修士学位論文

ATLAS実験ミューオントリガーシステムにおける運動量選別用ASICの開発

東京大学大学院 理学系研究科
物理学専攻

86100
戸谷 大介

2000年 1月

概 要

欧州原子核研究機構(CERN)において建設が予定されている大型陽子・陽子衝突型加速器LHCを用いて、標準理論の検証とそれを超える物理の探索を目的としたATLAS実験が2005年の開始を目指して準備が進められている。高い衝突頻度のLHCにおいて行なわれるATLAS実験では、膨大な数の反応の中から重要な事象を落さずに効率良くデータ収集する必要があるため、トリガー・システムの開発は重要である。このトリガー・システムの中で前後方部ミューオン・トリガーでは、ミューオンの運動量の大きさに対して閾値を設けるために、トロイド磁場による飛跡の曲り具合から運動量を測定する。また、バックグラウンドによる偶発トリガーを防ぐために複数の検出器の信号の同時性を要求する。本論文ではこの処理を行なう回路を含んだASICの設計と試作について述べる。また、前後方部トリガー・エレクトロニクス・システムの開発やデータの解析に必要なシミュレーション・コードのプロトタイプの設計・開発とASICの設計への応用について述べる。

目 次

1 序論 **3**
1.1 ATLAS 実験・LHC 計画 . . . 3
1.2 ATLAS 検出器 . . . 5

2 ATLAS 実験におけるトリガー・DAQ システムの構成 **9**
2.1 トリガーシステム全体 . . . 9
2.2 Level-1・ミューオン・トリガーシステム . . . 10

3 Slave Board High-P_T Board 用 ASIC の開発 **15**
3.1 Slave Board と High P_T Board の機能 . . . 15
3.2 Slave Board 用 ASIC の仕様 . . . 16
3.3 High-P_T Board 用 ASIC の仕様 . . . 22
3.4 試作の目的 . . . 24
3.5 VDEC に提出したチップのデザイン . . . 24
3.6 VDEC における試作の流れ . . . 25
3.7 シミュレーションによるデザインの検討 . . . 27
3.8 結論・今後の予定 . . . 30

4 TGC トリガーシミュレーション用プログラムの設計と開発 **32**
4.0.1 旧 ATLAS software の構成 . . . 32
4.1 新しい ATLAS Software の状況 . . . 32
4.2 オブジェクト指向解析・設計 . . . 33
4.2.1 UML の説明 . . . 33
4.3 トリガーシミュレーションプログラムの設計 . . . 35
4.3.1 プログラムの特徴・機能 . . . 35
4.3.2 解析 . . . 36
4.3.3 設計 . . . 37
4.4 実装とテスト . . . 45
4.5 トリガーエレクトロニクスの設計への応用 . . . 48
4.6 まとめ . . . 50

5 全体のまとめ **51**

6 謝辞 **53**

Appendix A VDEC に提出したチップの回路図 **54**

Appendix B トリガーシミュレーションプログラムについての補足 **62**
B.1 naming と numbering scheme . . . 62
B.2 まだハードウェアのデザインが固まっていない部分 . . . 62

1 序論

1.1 ATLAS 実験・LHC 計画

標準理論では物質の最小構成単位は 3 世代のクォークとレプトンと呼ばれるスピン 1/2 のフェルミオンである。また素粒子間の相互作用は電磁相互作用と弱い相互作用を統一した電弱相互作用と強い相互作用であり、それぞれ SU(2)×U(1),SU(3) の対称性をもつゲージ理論として記述される。この標準理論はこれまで行なわれた実験から得られる結果を良く説明し成功をおさめている。しかしその標準理論の中で電弱相互作用の対称性の破れを説明するヒッグス機構で登場するヒッグス粒子はまだ発見されていない。対称性の破れの機構を理解するためにはヒッグス粒子の発見とその性質の測定が不可欠であり、現在の高エネルギー物理の重要な課題である。

現在スイスのジュネーブにある欧州素粒子物理研究所 (CERN) において LHC(Large Hadron Collider) が 2005 年運転開始を目標に建設が進められている。LHC は重心系エネルギーが 14TeV の陽子陽子衝突型加速器で、世界最高エネルギーの加速器である。ATLAS 実験ではこの LHC を用いて標準理論の検証とより高いエネルギー領域における新しい現象の探索を行なう。

ATLAS 実験では標準理論のより正確な理解を目的としてヒッグス粒子の他に LHC で大量に生成されるトップクォークや B 中間子の研究が行なわれる。また LHC で探索可能な標準理論をこえる物理では超対称性模型がある。これらのトピックスについて簡単に説明する。

Higgs 粒子

LEP における直接探索で発見されない場合、最終的に 110GeV/c^2 程度までの質量のヒッグス粒子が除かれる。一方 LEP や SLD での電弱相互作用の精密測定から質量は 100∼300GeV/c^2 の間であると予測されている。素粒子としてのヒッグス粒子の質量の上限は 1TeV 程度であるいわれているが LHC では LEP による下限 100GeV からその理論的な上限の 1TeV 付近までの質量のヒッグス粒子をとらえることが可能である。

ヒッグス粒子の生成断面積と崩壊の分岐比は図 1,2 のようになっている。ヒッグスの質量が W^+W^-,Z^0Z^0 への崩壊の閾値より小さい領域では崩壊可能な粒子のうち、質量の最も大きいボトムクォークへの分岐比が最も大きい。質量の小さい領域ではこの崩壊モードで探索する (図 3)。ボトムクォークのジェットだけでは QCD ジェットのなかから区別してトリガーすることは難しいので、トップクォークの生成に伴って生成するヒッグス粒子からの崩壊を探す。しかし、図 1 を見ると分かるようにこれらの生成過程はグルーオン融合に比べて 1 桁から 2 桁ほど生成断面積が小さい。このため単独でトリガーできる 2 つの光子へ崩壊するチャンネルは、カロリーメーターが高いエネルギー・角度分解能を実現できれば有効なチャンネルである。質量が大きい領域では Z^0Z^0、W^+W^- への崩壊が可能になり発見しやすくなる。シミュレーションでは $H\to\gamma\gamma$ のモードを含めれば、年間 10fb^{-}1 の積分ルミノシティが得られるとすると 3 年間で 1TeV までの質量領域で 5σ でヒッグス粒子を発見できる。

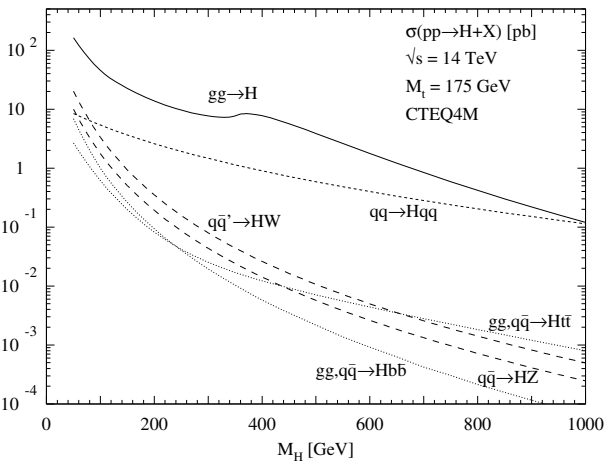


図 1: ヒッグス粒子の生成断面積 [1]

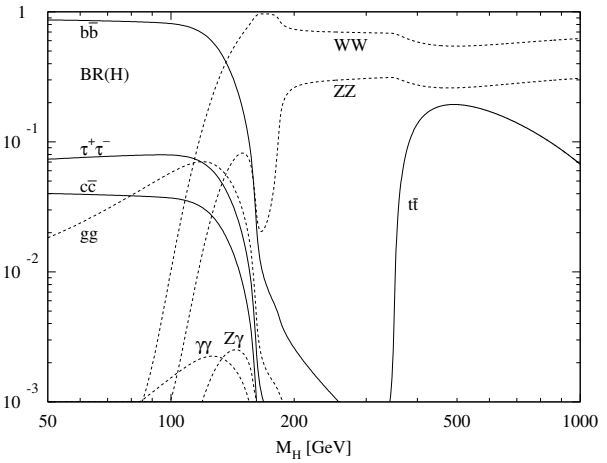


図 2: ヒッグス粒子の崩壊の分岐比 [1]

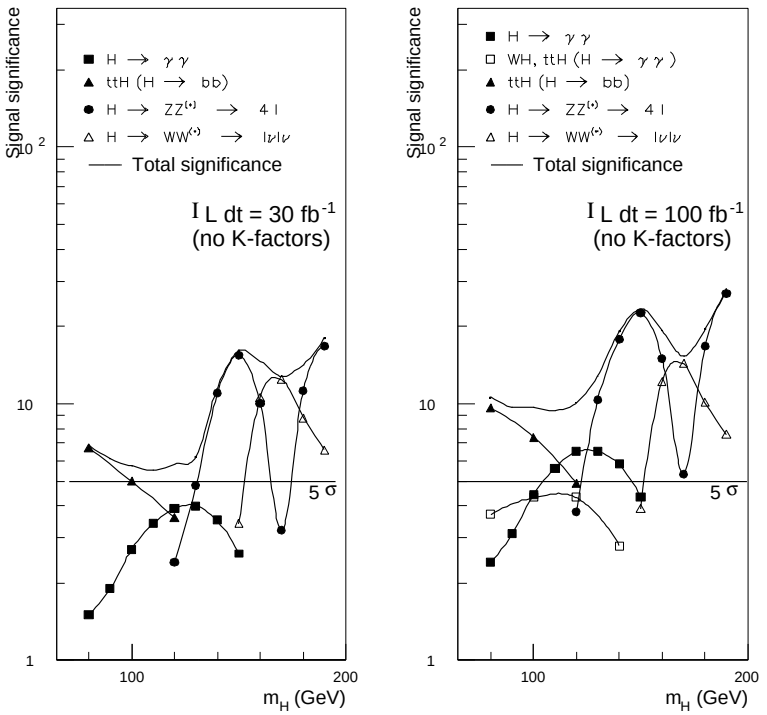


図 3: ATLAS でのヒッグス粒子の発見能力 [2]

Top quark の精密測定

トップクォークの質量は電弱相互作用の基本的なパラメータであり、その精密測定は特に電弱理論のヒッグス機構に関連する部分を調べる上で重要である。LHC では $t\bar{t}$ 対の生成断面積は Tevatron の 100 倍程度あるためにトップクォークの精密測定を行なうことができる。また大量に生成する上に崩壊して W,b を生成するために、ヒッグス粒子や SUSY 粒子の大きなバックグラウンドとなる。これらの粒子の発見のためには生成断面積や崩壊幅などの性質を詳しく調べる必要がある。現在のシミュレーションではトップクォークの質量は W の質量の測定誤差が 20MeV 以下であれば約 2GeV の精度で測定できると予想されている。

$t\bar{t}$ 対生成以外ではトップクォークの単独生成は $|V_{tb}|$ の決定や $\Gamma(t \to bW)$ の直接測定が可能になる点で重要である。

B 中間子の精密測定

ATLAS 実験では $B_d^0 \to \pi^+\pi^-$,$B_d^0 \to J/\psi K_s^0$, $B_s^0 \to J/\psi\phi$, の崩壊の非対称度を測定することで、それぞれユニタリー三角形の三つの角を決定できる。また B_s^0,B_d^0 のフレイバー振動の周期と $b \to ul\nu$ の崩壊率からユニタリー三角形の 2 つの辺 $|V_{td}|$,$|V_{ub}|$ が求まるので、間接的に CP の破れを測定できる。

超対称性粒子

LHC では超対称性粒子のうちクォークとグルーオンの超対称性パートナーであるスクォークとグルーイーノが大量に生成される。R パリティが保存する場合、超対称性粒子を示す信号はスクォークやグルーイーノがカスケード崩壊をして生じる複数のジェットとレプトン、最軽量超対称性粒子 (LSP) の最軽量ニュートラリーノによる missing E_T である。ニュートラリーノは B,W^0 の中性ゲージ粒子と H_u, H_d の中性の Higgs 粒子のスパートナーが電弱対称性の破れのために混合したものである。

1.2 ATLAS 検出器

図 4 に ATLAS 検出器の全体を示す。半径 11m、ビーム軸方向の長さはバレル部が 25m, 中心から最も外側に位置する muon chamber までの長さが 23m である。総重量は 7000ton である。

Magnet System Inner Detector のための Central Solenoid(CS) と Muon Spectrometer のための air-core toloid からなる。air-core toloid はバレル部とエンドキャップ部で分けられており、BT と ECT と呼ぶ。CS の中心での磁場は 2T である。BT と ECT の磁場の経路についての磁場の積分値はそれぞれ 2∼6Tm、4∼8Tm 程度である。

Inner detector 内部飛跡検出器は Pixel Detector,Semi Conductor Tracker(SCT),Trasition Radiation Tracker(TRT) からなる。(図 5)

Pixel Detector は Ineteraction point の近くで高精度の二次元の位置測定を行なう silicon detector である。バレル部では 3 層、エンドキャップ部では 5 層になっている。最も内側の層の半径は 4cm であり半径 3cm のビームパイプのすぐ外側に配置されている。

SCT は Pixel detector と TRT の中間の領域で高精度の位置測定を行なうことを目的とした silicon 検出器で運動量、インパクトパラメータ、セカンダリバーテックスの位置の測定に用いられる。$6.36 \times 6.40\text{cm}^2$ の silion micro strip detector を二枚重ねることで二次元の読み出しを行なっている。バレル部 4 層, エンドキャップ部 9 層になっている。

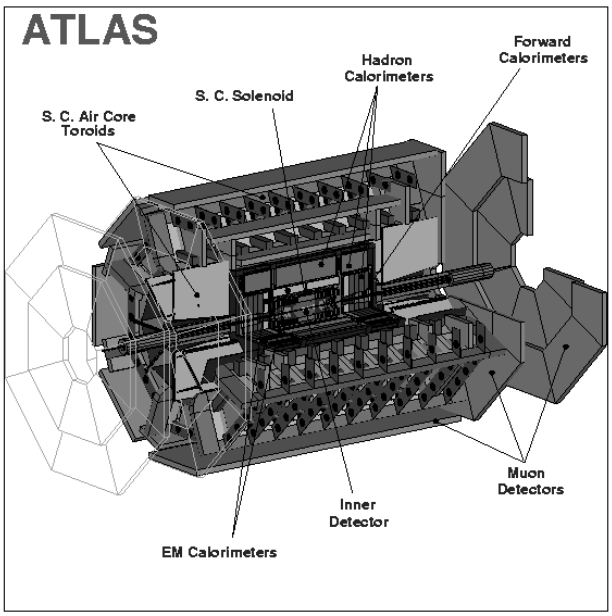


図 4: ATLAS 検出器

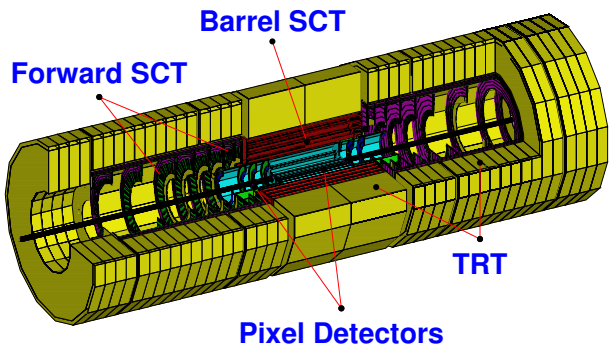


図 5: 内部飛跡検出器

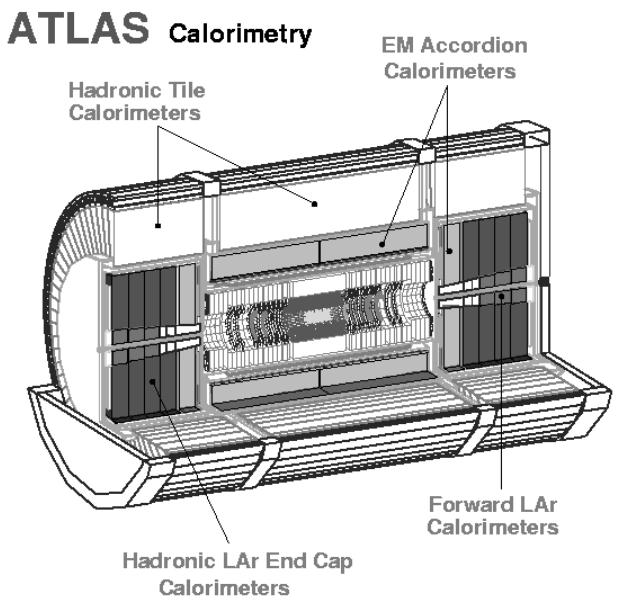


図 6: カロリーメータ

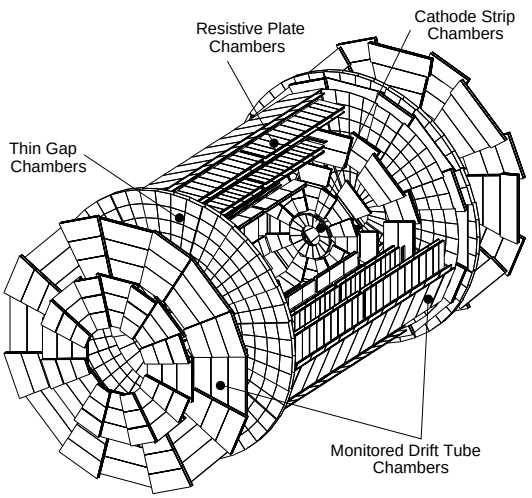


図 7: ミューオンシステム

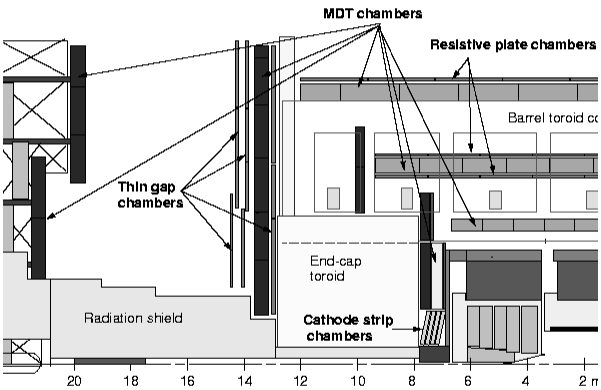


図 8: ATLAS 検出器の RZ 断面図。

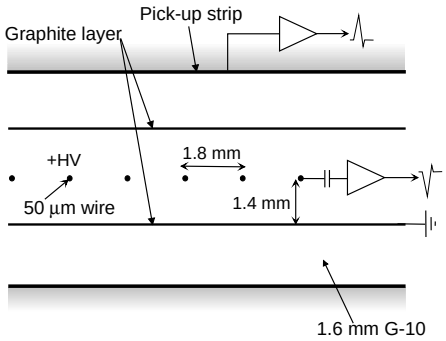


図 9: TGC の信号線の構造

TRT は直径 4mm、長さ最大 150cm の xenon ガスを封印した straw detector とその間の輻射体からなる飛跡検出器である。輻射体からの遷移輻射を straw detector で検出する。straw tube の半径が小さく、信号線ごとにガスの領域が分けられているために高い入射粒子頻度の環境でも動作が可能である。

Calorimeter　Electro-magnectic calorimeter として読みだし電極がアコーディオン状の鉛-液体アルゴン (LAr) の検出器が使われている。その前には presampler が配置され、カロリーメータより内側の内部検出器や coil でのエネルギー損失を補正するために使われる。バレル部のハドロンカロリーメータは鉄の吸収体にプラスチックシンチレータのタイルが埋め込まれたサンプリングカロリーメータである (図 6)。高い放射線耐性が要求されるエンドキャップ、フォーワード部では銅 LAr カロリーメータとタングステン LAr カロリーメータが使われる。

Muon Spectrometer　Muon spectrometer は高精度の運動量測定のための MDT (Monitored Drift Tube) と CSC(Cathode Strip Chamber) と、トリガーと第 2 座標測定のための TGC(Thin Gap Chamber) と RPC(Resistive Plate Chamber) がある (図 7)。MDT は直径 30mm で Ar 93% $CO_2$7% のドリフトチューブで、3 bar の圧力をかけて使用する。バレル、エンドキャップ部ではそれぞれトロイド磁場によるミューオンの飛跡のサジッタと変向角の測定を行ない半径方向の運動量成分を求める。CSC はミューオンシステムの中で最も衝突点に近く η の大きい領域での運動量測定を行なうストリップ読み出しの MWPC である。RPC はバレル部の muon trigger 用のベークライトの板の間のガス中で放電を起こすワイヤーのないガスチェンバーである。

TGC はエンドキャップ部のミューオントリガー用の薄型の MWPC である (図 8)。検出器の内側から 2 番目の MDT の隣に配置されている。TGC は最も内側に位置し 3 層の検出器からなる triplet chamber と MDT の外側に位置し 2 層の検出器からなる doublet chamber から構成されている。doublet chamber は 2 つ配置されている。それぞれの検出器からはワイヤーグループの信号とそれと垂直なカソードストリップからの信号を読み出すことができる (図 9)。ストリップの間隔は 4mrad(フォワード部では 8mrad) である。カソードストリップの信号も読み出し第二座標の測定に用いる。実験では 1 つの triplet chamber と 2 つの doublet chamber を配置し、合計 7 層の検出器からの wire group と strip の 2 方向の信号を読み出す。2 つの doublet は内側から middle(または inner) doublet chamber, pivot doublet chamber と呼ぶ。これ以降では triplet chamber, doublet chamber をそれぞれ triplet, doublet と省略する。また今後場所によってはワイヤーグループを単にワイヤ、カソードストリップをストリップと省略する場合がある。

2　ATLAS 実験におけるトリガー・DAQ システムの構成

2.1　トリガーシステム全体

ATLAS 実験では約 1GHz の反応率、40MHz のバンチクロッシングレートで発生する粒子の中から、重要な事象からの粒子を選別し、最終的にデータストレージへの書き込みが可能な 100Hz 程度のレートまで下げなければならない。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 (図 10)。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していく。

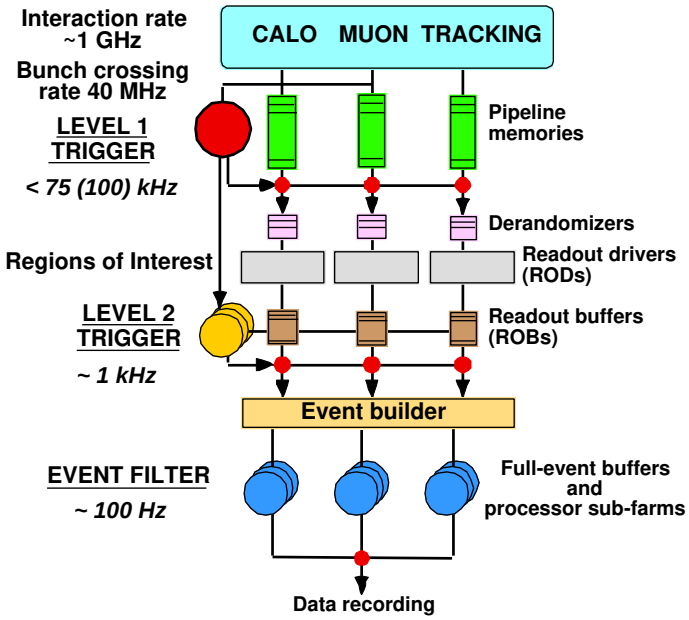


図 10: Muon trigger system の構成

Level-1 trigger

Level-1(LVL1) トリガーではトリガー用ミューオン検出器から得られる R 座標と ϕ 座標と横運動量と、全てのカロリー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ーメータのエネルギー情報は位置の精度を落している。この選別により 75kHz(最大 100kHz まで上げることは可能) までイベント・レートを落す。

Level-1 トリガーのバンチクロッシングからの Latency は 2.5μsec である。LVL1 での選別は主にコインシデンスとベト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ー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 トリガーでの判定が終るまで ROB に保持される。

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。

ミューオンについては 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。またカロリーメータの情報を合わせることで、ミューオンの 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。

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 1MB/1event なので 100MB/s に相当する。

2.2 Level-1・ミューオン・トリガーシステム

Muon trigger system の役割はミュー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し、高い P_T のミューオンを含む事象を選択することである。その際にミューオンの P_T について閾値を設けて、その閾値以上の P_T を持つミューオンだけを選択する。現在のデザインではミューオンの P_T の閾値は Low P_T 用 3 段階、High P_T 用 3 段階の計 6 段階が用意されている。

LVL1 ミュー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 P_T 閾値ごとのミューオンの数の算出
- 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。P_T 閾値の定義はこの値より大きな P_T のミューオンのトリガー効率が 90% 以上になる P_T の閾値である。

また性能に対する要求は

- P_T threshold を 6~35GeV の範囲で設定できる。
- $|\eta|$ <2.4 のラピディティの範囲においてミューオンに対する平均のアクセプタンスが 90% 以上である。

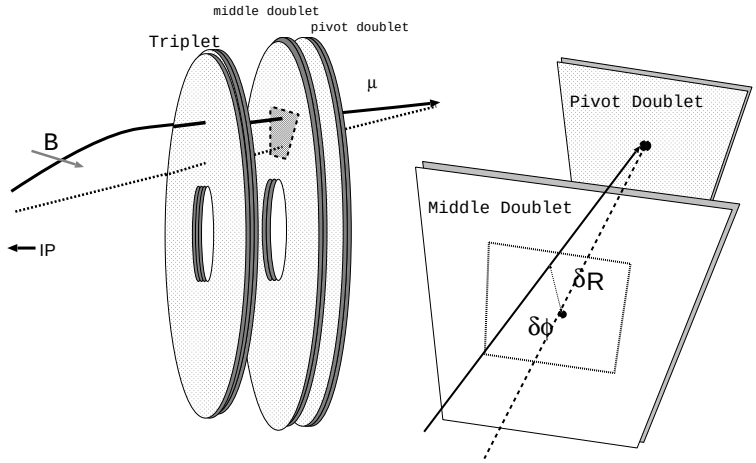


図 11: TGC での P_T による選別の仕組み。

- 設定した P_T 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 P_T が設定した threshold の値よりも大きいミューオンが 25% 以上あることなどである。しかしこの要求はミュー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ー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 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。ここでコインシデンスとは対応する (この場合 rapidity がほぼ同じ位置にある) 複数のチャンネルの中である数以上のチャンネルに信号があることを指す。例えば 2 out-of 3 のコインシデンス条件とは対応する 3 チャンネルの内 2 チャンネル以上で信号があるという条件のことである。このコインシデンスの判定は後述する Slave Board, High-P_T 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 P_T トラックと呼び、High-P_T Board のコインシデンスの条件を満たした信号を High P_T のトラックと呼ぶことにする。

P_T の閾値よる選別を行なうためには、Level-1 トリガーでもミューオンの P_T を求める必要がある。TGC では図 11 に示したようにトロイド磁場によるミューオンの変向角から P_T を測定する。

低い P_T のミューオンは Slave Board で求めた middle doublet と pivot doublet の間での R 方向、ϕ 方向のミューオンのヒット位置の差 $\delta R, \delta\phi$ 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 P_T のミューオンは High-P_T Board で求めた triplet と pivot doublet の間の $\delta R, \delta\phi$ 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-P_T Board からの $\delta R, \delta\phi$ について P_T の threshold の値ごとに $R-\phi$ coincidence を行なう。ここで P_T の大きさと $\delta R, \delta\phi$ 対応づけられ、閾値ごとにトリガーしたミューオンの数が決定する。

図 12 に Level-1 エンドキャップミューオントリガーのエレクトロニクスシステムの構成を示す。TGC からの信号は Patch Panel でバンチ クロッシング ID、OR'ing の処理された後、Slave Board では wire group,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 P_T Board ではトリプレットとダブレットのコインシデンスの結果を用いて wire group,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 wire group と stirp の信号、つまり R 座標と ϕ 座標についてのコインシデンス ($R-\phi$ コインシデンス) を判定する。最終的には P_T の閾値ごとの

トリガーしたミューオンの個数を Level-1 trigger の結果として出力する。

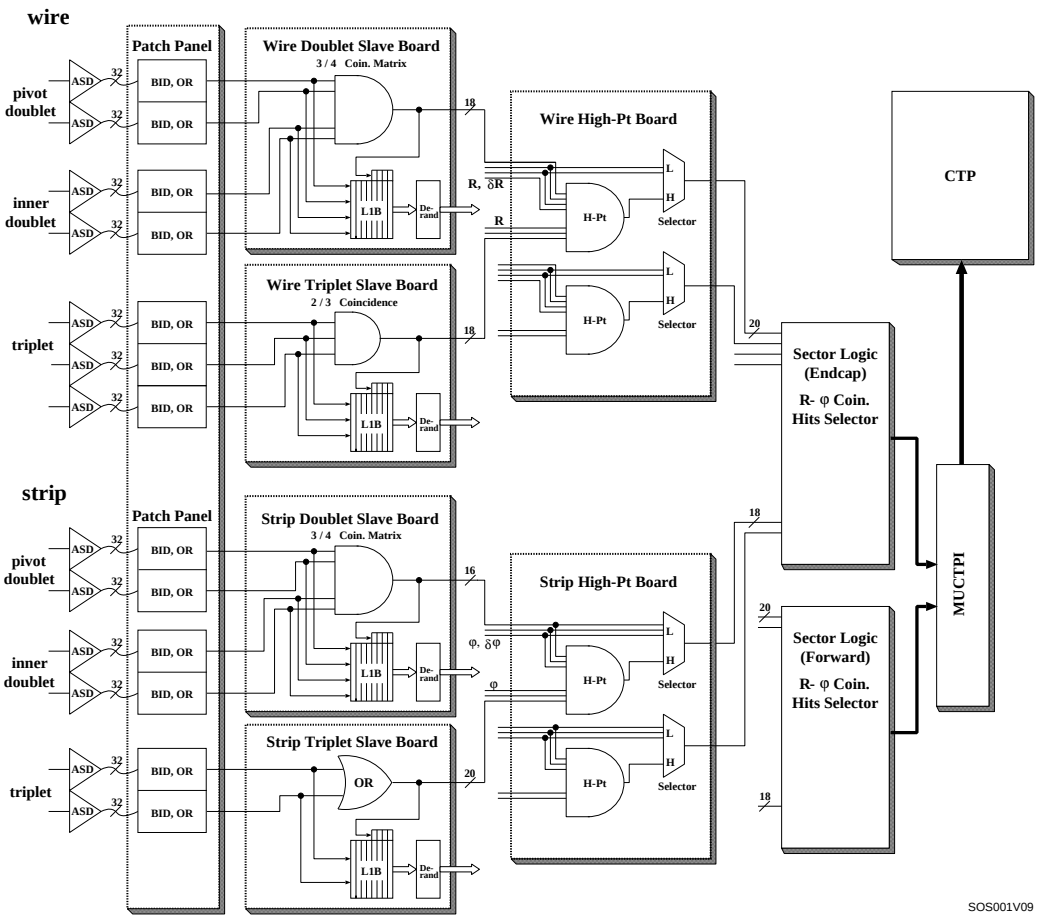


図 12: Level1 Endcap muon trigger の構成

次に構成するモジュールごとの説明をする。

Patch Panel

Patch Panel の主な機能はバンチクロッシングの ID、Slave Board への信号の分配、そして信号の OR'ing である。バンチクロッシング ID(BCID) とは信号がどのバンチの beam の衝突からのミューオンなのかを割り当てることを指す。TGC では ASD(Amplifier Shaper Discriminator) Board から出力される非同期の信号を LHC の Bunch crossing の周波数 (LHC clock) の 40MHz に同期させることを行なっている。Bunch ID された後の信号は Slave Board へ分配される。半径方向 (R 方向) の chamber の境界では inefficient な部分を作らないために wire group、strip の信号はともに SlaveBoard に送る前に OR が取られる。

この他の機能としては ASD への電源、ディスクリの閾値電圧、テストパルスの供給や Slave Board への電源、LHC clock を含む Timing and Trigger Control(TTC) 信号の供給を行なう。Patch Panel は次に述べる Slave Board とともに PS Pack と呼ばれるフレームに収められて TGC に面して配置される。

SlaveBoard・High-P_T Board

Slave Board と High-P_T Board については 3 で詳しく述べるので、ここではエンドキャップ・トリガーシステムにおけるこれらのモジュールの役割について述べる。これらのモジュールの役割は大まかに言ってワイヤーグループ, ストリップごとのコインシデンスと横運動量の測定である。正確には後に続く SectorLogic で行なうように R 方向と ϕ 方向の 2 方向の変向の情報が必要であるが、R 方向の成分がが横運動量をほぼ決定づけているので Slave Board と High-P_T Board が横運動量を決めていると考えて良い。

Low P_T のミューオンに対してはまず pivot doublet と middle doublet の計 4 層のチェンバーからのワイヤーグループ (ストリップ) の信号に対して、3 out-of 4 のコインシデンスをとる (図 12)。そしてコインシデンス条件を満たすトラックについて pivot doublet と middle doublet でのヒット位置の差 $\delta R(\delta\phi)$ を求める。triplet ではワイヤーグループ (ストリップ) の信号に対して 2 out-of 3(1 out-of 2) のコインシデンスをとり、ヒット位置だけを求める。ここで strip 信号用の Triplet Slave Board でのコインシデンスの条件が 1 out-of 2 になっているのは triplet のまんなかのチェンバーのストリップからの信号は読み出さないためである。高い P_T を持ったミューオンの場合 middle doublet と pivot doublet のヒット位置の差は小さいので、代わりに High-P_T Board で triplet と doublet でのヒット位置の差を求める。High-P_T Board は Sector Logic に位置 (R,ϕ), ヒット位置の差 (δR,ϕ) を送る。High-P_T Board では High-P_T Board でのコインシデンスがとれない場合には Slave Board での δR,$\delta\phi$ を sector logic に送る。これは High-P_T Board 上の selector で行なわれる。δR,$\delta\phi$ がどちらのボードから得られたものかを示すためにそれを区別するフラグも sector logic に送られる。

SectorLogic

Sector Logic の役割は R-ϕ coincidence と P_T の決定である。1 octant は図 13 のような Trigger Sector,SubSector に分けられており、この Trigger Sector のトラックから最も高い P_T を持つもの 2 つを選ばなければならない。High-P_T board までで Rϕ それぞれの方向に対して 2 つの SubSector ごとに 1 つのヒットまで絞られる。Endcap の場合最大で R 方向で 19 ヒット、ϕ 方向で 2×5 ヒットあることになるが、現在のデザインでは R 方向に 2 個 ϕ 方向に 4 個の SubSector の集まりである SubSectorCluster ごとに分けられこの中で一つのヒットについて、P_Ts 閾値のレベルごとに R-ϕ コインシデンスが判定される。PreSelector ではこの 19 個の SubSectorCluster について、P_T のレベルごとに R の大きい方から 2 番目までのヒットを選択する。さらに Selector では各レベルの PreSelector の出力の中から P_T の高い方から 2 番めまでのヒットを選択する。この 2 ヒットの P_T Level, 位置 (R,ϕ) が MUCTPI(Muon Central Trigger Processor) に送られる。

Level-2 trigger ではこの SectorLogic から得られる R, ϕ の座標からから RoI を決定する。

図 13: Sector の分割

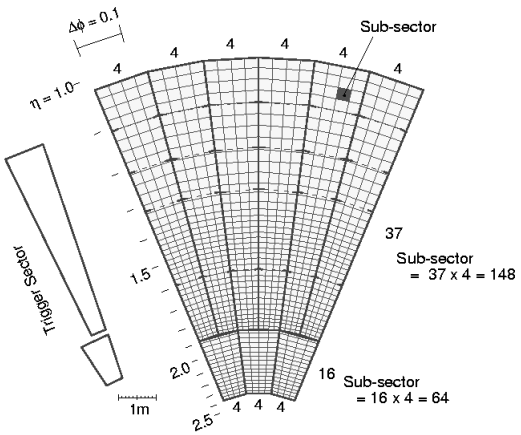
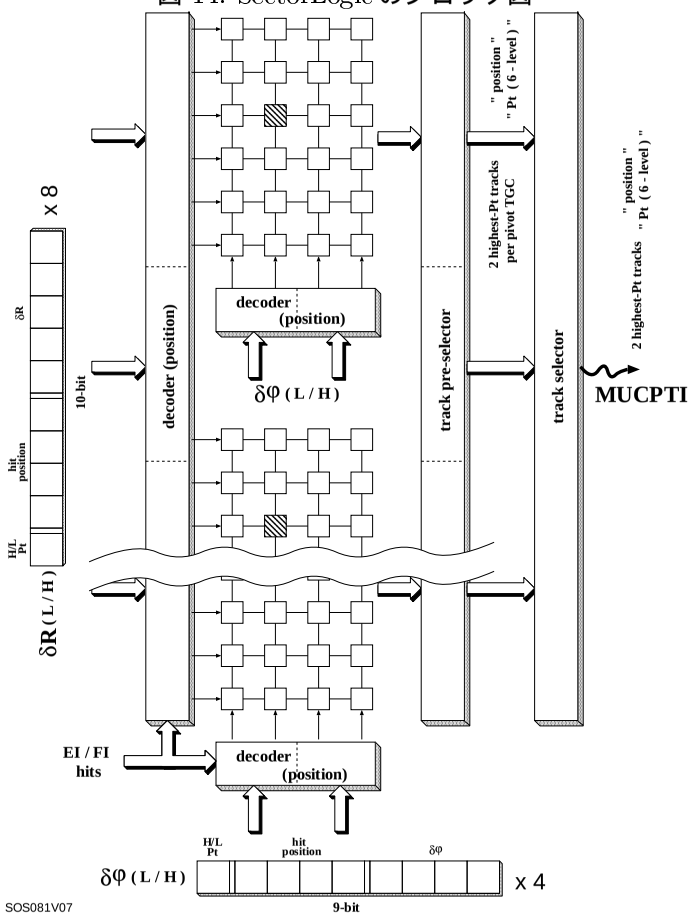


図 14: SectorLogic のブロック図



3 Slave Board High-P_T Board 用 ASIC の開発

トリガーのためのミューオンの運動量の測定と r, ϕ 方向ごとのコインシデンスの判定を行なっている Slave Board と High P_T Board の主要な機能を含んだ ASIC の試作を東京大学大規模集積回路設計教育センター (VDEC) で行なった。3.1-3.3 節で Slave Board と High-P_T Board の機能と仕様をまとめた後、3.4 節でこの試作について説明する。

3.1 Slave Board と High P_T Board の機能

トリガーシステムの中における SlaveBoard と High-P_T Board の役割は、2 章で説明したように運動量と wire group,strip それぞれの信号についてのコインシデンスの判定である。

Slave Board の機能は次のようにまとめられる。

- 2 層の doublet,triplet のそれぞれについてワイヤーグループ、ストリップの信号のコインシデンスを判定
- doublet 間のヒット位置の差の測定
- ノイズの多いチャンネルのマスク
- ASD から trigger logic への入力までのケーブルや配線などによる遅延の補償
- パッチパネルからの入力信号の読み出し
- ASIC またはボード上の回路の診断

Slave Board にはワイヤーグループとストリップの信号用の二種類の Triplet Slave Board(TSB) と二種類の Doublet Slave Board(DSB) の 4 種類 (WTSB,STSB,WDSB,SDSB) がある。

High-P_T Board の機能は次のようにまとめられる。

- pivot doublet と triplet の間のワイヤーグループ、ストリップの信号のコインシデンスを判定
- doublet-triplet 間のヒット位置の差の測定
- Sector Logic へ送るトラックの選択
- Slave Board の出力 trigger logic への入力までのケーブルや配線などによる遅延の補償
- ASIC またはボード上の回路の診断

High-P_T Board にはワイヤーグループとストリップからの信号用の 2 種類 (WHPB,SHPB) がある。

Slave Board と High-P_T Board では TTL と LVDS (Low Voltage Differential Signal) の変換を除いてほとんどの処理は ASIC で行なう。本番は 4 種類の Slave Board 用の回路が一つのゲートアレイに、2 種類の High-P_T Board 用の回路一つのゲートアレイに収められる予定である。

Slave Board 及び High-P_T Board 用 ASIC

SlaveBoard ASIC と High-P_T Board ASIC についての要請は以下の通りである。

動作周波数　LHC clock の 40.08MHz で安定に動作する。

処理時間　Level-1 trigger は 2.5μs から不測の事態のために取っておく 0.5μs を引いた 2.0μs の latency で設計されている。この内訳は MUCTPI までで 1.15μs(46bunch crossing),MUCTPI と CTP での処理のために 0.3μs(12bunch crossing), L1A(Level-1 Accept, イベントが Level-1 トリガーの条件を満たした時に CTP によって生成される信号) をフロントエンドのエレクトロニクスに戻すために 0.5μs(20bunch crossing) となっている。MUCTPI までの latency は表 1 のように分配されており、Slave Board については Phase Adjust を 1 clock、Coincidence Matrix から出力までの処理を 3 clock 以内で行なう必要がある。また High-P_T Board については、Phase Adjust を 1 clock, Coincidence Matrix から出力までを 4clock 以内で行なう必要がある。

放射線耐性　Slave Board と High-P_T Board が設置される TGC の付近での CMOS の IC に対する１０年間の放射線の許容値は、中性子は 1cm^2 あたり 3.9$\times$10^{12} 個,1MeV の中性子は 1cm^2 あたり 8.1$\times$10^{11} 個である。また、１０年間の吸収線量の許容値は 2.5$\times$10^{1}Gy である。TGC のエレクトロニクスでは特に放射線耐性を高くした IC を使うことは考えていない。これは選択肢が狭くなることやコストが高くなるためであり、代わりに十分動作を確認する。

磁場の影響　ホイール上ではトロイドマグネットから 100$\sim$1000Gauss の程度の洩れ磁場があり、ASIC はこの磁場のもとでも正しく動作する必要がある。

表 1: Level-1 trigger での Latency

Process/transmission	Time required (# of bunch-crossings)	Accumulated time
TOF to TGC	3	3
TGC response	1	4
ASD	1	5
Cable to Patch-Panel	1	6
Bunch-crossing ID and OR	2	8
Cable to Slave Board	1	9
Phase adjust	**1**	10
3/4 or 2/3 coincidence	**3**	13
Cable to high-P_T coincidence	3	16
Delay adjustment	**1**	17
High-PT coincidence matrix	**4**	21
Cable to USA15	16	37
Sector Logic processing	8	45
Cable to MUCTPI	1	46
Total delay sum	46 BC	1.15 μs

次にこれらの ASIC の仕様について説明する。

3.2 Slave Board 用 ASIC の仕様

Slave Board のトリガーロジック部の機能を図 15 に示す。

ASIC の Slave Board 用回路は TTL 信号のビットパターンを入力とする (表 2)。DSB では 4 種類の入力は pivot doublet からの信号と middle doublet からの信号の 2 つに分けられる。この表にあるように WTSB

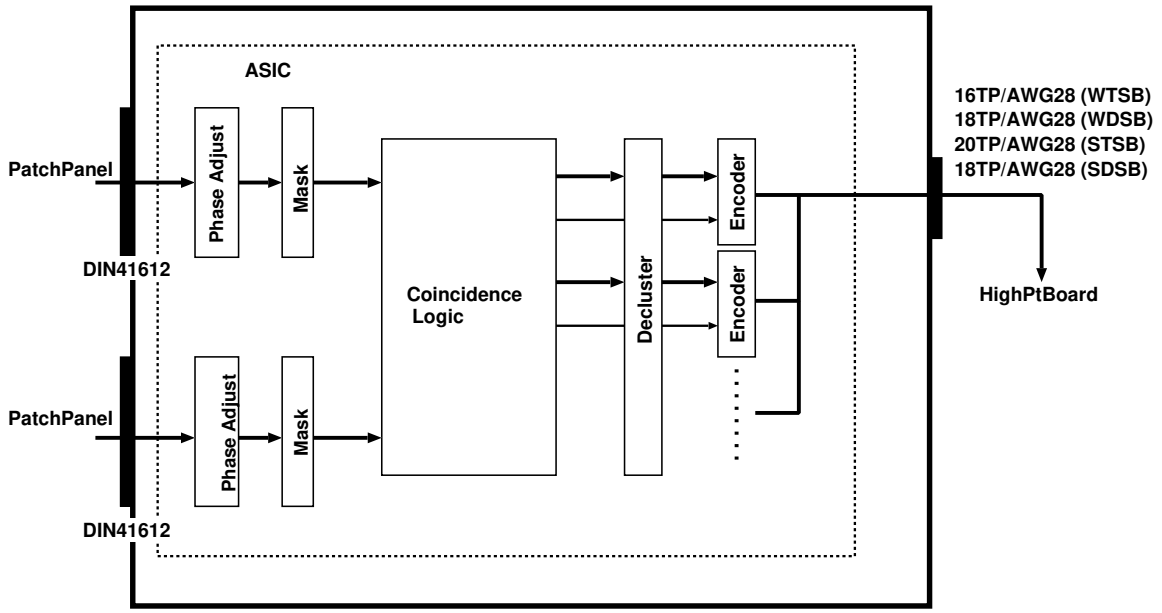


図 15: SlaveBoard のブロック図

からの信号は同じ長さの 3 つのパターンで、トリガーロジック部ではこれらの層の間でコインシデンスがとられる。同様に Strip Triplet Slave Board の 2 層の間、Doublet Slave Board の各 doublet の 2 層の間でコインシデンスがとられる。

コインシデンスが取れた信号のなかから、いくつかの領域ごとに 1 つのトラックを選択し、チャンネル番号をエンコードして出力する。チャンネル番号の他に TSB では各領域での信号の有無を表すビットを追加する。また、DSB では pivot doublet と middle doublet でのチャンネル番号の差 $(\delta R,\delta\phi)$ の値を含めてエンコードする。

表 2: Slave Board の種類ごとの入出力ビット数

種類	WTSB	WDSB	STSB	SDSB
入力	36bit×3	36bit×2(pivot) 44bit×2(middle)	32bit×4	32×2(pivot) 32bit×2(middle)
出力	1(hit) 5(channel) ×3	3(δR) 5(channel) ×2	1(hit) 4(channel) ×8	4($\delta\phi$) 5(channel) ×2

クロック信号は入力側のコネクタに接続された Patch Panel Board から与えられる。電源は Patch Panel から供給されるディジタル +3.3V DC 電圧と GND を使用する。

まず、全ての Slave Board に共通な部分は以下の通りである。

coincidence logic 入力された信号のうちコインシデンス条件を満たすものを選択する。バックグラウウンドの頻度が高い場合のために通常のコインシデンス条件の他に、より厳しい条件を要求するように設定できる。例えば通常の 2 out-of 3 のコインシデンス条件に対して 3 out-of 3 のコインシデンス条件が用意されている。このコインシデンスの条件の選択は JTAG[1]を用いて設定する。このコインシデンスロジックは DSB については二方向から入力するパターンのビットの組合せについてコインシ

[1] IC 内部の動作テストやプリント基板実装後のボードテストのためのテスト手法の規格で、規格の名称のかわりに規格化の作業グループの名称の JTAG(Joint Test Action Group) で呼ばれる。

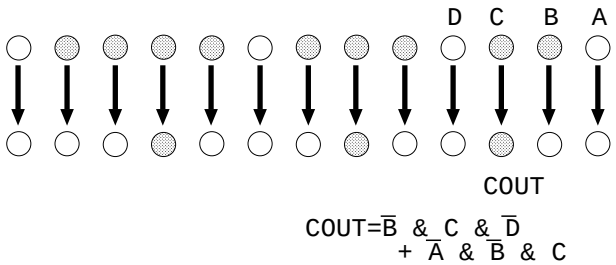


図 16: Declustring logic

デンスをとるマトリックス状になっているため、特にコインシデンスマトリックスと呼ばれる。ボードの種類ごとのコインシデンスロジックの違いについては後で説明する。

declustering 複数の連続したチャンネルでコインシデンスロジックを通った信号がある場合、その中から 1 つの信号だけを選択する。具体的には図 16 の様に信号の番号の小さい方から 2 番目の channel だけを選択する。

encoding declustering された信号のチャンネルの番号をエンコードする。この他にボードの種類ごとに異なる種類データがあるが、それは後で説明する。

phase adjust ASD のチャンネル間の信号の遅延の差は Patch Panel 上のサブ・ナノ単位のディレイで、可能な限り少なくなるように調整される。Patch Panel では ASD から PatchPanel までのケーブルごとの遅延の差は調整しないため、バンチ ID された信号の中には異なるクロックに同期しているものもある。Slave Board ではケーブルの遅延が異なる信号のコインシデンスを取るため、このクロックの差を補償するために 0-50ns の可変ディレイが用意されている。この目的のためにはクロック単位のディレイで十分であるが、シグナルがクロックの立ち上がりエッジと重なってしまう場合のために 12.5ns のステップになっている。この回路は図 17 の通りである。このディレイの値は JTAG によって設定される。

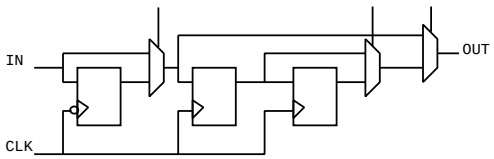


図 17: Phase Adjust 回路

masking 入力されたビットパターンを入力の値に関わらず off にする。マスクのパターンは 1 チャンネル毎に設定できる。このパターンはは JTAG によってロードされる。

JTAG ディレイ, マスクなどのパラメータの設定を行なう。また ASIC のテスト用にバウンダリスキャンの機能を持つ。バウンダリスキャンとはＩＣバウンダリ (入出力ピンと内部の回路の境界) 上の内部の回路を囲むレジスタのリングを使用して入出力信号を観測することである。

次に Slave Board の種類ごとの説明をする。

Wire Triplet SlaveBoard

coincidence logic WTSB 用の coincidence logic を図 18 に示す。マスク回路を通過した 3 つのビットのパターンから 2 out-of 3 コインシデンスの論理に従って 100 ビットのパターンを生成する。オプションと

して 3 out-of 3 のコインシデンス条件を用いることができる。

encoding　このパターンを declustering した後、32 チャンネルのブロックの中から最も番号の小さいチャンネルを選び、そのブロックの中での番号を 5 ビットにエンコードする。これに各ブロックでの信号の有無を表す 1 ビットを追加し、3 ブロック分を合わせて 18 ビットを WTSB ASIC の coincidence logic 部の出力とする。

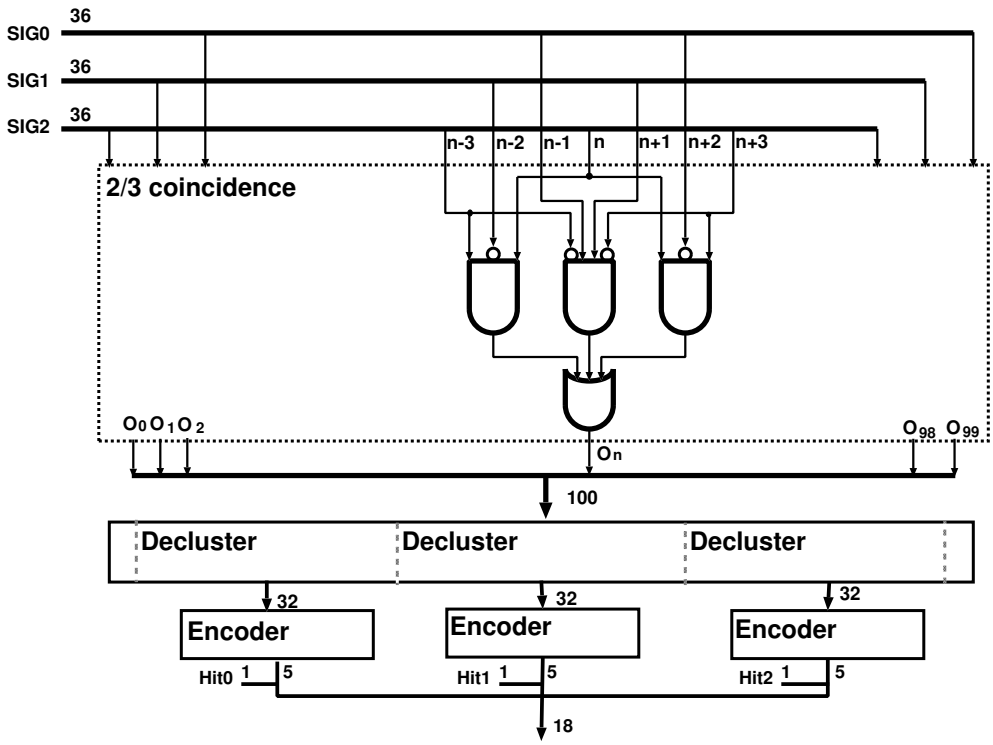


図 18: WTSB の概念図

Strip Triplet SlaveBoard

coincidence logic　STSB 用の概念図を図 19 に示す。Mask 回路を通過した 4 つの 32 ビットの pattern が 2 つづつ別々の 1 out-of 2 の論理に従ってそれぞれ 64 ビットの pattern を生成する。64 ビット分が 1 つのチェンバーからの信号に対応する。オプションとして 2 out-of 2 の coincidence 条件を用いることができる。

encoding　この pattern を declustering した後、16 channel の block の中から最も番号の小さい channel を選び、その block の中での番号を 4 ビットに encode する。この encode 各 block での信号の有無を表す 1 ビットを追加し、8 block 分を合わせた 40 ビットを STSB ASIC の出力とする。

Wire Doublet Slave Board

coincidence matrix　WDSB 用の概念図を図 20 に示す。WDSB ではコインシデンスの判定と一緒に pivot doublet と middle doublet からの信号のチャンネル番号の差 δR を計算し、その差の絶対値が最小のトラッ

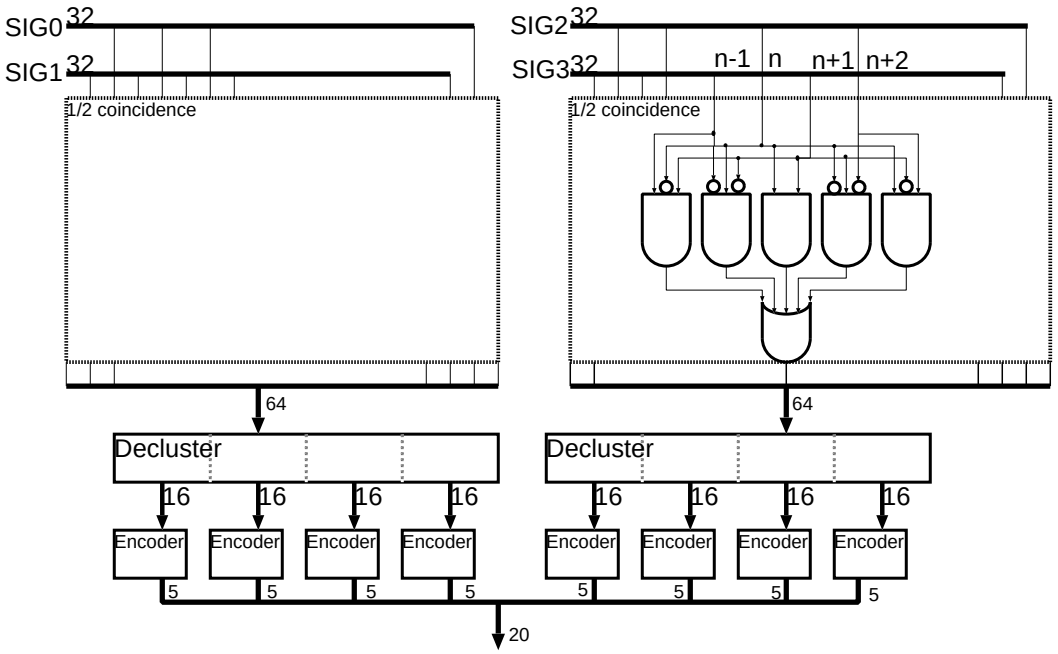


図 19: STSB の概念図

クのみを選択する。δR の範囲は-7 から 7 である。絶対値が同じで符合の異なる信号がある場合にはで正の符合のものを優先する。図 22 はコインシデンスマトリックスのローカルコインシデンスの部分を示す。

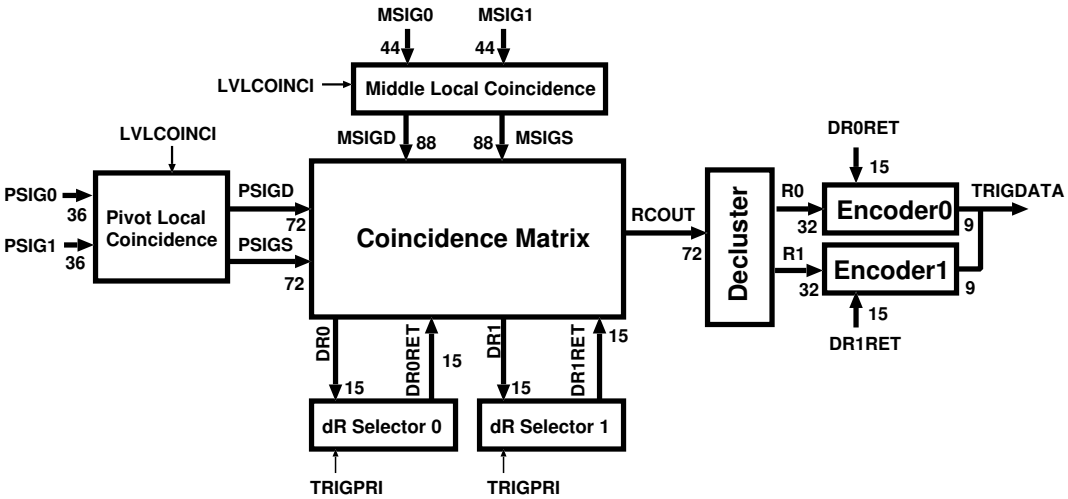


図 20: WDSB のブロック図 (全体)

図のように 1 つの層で連続した 2 ヒットまたは 1 ヒットあって、もう一方の層のでヒットがない場合に 1 ヒットがあったと判定する。また別々の層の隣合うチャンネルでヒットがあった場合、2 ヒットがあったと判定する。2 ヒットがある場合は PSIGD,MSIGD から信号が出力され、1 ヒットがある場合は PSIGS,MSIGS から信号が出力される。veto がイネーブルの場合 PSIGS,MSIGS の出力が off になり 1 ヒットの場合が除かれるので、コインシデンス条件が 3 out-of 4 から 4 out-of 4 に変わる。

図 23 はコインシデンスマトリックスの一つ一つの要素で上のローカルコインシデンスからの出力が入る。

出力 Y は δ が等しい斜めの列で 36 チャンネル分の or がとられて dR selector に送られる。dR selector で最も δ の信号が選ばれて送りかえされる。送り返された信号と Y が行ごとに or をとられてマトリックスの出力として送られる。

encoding　このパターンを declustering した後、2 つの 32 チャンネルのブロックの中から最も番号の小さいチャンネルを選び、そのブロックの中での番号を 5 ビットにエンコードする。15 ビットの δR とブロック内の信号の有無を表す 1 ビットを合わせた 16 ビットを 4 ビットにエンコードし追加する。2 ブロック分を合わせた 18 ビットを WDSB ASIC の出力とする。

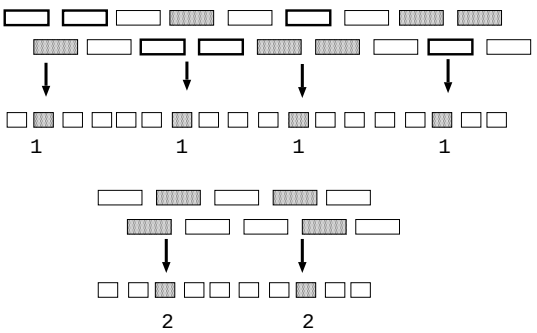


図 21: ローカルコインシデンスの説明

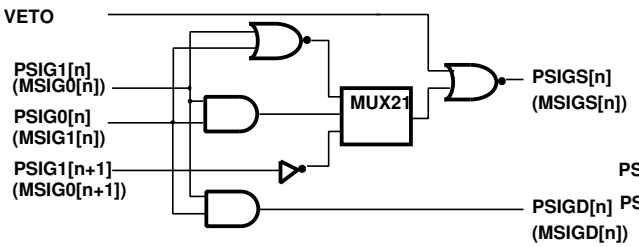


図 22: WDSB の Local Coincidence 部

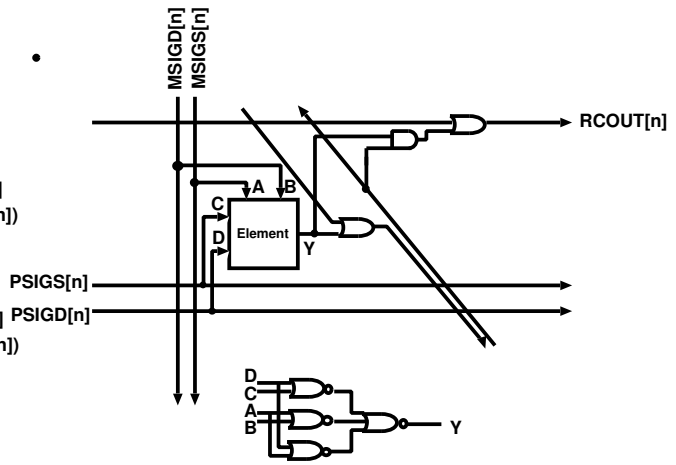


図 23: WDSB の Matrix 部 (注 : 図中の 2 つの or は実際には複数の Element module にまたがった多入力 or になっている。)

Strip Doublet Slave Board

coincidence matrix　SDSB でも WDSB と同様に pivot doublet と middle doublet の信号の差 $\delta\phi$ の最小の信号のみを選択する。$\delta\phi$ の範囲は-3 から 3 である。絶対値が同じで符合の異なる信号がある場合については、WDSB と同様に正の符合のものを優先する。

encoding　このパターンを declustering した後、32 チャンネルのブロックの中から最も番号の小さいチャンネルを選び、そのブロックの中での番号を 5 ビットにエンコードする。7 ビットの δR とブロック内の信号の有無を表す 1 ビットを合わせた 8 ビットを encode した 3 ビットを追加し、2 ブロック分を合わせた 16 ビットを SDSB ASIC の出力とする。

3.3　High-P_T Board 用 ASIC の仕様

High P_T Board の機能を図 24 に示す。ASIC の High P_T Board 用回路も TTL 信号のビットパターンをを入力とする (表 3)。HPB では入力信号は LVDS であるが、LVDS-TTL 変換の回路はこの ASIC とは別に用意する。HPB では入力は triplet からの信号と doublet からの信号の 2 種類がある。trigger logic 部ではこれらの入力信号をデコードしたのち DSB と同様の Coincidence Matirx で選別を行なう。

次に個々の機能について説明していく。

Phase Adjust　High-P_T Board でも Slave Board ど同様に SlaveBoard から High-P_T Board までの遅延の差を Phase Adjust 回路で補償する。High-P_T Board の場合、各ケーブルからの信号ごとに遅延を設定する。

Decoder　SlaveBoard からの信号を Coincidence Matrix に入力するためにデコードする。

Coincidence matrix　Doublet Slave Board と同じようにデコードされた doublet からの信号と triplet から信号のコインシデンスをとる。Strip High-P_T Board では選択する $\delta\phi$ の範囲は-7 から+7 までである。Wire High-P_T Board では選択する δR の範囲は-20 から+20 までであるが、+11 から+20,-11 から-20 までの信号は 2 つの δR,$\delta\phi$ ごとにまとめられるので、範囲は-15 から+15 になる。このように Doublet Slave Board のコインシデンスの場合にくらべて広くとっているのは、これは doublet 間の間隔が 0.5m であるのに対して、triplet と dublet の間隔 1.5m と広いためである。

Encoder　この Encoder ではコインシデンス条件を満たしたトラックの中から、32 チャンネルごとに区切られた領域ごとに 1 つのトラックが選ばれる。コインシデンス条件を満たすトラックが複数ある場合はチャンネル数が小さい方を優先して選択する。High-P_T でのトラックがあった場合、$\delta R, \delta\phi$ をエンコードしたものに pivot doublet でのヒットのチャンネルが選ばれた領域をさらに半分に分けた領域のどちらからのトラックかを表すビットをつけて出力する。この領域は 1 チップあたり 6 つづつあるので、最大 6 トラックが残る。

Hit Selector　さらに各領域で選ばれた計 6 トラックの中から doublet と triplet の $\delta R, \delta\phi$ が最も小さいもの 2 つを選択する。

クロック信号は High-P_T Board と同じクレートのボード上の TTCrx(TTC receiver) チップから各 High-P_T Board に分配される。

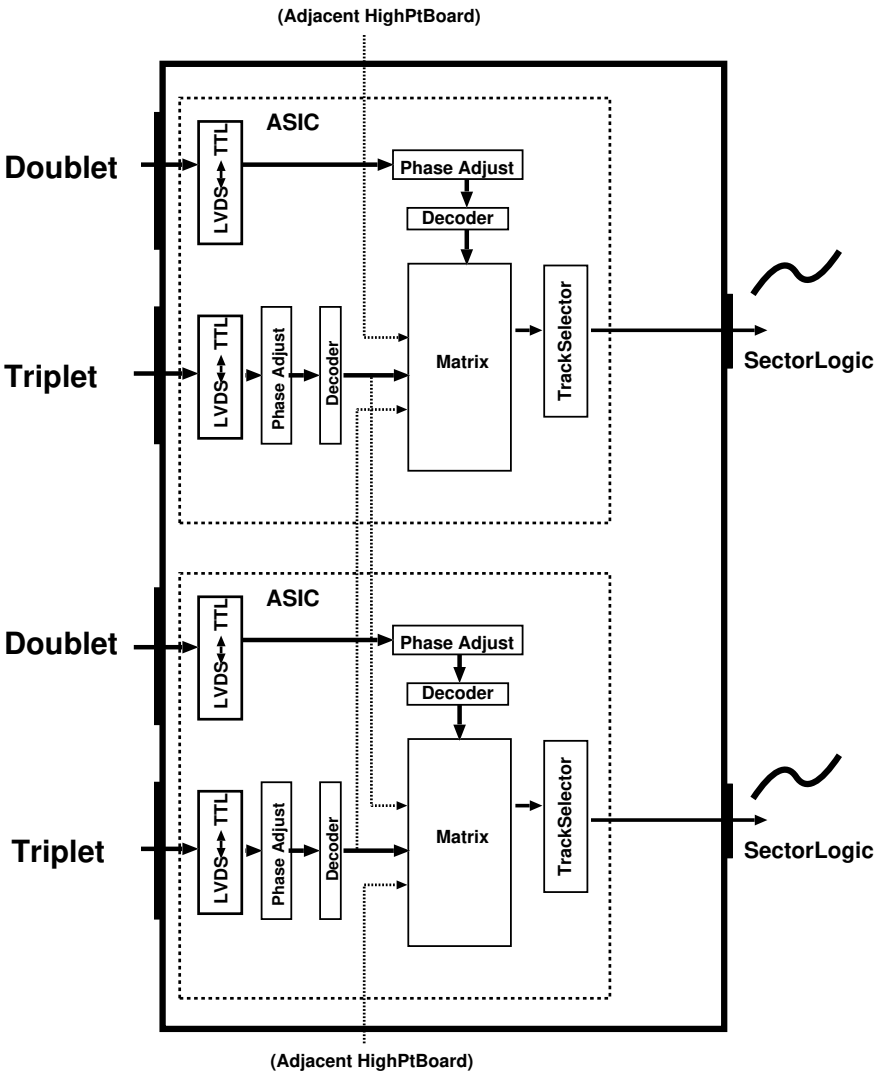


図 24: High-P_T Board の概念図

表 3: High-P_T Board の種類ごとの入出力のビット数

種類	Endcap WHPB	Forward WHPB	Endcap SHPB	Forward SHPB
入力	18bit×6 + 18bit× 6	18bit×3+18bit×4	18bit×6+18bit×4	18bit×3+18bit×3
出力	14bit×6	14bit×4	12bit×5	12×3

3.4 試作の目的

東京大学大規模集積回路設計教育センター (VDEC) で Slave Board の機能の一部を含んだ ASIC の試作を行なった[2]。この試作の主な目的は現在のデザインが動作周波数と Latency についての要求を満たしているかを確認することである。実際には 4 種類の Slave Board と 2 種類の High Pt Board を一つの gate array ASIC に収める予定だが、VDEC では試作できる chip の規模が Rohm に限れば最大でも 0.6μm のプロセスで 87 ピン,4 万 2000 ゲート (現在では 159 ピン、20 万 5000 ゲート) に限られている。デザインの検討のための試作なので、これらの回路のうち特に大きな配線遅延が予想される Doublet Slave Board や High Pt Board の coincidence matrix を入れた。しかしこの場合でもピン数の制限から全ての回路を入れることはできない。使用可能なピン数に合わせて入力を減らした回路を実装してもよいが、使用可能なゲート数からの制限はないので、直接コインシデンスマトリックス部分などの遅延を見積もれるように実際の入力数の回路を実装した。

3.5 VDEC に提出したチップのデザイン

このチップに含まれている機能は最も大きな遅延が予想される Doublet Wire Slave Board 用の coincidence matrix, declustering logic, encoder の部分である。この他に coincidence matrix の入力パターンを生成する回路が含まれている。全体のブロック図を図 25 に示す。

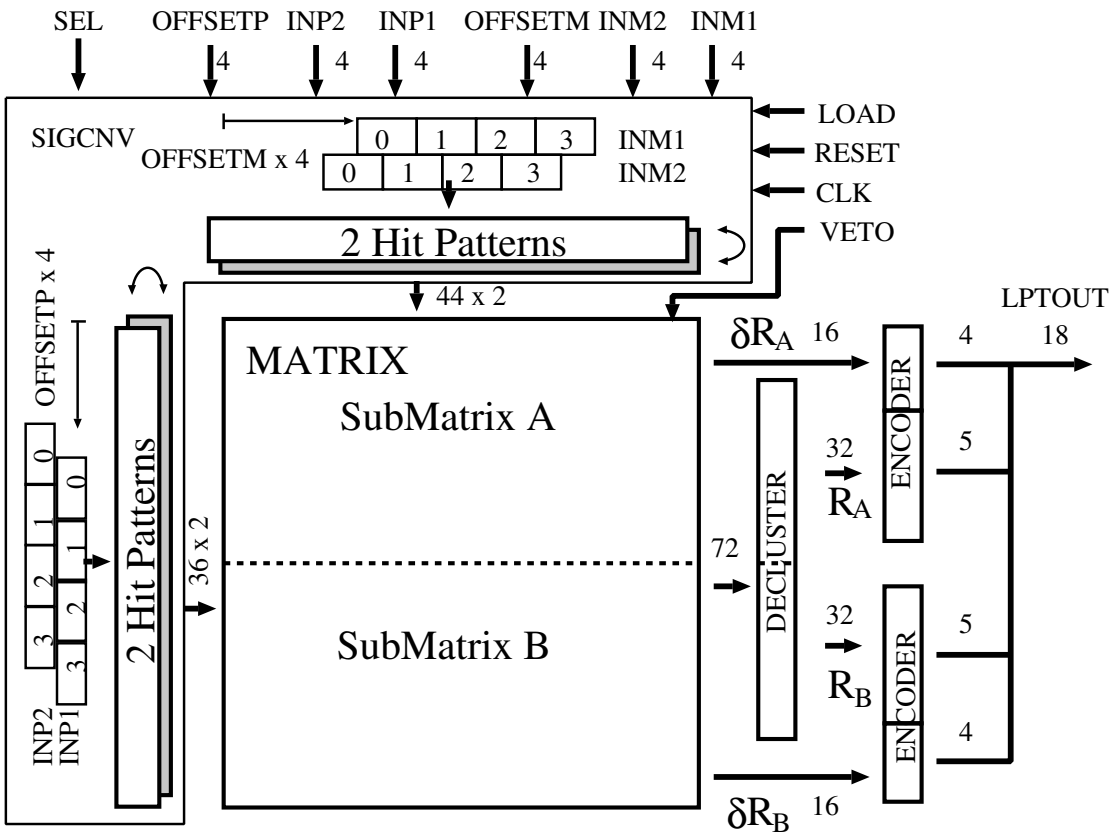


図 25: VDEC に提出したチップのブロック図

[2]本チップ試作は東京大学大規模集積システム設計教育研究センターを通し ローム (株) および凸版印刷 (株) の協力で行われたものである。

本来ならば 40MHz でヒットパターンを入力して動作テストを行なうべきであるが、このデザインではそれができないので 2 つのパターンを 40MHz で交互に入力することで動作周波数についてテストするようになっている。

チップの入出力は以下の通りである。

INP1,INP2 pivot doublet からの 4 ビット分の hit pattern が入力される。INP1,INP2 はそれぞれ doublet の 1 層目と 2 層目の hit pattern である。

OFFSETP INP1,INP2 の hit pattern を matrix に入力する時の offset の値。

INM1,INM2 middle doublet からの 4 ビット分の hit pattern が入力される。INM1,INM2 はそれぞれ doublet の 1 層目と 2 層目の hit pattern である。

OFFSETM INM1,INM2 の hit pattern を matrix に入力する時の offset の値。

LOAD 1 の場合 clock の立ち上がりのタイミングで 4 つの hit pattern が register に入力される。入力された pattern と既に入力されている pattern は or を取られて register に入力される。

SEL hit pattern を入力する register を指定する。1 の場合 register1 に、0 の場合 register2 に入力される。

RESET Register で保持されている 2 つの hit pattern を reset する。

CLK SIGCNV block で使われるクロックを入力する。MATRIX 以降は完全に組合せ論理なのでこのクロックは使われない。クロックごとに 2 つのパターンが交互に入力される。

VETO これを enable にすることで coincidence 条件が 4 outof 4 になる。

LPTOUT エンコードされた r、δr が出力される。

図 25 の中の回路の回路図は Appendix A に示す。

3.6 VDEC における試作の流れ

HDL による回路の記述 Verilog HDL を用いて回路を記述した。HDL(Hardware Description Language) とは論理回路を記述するための言語で、Verilog HDL とは HDL の一種で VHDL とならんで最も良く使われている HDL である。この時点では論理の確認のために Cadence 社の Verilog-XL 用いてシミュレーションを行なった。このシミュレーションには配線遅延は含まれていない。

論理合成 この回路は入力信号を生成する部分以外は組合せ論理なので、基本的にデータフローレベル[3]で記述し Synopsys 社の Design Analyzer を用いて論理合成を行なった。Design Analyzer では VDEC から提供されているライブラリを併用することでゲートの遅延と配線遅延を含めたシミュレーションが可能である。ここでは遅延を考慮した場合にも正しく動作しすることを確認し、配線遅延が最小になり、また遅延の最大値と最小値の差が大きくならないように最適化した。

レイアウト Avant 社の Apollo を用いてデバイス上にゲートを配置し配線した (図 26)。

レイアウト検証・デザインルールチェック 配置配線した結果 (レイアウト) がネットリスト[4]を正しく反映しているかを確かめるのがレイアウト検証 (LVS,Layout Vs Schematic) である。また、レイアウトがゲートの寸法や配線の線幅などの設計ルールを破っていないかを確認する作業が DRC(Design Rule

[3]ゲートを直接指定せずに信号の関係を記述する表現方法。各モジュールで入力が変化するたびに出力が再計算される。
[4]回路部品とその接続関係をテキストで表現したもの。論理合成による出力がネットリストである。

Check) である。Cadence 社の Dracula を用いて LVS,DRC を行ない、正しく行なわれていることを確認した。

提出 以上を行なった後 VDEC の計算機に設計データを送る。

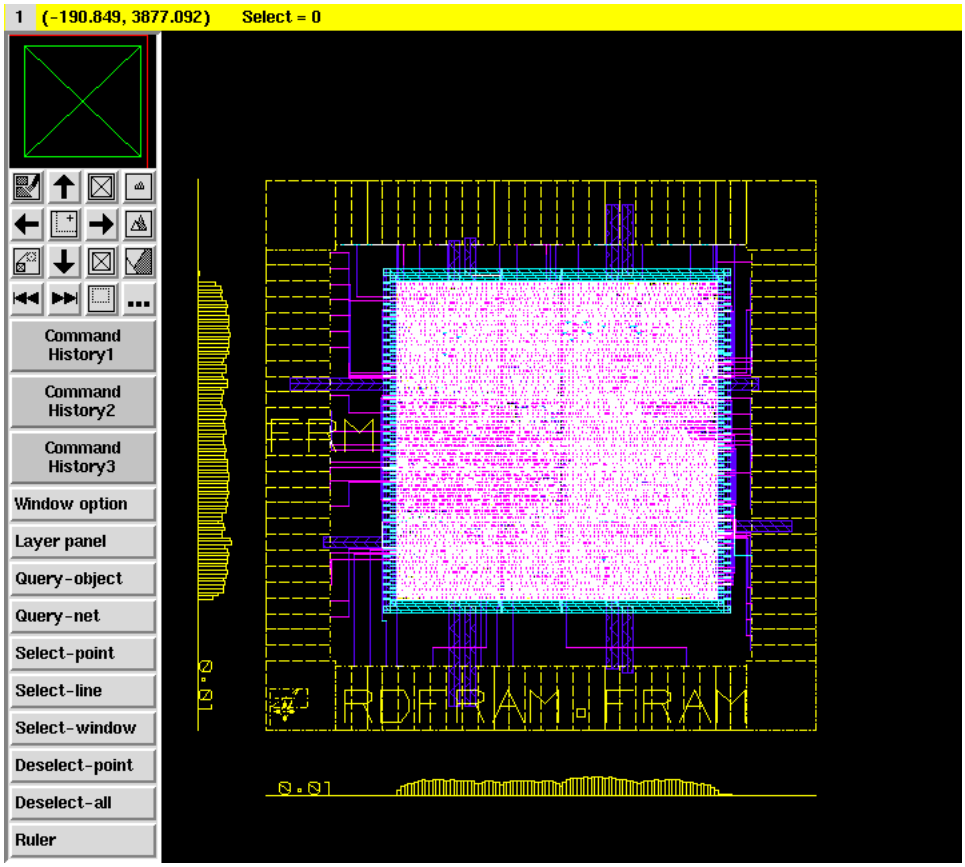


図 26: 配置配線後の Apollo のスクリーンショット

3.7 シミュレーションによるデザインの検討

論理合成の段階でのシミュレーションを説明する。図 27 のマトリックスの対角線上で δR が-7 から 7 までの範囲に、図 27 中に示した 4 種類の信号のパターンからなる pivot と middle の信号のパターンを入力した。4 種類のパターンの上段が 1 層目のパターンで、下段が 2 層目のパターンである。

このようなパターンを 1 万回入力した時の出力の分布が図 28 である。この図の上の段は 1 回当たり 1 パターンを入力した場合、下の段は 1 回当たり 5 パターンを入力した場合である。1 つのパターンを入れた場合の分布から、次のことが確認できる。

- δR の大小で選別することはないので、δR が-7 から 7 の範囲で δR の大きさに関係なく一様に分布する。
- トラックを選択する時に分けた 2 つの領域の境界 (R=32 の部分) など inefficient になっている部分がない。
- 3 out-of 4 のコインシデンス条件を満たすパターンの組合せは 4 分の 1 になるが、4 out-of 4 の場合の選択されたトラックの数も 4 分の 1 になっている。

5 つのパターンを入れた場合から、δR の絶対値が小さい信号を優先して選択していることが分かる。

しかし 3 out-of 4, 4 out-of 4 の両方の場合で 32 チャンネル付近で他のチャンネルよりもトラック数が多いチャンネルがある。この原因を説明する。図 29 に示すようにマトリックスの中央付近では subsubmatrix 半個分だけ余分に対角線上に or をとるようにしている。この余分に or を取っている部分の subsubmatrix の出力も同じチャンネルの位置の出力と or を取っているために図 28 のような double counting が生じた。このことに気づかずに設計を提出したので、出来上がる ASIC にはこのような double counting があるはずである。

middle と pivot の両方から 1 ヒットのパターンが入力された場合や、一方から 1 ヒットのパターン、もう一方から 2 ヒットのパターンが入力された場合には、選ばれるトラックがないことは確認した。

以上より、このチップのデザインで境界での double counting の問題を除いてを正しく機能することが確認できた。

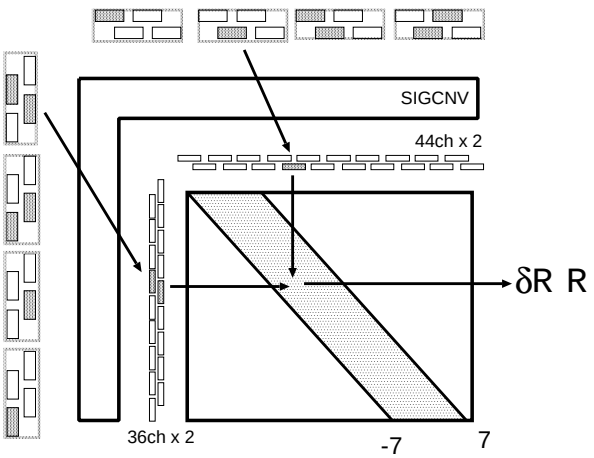


図 27: テストパターンによる確認の説明

このデザインで既に述べた要請のために最大遅延と、遅延の差の最大値には制限がある。最大遅延についての制限は表 1 にあるように、phase adjust より後の部分で 3 クロック分の 75nsec 以下というものである。全体の最大遅延が 25nsec を超えない場合には問題ないが、超える場合には途中でラッチをかけてタイ

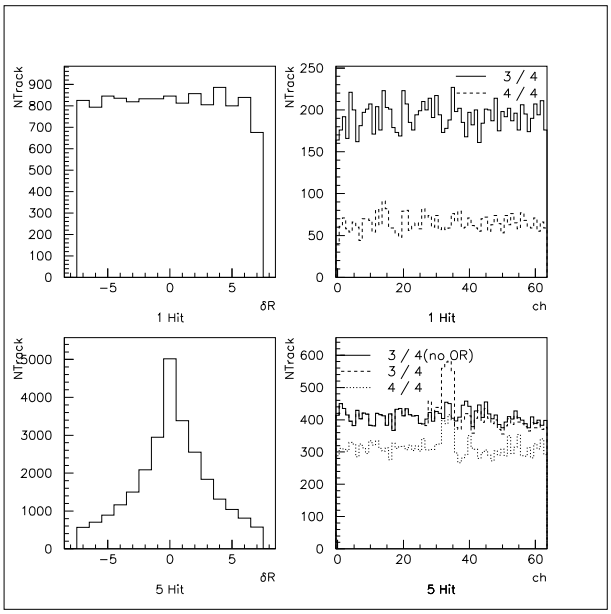


図 28: テストパターンを入力した場合の出力。3/4,4/4 はそれぞれ 3 out-of 4, 4 out-of 4 のコインシデンスを表す。

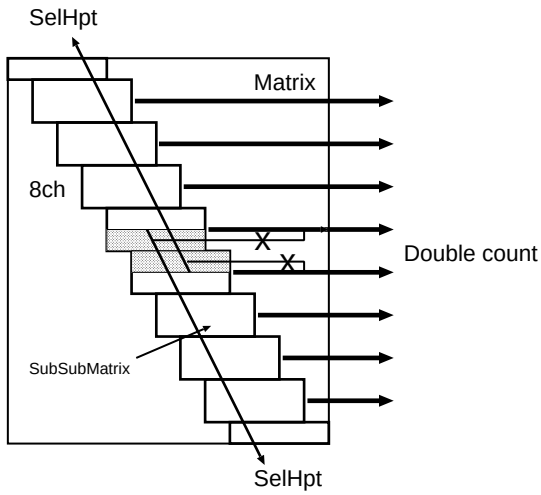


図 29: コインシデンスマトリクスの領域の境界での double counting。図はコインシデンスマトリクスを表し、その対角線上に配置された四角は subsubmatrix を表す。

ミングを合わせなければならない。このためラッチをかける前の段階で最大遅延と最小遅延の差について 25nsec を超えないという制限が加わる。

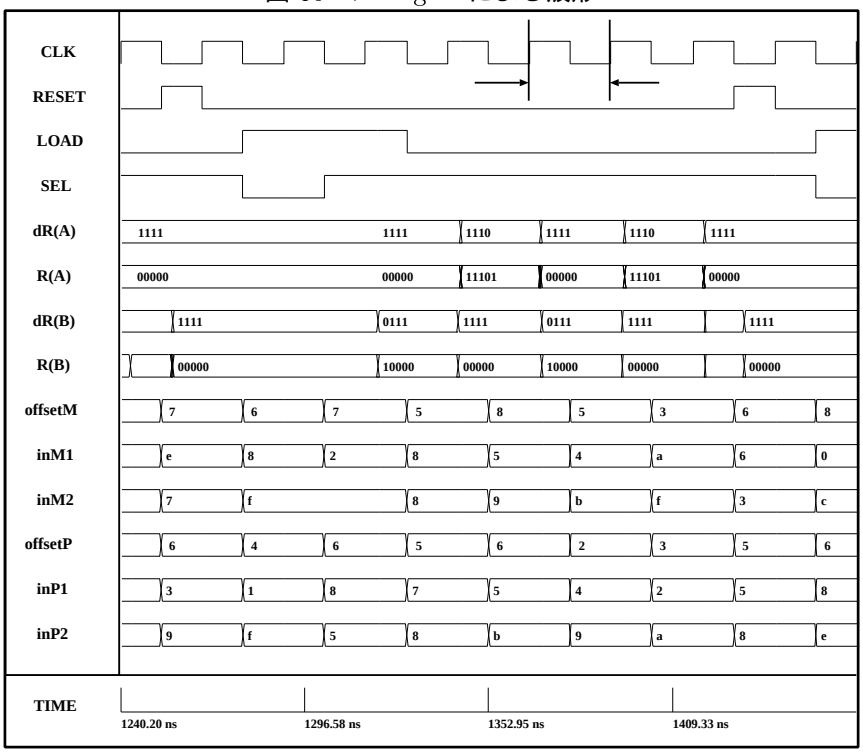
遅延の差が最も生じやすいのは matrix の部分などにある多数のチャンネルに渡って or が取られているところである。その場合でもほとんど問題になるほど遅延の差が大きくなる可能性はなく順に OR を取っていくようなことしない限り、遅延の差が大きくなることはない。多入力 OR を使うか、全てのチャンネルが同じ段数の OR を通るようにすれば問題はない。特にこのことに気をつけてセルを置いていくことも考えられるが、順に OR を取っていく様な回路は最大遅延が大きく、論理合成の際に最大遅延が小さくなるように最適化していれば落されるので特に気にする必要はない。実際に論理合成の結果 (Appendix A 図 56) を見るとどのチャンネルもほぼ同じ数の NOR を通っており、遅延の差についての問題は出ない。

図 30 は Verilog-XL の波形で dR(A),R(A),dR(B),R(B) はそれぞれエンコーダの出力 18 ビットのうち二つのヒットの δR,R にあたる部分である。既に説明したように LOAD が 1 の間に OFFSETM, OFFSETP, INM1, INM2, INP1, INP2 をつかってパターンを作る。CLK の立ち上がりのタイミングで Coincidence Matrix へのパターン切り替わっている。R=00000,dR=1111 はそのブロックにヒットがなかったことを示す。図 30 上の図の矢印で囲まれた 1clock 分を拡大したものが図 30 の下の図である。下の図から出力の dR,R をみるとクロックの立ち上がりのタイミングで信号が入力されてから 4ns 程度かかって出力が出そろっている。また、出力データの各ビットの遅延のばらつきは 1ns 程度以下であることが分かる。

最大遅延

Design Analyzer による配線遅延も含めたシミュレーションより、最大遅延は Coincidence Matrix 部では 3.65ns、Encoder 部では 1.70ns であり、最小遅延は Coincidence Matrix 部では 0.89ns、Encoder 部では 0.19ns であることが分かった (表 4)。この表で MaxRise(MaxFall) は入力信号の立ち上がりから出力信号

図 30: VerilogXL による波形



の立ち上がり (立ち下がり) までの遅延時間の最大値である。同様に MinRise(MinFall) は入力信号の立ち上がりから出力信号の立ち上がり (立ち下がり) までの遅延時間の最小値である。このチップでは入力ピンの数が少ないために 2 種類の Hit Pattern を一旦レジスタに貯める必要がある。そのための回路が SigCnv であり、ここでは関係ない。

Doublet Slave Board 用の回路では他に Phase Adjust 回路、Mask 回路が含まれるが、Coincidence Matrix と Encoder に比べて小さく問題にはならない。Phase Adjust 以降は 3clock 使うことができるが、このシミュレーション通りかまたはそれに近い最大遅延が得られれば、最大遅延に対する要求を十分に満たせることが分かった。またこの程度の最大遅延であれば、最大遅延と最小遅延の差は問題にならないので途中にラッチを置かずに,1clock で処理できる。

表 4: Desgin Analyzer で求めた部分ごとの最大遅延 (単位は ns)。

	MaxRise	MaxFall
sigcnv	0.58	0.55
matrix	3.60	3.65
encoder	1.70	1.67
total	5.90	5.91

また最大遅延経路に沿って各セルでの遅延とつながれている調べて、VDEC から提供されているセルの負荷容量による遅延を足し上げると、それぞれ 3.36ns と 1.91ns の合計 5.27ns になる。シミュレーションの結果と合わせて配線による遅延は 0.6ns 程度と見積もれる。

回路規模

Design Analyzer で見積もった今回試作したチップの Cell Area とこれから見積もった NAND ゲート換算の回路規模は表 5 の通りである。

表 5: Design Analyzer による Cell Area

名前	セル面積 [mm^2]	ゲート規模数
encoder	0.24	1.0K
matrix	4.94	21K
sigcnv	1.03	4.3K
total	6.21	26K

3.8 結論・今後の予定

エンドキャップミューオントリガー用の運動量選別用 ASIC の主な機能であるコインシデンスマトリックスを含むチップを VDEC を利用して試作した。シミュレーション上は

- 運動量の大きさによるミューオンの選別に必要な δR の小さいトラックを優先して選択する機能とコインシデンスによって選別する機能が動作している。
- コインシデンスマトリックスの領域の境界を除いた全てのチャンネルで等しく出力がある。
- 要求された動作周波数 40MHz で動作し、Latency 75nsec で動作する。

- VETO を有効にすることでコインシデンス条件が 4 out-of 4 に変更される。

ことが確認できた。今後チップが届き次第動作テストを行ない、必要な性能が出ていること確かめる予定である。

4 TGCトリガーシミュレーション用プログラムの設計と開発

高エネルギー物理実験ではこれまで Fortran が広く使われてきたが、現在 C++や Java などのオブジェクト指向言語で置き換える動きが進んでいる。それは世の中の流れがオブジェクト指向言語に向かっているためということもあるが、実際に実験の規模が大きくなるとともにソフトウェアの規模や開発者の数が大きくなり開発が複雑になっていきているためでもある。

ATLAS でもこれまでの Fortran による開発から C++ベースの開発に移っている。これまで Fortran ベースの ATLAS software を用いて検出器の性能や物理のシミュレーションが行なわれてきたが、Physics TDR[2] が完成後、実際の解析で使える software の開発の必要になってくると、オブジェクト指向化の必要性が認識されるようになった。またこれまで広く使われてきた検出器のシミュレーションプログラムである GEANT3 の新しいバージョンの Geant4 は C++で書き直されているおり、各検出器のシミュレーションも C++による開発に移行している。TGC のトリガーシミュレーションについては GEANT3 によるシミュレーションのための Fortran で書かれたプログラムがある。しかしこれは開発の初期の段階で必要な情報を得るためにトリガーシステムとしての機能を実現したものである。現在ではエレクトロニクスのデザインは固まってきており、今後システムの後段の設計のために実際の設計に則したプログラムが必要になってきている。ATLAS での流れにそってエンドキャップミューオントリガーのシミュレーションのプログラムも C++で書き直し始めるのが有効である。

この章では単にオブジェクト指向言語を用いるのではなくオブジェクト指向開発の流れにそって行なったトリガーシミュレーション用のプログラムの設計と開発について述べる。

4.0.1 旧 ATLAS software の構成

Fortran ベースの ATLAS Offline software は以下のプログラムで構成されている。

DICE GEANT のための検出器の記述

ATLSIM ATLAS での GEANT のインターフェース

ATRECON 内部検出器、カロリーメータ、ミューオンシステムでのイベントの再構成を行なう。

ATGEN イベントジェネレータへのインターフェース。

ATRIG トリガーシミュレーション

SLUG これらのプログラムの操作のためのプログラム。

4.1 新しい ATLAS Software の状況

新しい ATLAS Software の状況は ATLAS Architechture Task Force(ATF) によって ATLAS に共通な Framework が検討されている段階である。

この Architecture とはソフトウェアの高レベルのデザインを指す。また Framewark とはある目的 (例えば reconstruction) のために提供される class の集まりのことを指し、Architecture の重要な概念をコードとして実現したものである。

ATF のレポート [5] によると、ATLAS software は次の 3 種類の Library から構成される。

1. STL[5],CLHEP(CERN で開発されている高エネルギー実験のためのクラスライブラリ) などの基本的な C++のユーティリティのライブラリ。

[5] 標準テンプレートライブラリ。テンプレートは型などをパラメータ化した汎用的なクラス・関数の定義

2. ATLAS Common Framework。イベントデータや検出器を記述しているデータにアクセスするクラスなどを提供する。
3. イベントフィルタ、イベントの再構成、解析や可視化など一般的に定義されている目的のためのライブラリ。

ATF では 2 の ATLAS Common Framework を作るためにオブジェクト指向分析・設計を進めている。その成果としてユースケース、クラス図、シーケンス図などが示されている [6, 7]。

4.2 オブジェクト指向解析・設計

オブジェクト指向解析では対象とする領域をオブジェクトの観点から分析する。ここでいうオブジェクトとは現実の世界に対応するものがある概念をさす。具体的にはその問題領域の中の概念を探しだし、その概念の定義することを行なう。さらにその概念の間の関連を見つけ出し概念モデルを構築していく。オブジェクト指向設計では解析の段階で得られた概念と関連に対応するソフトウェアのオブジェクトやその関連を決定していく。この段階でそれらのオブジェクトを表すクラスを定義し、属性やメソッドを与える。オブジェクト指向プログラミングでは定義されたクラスをオブジェクト指向言語を用いて実装する。

これまで広く使われていた構造化分析 / 設計では問題領域を機能の面から分割していた。複雑な問題については機能を階層化することで対処してきた。これに対してオブジェクト指向分析 / 設計では境界のはっきりした扱いやすい実体に対応した概念で分割しており、より大きく複雑なソフトウェアの開発に有効である。

4.2.1 UML の説明

現在、概念モデルを記述するための言語として標準化されているのが UML(Unified Modeling Language) である。UML はデファクト・スタンダードを目指して Grady Booch, James Rumbaugh, Ivar Jacobson によって開発された。この UML が登場する以前では著者ごとに異なる表記法でモデリング手法を議論していが、UML が標準化されることによって表記法の説明や表記法自身についての議論が必要なくなった。

UML ではある観点から見た概念モデルを表すものをビューと呼んでいる。ビューにはユースケース・ビュー、論理ビュー、コンポーネント・ビューなどがあり、これらのビューはいくつかの決まったダイアグラムを用いて表現する。UML で用いられるダイアグラムにはユースケース図、クラス図、オブジェクト図、シーケンス図、コラボレーション図、状態図などがある。

ユースケース図 システムと外部アクタの相互作用を表現することでシステムの機能の記述を行なう。アクタとはシステムと相互作用するユーザや他のシステムを指す。

クラス図 クラスの属性や操作とクラスの間の関連を記述し、システムの静的な構造を表現する。

オブジェクト図 オブジェクト・インスタンス間の関連を記述する。実行時のシステムの構成の実例を表す。

シーケンス図 複数のオブジェクトの協調動作を表現する。特にメッセージの順序を強調したい場合に用いる。

コラボレーション図 シーケンス図と同様にオブジェクトの協調動作を表現するが、それだけでなくオブジェクト間の関連も記述することが可能である。

状態図 オブジェクトが取りうる状態とその間の遷移を表現する。これにより遷移に伴って起きるアクションを記述することができる。

トリガーシミュレーションのプログラムの説明でクラス図・シーケンス図を用いていているので、これらについて簡単に説明しておく。

クラス図　クラス図の構成要素はクラスとその間の関連で、クラスは図 31 のような 3 つに分割された四角の箱で表す。3 つの箱の中に順にクラス名と属性と操作を記入する。各属性と操作の前の+,-,#などの記号は可視性を表し、それぞれ public,private,protected であることを示す。

図 31: クラス図の要素の説明

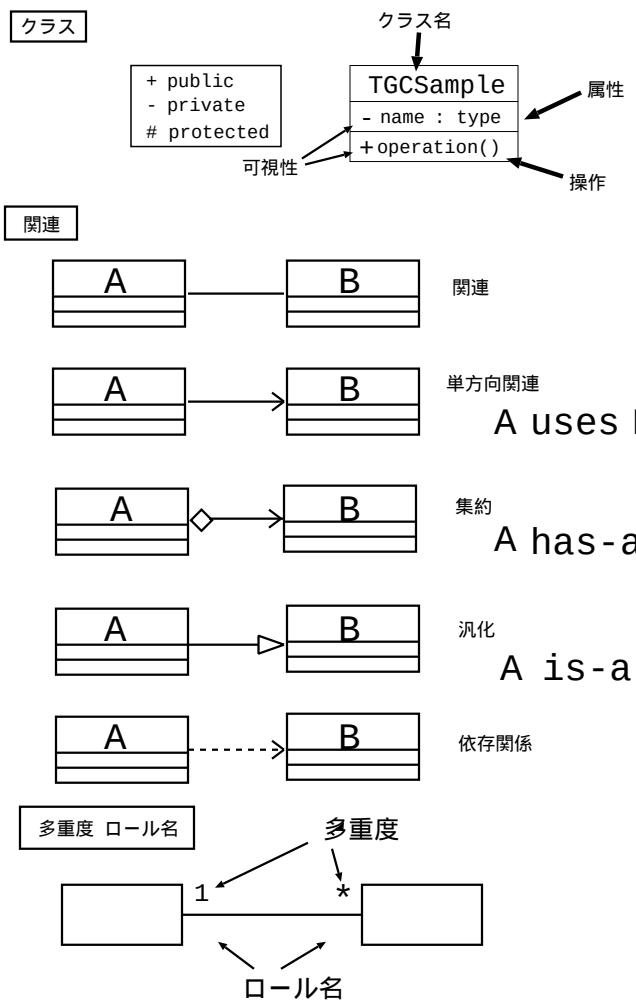
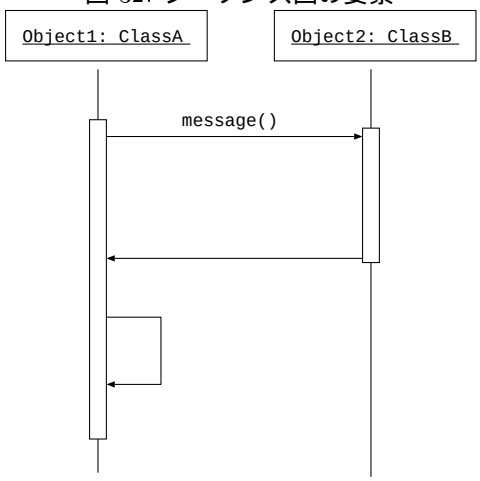


図 32: シーケンス図の要素



クラス間の関連には関連、単方向関連、汎化、依存関係、集約、コンポジッション集約がある。

関連 単に 2 つのクラスの間に何らかの関係があることを示し、クラスを実線で結ぶことで表される。

単方向関連 一方のクラスが他方を使用する関係を表す。使うクラスから出る実線の矢印を用いて表される。“uses” の関係とも呼ばれる。

集約 一方のクラスが他方のクラスを属性として持っている場合に用いられる。 後者から前者への菱形の頭の矢印で表す。“has a” の関係とも呼ばれる。

汎化　一方のクラスは他方のクラスの一種であるという関係を表す。前者のクラスから後者のクラスへのびる三角の頭の矢印で表す。“is a” の関係とも呼ばれる。

依存関係　一方のクラスが他方のクラスに何らかの形で依存してことを表す。後者の変更が前者に影響を及ぼす程度の弱い関連を表す。これは破線の矢印を用いて表される。

コンポジッション集約　一方のクラスは他方のクラスを用いて構成されている場合に用いられる。後者から前者への黒く塗りつぶされた菱形の頭の矢印で表す。“consist of” の関係とも呼ばれる。

関連を表す矢印の両端に数字が書いてあるが、これはそれぞれのクラスのインスタンスの多重度である。*は 0 以上の数を表し、... を用いて数の範囲を示す。

シーケンス図　シーケンス図では図 32 のように縦方向にのびている線がオブジェクトを表す。オブジェクトの名前とクラス名はその線の上にある箱に記入する。縦の線が細長い箱に変わっている部分はそのオブジェクトにフォーカスが当たっている、つまりそのコードが実行されているかメッセージを待っている状態であることを示している。オブジェクト間のメッセージは水平方向の矢印で示される。また、自分自身へのメッセージは曲ってもどってくる矢印で表される。

4.3　トリガーシミュレーションプログラムの設計

つぎに ATLAS のエンドキャップトリガーのシミュレーションプログラムの設計について述べる。

4.3.1　プログラムの特徴・機能

このトリガーシミュレーションのプログラムは

1. 設計の時点でエレクトロニクスのパラメータを見積つもり、設計上の落し穴などがないかの確認
2. 実験開始時にボードが正しく配置・接続されているかの確認
3. 解析の時点でのトリガーのシミュレーション

などを目的としている。1 や 2 の目的のためには当然ハードウェアとソフトウェアの対応が分かりやすい方が設計上の変更を反映させやすいため便利である。このプログラムでは後で述べるような実際のボード種類ごとに対応するクラスを作り、その個々のボードに対応してオブジェクトを作った。また各ボードはケーブルによって対応づけされているように、ボードの接続についての情報はそれぞれのオブジェクトが持つようにした。このプログラムには次のような特徴がある。

コネクションテーブル　このプログラム中ではボードとボードの接続やパッチパネル内部でのコネクションなどは初期化の時点で外部ファイルから読み込むようになっている。これらの外部ファイルは検出器とエレクトロニクス共通のデータベースが整った段階でハードウェアのボードと共通にする予定である。現実とソフトウェア上のボードとチャンネルが同じ Naming・Numbering で 1 対 1 に対応しているため、変換が必要ないという利点がある。

ASIC の設計のためのテストベクタの生成　個々のボードの入力・出力信号に対応した TGCPatchPanelOut, TGCSlaveBoardOut があるのでこれをテストするボードに対応するものだけ出力することで、secor logic などのテスト用のテストベクタを容易に作ることができる。

マルチバンチの扱い　検出器の調整のために、一度に複数のバンチのデータを取ることがある。このような場合のためにこのシミュレーションではPatchPanelにTGCBIDOutを貯める可変のバッファをオプションでつけることで対処する。

4.3.2　解析

概念の抽出、関連・属性の発見　Level-1トリガーのMuon triggerのTDRをもとにエレクトロニクス・システムの中でトリガーシミュレーションに関係する概念を抽出した。ICに含まれる一部の回路を概念としてあるが、これはこれらの回路が他に比べてが大きくなる可能性があるためである。

分類	概念
物の集合	エレクトロニクスシステム、トリガーセクター、 octant, event
信号・信号のパターン	ASD,PatchPanel, SlaveBoard, HighPtBoard, SectorLogic, BCID回路の出力信号, マスクパターン、R-ϕ coincidenceのマップ
回路	sector logicのPreSelector, Selector, SubSectorController, R-ϕ coincidence Matrix, PatchPanelのBCID回路
基板・モジュール	Wiregroup/Strip,triplet/doublet用のPatchPanelとSlaveBoard, Wiregroup/Strip用のHigh-P_T Board, SectorLogic
コネクタ・ピンの接続	ASDからPatchPanelへの接続、PatchPanelの入力pinと出力pinの接続, PatchPanelからSlaveBoard、SlaveBoardからHigh-P_T Board、High-P_T BoardからSectorLogicの接続

これらの概念についてクラスを作り、関連と属性を加えたモデルを作った。クラス図34-40に、シーケンス図を41-43に示す。

TGCEvent,TGCASDOut(図33) ASDの出力のクラスTGCASDOutを作った。TGCEventはTGCASDOutの集合である。属性signalTypeはwiregroupとstripのどちらの信号かを示す。hitIDはモジュールのηの小さい方から数えたチャンネルの続き番号をしめす。channelは各チェンバーごとのチャンネル数をしめす。hitToFはそれぞれのヒットのTime of Flightをしめす。Numberingについて情報はTGCReadOutIndexが持っている。

TGCReadOutIndex(図33) これらの検出器を識別する変数についてはAppendix Bを参照。

TGCElectornicsSystem(図34) この4種類のPatchPanelのクラスの汎化としてTGCPatchPanelを作った。隣に位置するPatchPanelとの一部の信号の交換があるので、そのための関連を付け加えた。TGCBIDOutはBunchIDの属性が付け加わったTGCASDOutのサブクラスとした。PatchPanelの役割の一つの信号の配分のための入力チャンネルと出力チャンネルの対応はTGCConnectionInPPが持っている。

TGCSlaveBoard(図36) 2種類のDoublet SlaveBoardのクラスの汎化としてTGCDoubletSBクラスを作った。これはWireDoubletSlaveBoardとStripDoubletSlaveBoardの処理がほとんど共通であったためである。また、TGCDoubletSBクラスと2種類のTGCTripletSBクラスの汎化としてTGCSlaveBoardクラスを作った。

TGCHighPtBoard(図38) TGCWireHighPtBoardとTGCStripHighPtBoardの汎化としてTGCHighPtBoardクラスを作った。エンドキャップのWire HighP_TBoardでは隣合うボード間で信号のやり取りがあるために隣のHighPtBoardを示す属性を持っている。

TGCSectorLogic(図39)) TGCSSCControllerがHighP_TBoard,SlaveBoardとの信号を受けとる。

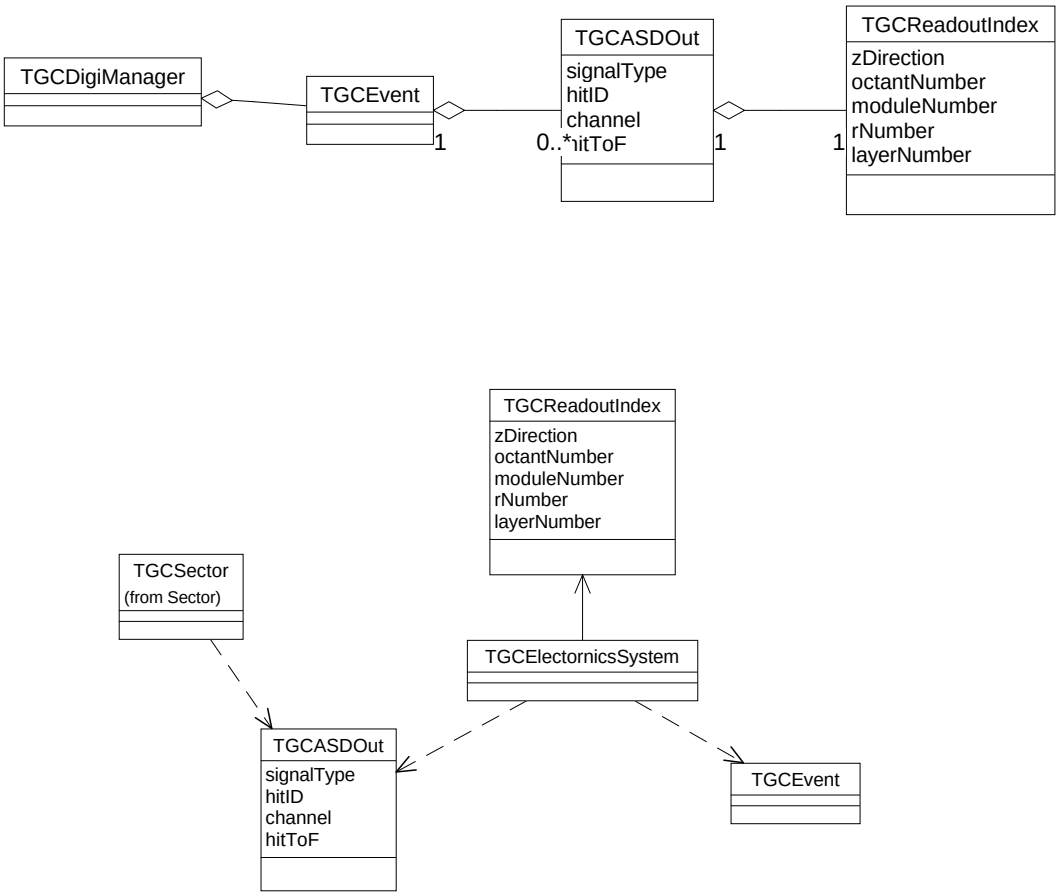


図 34: ElectronicsSystem のクラス図

4.3.3 設計

このプログラムでは Timing Manager を作り、この Timing Manager が PatchPanel から順に処理の開始を命令していく。SectorLogic から順に必要なデータを前のモジュールに取りにいけばこのような Manager を作る必要はなく簡潔になるが

- Patch Panel や HighP_T Board では隣合うボードの入力信号を使うボードがあり、シミュレーション上そのようなボードでは隣のボードが入力信号を全て持っていなければ処理を始めることができない。
- このトリガーシミュレーションの場合、一つのトリガーセクターに限ったとしても信号が入る PatchPanel は多くないと考えられる。後ろからとりにいくデータの入らないボードも呼び出してしまうので無駄が多い。
- 一旦 PatchPanel まで遡った後に、処理を行ないながらまた SectorLogic に戻って来なければならない。

などの欠点があるので、そのような方法はとらなかった。

クラスの作成　ここでは概念モデルのクラスの他にエレクトロニクスシステムに対応するオブジェクトがないが必要なクラスを追加する。

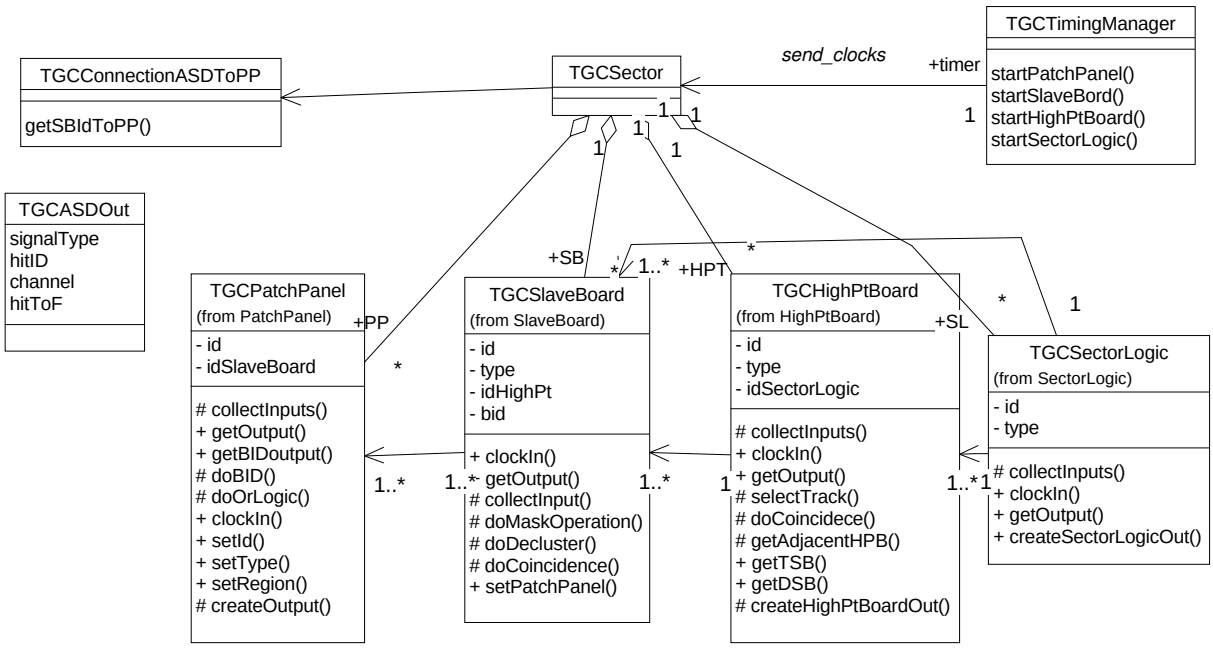


図 35: Sector のクラス図

TGCTimingManager 実際のエレクトロニクスシステムには存在しないが、各モジュールにクロックに相当する、スタート信号を分配する。また、Patch Panel や High-P_T Board では各ボードが信号を集め終った後に信号の処理を開始するための信号を分配する。

TGCDatabaseManger ボード間のコネクションやマスク、r-ϕ コインシデンスマップを管理するクラス。これらのデータは TGCDatabaseManager の初期化の時にファイルから読み込まれる。

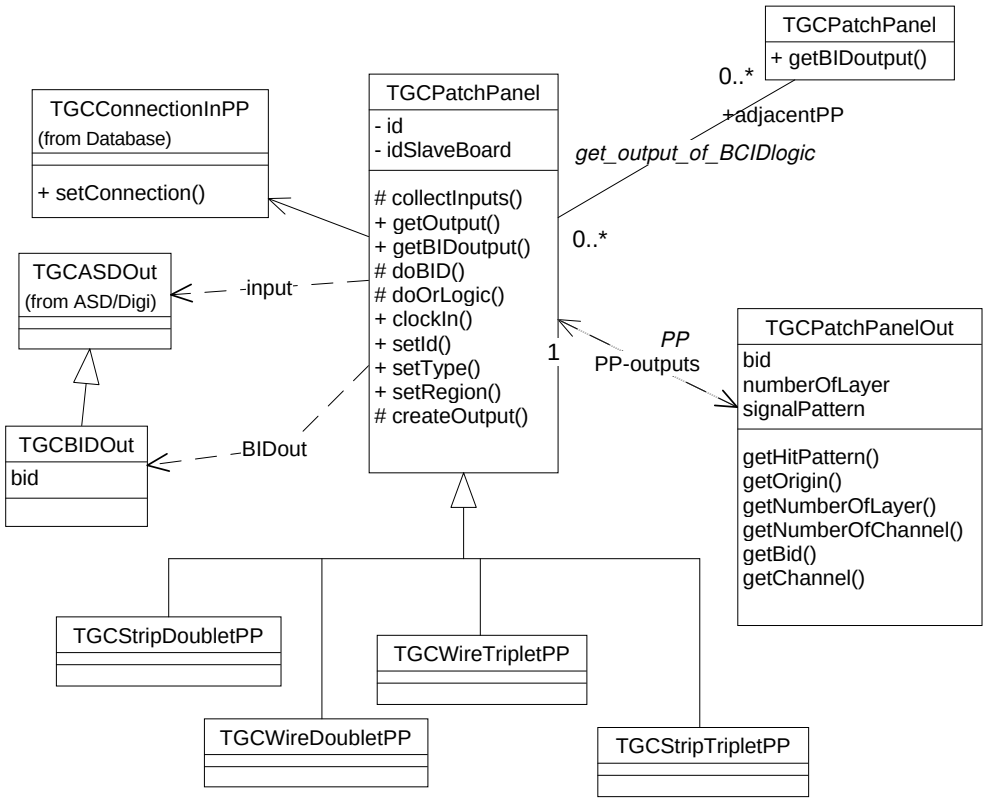


図 36: PatchPanel のクラス図

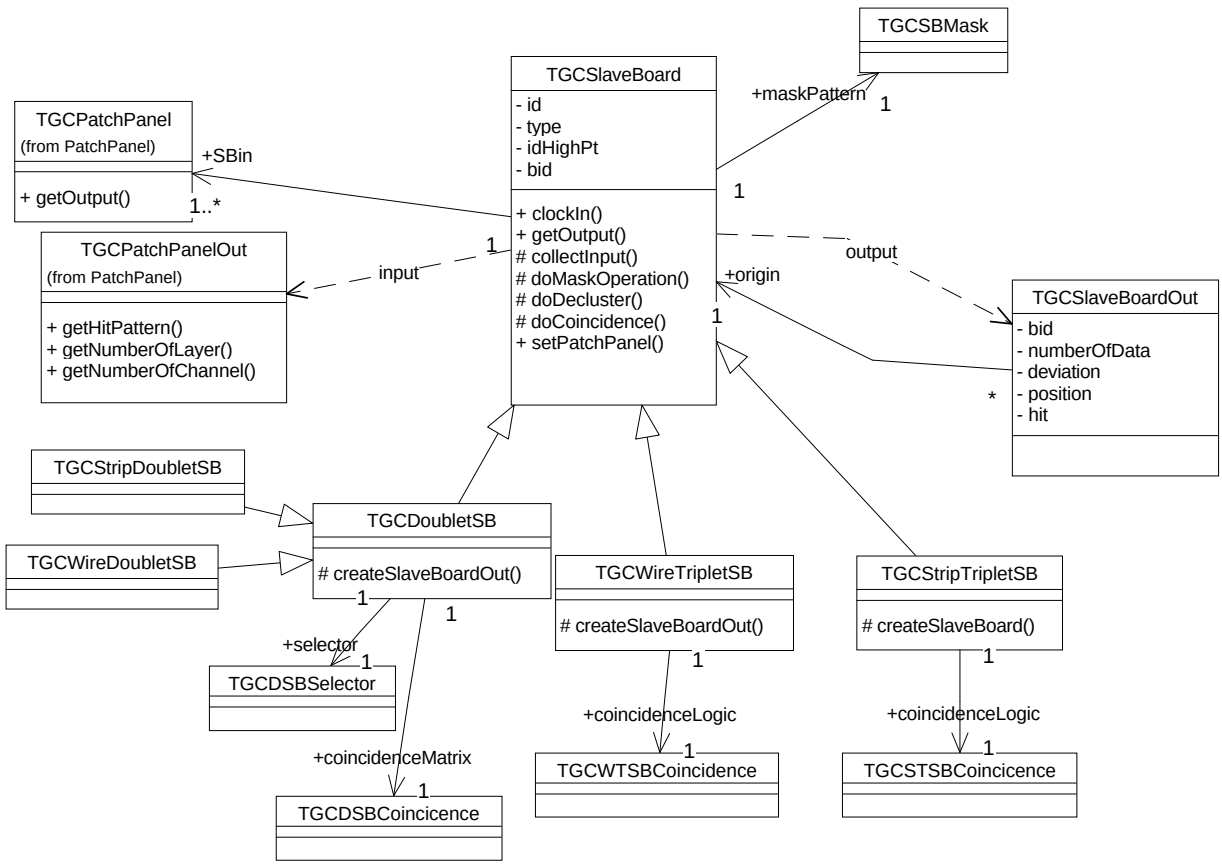


図 37: SlaveBoard のクラス図

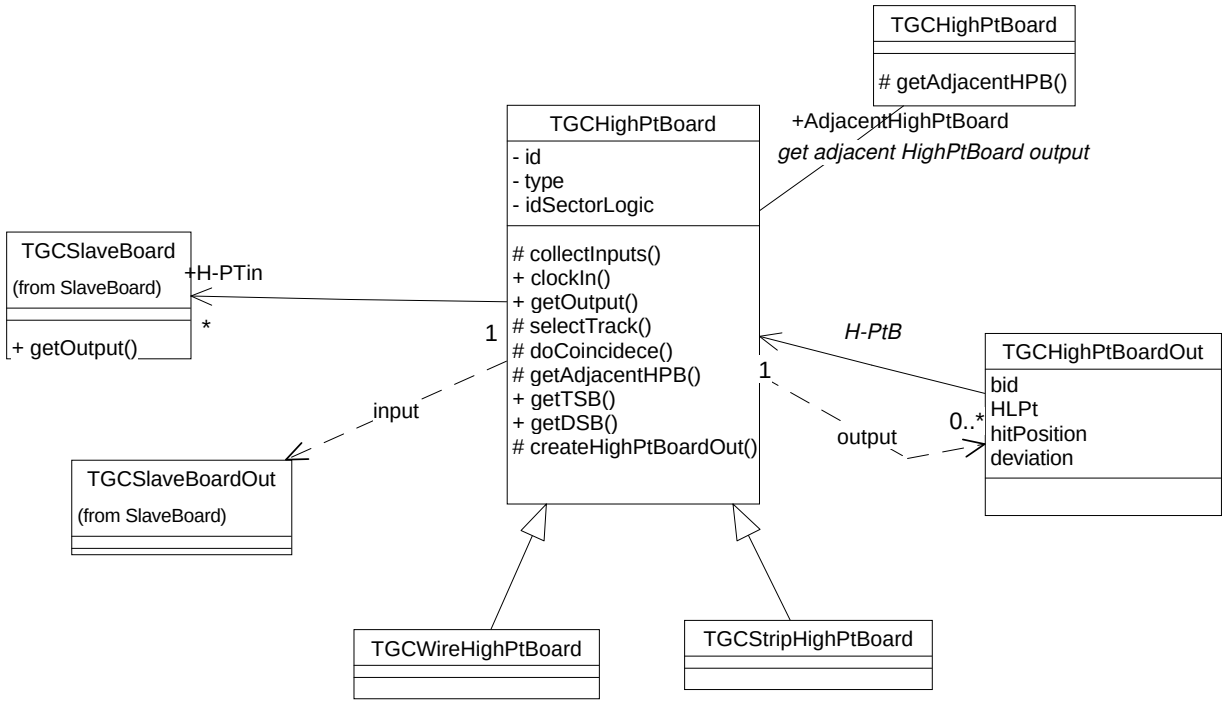


図 38: High-P_T Board のクラス図

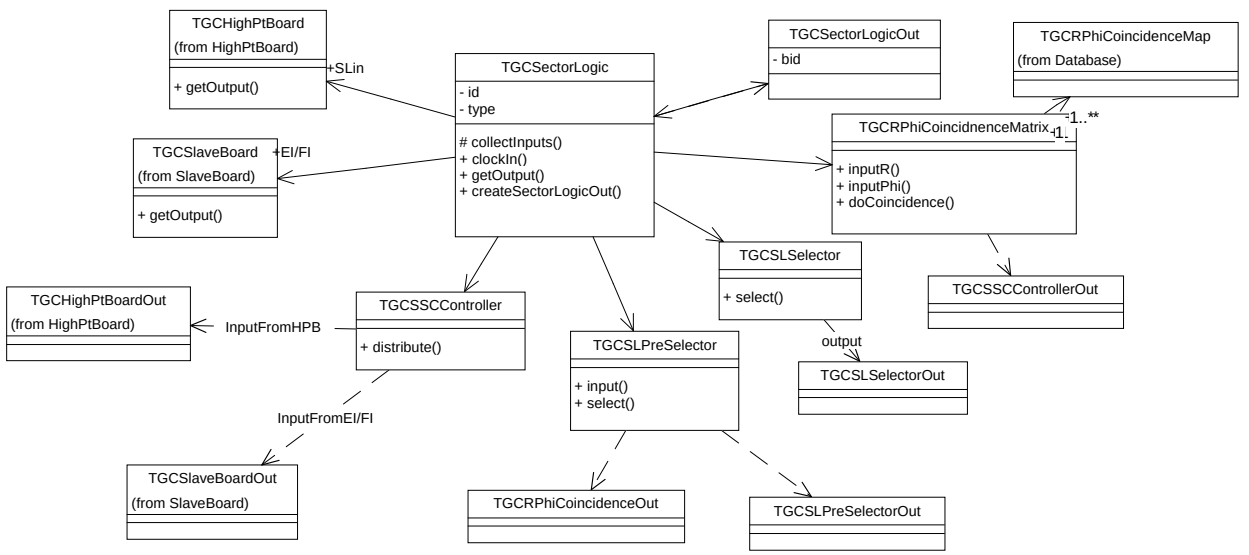
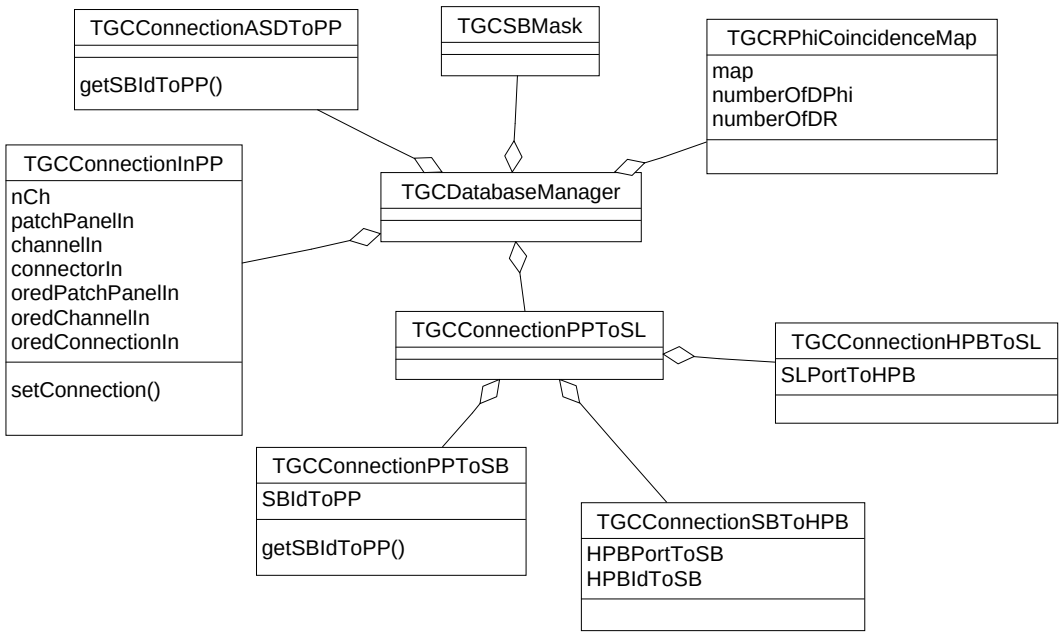


図 39: Sector Logic のクラス図



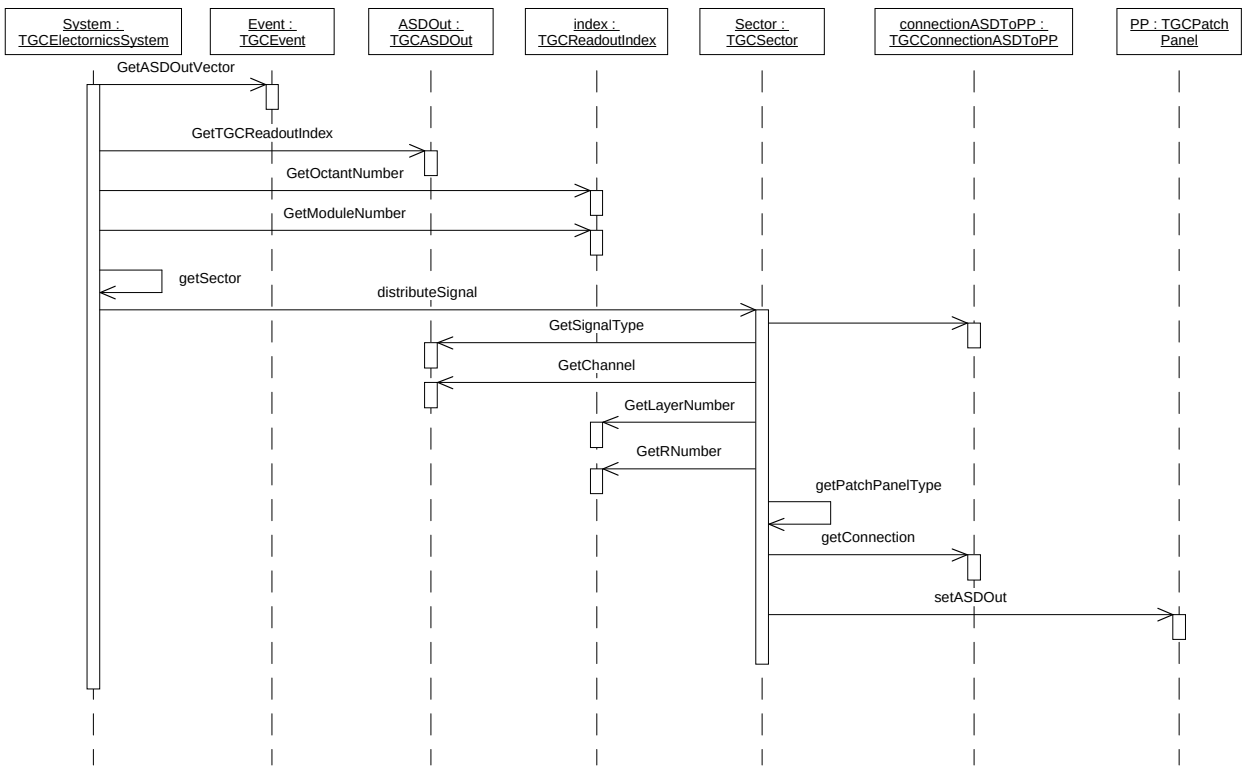


図 41: TGCASDOut の受渡しについてのシーケンス図

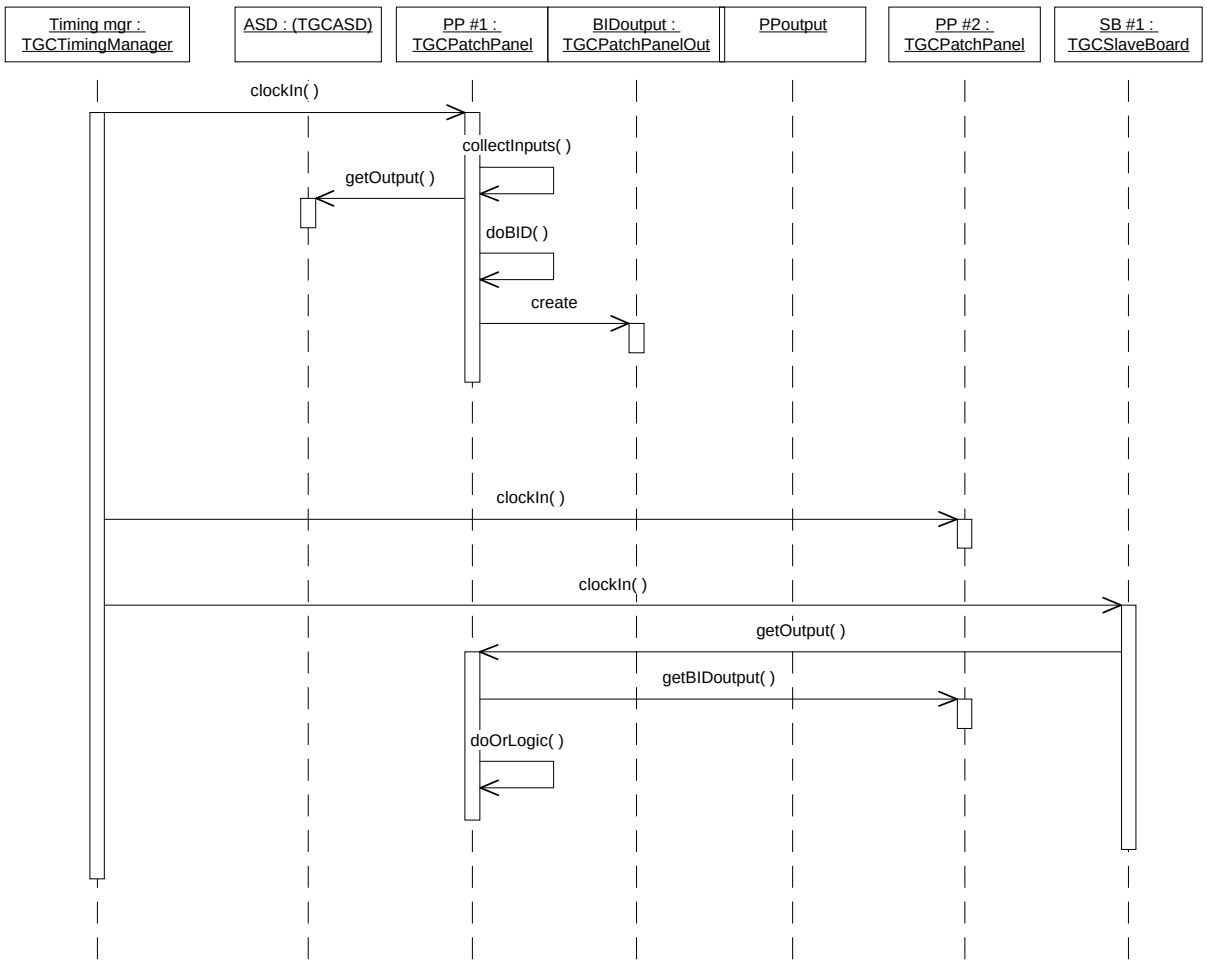


図 42: PatchPanel でのバンチ ID のシーケンス図

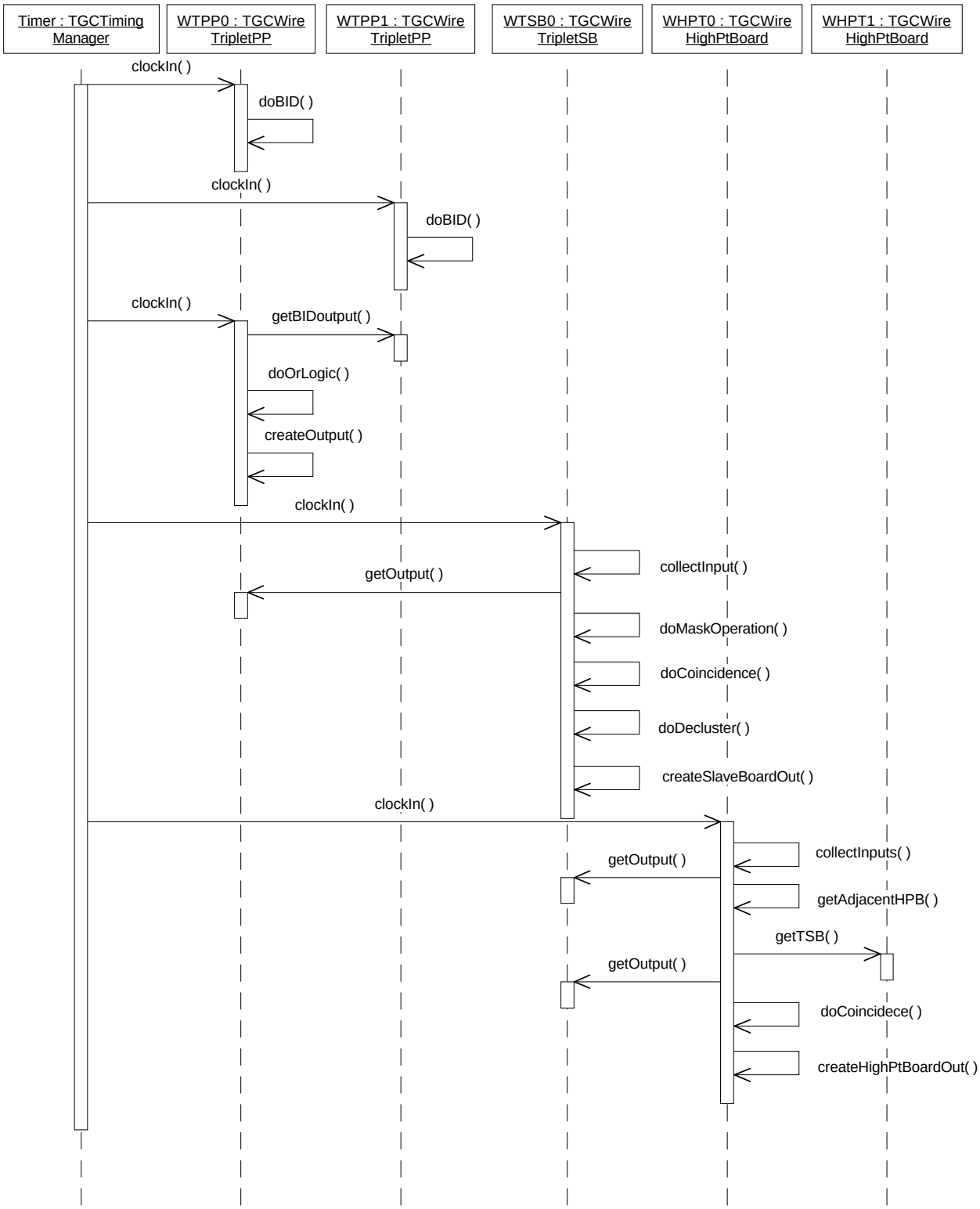


図 43: Patch Panel、SlaveBoard、High-PT Board での処理のシーケンス図

4.4 実装とテスト

4.3 章で説明した設計に基づいて、まずエンドキャップについて Patch Panel, Slave Board, High-P_T Board の部分を実装した。組み立てたプログラムのテストを行なった。まず、プログラムが期待通りに動作しているかを確かめるテストを行なった。これはボードとにテストパターンを入力して行なった。

プログラムとしては正しく動作していても、設計自身に不足があるためにトリガーシミュレーションのプログラムとしては間違っている場合がある。このような間違いがないこと用いたモンテカルロデータは DICE で生成した $P_T = 6, 20\mathrm{GeV}$ の μ^+, μ^- の 4 種類それぞれ約 1 万イベントづつである。η, ϕ について均一に $\eta = 1.2 \sim 2.0$、$\phi = 0.39 \sim 1.18\mathrm{rad}$ の 1octant 分の範囲で発生させた。

P_T	charge	Nev(in Endcap)
20	+	9222
20	−	9131
6	+	8743
6	−	8671

Patch Panel

PatchPanel について確認するべきところは, テーブルを用いているレイヤー, チェンバー, チャンネルの番号と PatchPanel の ID(PPID), コネクタ, チャンネルの番号の対応が正しいかということと、PatchPanel の入力 (PPID, コネクタ, チャンネル) と出力 (PPID, コネクタ, チャンネル) の対応が正しいかという点である。また、チャンネルによっては正しく OR が取られているかということも確かめなければならない。4 種類の PatchPanel のコネクションは全てチャンネルの ASD の出力を 1 チャンネル分づつ順に入力していき対応した出力があることを確認した (図 44)。図中で ID となっているのは PP の種類ごとの番号、con となっているのがコネクタの番号である。2 ヒット同じチャンネルに入っているのは、信号の OR'ing がある場所であることは確認した。

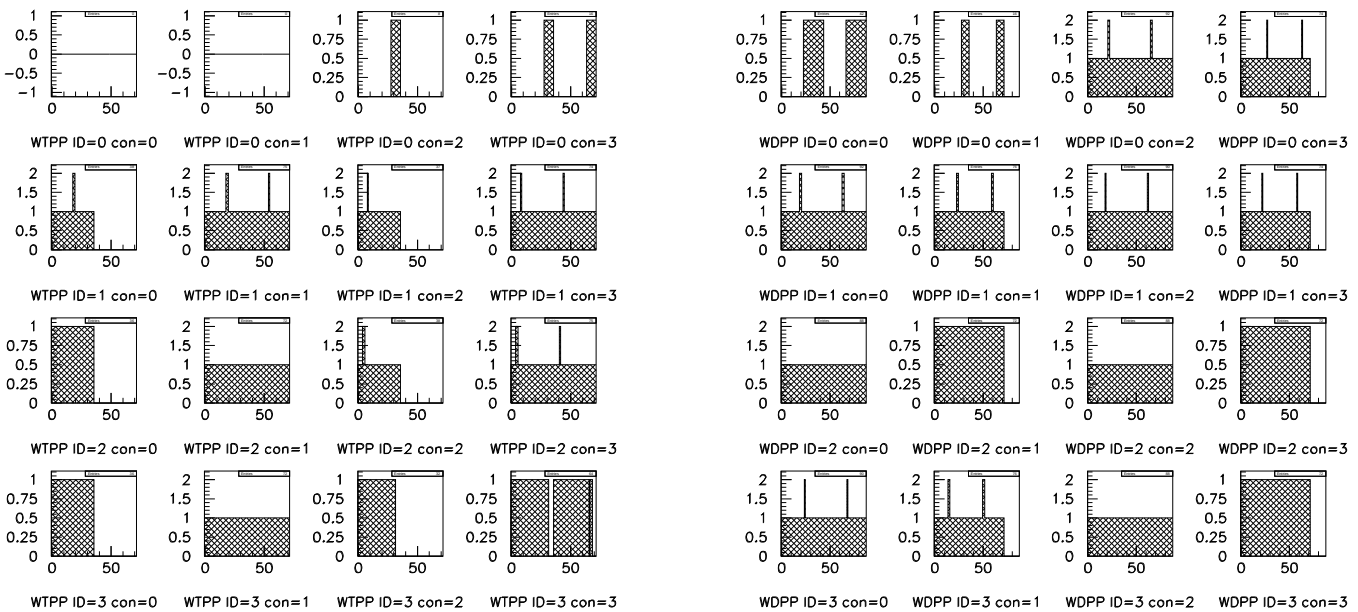


図 44: PatchPanel のテスト

Slave Board

図 45 は SlaveBoard に図??のテストパターンを一つづつ入力した場合の出力の例である。全てのチャンネルで一様に出力しているのが分かる。StripDoubletSlaveBoard の 63 チャンネル目が欠けているが、は middle のテストパターンを 2 ヒットのパターンに絞っているためである。

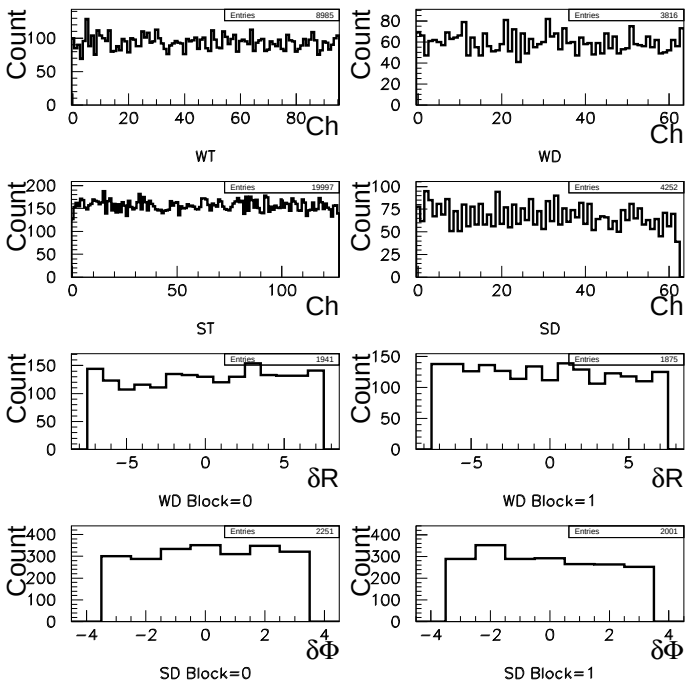


図 45: SlaveBoard にテストパターンを入力した場合の出力の例。StripDoubletSlaveBoard の 63 チャンネル目は middle のテストパターンを 2 ヒットのパターンに絞っているためにかけている。

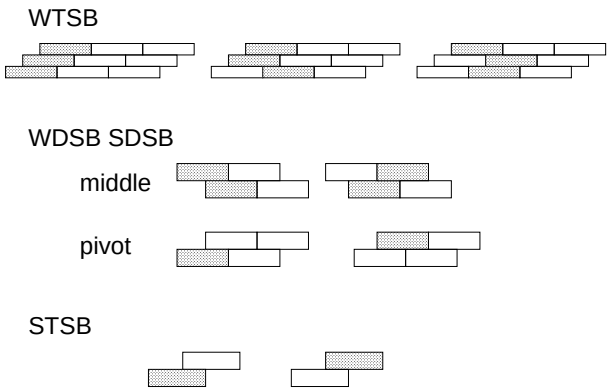


図 46: Slave Board に入力したパターン。四角の箱がチャンネルを表す。塗りつぶされているものは信号のあったチャンネルを表す。

図 47 は 6GeV,20GeV のミューオンに対する、SlaveBoard の δR の分布である。mu^+,μ^- の 6GeV のの δR の分布の中心はそれぞれ 3,-4 であり、20GeV のミューオンの δR はともに 1 である。

ATLAS のトロイド磁場は図 51 のようになっていて、Transition Region を除けば TGC のカバーする領域では 2.0 から 6Tm である。この Slave Board は大体 $\eta = 1.6$ のところにある。この部分の磁場の強さは

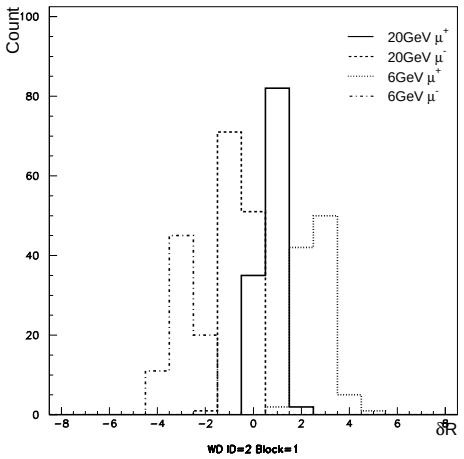


図 47: P_T=6,20GeV のミューオンについての δR 分布

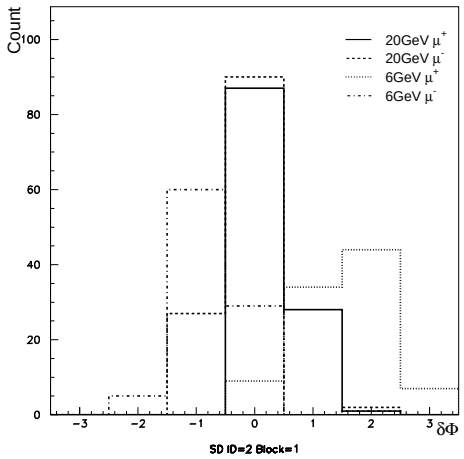


図 48: P_T=6,20GeV のミューオンについての $\delta\phi$ 分布

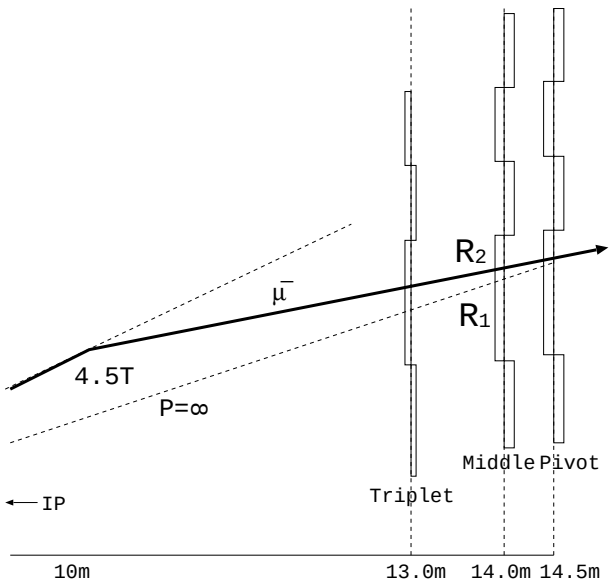


図 49: δR の計算の時に仮定した検出器の配置。

大体 4.5Tm である。この値を用いて 6GeV,20GeV のミューオンが通過する middle doublet 上のヒットの位置の R_1 と pivot doublet でのヒットの位置から求めた middle doublet 上での位置 R_2、それぞれの位置の rapidity の差 $\delta\eta$ は

	R_1[m]	R_2[m]	$\delta\eta[\times 10^{-3}]$	δR
6GeV μ^+	6.311	6.353	6.1	4.0
6GeV μ^-	5.484	5.458	5.0	3.3
20GeV μ^+	6.000	6.024	2.5	1.7
20GeV μ^-	5.761	5.748	2.1	1.4

である。

doublet chamber では singlet (doublet chamber の内の 1 層分) での wire group の中心間隔は η で約 0.003 である。Slave Board の出力で示しているチャンネル数は 2 つの singlet のチャンネルを交互に重ねた時の位置を表しているので、チャンネルの間隔は singlet の wire group の間隔の半分の 0.0015 である (図 52)。このように磁場とチャンネルの間隔から概算した δR はほぼ一致している。一方、ϕ 方向にはきれいな磁場は存在しないために図 48 のように正と負の $\delta\phi$ に広がった分布になる。

High-P_T Board

High-P_T Board では High-P_T Board でのコインシデンスの条件が満たされた場合 High P_T トラックとして High P_T Board のコインシデンスマトリクスで得られた δR をそのトラックの δR とする。一方条件を満たさなかったトラックは Low P_T トラックとして Slave Board で δR,$\delta\phi$ の値が用いられる。図 53 に High P_T トラックの δR の分布を示す。High P_T のコインシデンスの条件をを満たす、6GeV のミューオンによるトラックもあるが、δR の大きさから分離することが可能である。

High-P_T Board でも Slave Board の場合と同じように簡単に δR, $\delta\phi$ の値を見積もることはできる。triplet のそれぞれの singlet でのワイヤーグループの大きさは $\delta\eta$ =0.005 で、triplet 全体でのワイヤーグループの大きさはその 3 分の 1 である。それぞれの場合での δR の値は表 6 の通りである。分布の方は中心がずれているが、それを直せば δR はほぼ一致する。

表 6: High-P_T Board での δR

	R_1[m]	R_2[m]	$\delta\eta[\times 10^{-3}]$	δR
6GeV μ^-	5.164	5.059	19.3	11.5
6GeV μ^+	5.783	5.891	16.9	10.1
20GeV μ^-	5.371	5.337	5.8	3.4
20GeV μ^+	5.556	5.585	4.9	2.9

4.5 トリガーエレクトロニクスの設計への応用

今後プログラムの結果を確かなものにして次のようなことに応用していく。

High-P_T での Hit selection　ここで行なったモンテカルロのデータを用いたシミュレーションでは、バックグラウンドや偽のミューオンによる信号を入れていなかった。これらの信号を含めた場合、High-P_T で

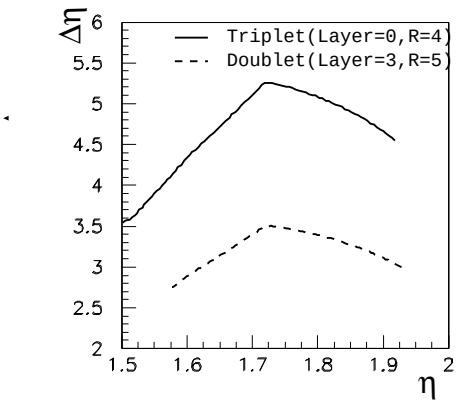


図 50: η で表したチャンネルの間隔。

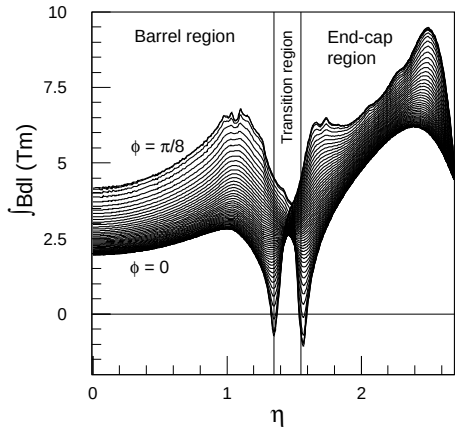


図 51: η に対するトロイド磁場の強さ [4]

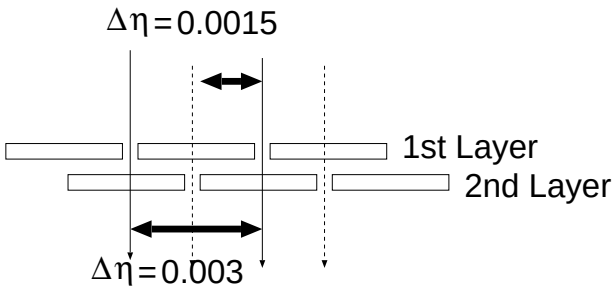


図 52: doublet での η で表したチャンネル間隔

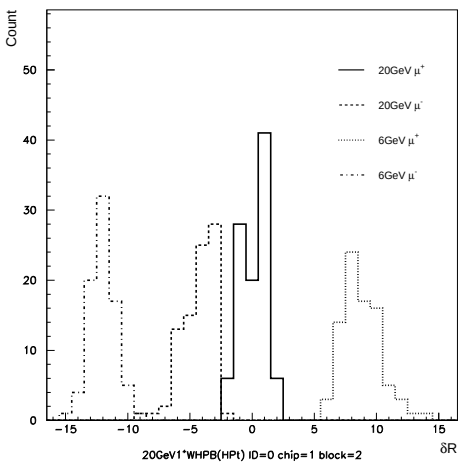


図 53: High-P_T トリガーの場合の HighP_TBoard での δR の分布

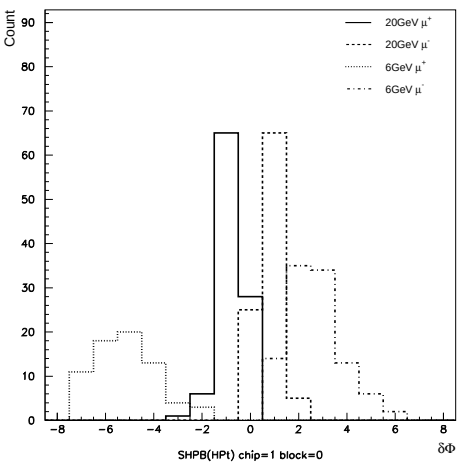


図 54: High-PT トリガーの場合の High-P_T Board での $\delta\phi$ の分布

のチップごとのトラックの選択の時点で本物のミューオンが落されてしまう可能性が出てくる。この確率はどの程度であるか調べ、大きい場合には設計を変更する必要がある。

R-ϕ coincidence map　Sector Logic で本当に全ての SubSector の全ての閾値レベルで別々のマップを用意する必要があるのか、マップの数を減らすことはできないかを検討する。

4.6　まとめ

オブジェクト指向解析・設計の手順にそって、エンドキャップミューオントリガーシミュレーションに必要な概念とその属性を見つけ、クラスにまとめた。またそのクラスのオブジェクトの振舞いをシーケンス図にまとめた。

次に設計に基づいて、プログラムの基本的な部分を組み立てた。プログラム自身の動作を確かめた上で、トリガーのシミュレーションのプログラムとして問題がないかを確認した。またモンテカルロのデータを用いて Slave Board, High-P_T Board での δR,$\delta\phi$ を求めた。今後 sector logic などを実装し、まずエレクトロニクスの設計に応用していく予定である。

5　全体のまとめ

高い衝突頻度の LHC において行なわれる ATLAS 実験では、膨大な数の反応の中から重要な事象を落さずに効率良くデータ収集する必要があるため、トリガーシステムの開発は重要である。ATLAS のトリガーシステムの中で前後方部ミューオントリガーでは、ミューオンの運動量の大きさに対して閾値を設けるために、トロイド磁場による飛跡の曲り具合から運動量を測定する。また、バックグラウンドによる偶発トリガーを防ぐために複数の検出器の同時性 (コインシデンス) を要求する。

本論文ではまず、この処理を行なう回路を含んだ ASIC の設計と試作について述べた。シミュレーション上、飛跡の曲り具合の小さいトラックを優先する機能とコインシデンスを判定する機能が動作することを確認した。また動作周波数と処理に必要な時間に対する条件の 40MHz と 75nsec を満たすことが確認できた。今後動作テストを行ないシミュレーションとの比較を行なう。

次にエレクトロニクスの設計や実験での解析に必要な、トリガーシミュレーション用のプログラムの設計と開発を行ない、オブジェクト指向解析・設計においてクラスとその関連を表すクラス図や、オブジェクトの動的な振舞いを表すシーケンス図にまとめた。また、その設計をもとにプログラムを製作し、その基本的な機能についてテストを行なった。そのテストではミューオンの飛跡の曲り具合を知ることができ、その値は磁場大きさと検出器の配置から予想される値と一致していることを確認した。今後現行のプログラムとの比較を十分に行なった後、モンテカルロのデータを用いたシミュレーションが必要な回路の設計に利用する予定である。

参考文献

[1] Michael Spira and Peter M. Zerwas, hep-ph/9803257

[2] ATLAS Collaboration, ATLAS detector and Physics performance Technical Design Report, ATLAS TDR14, CERN/LHCC/99-14(1999)

[3] ATLAS Collaboration, First-Level Trigger Technical Design Report, ATLAS TDR12, CERN/LHCC/98-14(1998)

[4] ATLAS Collaboration, Muon Spectrometer Design Report, CERN/LHCC/97-22(1997)

[5] Reprot form the ATLAS Architecture TaskForce Version 2.6(10/11/99)

[6] The Architecture TaskForce, Use-case Model and Supplementary Requirements of the ATLAS Common Framework version 1.4(10/11/99)

[7] The Architecture TaskForce, First Attempt at Analysis and Design of the ATLAS Common Framework version 1.4(10/11/99)

6 謝辞

本研究を最後まで指導して下さった佐々木修氏に感謝します。京都大学 坂本宏氏 都立大学 福永力氏をはじめとする TGC エレクトロニクスグループの皆様には感謝します。神戸大学 蔵重久弥氏にはトリガーシミュレーションのプログラムの設計について指導していただいたことを感謝します。近藤敬比古氏, をはじめとする高エネルギー加速器研究機構の皆様には貴重な意見を頂き感謝しております。トリガーシミュレーションにおいて支援していただいた、CERN の長谷川庸司氏, 京都大学 村上光一氏, 農工大 伊藤倫太郎氏には大変感謝しております。学振特別研究員 蓮子和己氏, には研究全般に渡って相談にのっていただきました。感謝します。都立大学 狩野博之氏、京都大学 西田 昌平氏には ASIC の試作などにおいて協力して頂きありがとうございました。研究生活をともにして下さった小林研学生の 陣内修氏、吉田光宏氏、佐藤構二氏、津野総司氏、松浦聡氏、仁木太一氏、香取勇一氏、長島壮洋氏、南條創氏に心から感謝します。最後に、二年間の研究生活において適切な指導並びに助言を頂いた指導教官 小林富雄教授に深く感謝致します。

Appendix A　　VDEC に提出したチップの回路図

matrix, encoder, decluster の部分は仕様にあるものとほぼ同じなので簡単に説明するだけにとどめる。

- Matrix トリガーマトリックス全体 (図 55)
  - SubMatrix (図 56) matrix の半分の部分。2 つの submatrix からそれぞれ 1 ヒット選ばれる。
  - SubSubMatrix (図 57) pivot doublet の信号 8 チャンネルと middle doublet の信号 12 チャンネルを単位としたマトリクス。SubMatrix は SubSubMatrix10 個からなる。
  - Matrix_Core (図 58) SubSubMatrix からローカルコインシデンスの回路を除いた部分。
  - Matrix_Column (図 59) δR,$\delta\phi$ が等しくなる Matrix Element を 1 列分並べたもの。これからの出力が SelHpt に入り、その後 SelHpt で選択された Matrix_Column にだけ信号が戻る。
  - Matrix_Element (図 63) 3 out-of 4 の coincidence の判定を行なう。
  - SelHpt 最も δR の小さい列に信号を返す。δR の正の符合のものを優先する。
- Encoder(図 60)
  - Decluster(図 61,62,64) Matrix から出力された全てのチャンネルについて declustering を行なう。
  - Encoder16to4(図 65) 16 ビットのプライオリティ・エンコーダ。δR の値をエンコードする。encoder32to5 の empty flag を 16 ビット目にに入力する。
  - Encoder32to5(図 66) 32 ビットのプライオリティ・エンコーダ。R の値をエンコードする。empty flag がある。

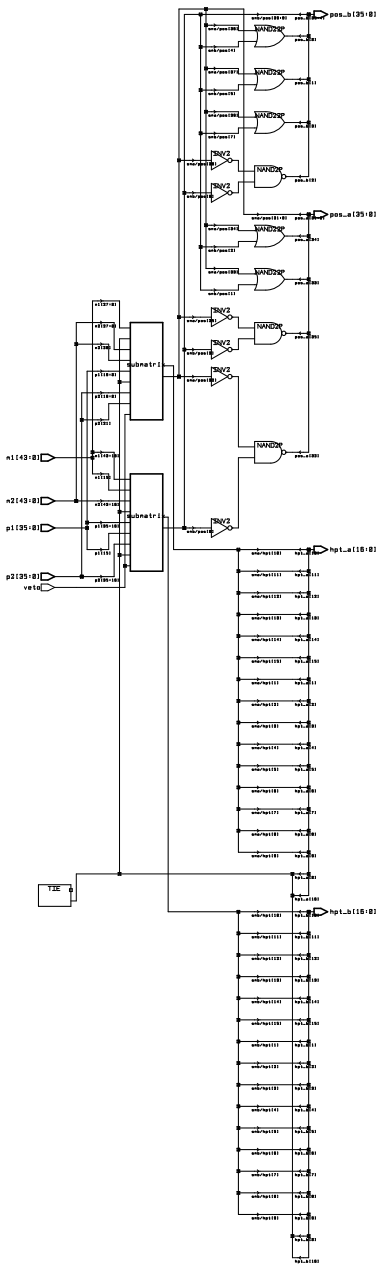

pos_b[35:0]
NAND2X
INV2
pos_a[35:0]
submatrix
n1[43:0]
n2[43:0]
p1[35:0]
p2[35:0]
vote
hpt_a[15:0]
TIE
hpt_b[15:0]

図 55: Matrix

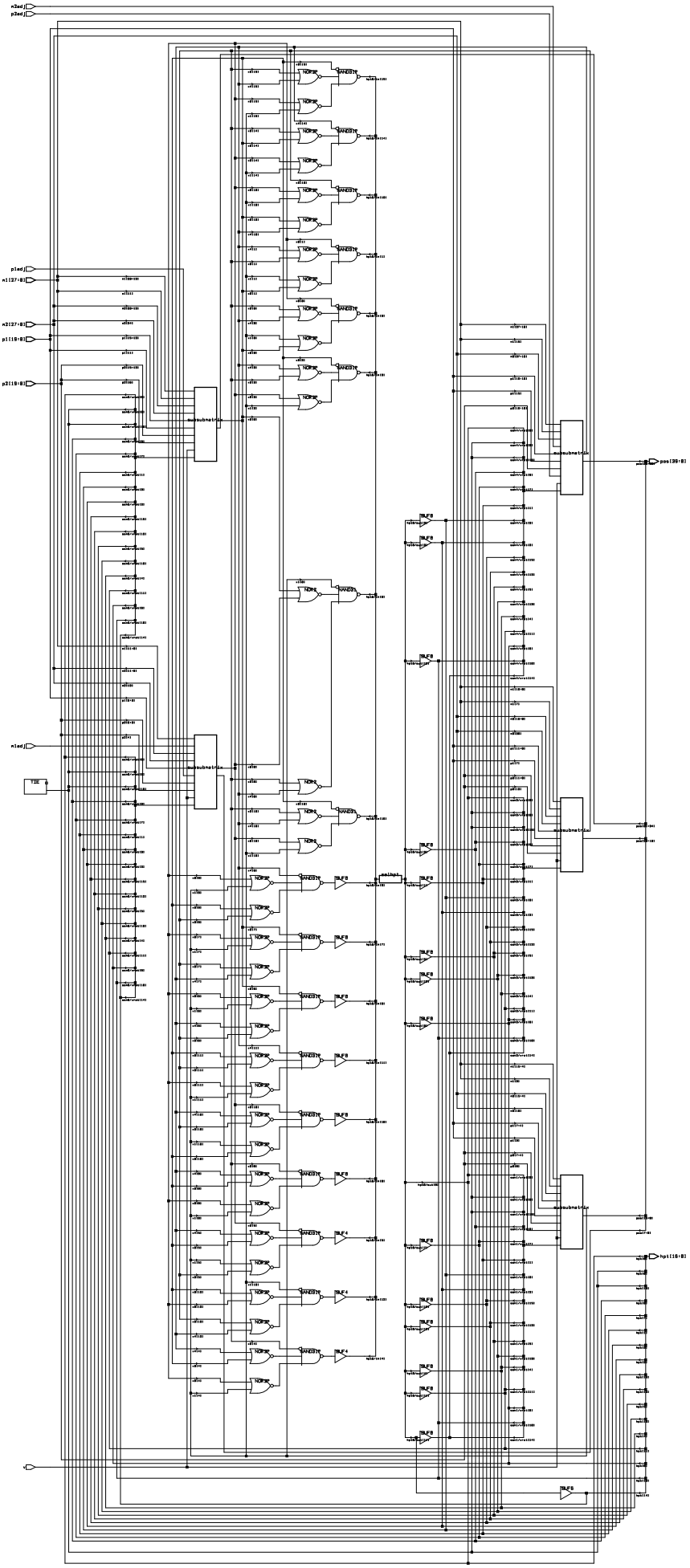


図 56: SubMatrix

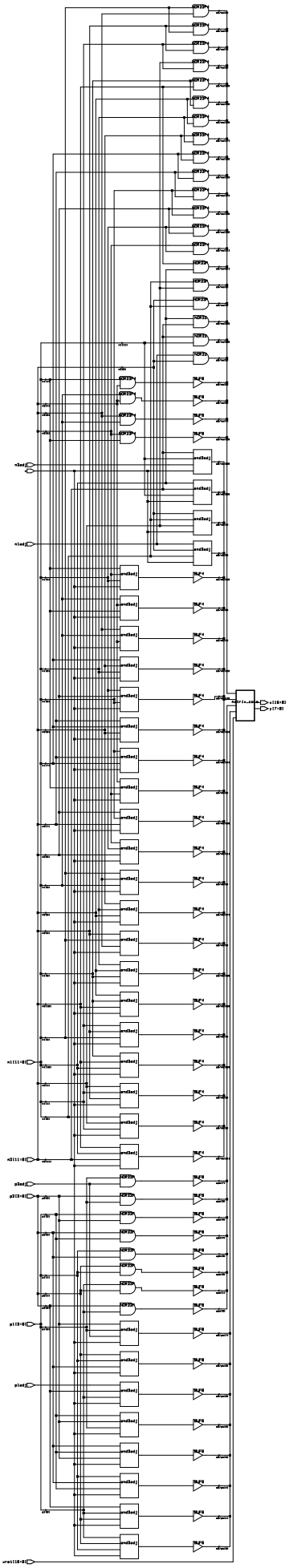


図 57: SubSubMatrix

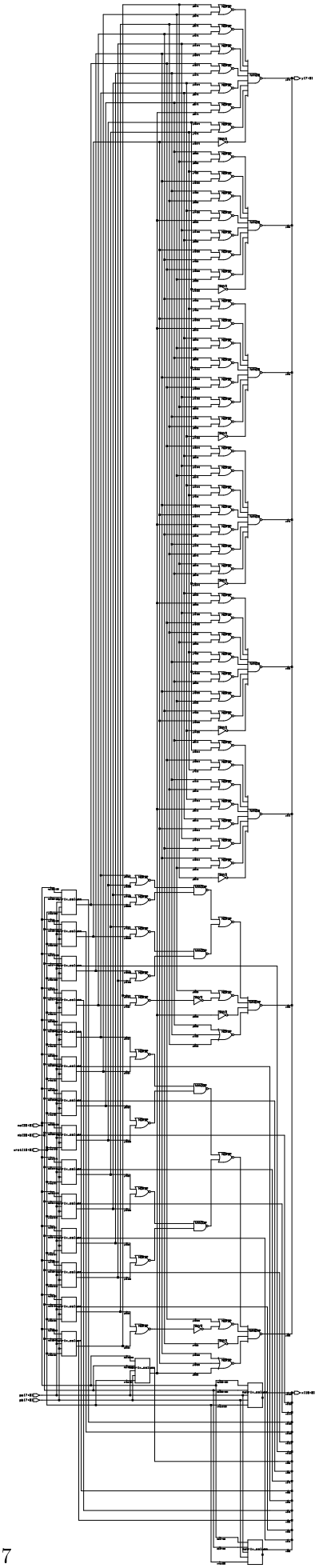


図 58: MatrixCore

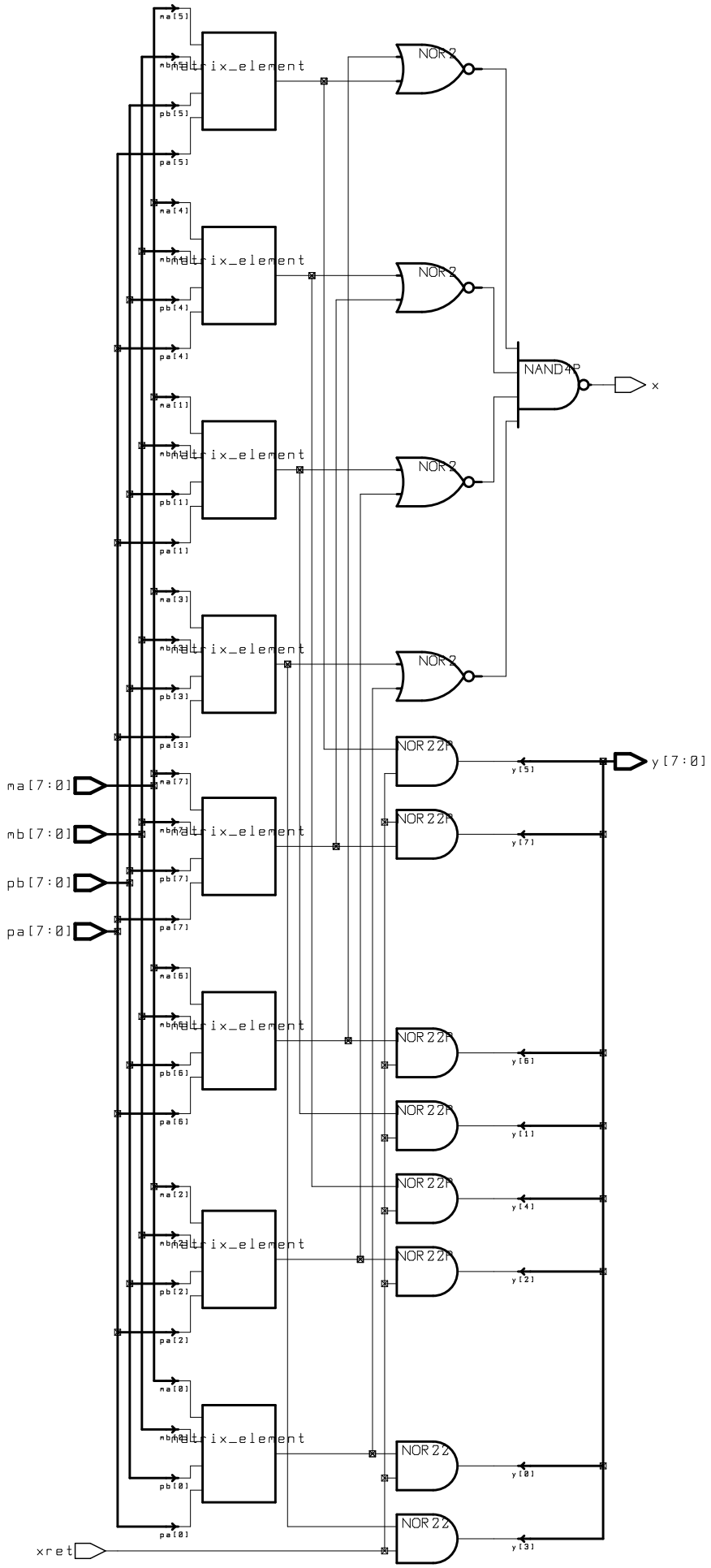


図 59: MatrixColumn

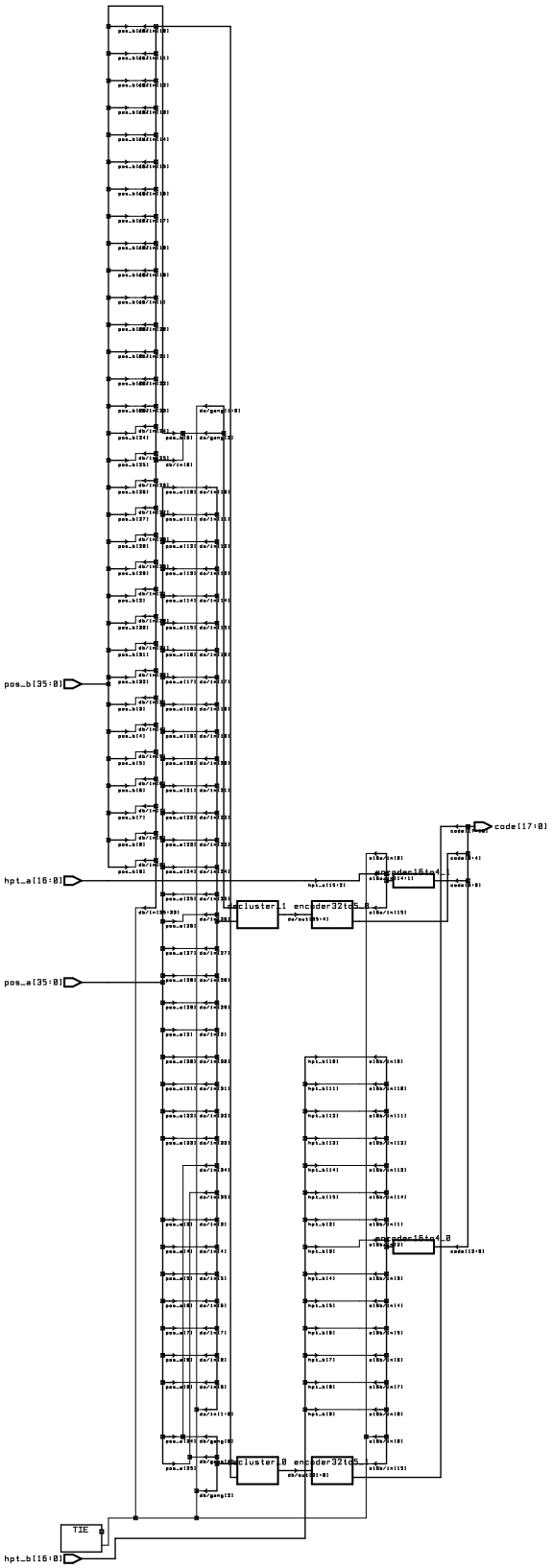


図 60: Encoder

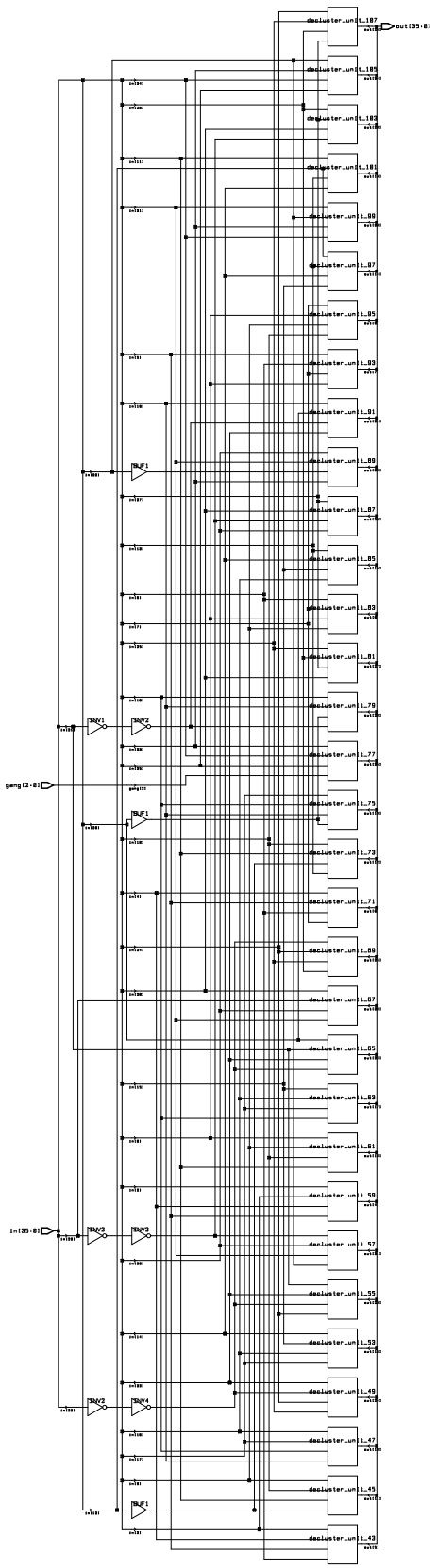


図 61: Decluster

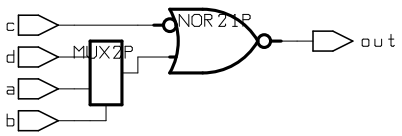


図 62: DeclusterUnit

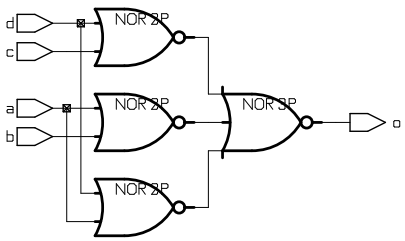


図 63: MatrixElement

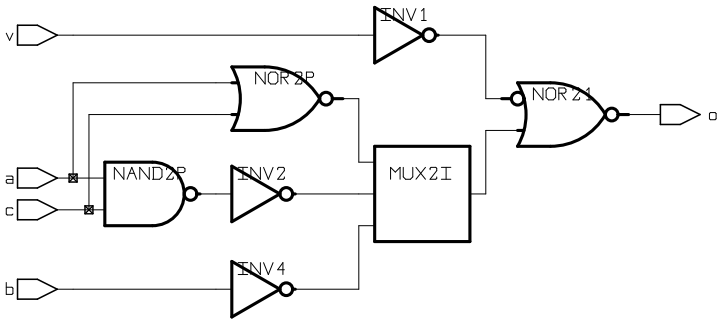


図 64: And3Adj

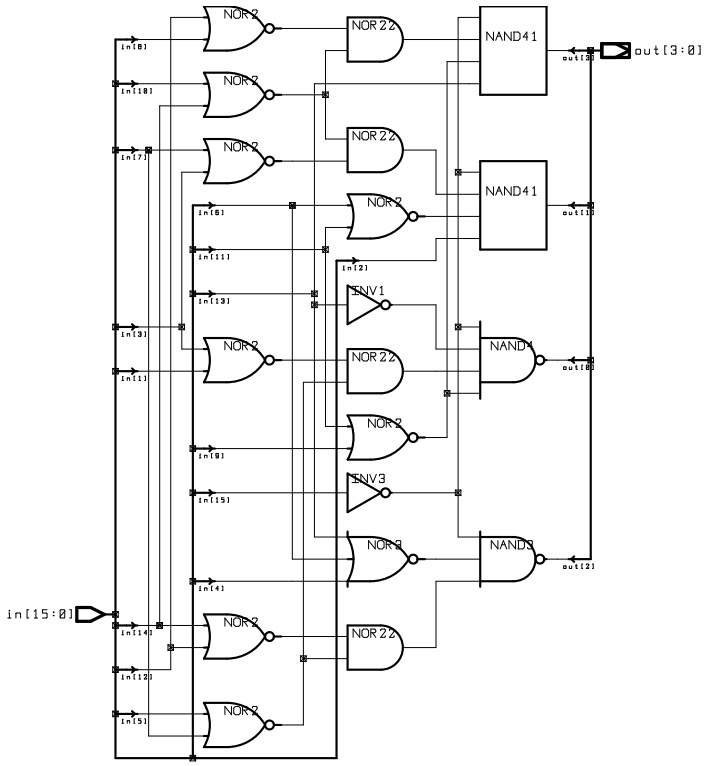


図 65: Encoder16to4

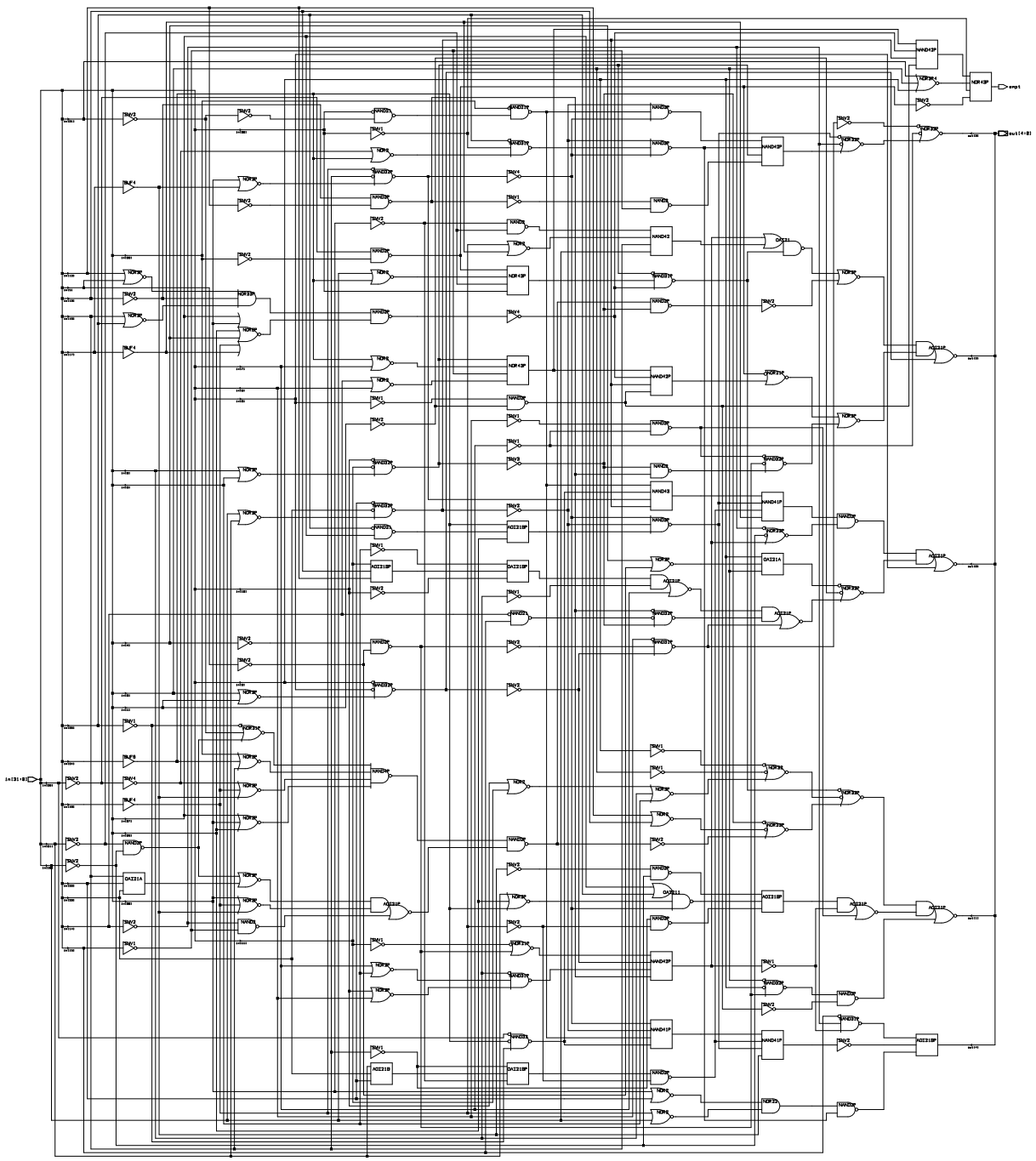


図 66: Encoder32to5

Appendix B　トリガーシミュレーションプログラムについての補足

B.1　naming と numbering scheme

このシミュレーションで用いている TGC の naming と numbering の枠組は以下の通りである。

Backward/Forward ATLAS では η で分けた検出器の領域の名前として"Forward" を使っているので、z 座標で分けた検出器の領域を表すのに Forward/Backward を使うことはできない。代わりに A,B,C を使っている。A,B,C はそれぞれ $z > 0, z = 0, Z < 0$ の領域を表す。

Octant ビーム軸を中心として ϕ 方向に 8 等分した領域。+x 軸上にある Octant を 0 番として ϕ が大きくなる方向に番号を増やしていく。

Module octant を endcap で 6 等分、forward で 3 等分した領域を module と呼んでいる。番号は図??の通りである。

ch Number module ごとに η の小さい方から 0 から順につけていく。Triplet については 1 から順につけていく。

chamber module ごとに η の小さい方から 0 から順につけていく。

projection wire group,strip に対してそれぞれ w,s をつける。

B.2　まだハードウェアのデザインが固まっていない部分

既に述べたように SectorLogic のデザインでは full simulation のデータを使う必要があるため、現時点ではデザインは提案されてはいるがまだ決定していない。また HighPtBoard や SlaveBoard についてはほぼデザインが決まっているが、一部まだ検討されている点がある。このような未決定のデザインの中で主な事項について、この simulation でどのように仮定しているかを記しておく。

- HighPtBoard から送られるデータの数は SectorLogic との間を結ぶ optical link の本数に大きく影響するため、検討が必要であり変更の可能性がある。現在の HighPtBoard で最終的に残る hit の数は下の表の通りである。このシミュレーションでもこれに従っている。

Endcap	wiregroup	8 hits
	strip	4 hits
Forward	wiregroup	4 hits
	strip	2 hits

- HighPtBoard の coincidence matrix の δ の範囲は-20~20 で、-20~-16,16~20 は 2 channel づつ or をとっている。
- 既に示した SlaveBoard の出力と HighPtBoard の入力の対応は仮のもので今後変更の可能性がある。その場合には coincidence matrix への入力が半端になっている R 方向の両端での結果に影響を与える可能性がある。

表 目 次

1 Level-1 trigger での Latency . . . 16
2 Slave Board の種類ごとの入出力ビット数 . . . 17
3 High-P_T Board の種類ごとの入出力のビット数 . . . 23
4 Desgin Analyzer で求めた部分ごとの最大遅延 (単位は ns)。 . . . 30
5 Design Analyzer による Cell Area . . . 30
6 High-P_T Board での δR . . . 48

図 目 次

1 ヒッグス粒子の生成断面積 [1] . . . 4
2 ヒッグス粒子の崩壊の分岐比 [1] . . . 4
3 ATLAS でのヒッグス粒子の発見能力 [2] . . . 4
4 ATLAS 検出器 . . . 6
5 内部飛跡検出器 . . . 6
6 カロリーメータ . . . 7
7 ミューオンシステム . . . 7
8 ATLAS 検出器の RZ 断面図。 . . . 8
9 TGC の信号線の構造 . . . 8
10 Muon trigger system の構成 . . . 9
11 TGC での P_T による選別の仕組み。 . . . 11
12 Level1 Endcap muon trigger の構成 . . . 12
13 Sector の分割 . . . 14
14 SectorLogic のブロック図 . . . 14
15 SlaveBoard のブロック図 . . . 17
16 Declustring logic . . . 18
17 Phase Adjust 回路 . . . 18
18 WTSB の概念図 . . . 19
19 STSB の概念図 . . . 20
20 WDSB のブロック図 (全体) . . . 20
21 ローカルコインシデンスの説明 . . . 21
22 WDSB の Local Coincidence 部 . . . 21
23 WDSB の Matrix 部 (注 : 図中の 2 つの or は実際には複数の Element module にまたがった多入力 or になっている。) . . . 21
24 High-P_T Board の概念図 . . . 23
25 VDEC に提出したチップのブロック図 . . . 24
26 配置配線後の Apollo のスクリーンショット . . . 26
27 テストパターンによる確認の説明 . . . 27
28 テストパターンを入力した場合の出力。3/4,4/4 はそれぞれ 3 out-of 4, 4 out-of 4 のコインシデンスを表す。 . . . 28
29 コインシデンスマトリクスの領域の境界での double counting。図はコインシデンスマトリクスを表し、その対角線上に配置された四角は subsubmatrix を表す。 . . . 28

30 VerilogXL による波形 . . . 29
31 クラス図の要素の説明 . . . 34
32 シーケンス図の要素 . . . 34
33 ASD のクラス図 . . . 37
34 ElectronicsSystem のクラス図 . . . 37
35 Sector のクラス図 . . . 38
36 PatchPanel のクラス図 . . . 39
37 SlaveBoard のクラス図 . . . 40
38 High-P_T Board のクラス図 . . . 41
39 Sector Logic のクラス図 . . . 41
40 Database Manager のクラス図 . . . 42
41 TGCASDOut の受渡しについてのシーケンス図 . . . 42
42 PatchPanel でのバンチ ID のシーケンス図 . . . 43
43 Patch Panel、SlaveBoard、High-PT Board での処理のシーケンス図 . . . 44
44 PatchPanel のテスト . . . 45
45 SlaveBoard にテストパターンを入力した場合の出力の例。StripDoubletSlaveBoard の 63 チャンネル目は middle のテストパターンを 2 ヒットのパターンに絞っているためにかけている。 46
46 Slave Board に入力したパターン。四角の箱がチャンネルを表す。塗りつぶされているものは信号のあったチャンネルを表す。 . . . 46
47 P_T=6,20GeV のミューオンについての δR 分布 . . . 47
48 P_T=6,20GeV のミューオンについての $\delta\phi$ 分布 . . . 47
49 δR の計算の時に仮定した検出器の配置。 . . . 47
50 η で表したチャンネルの間隔。 . . . 49
51 η に対するトロイド磁場の強さ [4] . . . 49
52 doublet での η で表したチャンネル間隔 . . . 49
53 High-P_T トリガーの場合の HighP_TBoard での δR の分布 . . . 49
54 High-PT トリガーの場合の High-P_T Board での $\delta\phi$ の分布 . . . 49
55 Matrix . . . 55
56 SubMatrix . . . 56
57 SubSubMatrix . . . 57
58 MatrixCore . . . 57
59 MatrixColumn . . . 58
60 Encoder . . . 59
61 Decluster . . . 59
62 DeclusterUnit . . . 60
63 MatrixElement . . . 60
64 And3Adj . . . 60
65 Encoder16to4 . . . 60
66 Encoder32to5 . . . 61