[SSC7-12]] × 2 [pos 12 δR 5bit[SSC13-18]] × 2

EndCapStrip

[pos 8 δ φ 5bit[Chamber0-1]] × 2 [pos 12 δ φ 5bit[Chamber2-4]] × 2

ForwardWire

[pos 12 δR 5bit[SSC0-5]] × 2 [pos 4 δR 5bit[SSC6-7]] × 2

ForwardStrip

[pos 4 δ φ 5bit[Chamber0]] × 2

図 48 : SectorLogic の Input-bit 数

2.4.7.2. SSC

TriggerSector は SubSector と呼ばれる領域に分割されているが、その数は Forward 部で 64 個、Endcap 部では 148 個とかなりの数になる。そのため、全ての SubSector で独立に運動量測定を行った場合、回路が大きくなってしまい不都合であるため隣接する 2 つの

SubSector を R 方向に 2SubSector, 方向に 4SubSector のブロックにまとめて扱い、その中から 1 つのミューオン track の運動量測定を行うことにしている。

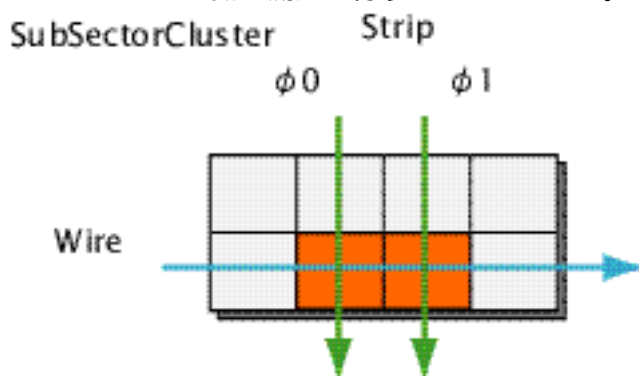


図 49 : SubSectorCluster の構造

HighPrBoard からの入力は SSC(SubSectorCluster)に対して R 方向で HitID が SSC を指し、pos が上半分か下半分かを示す。方向は HitID が 0²¹であるか 1 であるかを指し、pos が右半分か左半分かを示す。

2.4.7.2.1. Selection Rule

SSC では最大で R 方向から 1 つ 方向から 2 つの入力があり、SSC 内で 2 つのヒットが確定する場合がある。この時、運動量大きい方のヒットを出力するが、運動量が同じだった場合はどちらを出力するか決まっていない。暫定的にシミュレーションでは の小さいほうを出力することにした。

2.4.7.2.2. TGC と SSC の対応

TGC と SSC の対応について図 50 : TGC と SSC の対応を示す。R 方向は矢印で、 方向は SSC の背景色で表した。この図でわかるように、HighPt に送られるまでに R 方向の入力は、基本的に 32 チャンネルに分けられて (PatchPanel、SlaveBoard を経由して) 別々の HighPt モジュールの ASIC Chip に送られる。Endcap 領域では、4 個の HighPt ASIC で、Forward 領域では、2 個の HighPt ASIC で処理される。HighPt ASIC では隣同士の SubSector から 1 つのヒットを選び出したあと、track selector でカバーする範囲内のヒットを 2 個まで減らして SectorLogic に送っている。図 50 : TGC と SSC の対応の HighPt ASIC に付いている吹き出しは、track selector への入力数と出力数を表したものである。

例えば、 =1.05-1.37 について見てみる。この領域をカバーする R 方向の High-Pt の ASIC は、4 枚の TGC からの入力がある。図 TGC と SSC の対応によると、この領域に対応する SSC は、SSC1 から SSC6 までの 6 個である。SSC の R 方向の大きさは、Hi-PT が 1hit を出力する大きさに取ったので、この Hi-PT ASIC では最大 6hit を発生する。Hi-PT ASIC の track selector は、これから 2hit を選出する。そして、その結果が Sector Logic に送られる。一方、 方向には、この領域は 4 個の TGC から構成されている。したがって、各 TGC から 2hit 発生するため、合計 8hit が発生する。この領域は 2 個の 方向の Hi-PT ASIC によりカバーされており、それぞれで出力を 2hit に減らして Sector Logic に送っている。

²¹ 0, 1 とは図 49 : SubSectorCluster の構造に記されているように、SSC 内で左側の SubSector 2 つ分の領域を 0, 右側 2 つ分を 1 としている。

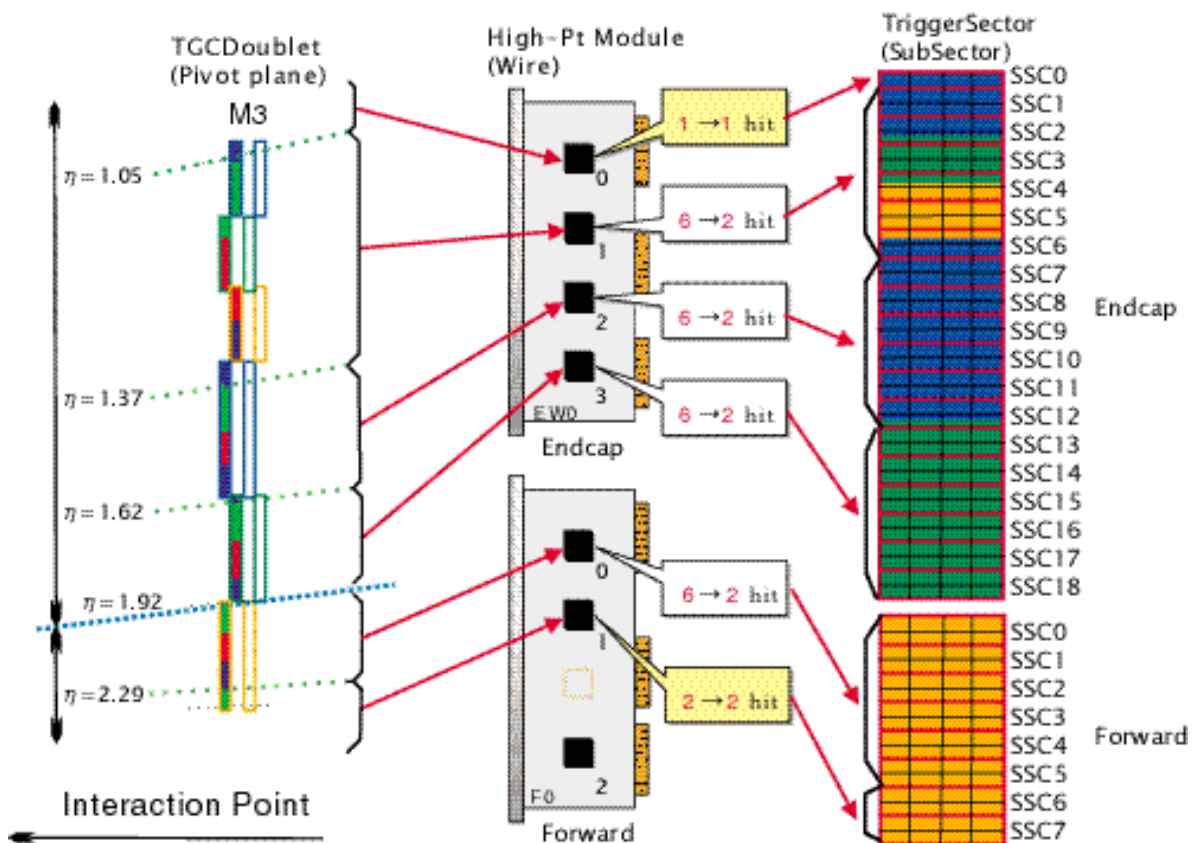


図 50 : TGC と SSC の対応

2.4.7.3. TGC 境界の問題

Forward 部の Trigger Sector は Pivot Plane において 1 枚の TGC Doublet であるのに対し、Endcap 部の Trigger Sector は 5 枚の TGC Doublet で構成される。R 方向に 4 つある TGC の境界と subsector の境界はどれも一致していない(R 方向は、TGC の境界と Trigger Sector の境界が一致している) ため、図 TGC と SSC の対応のように SSC2, 4, 6, 12 では TGC の境界を内部に持つことになる。

TGC の境界では、Patch Panel で overlap 部の OR を取ることによって double count を防止するとともに、全体が 1 つの大きな TGC として扱えるように配線されている。そのため、これより下流の SlaveBoard, Hi-PT では、これらの領域を特別扱いせずに処理が行われている。したがって、この TGC 境界を内部にもつ subsector に R 方向の hit が存在した場合、どちらの TGC 由来の hit か(あるいは両方か)という事は分からない。このことは、R- Coincidence Logic で R 方向の情報と合わせて運動量測定を行う時に問題となる。

R 方向の信号は TGC ごとに読み出されるため、その subsector を含む SSC2, 4, 6, 12 には同じ座標に対して R 方向の Hi-PT からの入力が入力が 2 個存在する。そのため、1 つの subsector で 2 通りの運動量測定を行い、その結果から 1 つのミュオンの track 候補を選ぶ必要がある。また、その SSC において、R 方向の入力が TGC 境界を含まない subsector 側にあった時、これらの subsector に属さない側の TGC から R 方向の入力があつたとしても、両者からは 1 つのミュオンの track を構成出来ないため、Track 候補から外さなければならない。一方、SSC 単位で R- Coincidence Logic を回路に実装することを考慮すると、このような複雑な処理を行う TGC 境界を含んだ SSC をそのまま設計すると、他の SSC に比べて大きく複雑な回路になり、速度面でボトルネックになってしまい、好ましくない。そのため、

- SSC2 SSC2a + SSC2b
- SSC4 SSC4a + SSC4b
- SSC6 SSC6a + SSC6b
- SSC12 SSC12a + SSC12b

このように SSC をそれぞれの TGC からの 方向の入力ごとに独立した 2 つの SSC に分割し、1 つの subsector で 2 通りの運動量測定を行う必要をなくして TGC 境界を持たない SSC と同等に扱えるようにしている。また、これにより後段の Track Selection Logic に入力される track 候補は 19 から 23 に増加している。

2.4.7.4. Decoder

HighPt から送られてくる HitID、pos を該当する SubSector に、5bit に encode された R、を -15 ~ +15, -7 ~ +7 に decode する。

図には HitID、pos が decode され対応される SubSector を示した。Wire では pos がそのまま SSC 内の r に対応して SubSector が決まり、Strip では 1 Chamber 内で HitID の小さいほうが 0、大きいほうが 1 となり、さらに pos により SubSector が決まる。

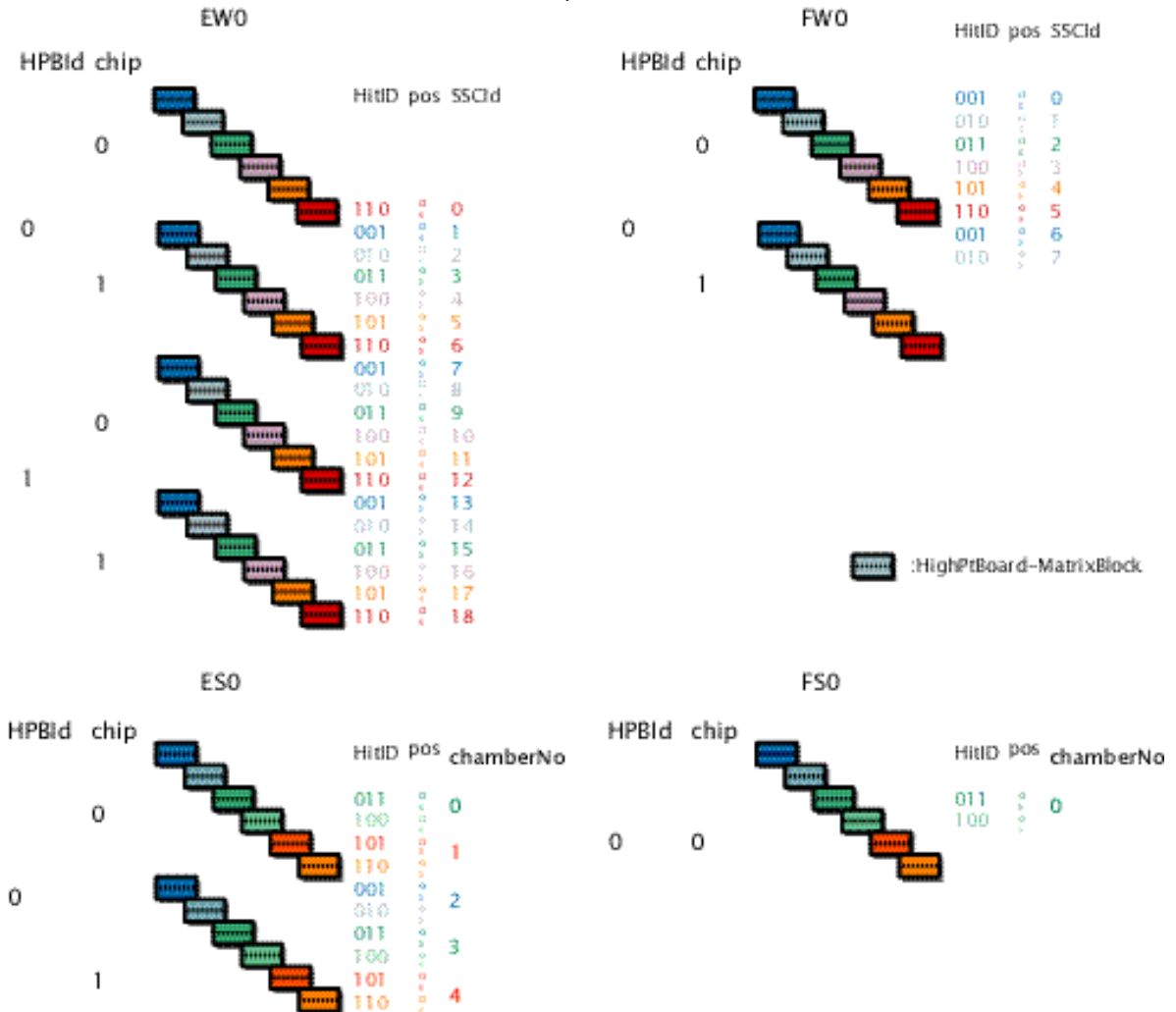


図 51 : HitID と pos の decode

2.4.7.5. R- Coincidence Logic

R- CoincidenceLogic の役割は、R 方向(Wire)の R と 方向(Strip)の から入射ミューオンの横運動量を測定し、6 段階の横運動量の閾値に分類して出力することである。R 方

向と 方向の入力が存在し LUT (LookUpTable) から Coincidence が取れる track 以外はノイズや fake hit とし track 候補から排除する。図 52 : LookUpTable には LUT の概念図を示した。Map 内の黒色の部分が true、白色の部分が false であり、入力された R、は LUT と Coincidence(and)を取り、Coincidence が成立すればそれに応じた Pt Level を true として出力する。

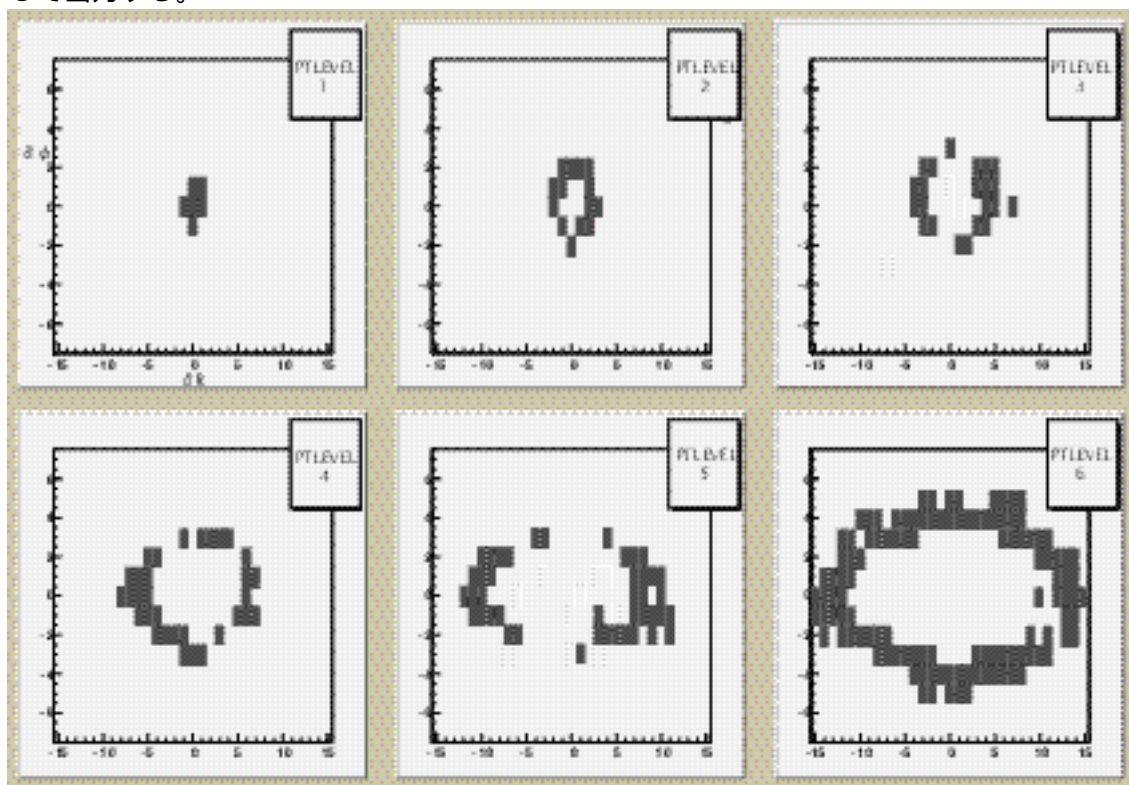


図 52 : LookUpTable

Track に影響を与えるトロイダル磁場は TGC で言う octant に対し対称となっているため 1octant 内全ての SubSector 分(19(SSC) × 8(SubSector)) × 6(Sector)[Endcap]+ (7(SSC) × 8(SubSector)) × 3(Sector)[Forward]個それぞれ別の LUT を用意し構造の差を吸収する。

2.4.7.6. Track Pre Selector

Pre Selector は TriggerSector 内にある SSC の数だけある R- Coincidence Logic からの入力から、各 PtLevel ごとに最大 2 つの Track を選び出し SSC の位置情報が付加され出力される。この時 の小さい(SSC#の小さい)track が優先される。

2.4.7.7. Track Selector

PtLevel ごと 6 個の Pre Selector から合計 12 個の track 候補が入力される。この中から Track が 2 個選ばれる。PtLevel が高い方、PtLevel が同じ場合は SSC#の小さい方が優先される。

2.4.7.8. Encoder と Output

SectorLogic が選んだ 2 個の Track の PtLevel を 3bit に、SubSector 単位の位置情報を R 6bit 2bit に encode し出力する。

2.4.8. モジュールの配置

3. TGC トリガーシミュレーション

3.1. 目的

- トリガーエレクトロニクスシステムが実際に扱うデータと同じデータでシミュレーションを行うことでテストボードで使用するテストベクタを生成する。
- DICE から得たデータからエレクトロニクスシステムの各モジュールに対し実験時に予想されるデータでモジュールの動作、設計の妥当性についての検証を行う。
- SectorLogic で実際に使用する R-PhiCoincidenceMap を作る。
- 最終的にはトリガー効率をシミュレーションによって計算しトリガーエレクトロニクスシステムへの要求を満たしているか検証する。

3.2. 構成

3.2.1. Geant3 DICE

Geant3 の目的は KeV ~ TeV 領域の高エネルギー粒子と検出器等の物質の相互作用をシミュレートし、検出器の性能評価をおこなうためのものであり、DICE では Geant3 によって陽子の衝突後から TGC に到達するまでのシミュレーションを行っている。このときに TGC を通過した場所の情報も持っているためトリガーシミュレーションではこの位置情報を使う。

3.2.2. Digitize

DICE の出力は座標であるため TGC の WireGroup、Strip Number に変換する必要がある。

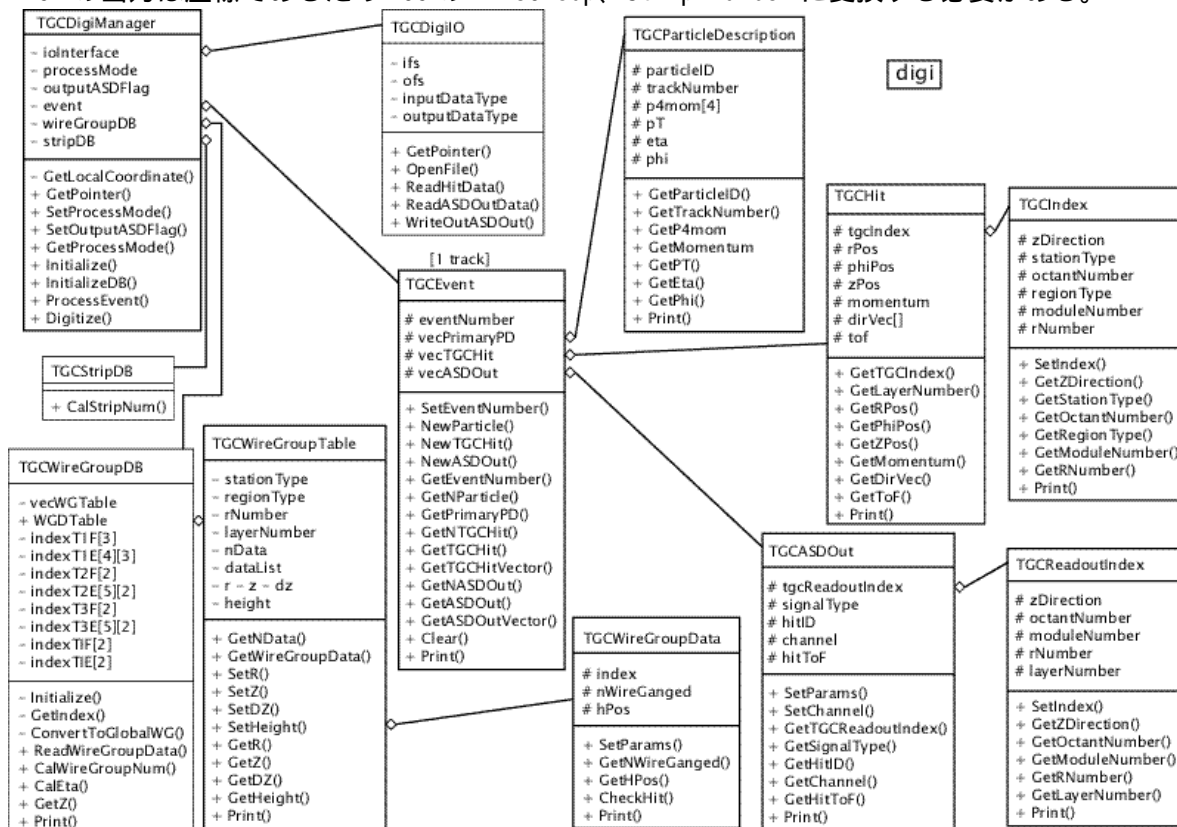


図 53 : digitize に関連するクラスのクラス図

クラス図のように digitize は TGCManager によって DICE の出力した位置情報をもつ digi.dat を読み込み 1 bunch 分を TGCEvent に持つ。これを WireGroupNumber のデータベース tgc_wg.dat から読み込んだ TGCWireGroupTable によって GlobalWireGroupNumber に変換し、TGCASDOut に収める。

3.2.3. Trigger Electronics Modules

トリガーエレクトロニクスシミュレーションが満たすべき仕様は 2 章で示した。

TGCElectronicsSystem で作られた各ボードクラスは、clockin() で開始され Collect Input() によって前段のボードのアウトプットを集め、Coincidence を行いアウトプットを作るという流れになっている。

3.2.3.1. PatchPanel 関連クラス

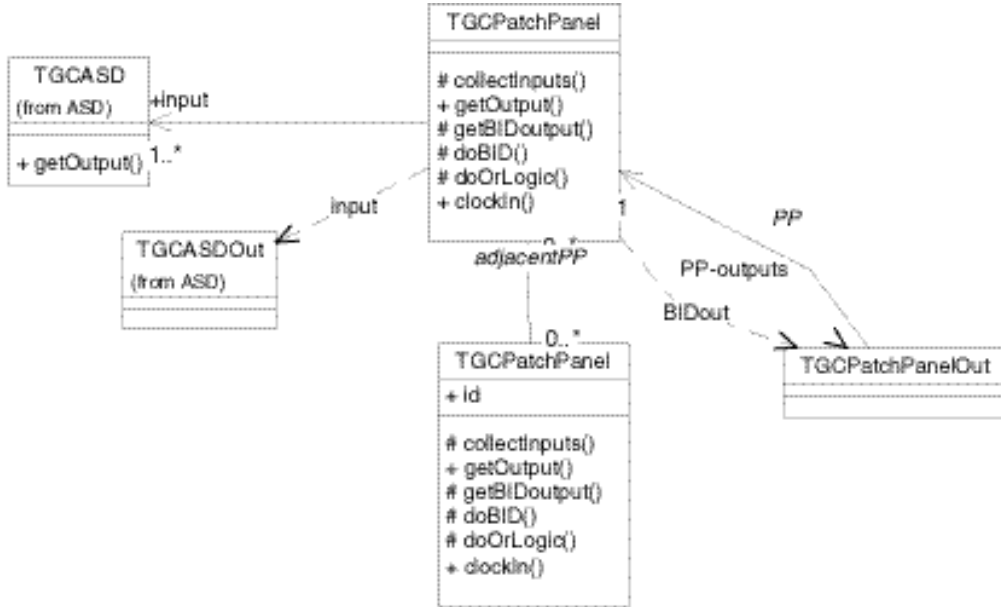


図 54 : TGC PatchPanel のクラス図

TGC PatchPanel クラスは TGCASDOut を collectInputs() によって集め、doOrLogic() によって OR'ing を行う。

3.2.3.2. SlaveBoard 関連クラス

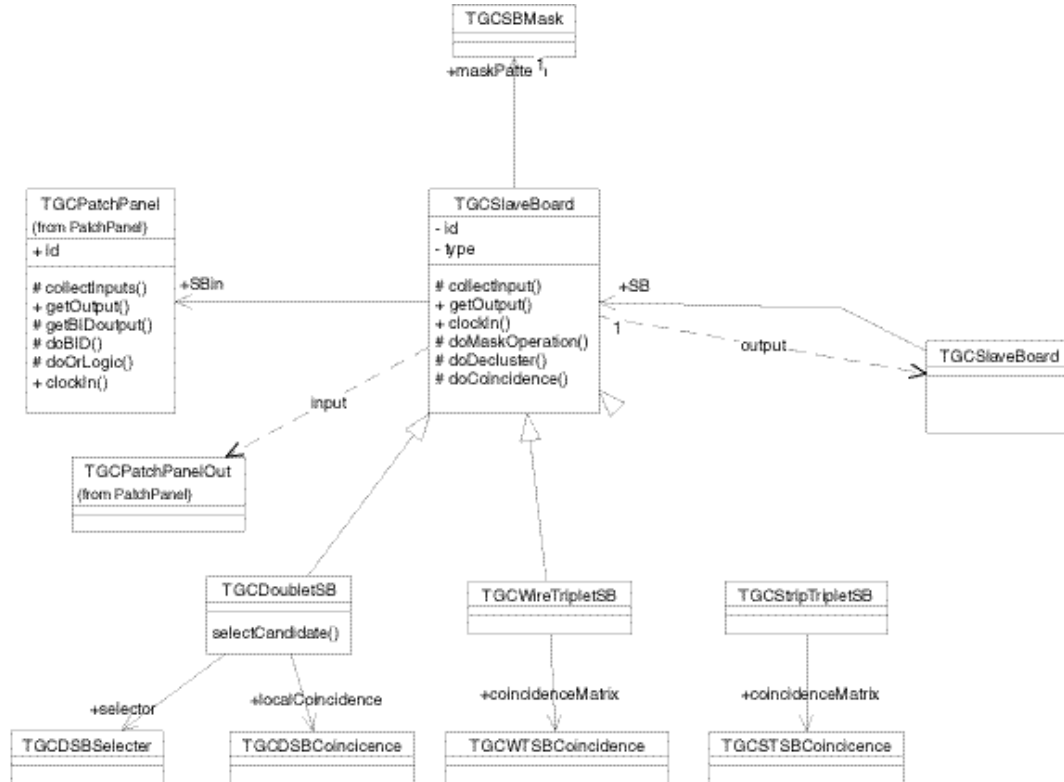


図 55 : SlaveBoard に関連するクラスのクラス図

TGCSlaveBoard クラスは TGCPatchPanel を collectInput() によって集め、doCoincidence(), doDecluster() によって Coincidence と Declustering を行う。

3.2.3.3. HighPtBoard 関連クラス

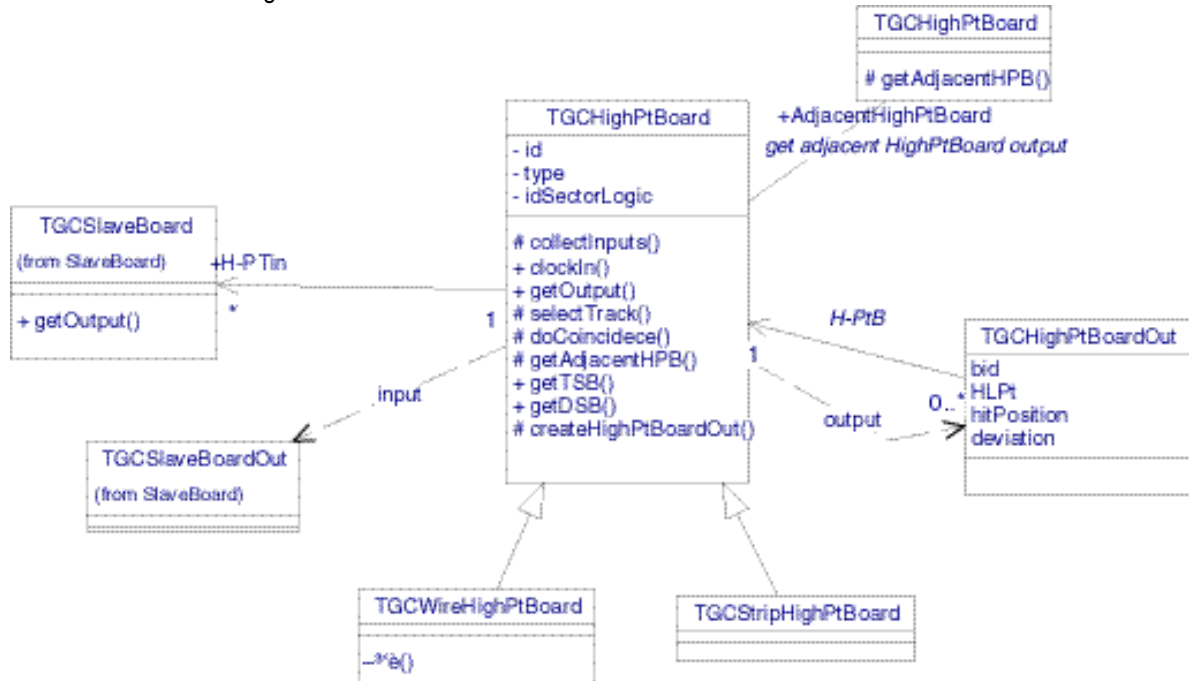


図 56 : HighPtBoard に関連するクラスのクラス図

3.2.3.4. SectorLogic 関連クラス

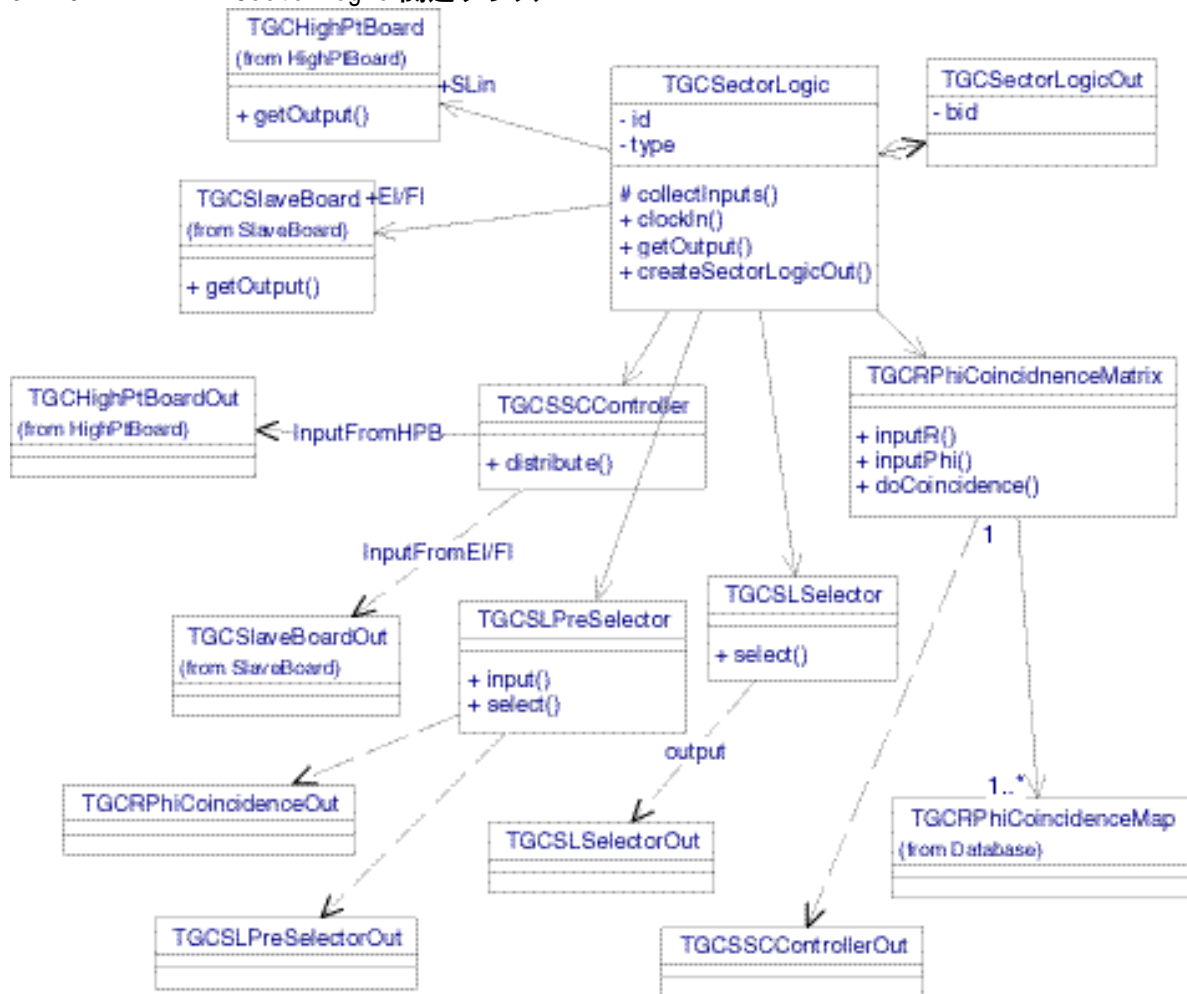


図 57 : SectorLogic に関連するクラスのクラス図

3.3. ソフトウェアのテストと評価

各モジュールのテストを行った。

3.3.1. Digitize、DICE

Digitize の出力の検証を以下の方法で行った。

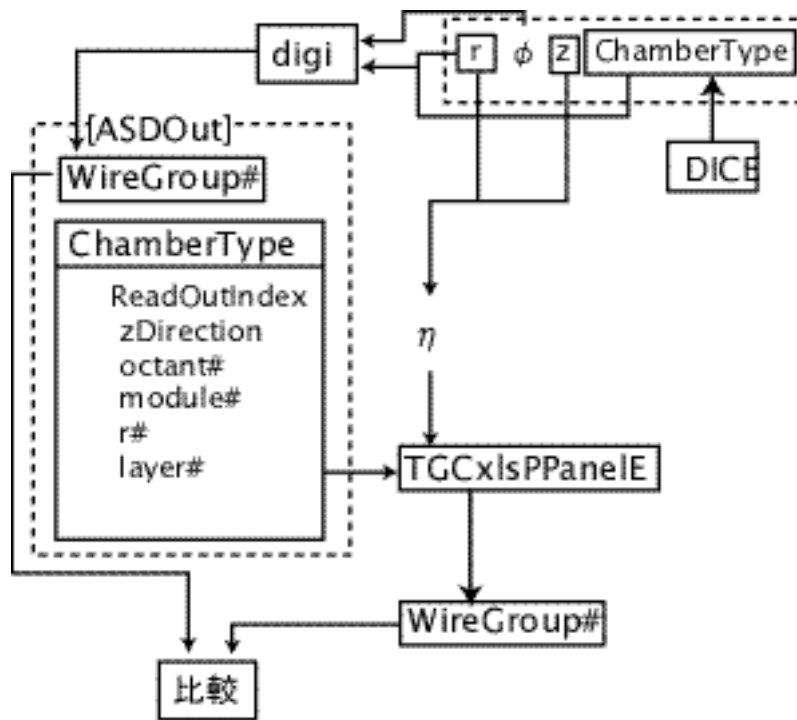


図 58 : 比較の方法

テストの方法はdigi の出力する GlobalWireGroup# が tgc.xls²²のデータベースが想定しているものと一致しているかどうか比較することで行う。
Tgc.xls は TGCxIsPPaneIE クラスとして実装した。

²² TGC のジオメトリーや Wire の Ganging 等多岐にわたり定義している

- ASDOutのWireGroup#が正しいことを確認するために ASDOut.lwgと tgc.xls.PPanelE.wgとを比較する
- tgc_xls_PPanelE.dat table
- TGCxlsPPanel class
- TGCWireGroupDBに実装

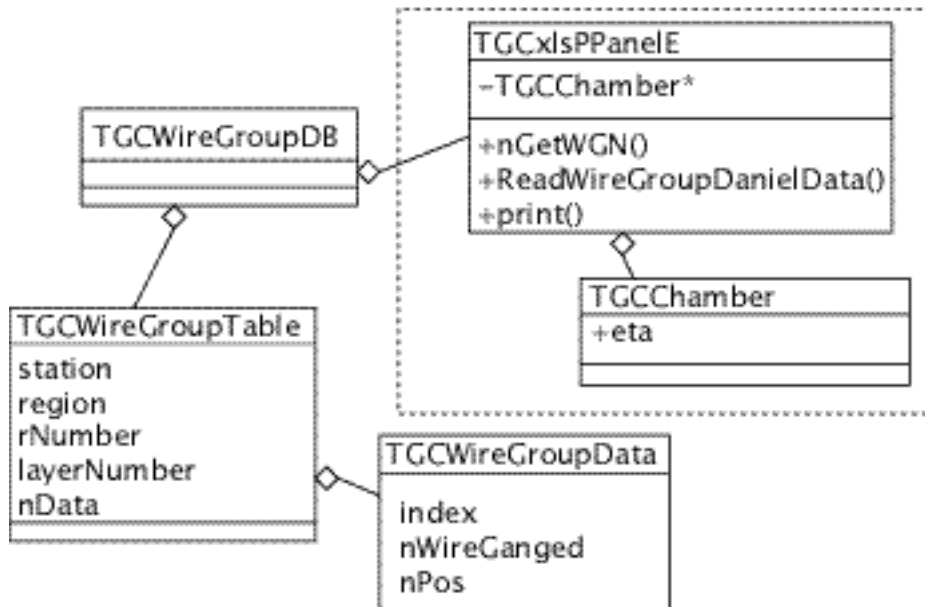


図 59 : 実装したクラス

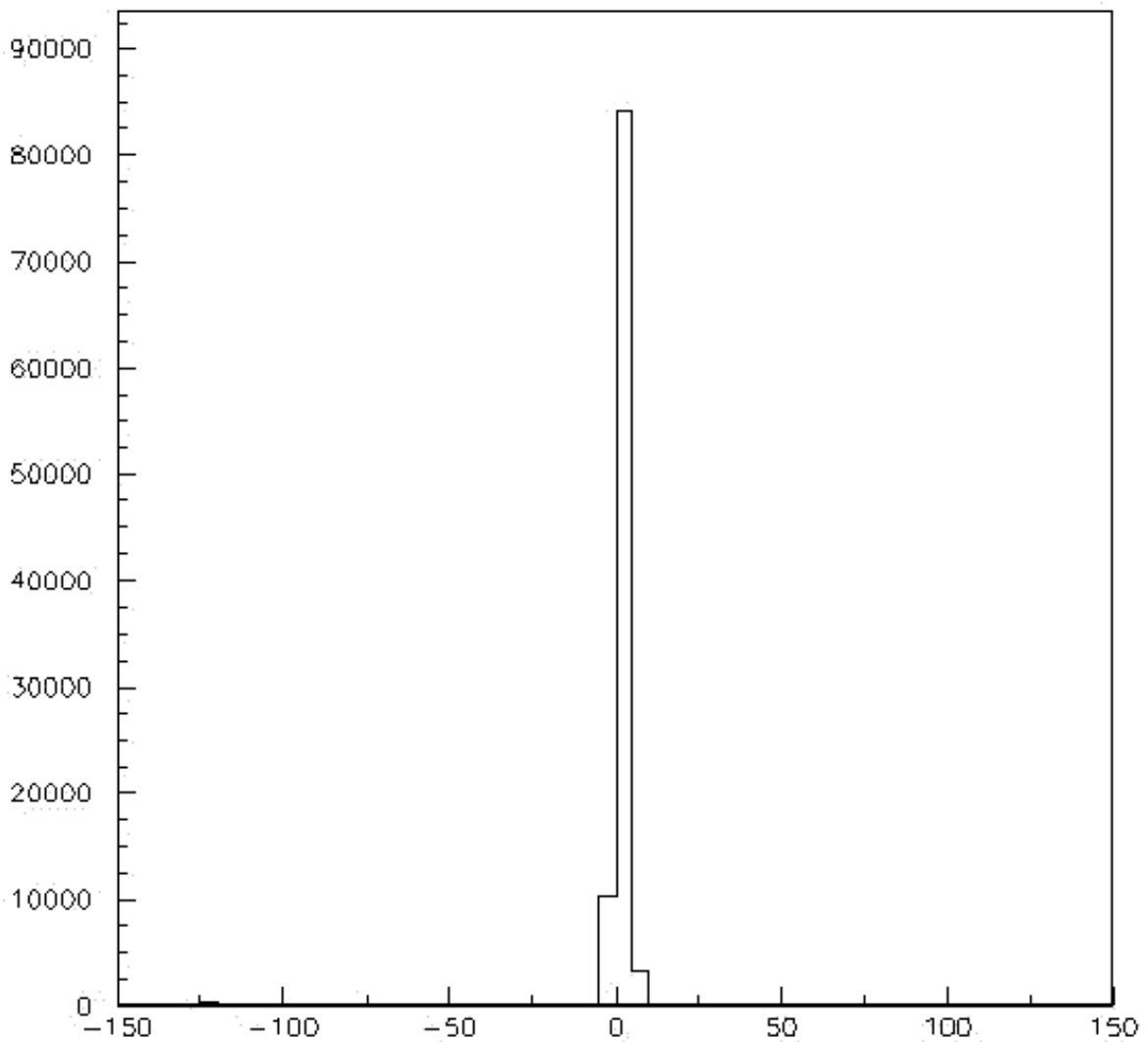
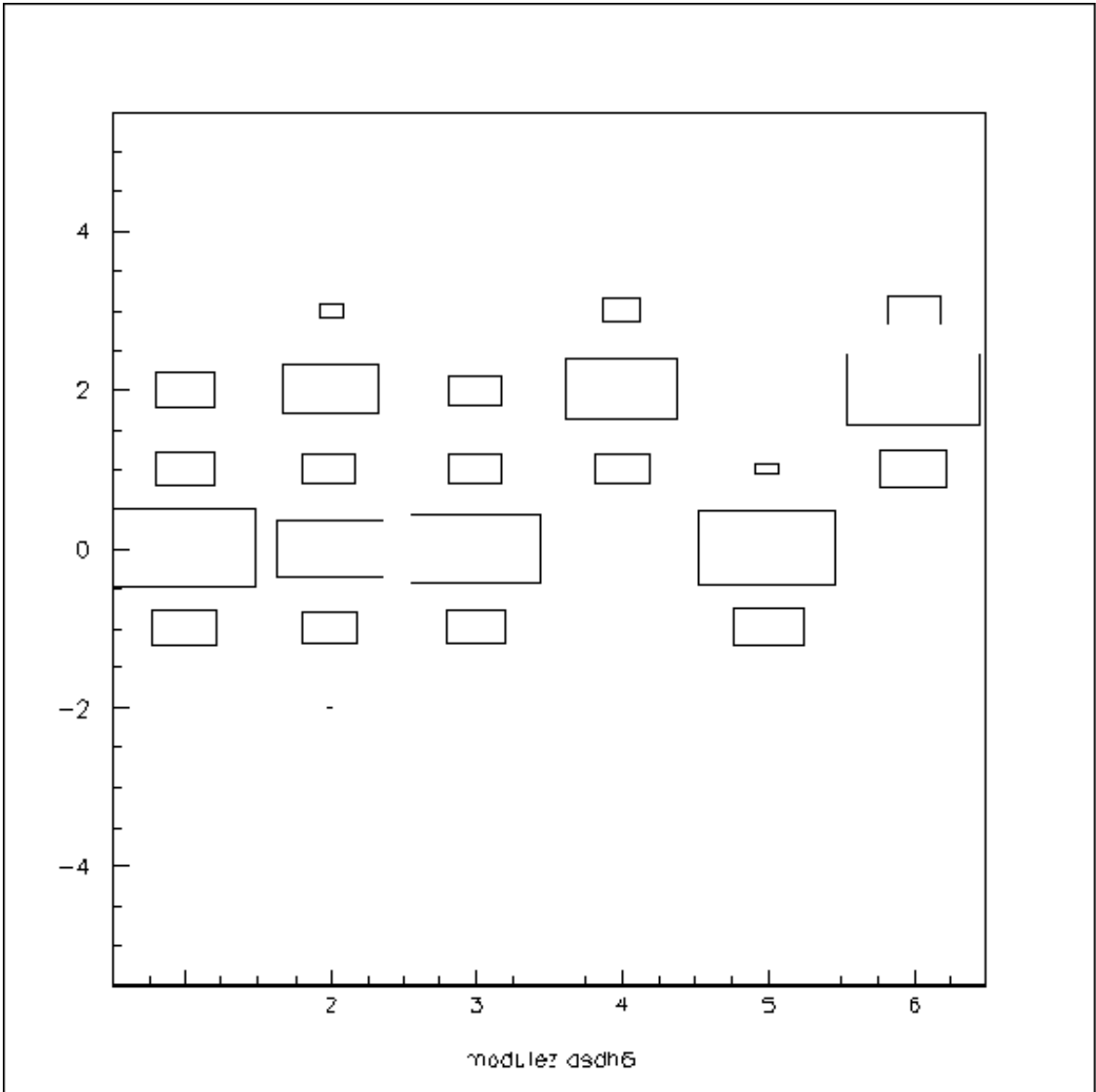


図 60 :digi の出力と tgc.xls の PPanelE との比較結果

図 60 : digi の出力と tgc.xls の PPanelE との比較結果では、縦軸カウント数、横軸 digi の出力と、PpanelE の差としている。-120 の付近に小さい山が見られるがこれはトラックがチェンバーからそれているときに本来-1 を返すべきところで初期化されていない値が返されていたため、現在問題なくなっている。

次に 0 付近を詳しく見ていくことにする。図 61 : 0 付近を Sector ごとに分けた比較結果は横軸 Sector、横軸 digi の出力と、PpanelE の差としている。



module = 2,4,6はz = c,d

図 61 : 0 付近を Sector ごとに分けた比較結果

Forward と Endcap 部を分けていないため 1,2,3 で重なってしまっているが、`tgc_wg.dat(hlo.txt)`はa,cを同じに扱っている。これが+2に山ができる原因と考えられる。

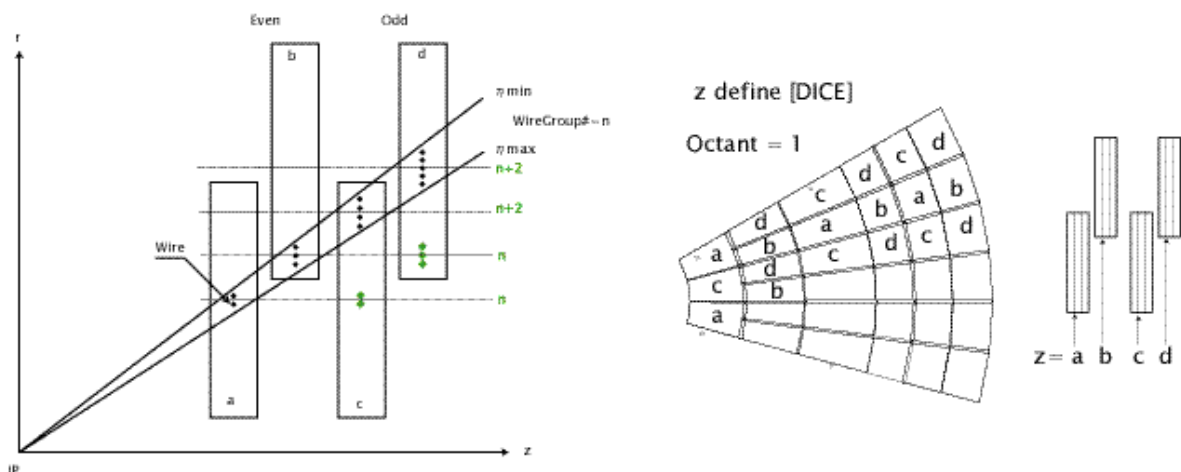


図 62 : チェンバーの z 座標の定義

Odd Sector 部分では Wire Ganging, channel number が Even Sector と同じ構成になっているので min ~ max のとき Odd Sector では channel が n+2(程度)になる。つまり+2にある山は問題がないことわかる。

±1 の山に関しては z の定義が完全に一致せず微小なずれがあることが考えられるが、この比較の目的はプログラムの問題を見つけるためであり、プログラムでデータベースの読み込みに問題がある場合やコネクションに問題がある場合などは適当な値でのずれやセクターごとにばらつきがあるなどの結果になるはずであり、問題がないことが確認できる。

なお Even Sector と Odd Sector にトラックがまたがった場合トリガーすることが出来なくなる。digi.dat で EVEN ODD 間に track がまたがるケースがどれくらいあるか確認した。

atlas-mu-6.dat 148/9993 [track] = 1.48%

atlas-mu+6.dat 100/9504 = 1.05%

atlas-mu-20.dat 40/9813 = 0.40%

atlas-mu+20.dat 10/9994 = 0.10%

Pt が小さいほど増えると思われるので最大 6GeV で 1.48%が捨てられることがわかる。

3.3.2. SlaveBoard, HighPtBoard -TestBoard との比較

SlaveBoard、HighPtBoard はテストボードが完成しているので、シミュレーションの Logic は実際にボードの出力と比較することでテストを行う。Logic の妥当性の検証は DICE の出力をシミュレーションへ入力し、SectorLogic へ入力された R, によって調べる。

テストボードへの入力データは DICE によって生成してもよいが、時間がかかることやさまざまな Pt のトラックを作るのには適していないこと、TGC の設置している領域では磁場がほとんどないためトラックが直線であり DICE に頼る必要がないことなどから、テストボードとの比較用のトラックジェネレータを作った。

テストボードは

PPID Triplet : FWT1, Doublet:FWD1

SLBID Triplet : FWT2, Doublet:FWD3

HPTID EWD1

を想定し $2.36 < \theta < 2.49$, $0 < p < 0.1309$ の領域で triplet layer0, pivot layer1 上にランダムな点を選びその 2 点を通るトラックを作った。

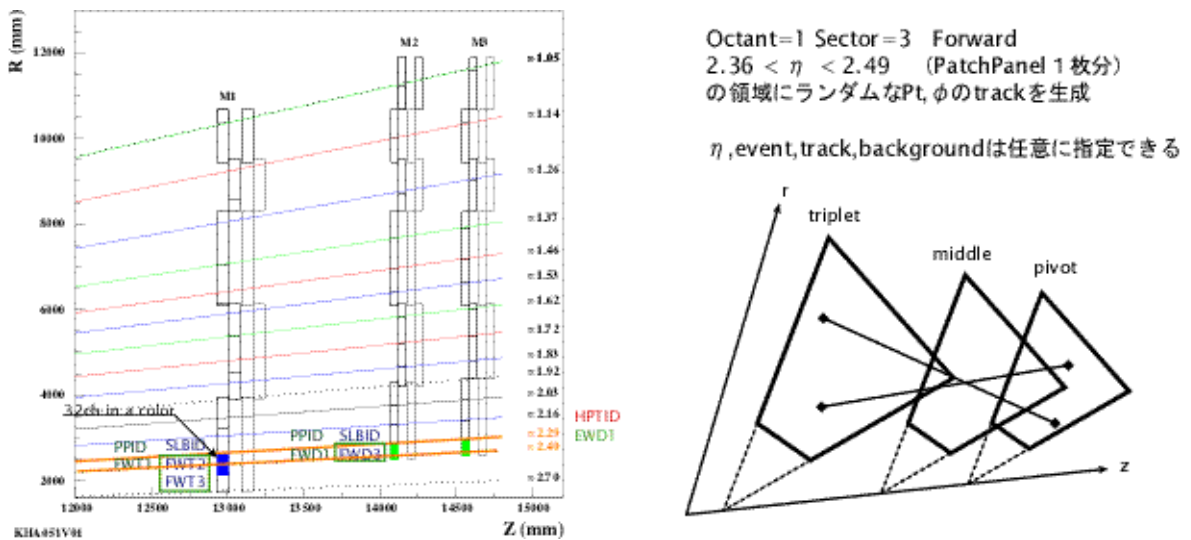


図 63 : テストボード比較用トラックジェネレータ

このトラックジェネレータの出力は DICE の出力と同じフォーマットにしており、トリガーシミュレーションに入力することが出来る。これを digitize, PatchPanel に通し PatchPanel の出力を SlaveBoard のテストボードへと入力した。

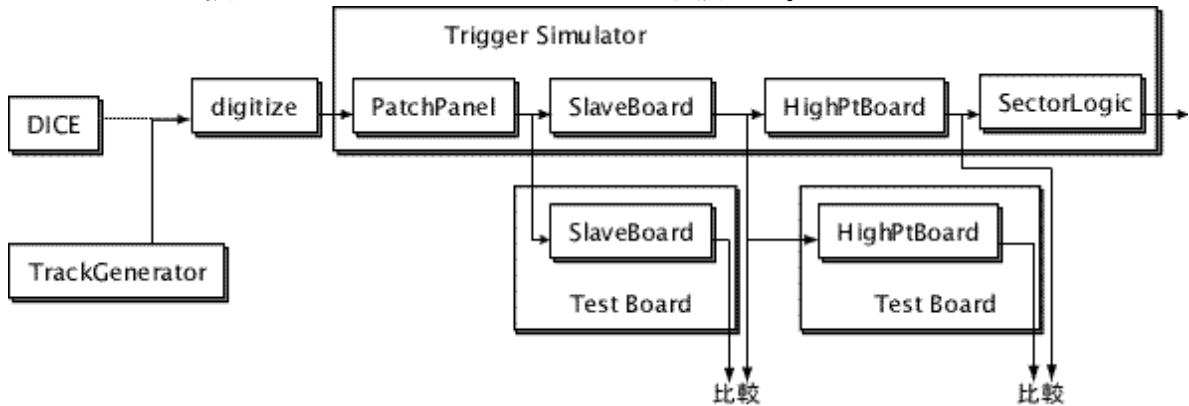


図 64 : Trigger Simulation の概念図

トラックは

| イベント数 | トラック数 | バックグラウンド |
|-------|-------|----------|
| 10000 | 1 | 0 |
| 10000 | 2 | 0 |
| 10000 | 6 | 0 |

で比較を行い一致していることを確認した。

3.3.2.1. TestVector

全てのボードのクラスには TGC **Out というアウトプットのクラスがあるためこれを decode すればそのまま実際のボードとの比較に使うことが出来る。一方入力の前段のアウトプットクラスを幾つか集めて必要なデータに加工²³し入力しているためそのまま使うことが出来ない。インプットクラスがあったほうが良いが現在の TestVector 生成メソッドで

²³ 例として HighPtBoard は Triplet SlaveBoard の隣り合う 4 つの出力を集め、端の 2 つに関してはそれぞれ 3 つあるブロックから 1 つだけを HighPtBoard の入力としている。

は前段のアウトプットを集め加工している。

3.3.3. 運動量 P のトラックでのテスト

SectorLogic を完成させるにあたって R-PhiCoincidenceMap の生成が必要となる。TGCSSCControllerへ入力された R , ϕ を使いMapを生成することになるが、この R , ϕ に問題がないかテストを行う。まず P のトラックを使うことにする。これは R , ϕ が0であるべきだと考えられるため、容易に確認することができ、digitize から SSC の入力までを通したテストにもなる。

P のトラックの生成には trgen を作ったのでこれを使う。別々のツールを使い比較を行う際にはチェンバーのジオメトリ等、各パラメータのコンシステンシーが取れているかどうかといった問題が常に付きまとうため、trgen ではトリガーシミュレーションで使用しているデータベース tgc_wg.dat を参照するよう変更している。

R の出力結果を図に示す。横軸が R 縦軸が η としている。

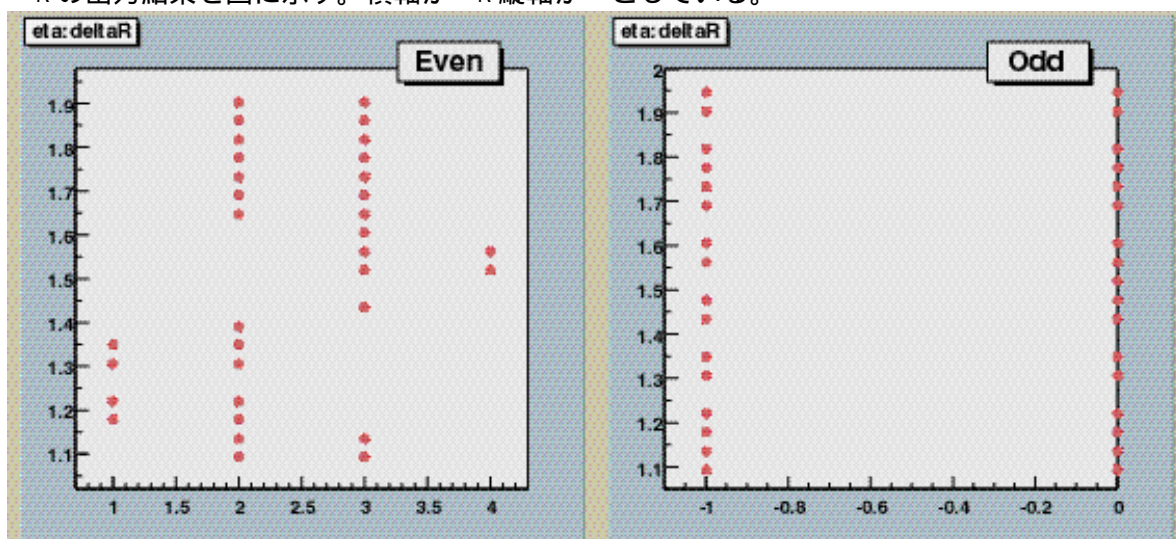


図 65 : シミュレーションが出力した R

これは R の出力を Even、Odd の Sector 別に縦軸に η をとってプロットしている。図を見ればわかるように、Odd Sector では $R=-1,0$ と問題ないものの Even Sector では $R=+2 \sim +4$ となってしまうっており、大幅にずれていることがわかる。この問題が起きるケースについて調べる。

| Sector | | 3 even | | | 4 odd | |
|--------|---|-----------|---|-----|----------|----|
| T1 | 0 | 12 | 3 | } 7 | } | 9 |
| | 1 | 11 | 2 | | | 9 |
| | 2 | 11 | 2 | | | 9 |
| T2 | 0 | 22 | 2 | } 5 | } | 20 |
| | 1 | 22 | 3 | | | 19 |
| T3 | 0 | 14 | 3 | } 5 | } | 11 |
| | 1 | 13 | 2 | | | 11 |

| | | | | | |
|----|---|----|------------|----|------------|
| T1 | 0 | 8 | } [22-5=17 | 5 | } [19-5=10 |
| | 1 | 7 | | 5 | |
| | 2 | 7 | | 5 | |
| T2 | 0 | 13 | | 11 | |
| | 1 | 13 | | 10 | |
| T3 | 0 | 10 | } [19-4=15 | 7 | } [14-4=10 |
| | 1 | 9 | | 7 | |

deltaR 17-15=2 10-10=0

図 66 : Even Sector と Odd Sector の違い

上段の 12,11,11 22,22 14,13 はそれぞれ sector 内でのグローバルな WireGroupName で下段は SlaveBoard に入力された channel である。22-5=17 の 17 が slaveBoard の出力であり、最終的に問題の Even Sector の R は ST1 と ST3 の差をとった 2 となる。問題の sector1,3,5 は Even sector であり Even Sector であることが絡んでいると思われる

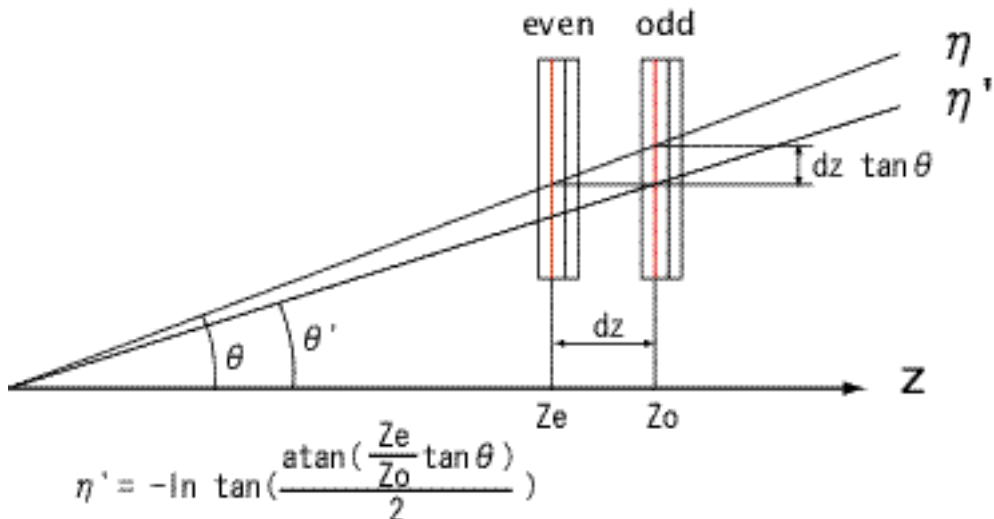


図 67 : Even Sector で問題が出る原因

本来 である track が Even Sector にあたった場合 'として PatchPanel に入力されてし

まう。つまり 9,9,9 が 12,11,11 となっているところである。問題となるのは ST1 では差の合計が 7 であるのに対し ST3 ではこれが 5 となっているところである。これはシミュレーションのバグではなく原理的におきてしまう問題だと思われるので tgc.xls の WireGroup ととの対応表を元に各 WireGroup それぞれの ΔR を計算し、その WireGroup 付近の 1channel が何に相当するかという数値で割ったもの-つまりその WireGroup で何 channel ずれるか-を全ての WireGroup に対して計算した。

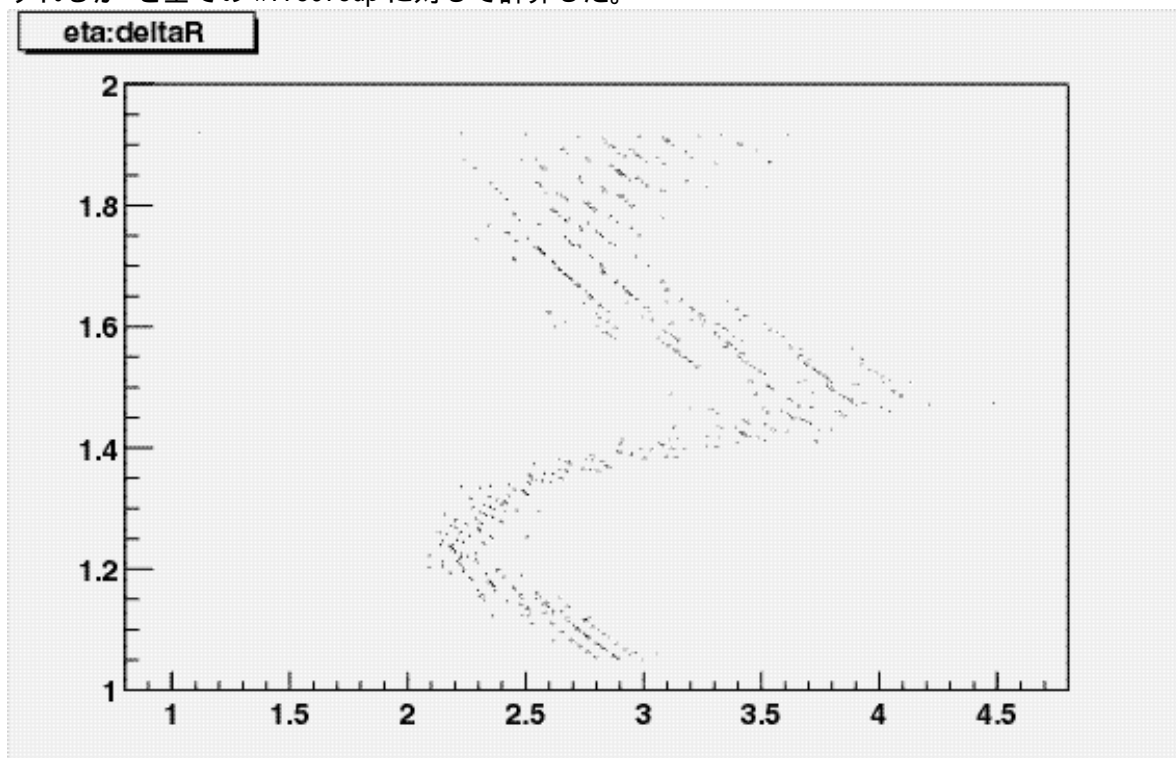


図 68 : Even Sector での ΔR

横軸 R 、縦軸 ΔR としている。なぜこのようにずれてしまうか説明する。同じ P の track があり $\Delta\theta = 7.5$ 度だけずれていて、それぞれ Even Sector と Odd Sector を通過したものとする。Odd Sector を通過した track では ST1 には WireGroup 9 9, ST3 : 11 11 を通過するが Even Sector を通過した track では

$(\Delta R) / ((n[\text{WireGroup}] - (n+1)) [\text{channel}])$ だけずれ ST1 Even : WireGroup 12 11 11, ST3 Even : 14 13 にあたる。Odd Sector にあたった場合の channel のずれは ST1 : WireGroup 3 2 2, ST3 : 3 2 となる。HighPt の ΔR の出力は Odd が 0 であるのに対し Even では $3+2+2 - 3+2 = 2$ となるこの 2 すなわち

$$\sum_{\text{layer}} \frac{(n, \text{layer}, ST1) - (n+1, \text{layer}, ST1)}{(n, \text{layer}, ST1) - (n+1, \text{layer}, ST1)} - \sum_{\text{layer}} \frac{(n, \text{layer}, ST3) - (n+1, \text{layer}, ST3)}{(n, \text{layer}, ST3) - (n+1, \text{layer}, ST3)} (= 2)$$

を各 P でプロットしたものがこの図 68 : Even Sector での ΔR である。

なお ST2、ST3 間でも同様に計算したところ、こちらのケースはほぼ $R=0 \sim 0.8$ と問題のないことが確認された。

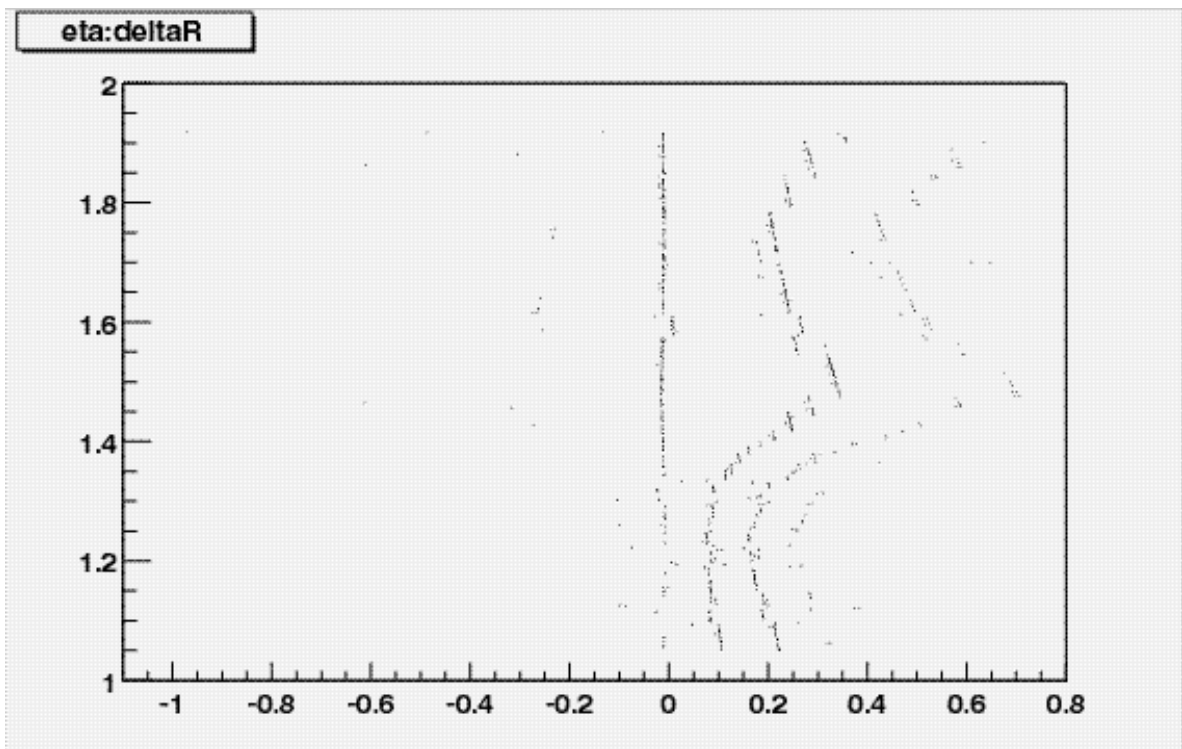


図 69 : ST2, ST3 間の R

さて、図 69 : ST2, ST3 間の R での R に示された ST1, ST3 間のずれが、先のシミュレーションの出力に表れたものであると考えられる。

解決方法は ST1 の差の合計を減らす (Triplet の Even と Odd 間の距離を縮める) か ST3 の合計を増やす (Pivot の Even と Odd 間の距離をひろげる) のだが、残念ながら TGC を移動することは出来ない。

しかしシミュレータに問題ないかを検証するために ST3 Odd を移動させ、 $R=0$ となることを確認する。現在 tgc.xls では ST3 の even-odd の距離は 125mm となっている。125mm から 225mm までを 1mm 間隔で移動させ計算した。

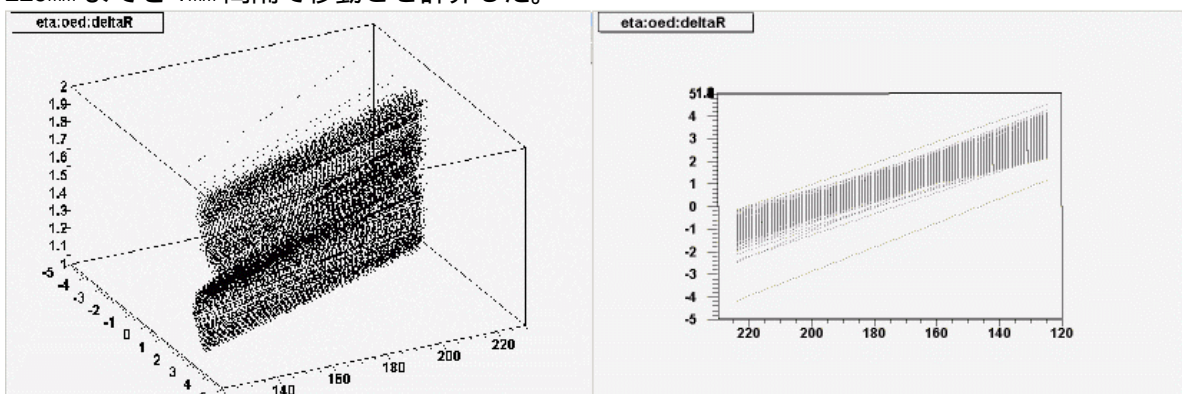


図 70 : ST3 の Even と Odd を移動させた場合の R

この図から 190mm にすれば -1 から +1 の間に収まりそうであることがわかる。ST3 Even を -74mm 移動した結果シミュレーションの出力も -1 から +1 の間に収まった。これによりシミュレータには問題はないと考えられる。

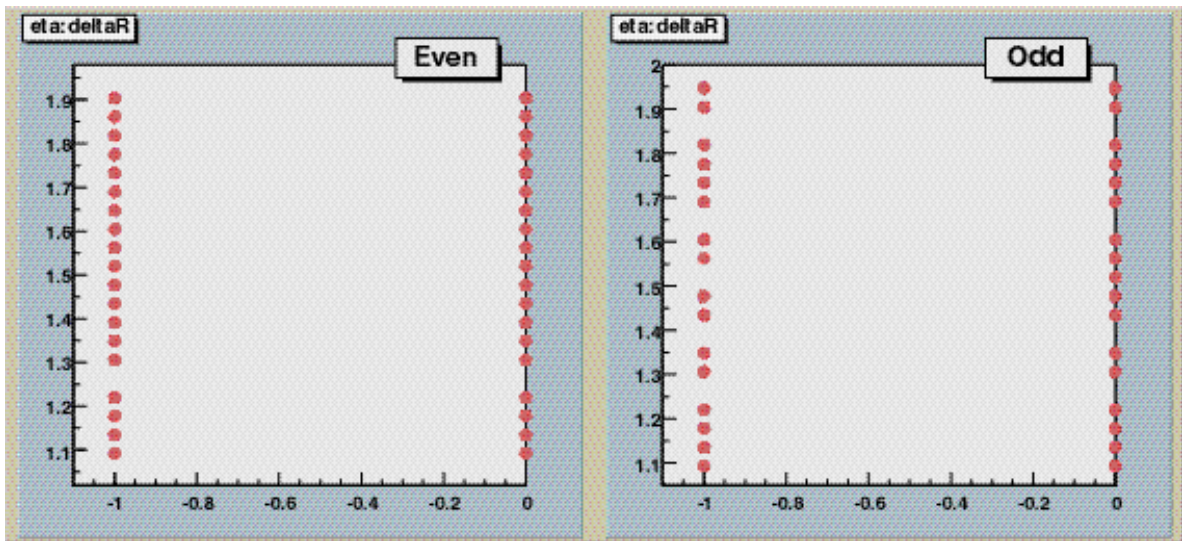


図 71 : ST3 Even Sector 移動後のシミュレーションが出力した R

しかし TGC を移動できない以上 $R=2\sim 4$ ずれてしまう問題は残るが、この問題を考える前に ΔR について検証する。

Pt のトラックに対し、R の出力と同時に ΔR も出力される。その結果 ΔR 全てが +1 であることが確認された。これは 2.3.5.2.2. 章の StripDoubletSlaveBoard で説明したことが原因となっている。

以上のことから最大で $R=4$, $\Delta R=+1$ のずれが生じるが、1 Octant 内での全ての SubSector ごとに別々の R-PhiCoincidenceMap を持つことが出来ること、図 Even Sector での R から任意の $\Delta R \sim +0.015$ 間での R の幅が幸いにも 1 以下であることから、どちらも R-PhiCoincidenceMap によって吸収することにする。しかし 6GeV 以上でも Pt の低いトラックでは $-15 < R < 15$, $-7 < \Delta R < 7$ から外れてしまうケースも少なからず存在し、これをトリガーすることは出来ない。

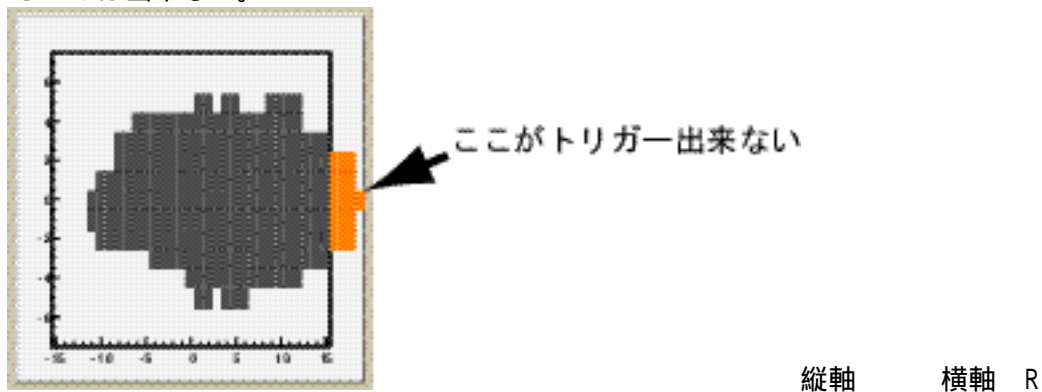


図 72 : R がずれた場合に問題となる点

これによってトリガー効率が落ちてしまうことが予想される。

3.3.4. 視覚化

データベースが正しく構築されているかを確認するために TGC のデータベース tgc_wg.dat を読み込み 3D で表示するツールを作った。のちに digi.dat のトラックを読み込み Track Viewer としての機能も持たせた。

開発には OpenGL という OS に依存しないグラフィックスライブラリを使用し、ウィンドウ制御とユーザーインターフェイスにはそれぞれ GLUT, GLUI を使用している。Tgc_wg.dat の

Viewer では TGCWireGroupDB クラスからデータを取得し表示している。図 73 : tgc_wg.dat の視覚化-1 では台形の1つがTGCとなっている。

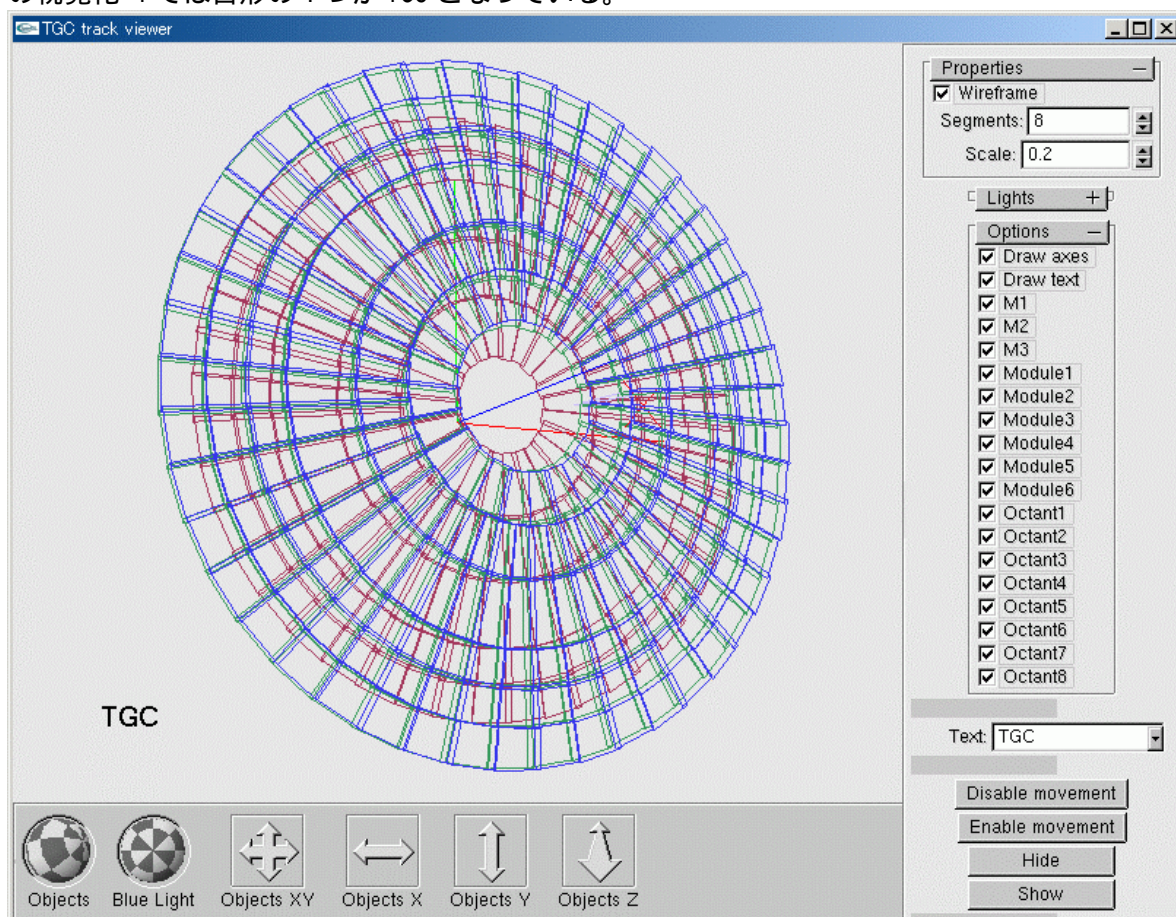


図 73 : tgc_wg.dat の視覚化-1

これによって tgc_wg.dat の持つチェンバーのジオメトリー z , r , h に異常がないことが確認できた。

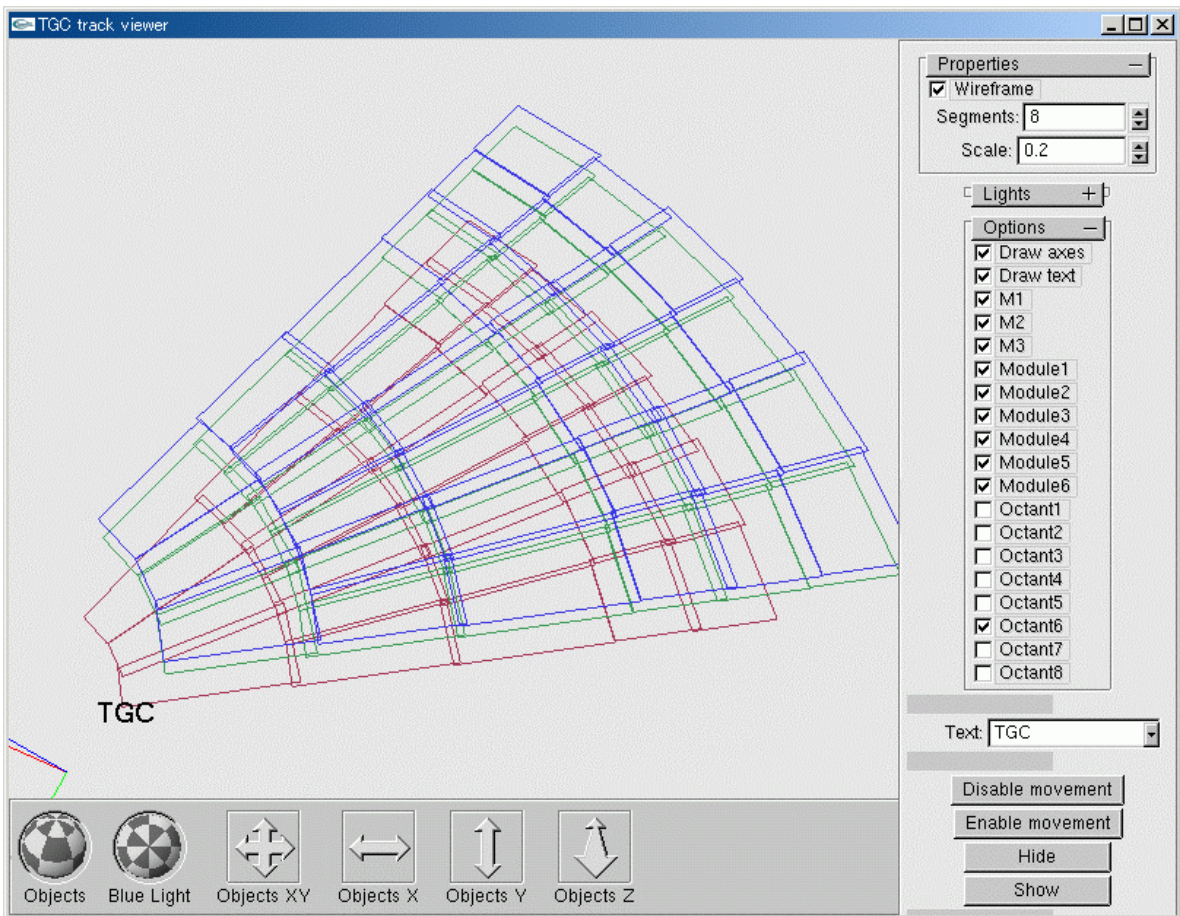


図 74 : tgc_wg.dat の視覚化-2

ウィンドウ左下に表示されている Rotator と Translator の操作で任意の視点からの確認や Octant, Sector の選択を可能にしている。

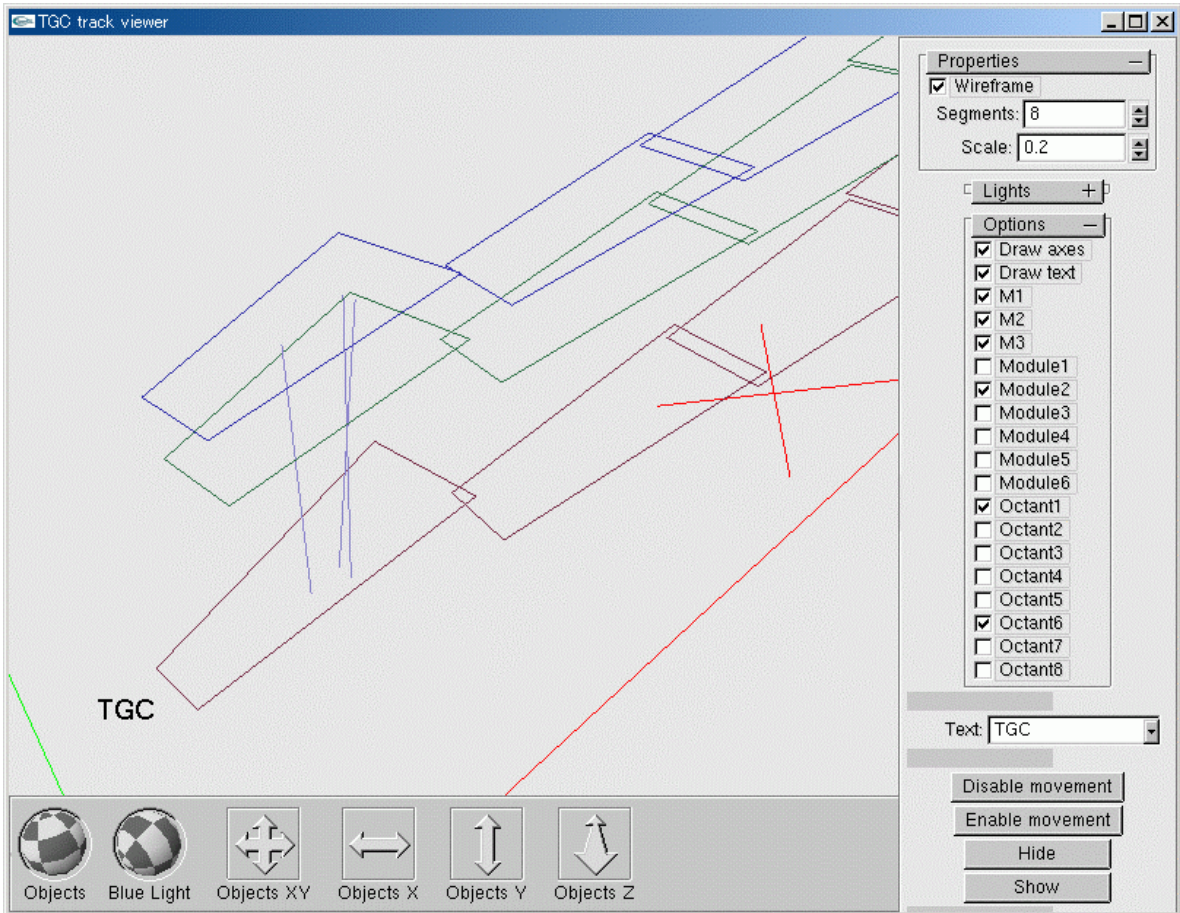


図 75 : digi.dat(track)の視覚化

図のように track がどこを通過したか確認することが出来る。図にはテストボードとの比較で使用した $2.36 < \theta < 2.49$, $0 < \phi < 0.1309$ の領域へ飛ばしたトラックが表示されているが、Octant=1, Forward, Sector=2 を想定どおり通過していることが確認できる。

現在 TGCEvent クラスからトラック情報を取得し表示するよう変更中である。

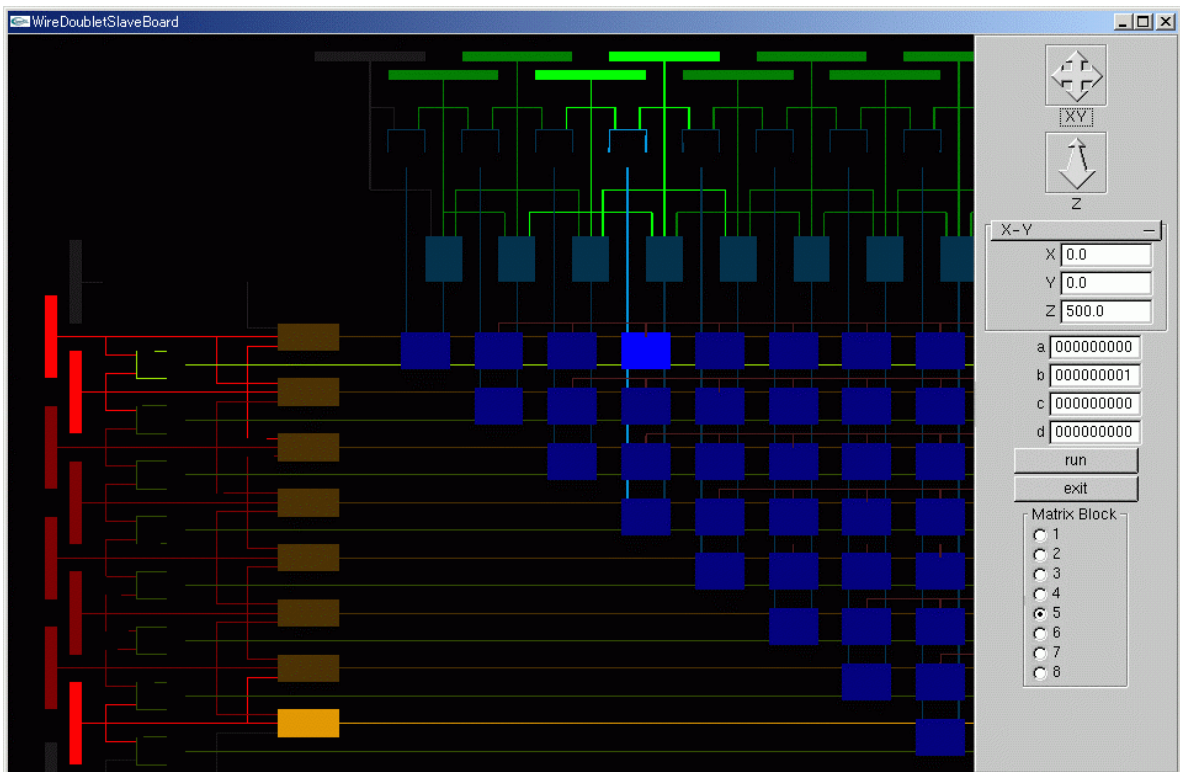


図 76 : TGCWireDoubletSlaveBoard クラスの視覚化

デバッグを容易にするため論理回路の視覚化を行った。これにも OpenGL を使用した。信号が入力された WireGroup や Coincidence の成立した Logic が表示されている。

3.4. R-PhiCoincidenceMap

TGCSSController に入力される R、 から R-PhiCoincidenceMap を生成する。今回は $\pm 6\text{GeV}$ 、 $\pm 20\text{GeV}$ のトラックから 2 段階の PtLevel をもつ Map を作る。R、 の入力を全ての SubSector で重ねて出力したのが次の図である。当初 R-PhiCoincidenceMap の数を減らすことが出来ないか検討をしていたが、Even-Odd 等の問題から全ての SubSector で Map を作るをえなくなった。1 Octant すべての SubSector の個数はかなりの数になるため一定の生成ルールを決め、それに従って Map を作り出すツールが必要である。

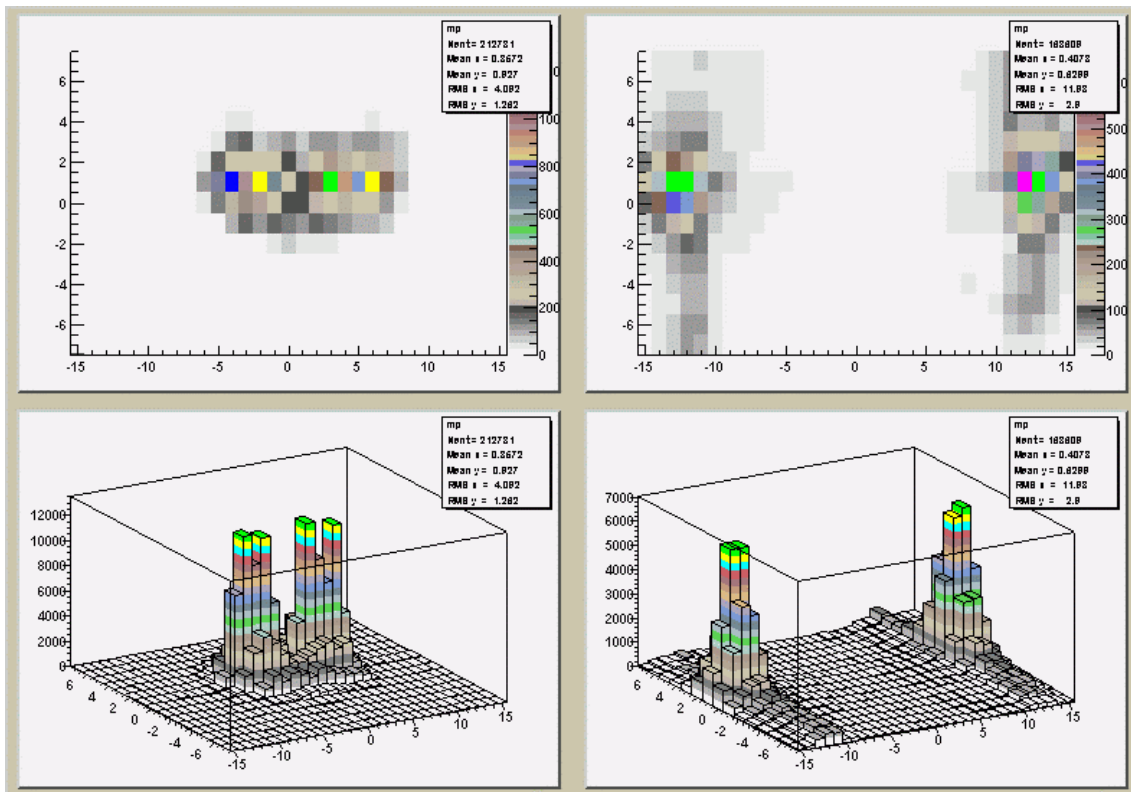


図 77 : TGCSSController に入力された R、

3.4.1. 生成ルール

Map 上に各 PtLevel の入力を置いていくと、次の 3 つの状態が起きると考えられる。

- セル²⁴に 1 つの PtLevel の入力しかない
- セルに複数の PtLevel の入力 (重なる) がある
- セルに何も入力がない

3.4.1.1. セルに 1 つの PtLevel の入力しかないケース

何にも迷う必要はない。その PtLevel に決定する。

3.4.1.2. セルに複数の PtLevel の入力 (重なる) があるケース

重なっている場合一番入力の多かった PtLevel を選ぶ。ただし PtLevel が小さくなるにつれ、同じ入力数が有っても広い領域に分散されてしまう。そこで PtLevel ごとにセル数をカウントしこれによってウェイトを決定し乗じたものを比較する。

3.4.1.3. セルに何も入力がないケース

この場合 2 つのケースが考えられる。

²⁴ Map 上の R、 1 マスをセルと呼ぶことにする。

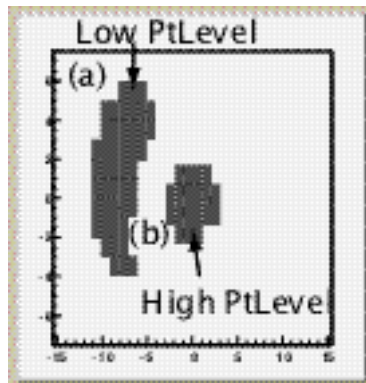


図 78 : 入力がないケース

- (a) そのセルには本当に飛んでこない。図に示した(a)の領域。
 (b) 低い PtLevel と高い PtLevel の山が分かれてしまっている場合、その間の図 78 : 入力がないケースの(b)のような領域は High と Low の中間の Pt を持った粒子が飛んでくる可能性があるの Low PtLevel とすべきである。

何もないセルが(a)に該当するのか(b)に該当するのかの判断は、これまでに挙げてきた生成ルールに従い生成したあと、PtLevel の高い順に1つのセルをとり $R, =0$ の点とに図 79 : 入力がない場合の判定方法のようにラインを引きそのライン上にあるセルがどの PtLevel にも属していない場合その選択されている PtLevel に決定することで判定する。

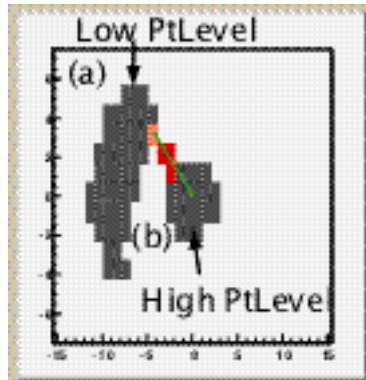


図 79 : 入力がない場合の判定方法

4. まとめ

3章で見てきたように digitize の入力から SectorLogic までの動作の確認を行い

- シミュレータに異常がないこと
 - TGC の配置によって Even sector では R が+2~+4 程度のずれが生じること
 - StripDoubletSlaveBoard の Logic の問題によって が+1 ずれること
- が確認された。

最終的には DICE からのデータを使用してトリガー効率を確認し、R、 のずれによつてトリガーシステムが要求を満たすことが出来るかどうかを示す必要がある。

図表目次

| | |
|---|----|
| 図 1 : L H C | 3 |
| 図 2 : トップクォークの崩壊モード | 4 |
| 図 3 : ゲルオンゲルオン融合 | 4 |
| 図 4 : WW、ZZ 融合..... | 4 |
| 図 5 : W/Z を伴う生成 | 5 |
| 図 6 : τ を伴う生成..... | 5 |
| 図 7 : SM ヒッグス粒子の生成過程別の散乱断面積..... | 6 |
| 図 8 : SM ヒッグス粒子の主な崩壊過程における分岐比 | 7 |
| 図 9 : ATLAS 実験の H_{SM}^0 発見能力 | 8 |
| 図 10 : ヒッグス粒子の質量測定誤差 | 9 |
| 図 11 : ATLAS 検出器..... | 10 |
| 図 12 : ミューオン検出器..... | 11 |
| 図 13 : TGC の構造..... | 12 |
| 図 14 : TGC の TRIPLET と DOUBLET | 12 |
| 図 15 : トリガーシステムの構成..... | 13 |
| 図 16 : TRIGGER SECTOR の構成 (X-Y 断面) | 16 |
| 図 17 : TGC の重なり | 16 |
| 図 18 : TGC の配置 (R-Z 断面)..... | 17 |
| 図 19 : TGC の構成 X-Y 断面..... | 18 |
| 図 20 : TRIGGER の方法 | 19 |
| 図 21 : ミューオントリガーシステムの構成..... | 20 |
| 図 22 : ボードの接続とネーミング | 21 |
| 図 23 : ENDCAPWIREDOUBLETPATCHPANEL | 22 |
| 図 24 : ENDCAPWIRETRIPLETPATCHPANEL | 23 |
| 図 25 : FORWARDWIREPATCHPANEL..... | 24 |
| 図 26 : STRIPDOUBLETPATCHPANEL | 25 |
| 図 27 : STRIPTRIPLETPATCHPANEL | 26 |
| 図 28 : SLAVEBOARD の INPUT-BIT 数..... | 28 |
| 図 29 : DSB での入力から COINCIDENCEMATRIX への接続..... | 29 |
| 図 30 : WDSB の COINCIDENCEMATRIX | 30 |
| 図 31 : が 1 となる問題 | 30 |
| 図 32 : 変更後の SDSB の COINCIDENCEMATRIX..... | 31 |
| 図 33 : WTSB の COINCIDENCEMATRIX | 32 |
| 図 34 : STSB の COINCIDENCEMATRIX | 33 |
| 図 35 : DECLUSTERING LOGIC..... | 34 |
| 図 36 : SLAVEBOARD の OUTPUT-BIT 数 | 34 |
| 図 37 : WIREHIGHPTBOARD の概要..... | 35 |
| 図 38 : HIGHPTBOARD の INPUT の BIT 数 | 36 |
| 図 39 : SLAVEBOARD との接続 | 36 |
| 図 40 : WIREHIGHPTBOARD の概要..... | 37 |
| 図 41 : STRIPHIGHPTBOARD の概要 | 37 |
| 図 42 : WIREHIGHPTBOARD のマトリックス部 | 38 |
| 図 43 : HIGHPTBOARD の MATRIXSTRUCTURE | 39 |
| 図 44 : HIGHPTBOARD の OUTPUT の BIT 数 | 40 |
| 図 45 : HIGHPTOUTPUT..... | 40 |
| 図 46 : SECTORLOGIC の概要 | 41 |

| | |
|--|----|
| 図 47 : HIGHPTBOARD との接続 (ENDCAP) | 42 |
| 図 48 : SECTORLOGIC の INPUT-BIT 数 | 42 |
| 図 49 : SUBSECTORCLUSTER の構造 | 43 |
| 図 50 : TGC と SSC の対応 | 44 |
| 図 51 : HITID と POS の DECODE | 45 |
| 図 52 : LOOKUPTABLE | 46 |
| 図 53 : DIGITIZE に関連するクラスのクラス図 | 47 |
| 図 54 : TGCPATCHPANEL のクラス図 | 48 |
| 図 55 : SLAVEBOARD に関連するクラスのクラス図 | 49 |
| 図 56 : HIGHPTBOARD に関連するクラスのクラス図 | 49 |
| 図 57 : SECTORLOGIC に関連するクラスのクラス図 | 50 |
| 図 58 : 比較の方法 | 51 |
| 図 59 : 実装したクラス | 52 |
| 図 60 : DIGI の出力と TGC.XLS の PPANELE との比較結果 | 53 |
| 図 61 : 0 付近を SECTOR ごとに分けた比較結果 | 54 |
| 図 62 : チェンバーの z 座標の定義 | 55 |
| 図 63 : テストボード比較用トラックジェネレータ | 56 |
| 図 64 : TRIGGER SIMULATION の概念図 | 56 |
| 図 65 : シミュレーションが出力した R | 57 |
| 図 66 : EVEN SECTOR と ODD SECTOR の違い | 58 |
| 図 67 : EVEN SECTOR で問題が出る原因 | 58 |
| 図 68 : EVEN SECTOR での R | 59 |
| 図 69 : ST2, ST3 間の R | 60 |
| 図 70 : ST3 の EVEN と ODD を移動させた場合の R | 60 |
| 図 71 : ST3 EVEN SECTOR 移動後のシミュレーションが出力した R | 61 |
| 図 72 : R がずれた場合に問題となる点 | 61 |
| 図 73 : TGC_WG.DAT の視覚化-1 | 62 |
| 図 74 : TGC_WG.DAT の視覚化-2 | 63 |
| 図 75 : DIGI.DAT (TRACK) の視覚化 | 64 |
| 図 76 : TGCWIREDOUBLETSLAVEBOARD クラスの視覚化 | 65 |
| 図 77 : TGCSSCONTROLLER に入力された R、 | 66 |
| 図 79 : 入力がないケース | 67 |
| 図 80 : 入力がない場合の判定方法 | 67 |
| 表 1 : LHC の主要パラメータ | 3 |

参考文献

- [1]ATLAS Collaboration, ATLAS detector and Physics performance Technical Design Report, ATLAS TDR 14,15 Rev.01 (2000/6/9)
- [2]ATLAS Collaboration, First-Level Trigger Technical Design Report, ATLAS TDR 12 Rev.01 (2000/6/9)
- [3]Naming and numbering scheme for the Endcap muon trigger system version 1.0(2000/6/5)
- [4]戸谷大介, ATLAS 実験ミュオントリガーシステムにおける運動量選別用 ASIC の開発, Master Thesis, 2000

謝辞

TGC エレクトロニクスグループの佐々木修氏、坂本宏氏、福永力氏、狩野博之氏には貴重な意見を頂きました。感謝致します。研究全般にわたり助けていただいた蓮子和己氏に深く感謝します。ソフトウェアについて様々な助言を頂いた神戸大学 蔵重久弥氏に深く感謝します。

戸谷大介氏、香取勇一氏、陣内修氏には適切なアドバイスを頂きました。感謝します。

研究生生活を共にし、良い意味でアドレナリンの分泌を促してくれた一宮亮氏、中村佳央氏、溝内健太郎氏、石田康明氏、田中賢一氏、松本悠氏、小松知氏、辻伸介氏、墨田土詞氏に感謝します。

信州大学長谷川庸司助手には、行き詰まった時に必ず適切な助言を頂きました。深く感謝します。

そして最後に、長い研究生生活をサポートして頂いた指導教官 竹下徹助教授に深く感謝いたします。