

修士学位論文

ATLAS 実験におけるエンドキャップ
レベル1 ミューオントリガーシステムの統合テスト

東京都立大学大学院 理学研究科
物理学専攻
小松知

2003年1月10日

論文要旨

ATLAS 実験は CERN (欧州原子核素粒子研究機構) において建設が進められている LHC(The Large Hadron Collider) を用いて行われる実験である。LHC は重心系衝突エネルギーが 14TeV の陽子・陽子加速器であり、2007 年に運転開始を予定している。この実験で用いられる ATLAS 検出器は LHC 周上に置かれる大型汎用検出器であり、主に Higgs 粒子や超対称性粒子の発見などを目的としている。

LHC では 40.08MHz という高頻度で衝突が行われるため、ATLAS 検出器は効率的に有効なイベントを選択する必要があり、3 段階のトリガー判定システムを設けレートを徐々に下げながらイベント選定を行う。そのなかで、最初に行われるのがレベル 1 トリガーというもので、透過力が高く、高精度で運動量測定ができ、かつ重要な物理現象に関わるミュオンを用いて行われる。そのためのミュオン検出器のひとつに TGC(Thin Gap Chamber) があり、私はこのなかでもエレクトロニクス部分に携わっている。このエレクトロニクスはレベル 1 トリガーの要求を満たすためハードウェアでの実装が必要となり、特定用途向け IC である ASIC によってトリガー処理機能を実現させている。

TGC エレクトロニクス・グループでは、開発されてきた ASIC のほぼ全てが完成段階に入り、これらが搭載されるモジュールのプロトタイプが製作され、単体評価を行ってきた。次段階として、各モジュールを接続しての統合テストが必要になり、テストシステムの構築が不可欠となった。このテストシステムは様々なソフトウェアから構成されるものである。私は各モジュールのコントロールプログラムの統合、及び統合テストのコントロールプログラム開発、周辺プログラムの開発などを行い、統合テストを効率よく行える環境を構築した。さらにこのシステムを ATLAS 検出器全体のコントロールシステムへの組み込みを行い、実際の実験で使用できる TGC エレクトロニクス・コントロールシステムの礎を築くことができた。また、このシステムを用いて TGC エレクトロニクスのトリガーパート統合テストによって動作検証を行った。動作検証はエレクトロニクスの出力とシミュレーションの出力を比較することにより行った。これにより両者の新たな問題点を見つけ出し、修正を積み重ねることによって、最終的には両者の出力は一致し、エレクトロニクスおよびシミュレーションの双方が正確に動作することを確認した。

目次

第 1 章	序論	1
第 2 章	ATLAS 実験	3
2.1	LHC 計画	3
2.2	ATLAS 実験の物理	4
2.2.1	標準理論 Higgs 粒子	4
2.2.2	超対称性 Higgs 粒子	6
2.2.3	超対称性粒子 (SUSY)	7
2.2.4	その他の物理	8
2.3	ATLAS 検出器	9
2.3.1	Inner Detector	10
2.3.2	カロリメータ	11
2.3.3	ミュオン・スペクトロメータ	12
2.3.4	マグネットシステム	16
第 3 章	ATLAS トリガーシステム及び DAQ システム	17
3.1	トリガーシステム	17
3.2	DAQ システム	18
3.3	レベル 1 ミュオントリガーシステム	19
第 4 章	前後方エンドキャップ・ミュオントリガー・システム	22
4.1	TGC の構造	22
4.2	TGC の配置	25
4.3	トリガースキーム	25
4.4	エレクトロニクスシステム	27
4.4.1	Amplifier Shaper Discriminator Board (ASD)	29
4.4.2	PS Board	29
4.4.3	Service Patch Panel Board (SPP)	33
4.4.4	High-pT Board (HPT)	34
4.4.5	Sector Logic Board (SL)	35
4.4.6	Star-Switch Board (SSW)	37
4.4.7	Read Out Draiver Board (ROD)	38
4.4.8	High-pT Star-switch Contrller Board (HSC)	39
4.4.9	Crate Control Interface Board (CCI)	39
4.5	設置環境	41
4.5.1	放射線の影響	41

第 5 章	統合テスト	45
5.1	テスト方法	45
5.2	トリガーシミュレーション	45
5.3	Software	47
5.3.1	テストベクタ変換ソフトウェア	47
5.3.2	Property Control ソフトウェア	47
5.3.3	Run Control ソフトウェア	47
5.3.4	周辺ソフトウェア	47
第 6 章	第 1 回統合テスト	49
6.1	セットアップ	49
6.2	CORBA(Common Object Request Broker Architecture)	51
6.3	Software System	53
6.3.1	Property Control	53
6.3.2	Run Control	57
6.4	テスト結果	60
6.5	第 1 回統合テストのまとめ	62
第 7 章	第 2 回統合テスト	64
7.1	セットアップ	64
7.2	ATLAS Online Software	66
7.3	Software System	68
7.3.1	Configure Database	68
7.3.2	Crate Controller	70
7.4	テスト結果	72
7.5	第 2 回統合テストのまとめ	77
第 8 章	まとめ	78
付録 A	ROD(Read Out Driver) の開発	80
A.1	ROD の役割	80
A.2	基盤実装	80
A.3	ソフトウェアの開発	82
A.4	性能測定とソフトウェア改良	84
A.5	改良点	86
A.6	ROD のまとめ	86
	謝辞	89

目次

2.1	LHC	4
2.2	Higgs 粒子の生成過程	5
2.3	Higgs 粒子生成断面積	5
2.4	Higgs 粒子の質量と崩壊分岐比	6
2.5	Higgs 粒子の発見ポテンシャル	6
2.6	超対称性理論による力の統一	8
2.7	ATLAS 検出器	9
2.8	Inner Detector	10
2.9	Calorimeter	11
2.10	Muon Spectrometer $r-z$ 断面図	12
2.11	トリガー用 Muon Detector	13
2.12	MDT の構造	13
2.13	CSC の断面図	14
2.14	RPC の断面図	14
2.15	TGC の構造 [4]	15
2.16	マグネットシステム	16
2.17	積分磁場強度	16
2.18	トロイダルマグネットの磁束 ($R-\phi$ 平面)	16
3.1	Trigger Levels	18
3.2	レベル 1 トリガーの流れ	19
3.3	レベル 1 ミューオントリガー	20
3.4	TTCvi Board	20
3.5	TTCrx Board	20
4.1	TGC の配置とその構造	22
4.2	TGC の断面図	23
4.3	TGC Triplet、Doublet 構造	24
4.4	TGC のタイムジッター	24
4.5	TGC $r-\phi$ 平面図	25
4.6	TGC $r-z$ 平面図	26
4.7	Sector	26
4.8	Trigger Scheme	27
4.9	TGC エレクトロニクス分類図	29
4.10	ASD の写真	30
4.11	ASD ASIC のコアの写真	30
4.12	PS Board の写真	30

4.13	PP ASIC のコアの写真	31
4.14	SLB ASIC のレイヤー図	31
4.15	SLB 3 out-of 4 ブロック図	32
4.16	SLB 2 out-of 3 ブロック図	32
4.17	SLB 1 out-of 2 ブロック図	32
4.18	SLB EI/FI ブロック図	32
4.19	JRC の内部模式図	33
4.20	JRC ASIC のコアの写真	33
4.21	SPP の写真	34
4.22	HPT の写真	35
4.23	HPT ASIC のコアの写真	35
4.24	HPT Wire ブロック図	35
4.25	HPT Strip ブロック図	35
4.26	SL の写真	36
4.27	SL ブロック図	36
4.28	SSW の写真	37
4.29	ROD の写真	38
4.30	HSC の写真	39
4.31	CCI の写真	40
4.32	HSC-CCI の接続	40
4.33	トリガーエレクトロニクスの配置	41
4.34	多数決論理回路 (Voting Logic)	44
5.1	トリガーシミュレーションの流れ	46
6.1	第 1 回スライステスト セットアップ図	49
6.2	ORB を中心とした CORBA の関連図	52
6.3	ORB を中心とした CORBA の要求、応答の流れ	53
6.4	モジュールコントロール・ソフトウェアの関係図	54
6.5	各モジュールごとの XML ファイルの記述例	55
6.6	アクセスの選択の流れ	56
6.7	第 1 回スライステスト GUI	56
6.8	Run Control Diagram	58
6.9	Data Base Diagram	59
6.10	ランコントロール用 XML ファイルの記述例	59
6.11	第 1 回スライステスト セットアップ写真	60
6.12	PS Board Wire Doublet Patch	61
6.13	PS Board Wire Triplet Patch	61
6.14	HPT ASIC - G-Link 間の接続と HPT Board のアウトプット (Wire 側の第一候補の出力)	61
7.1	第 2 回スライステスト セットアップ図	65
7.2	ATLAS Online Software relation	67
7.3	ソフトウェアの引数の比較	68
7.4	パーティションに関する XML ファイルの記述例	69
7.5	ハードウェアに関する XML ファイルの記述例	69

7.6	ソフトウェアに関する XML ファイルの記述例	69
7.7	モジュールに関する XML ファイルの記述例	70
7.8	Crate Controller Diagram	71
7.9	Run Contrller GUI	72
7.10	第 2 回スライステスト セットアップ写真	73
7.11	PS Board(Strip) の出力	74
7.12	Strip の位置	75
7.13	HPT ASIC - G-Link 間の接続と HPT のアウトプット (Wire 側の第一候補の出力と Strip 側の第一候補の出力)	76
A.1	ROD の構成図	82
A.2	ROD の SH-4 に対するプログラムの流れ	83
A.3	ROD 性能評価試験のセットアップ	84
A.4	CPU から FIFO へのアクセスサイクル	85

表目次

2.1	LHC の主要パラメータ	3
3.1	TTC の主要信号	21
4.1	TGC の主要パラメータ	23
4.2	TGC エレクトロニクスボードと ASIC の種類	28
4.3	パートごとのモジュールの分類表	28
4.4	SSW _{rx} 、SSW _{tx} の機能分類	37
4.5	シミュレーションによる放射線量の見積もり (RHA グループ)	42
4.6	安全係数	42
5.1	周辺ソフトウェア一覧	48
6.1	SLB Wire Doublet フォーマット	62
6.2	SLB Wire Doublet ハードウェア出力	62
6.3	SLB Wire Triplet フォーマット	62
6.4	SLB Wire Triplet ハードウェア出力	62
6.5	HPT の出力フォーマット	62
6.6	第 1 回スライステストの最終結果	63
7.1	SLB Strip Doublet フォーマット	73
7.2	レイテンシーの実測値と要求値の比較	77
7.3	第 2 回スライステストの最終結果	77
A.1	TTC _{rx} トリガー情報	81
A.2	SH-4 の特徴	81
A.3	ROD の処理速度測定値	84

第1章 序論

CERN(欧州原子核素粒子研究機構)において建設が進められている陽子・陽子衝突型加速器 LHC(The Large Hadron Collider) を用いた実験のひとつに ATLAS 実験がある。LHC は 14TeV の重心系衝突エネルギーを引き起こし、今までにない高エネルギー領域での実験を可能にする。ATLAS 実験はこの高エネルギーを利用して、標準理論で予言されている Higgs 粒子や超対称性理論で予言されている粒子の探索を主な目的としている。ATLAS 実験の検出器は LHC 周上の衝突点に設置されるもので、その大きさは長さ 44m、直径 22m、総重量 7000t にも及ぶ巨大なものである。

LHC のイベントレートは 40.08MHz と非常に高く、また 14TeV という高エネルギーのため大量のバックグラウンド粒子が生成されるため、ATLAS 実験では効率よく有効なデータを収集するためのトリガー・システムおよびデータ収集システムが必要となる。このため ATLAS 検出器では、3 段階のトリガー・システムを設け、徐々に有効なイベントを選別してレートを下げている。このトリガー・システムの中で、一番最初に行われるのがレベル 1 トリガーと呼ばれるものである。この一部に前後方(エンドキャップ)ミュオントリガー検出器 TGC(Thin Gap Chamber) があり、トリガー処理のためのエレクトロニクスと共にレベル 1 トリガーの一翼を担っている。エレクトロニクスのトリガー判定回路は、独自のものであり、かつ放射線耐性を備えていなければならないため、その多くが ASIC(Application Specific IC) と呼ばれる特定用途向けの集積回路によって実現される。そのため、TGC エレクトロニクスでは数種類の ASIC を開発してきた。開発してきた ASIC はほぼ全てが完成段階に入り、これらが搭載される数種類のモジュールのプロトタイプを製作し、モジュール単位ごとに動作検証を行ってきた。製作されたモジュールの単体評価も終わり、次なる段階としてこれらのモジュールを接続しての動作検証である統合テストを行うことにした。

まず統合テストを行うためには、システム全体をコントロールするためのソフトウェアの整備が必要となる。そのため、モジュールのコントロール・プログラムの統合、統合テストのラン・コントロール・プログラムの開発、入力するテストベクタの編集プログラムの開発、その他必要な周辺プログラムの開発などを行った。これらをまとめることにより、統合テストを効率よく行うシステム環境を構築した。このシステムは、エレクトロニクスが量産段階になったときの検査体系としてや、TGC を接続してのビームテスト時にも応用でき、これから益々重要性を増してくるものと思われる。また開発したソフトウェアを、本実験の際の TGC エレクトロニクスのコントロールとしても使用できるように、ATLAS 実験共通の ATLAS Online software への導入も行った。今後さらに改良を加えていけば、本実験でも十分に使用できるとと思われる。

次に開発したシステムを用いての統合テストによる動作検証を行った。検証方法は、別に開発されたトリガーシミュレーションの出力とエレクトロニクスからの出力を比較することにより行った。この比較により、双方の新たな問題点を洗い出すことができ、今後の開発に役立てることができた。可能な限り修正を施していった結果、最終的に両者の出力は一致し双方の正当性を確かめることができた。

本論では、まず次章に LHC の特徴や期待される物理、ATLAS 検出器の概要について述べ、第 3 章で ATLAS 検出器のトリガー・システムについて説明する。続いて第 4 章では前後方エンドキャップ・ミュ-

オントリガーの検出器とエレクトロニクスについて説明する。第4章では統合テストに関するソフトウェアに関してふれ、第5章、第6章で2回に渡った統合テスト環境の構築と統合テストに関して述べる。第7章では本論全体を通してのまとめを述べる。

第2章 ATLAS 実験

2.1 LHC 計画

LHC(Large Hadron Collider) は世界最大規模の大型陽子陽子衝突型加速器であり現在スイスとフランスの国境に位置する CERN において建設中である。LHC は LEP(Large Electron Positron Collider) で使われていた周長約 27km のトンネルを利用して建設されているが、LEP とは異なりハドロンコライダーであるためシンクロトロン放射によるエネルギー損失が少ないことにより完成すればかつて無い高エネルギー領域の実験を可能にする。そのエネルギーは重心系で 14TeV にのぼり、ルミノシティは最大 $10^{34}cm^{-2}s^{-1}$ である。これにより TeV 領域での素粒子実験が可能となり、理論上の上限が 1 TeV とされている ヒッグス粒子の発見、対称性粒子の発見、また標準理論の検証、更には未知の事象の発見など様々な成果が期待される。表 2.1 に LHC の主要パラメータを示す。

リング周長	26.66km	衝突頻度	40.08MHz
陽子ビームエネルギー	7.0TeV	バンチ間隔	24.95ns
重心系エネルギー	14.0TeV	バンチ陽子数	1.1×10^{11} 個
低ルミノシティ	$10^{33}cm^{-2}s^{-1}$	ビーム衝突角度	$200\mu rad$
高ルミノシティ	$10^{34}cm^{-2}s^{-1}$	衝突点でのビーム半径	$16\mu m$

表 2.1: LHC の主要パラメータ

LHC には 4 ケ所のビーム衝突点が設けられそれぞれのポイントに検出器が設置される。(図 2.1 参照)

- ATLAS (A Troidal LHC Apparatus)
大型汎用検出器
- CMS (The Compact Muon Solenoid)
ATLAS より小型の汎用検出器
- ALICE (A Large Ion Collider Experiment)
重イオン衝突実験用検出器
- LHC-B
B-Physics に特化した検出器

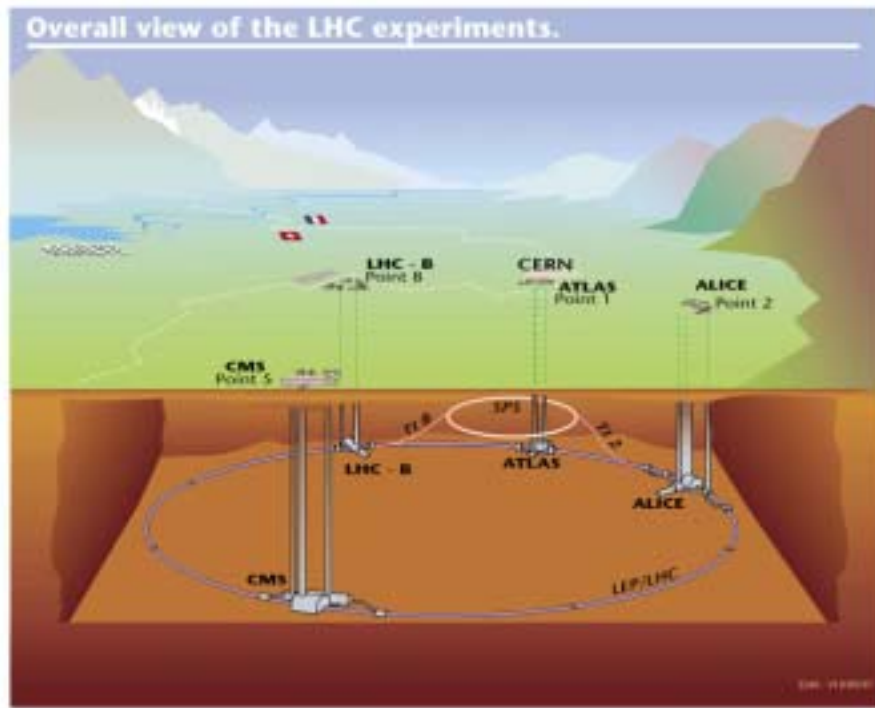


図 2.1: LHC

2.2 ATLAS 実験の物理

ここでは ATLAS 実験がどのような物理を目的として計画されているか、その目指す物理現象に関して述べる。

2.2.1 標準理論 Higgs 粒子

まず ATLAS 実験の最大の目的としてあげられるのが Higgs 粒子の発見である。Higgs 粒子は標準理論において電弱対称性の破れのメカニズムを理解する上で非常に重要な粒子であり、その存在はスカラー場二重項の寄与により中性スカラー粒子として存在するということが予言されている。

まず Higgs 粒子の生成過程に関して述べる。Higgs 粒子は重い粒子と結合しやすいという特徴があるため、主に次の 4 つの生成過程が考えられる。

4 種類の Higgs 粒子生成過程のファインマンダイアグラムを図 2.2 に、またその生成断面積と質量の関係を図 2.3 [1] に記した。

1. グルーオン融合 (gluon fusion) $gg \rightarrow H_{SM}^0$
 トップクォークまたはボトムクォークのループを介した過程で、最も生成断面積が大きい。しかし、Higgs 粒子が崩壊してできる粒子以外に大きな横運動量をもつ粒子がなく、バックグラウンドも多いため選別が厳しい。
2. W/Z 融合 (W/Z fusion) $qq \rightarrow qqH_{SM}^0$

クォークから放出されたゲージボソンから生成される過程で、2 番目に生成断面積が大きい。ゲージボソンを放出したクォークが反跳し、大きな横運動量を持つ 2 本のジェットが観測されるのが特徴で、イベント選別が比較的行きやすい。

3. W/Z 随伴生成 (W/Z associate production) $qq \rightarrow (W/Z)H_{SM}^0$

クォークペアの対消滅で生成されたゲージボソンから Higgs 粒子が放射される過程。終状態でゲージボソン (W/Z) が観測される特徴がある。また、荷電粒子から光子が放出されるシンクロトロン放射 (Bremsstrahlung) に似ていることから "Higgsstrahlung" とも呼ばれる。

4. トップクォーク随伴生成 (top associate production) $qq/gg \rightarrow ttH_{SM}^0$

グルーオンから対生成されたトップクォークから Higgs 粒子が放出される過程。生成断面積は小さいが特徴のあるトップクォークのペアを終状態に含むため、バックグラウンドを減らすことができる。

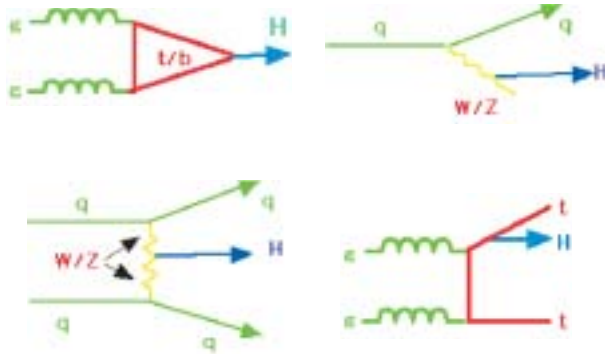


図 2.2: Higgs 粒子の生成過程

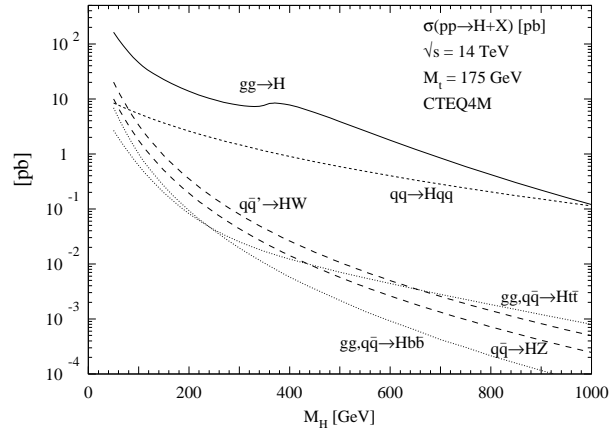


図 2.3: Higgs 粒子生成断面積

次に Higgs 粒子の崩壊過程に関して述べる。崩壊過程の分岐比、崩壊幅は Higgs 粒子の質量によって異なっており、図 2.4 [1] に示すように各質量領域で特徴的な崩壊過程が存在することがわかる。そのため、探索方法も質量によって異なってくる。以下にそれぞれの崩壊過程を簡単に示す。

• $H \rightarrow \gamma\gamma$ ($m_H < 150\text{GeV}$)

この質量領域での主な崩壊過程は $b\bar{b}, c\bar{c}, \tau^+\tau^-$ である。このとき Higgs 粒子がグルーオン融合過程によって生成されていた場合は、QCD によるバックグラウンドが大きくなるため選別が難しい。しかし希崩壊過程である $H \rightarrow \gamma\gamma$ を観測し、不変質量 $m_{\gamma\gamma}$ の分布を求めると Higgs 粒子の質量が鋭いピークとしてあらわれる。だが、この場合も $q\bar{q} \rightarrow \gamma\gamma, g\bar{g} \rightarrow \gamma\gamma$ という QCD からのバックグラウンドがあるため、このバックグラウンドとの選別のために角度分解能、およびエネルギー分解能の優れた電磁カロリメータが必要となる。

• $H \rightarrow ZZ^* \rightarrow 4l^\pm$ ($120\text{GeV} < m_H < 180\text{GeV}$)

この領域では m_H が m_W や m_Z よりも大きくなるため WW^* や ZZ^* への崩壊が始まり、W への崩壊の場合は $tt \rightarrow WWbb$ がバックグラウンドとなり選定が困難になるため Z への崩壊を調べるのが重要となる。この Z への崩壊過程はきれいなピークを得ることができるモードである。この場合 2 つのレ

プトン対のうち1つのレプトン対に対してはその不変質量が m_Z と等しくなるという条件を課すことができるが、 Z^* が仮想粒子であるためもう一方のレプトン対の不変質量には制限がない。そのため運動量やエネルギーに対する高い分解能を持つ検出器が必要となる。

- $H \rightarrow ZZ \rightarrow 4l^\pm$ ($180\text{GeV} < m_H < 800\text{GeV}$)

この崩壊過程はこの質量領域で最もきれいなピークを得ることができるモードで、ほかの質量領域の崩壊過程と比べても最もはっきりとあらわれるモードである。2つのレプトン対の不変質量が共に m_Z に等しいという条件を課すことができるため、信頼性の高いモードと言える。しかし、この質量領域の中でも高い領域では、Higgs 粒子の崩壊幅が急激に大きくなるためバックグラウンドとの選別が困難となる。そのためこの崩壊だけでは十分な統計を得ることができないため以下の2つの崩壊が重要となる。

- $H \rightarrow ZZ \rightarrow ll\nu\nu$ ($400\text{GeV} < m_H$)

この質量領域では上記の崩壊よりもこの崩壊のほうが約6倍分岐比が高くなる。この崩壊の場合 $\nu\nu$ の不変質量は再構築することができないが $\nu\nu$ に起因する消失横方向エネルギー (missing- E_T) を精密測定することにより補う。

- $H \rightarrow ZZ \rightarrow lljj, H \rightarrow WW \rightarrow lvjj$ ($600\text{GeV} < m_H$)

この領域では、これらの崩壊の分岐比が非常に高くなっていくため重要なイベントである。これらの崩壊においてバックグラウンドとの選別は、Higgs 粒子が W/Z 融合によって生成された場合を考える。この場合、散乱角前方にクォークによる2つのジェットがあらわれるのが特徴で、このジェットを観測することによってバックグラウンドを排除する。

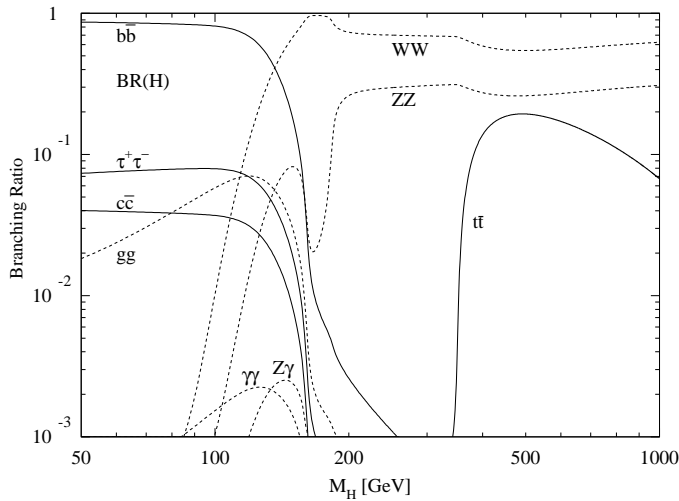


図 2.4: Higgs 粒子の質量と崩壊分岐比

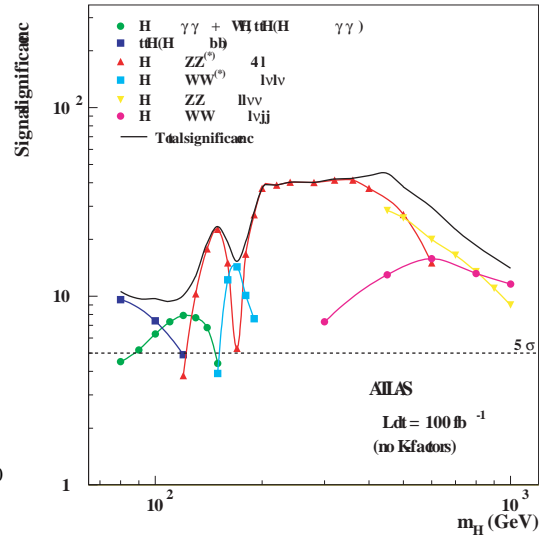


図 2.5: Higgs 粒子の発見ポテンシャル

2.2.2 超対称性 Higgs 粒子

Higgs 粒子は、超対称性理論でもその存在が予言されている。超対称性理論の中で最も単純な理論として超対称性標準理論 (Minimal Supersymmetric Standard Model) があげられる。この理論では2つの Higgs2 重

項が要求され、これにより 5 つの Higgs 粒子の存在が予言されている。その 5 つとは H^\pm (電荷スカラー)、 h (中性軽スカラー)、 H (中性重スカラー)、 A (中性擬スカラー) である。以下に MSSM で特徴的な崩壊モードに関して述べる。

- $H/A \rightarrow \tau\tau$
標準理論の Higgs 粒子の場合、 $H \rightarrow \tau\tau$ モードは分岐比が低く測定には適さないが、MSSM の場合は高い分岐比が予測されており観測が期待される。生成された 2 つの τ 粒子がレプトンに崩壊する場合と一方がハドロンに崩壊する場合の 2 種類のモードが利用できる。
- $H/A \rightarrow \mu\mu$
上記のモードと比べて、分岐比は $(m_\mu/m_\tau)^2$ 倍低いですが、精度よく測定を行うことが可能で、 $\tau\tau$ モードの測定を補う役割が期待される。
- $H \rightarrow hh$
崩壊モードは $hh \rightarrow bb\bar{b}\bar{b}$ が支配的だが、このモードの場合は効率のよいトリガーが行えないため、 $hh \rightarrow \gamma\gamma b\bar{b}$ モードで観測されることが期待される。イベントレートは低いですが、2 つの異なる Higgs 反応といい意味で非常に興味深いモードである。
- $A \rightarrow Zh$
2 つの Higgs 粒子が関係した反応として興味深いモードである。 $Zh \rightarrow llb\bar{b}$ など Z の崩壊で生じる 2 つのレプトンでトリガーを行う方法が有効である。

2.2.3 超対称性粒子 (SUSY)

超対称性は重力まで統一する理論として最も有望とされている理論である。LEP 実験において Z^0 粒子の崩壊から、相互作用の大きさを正確に測定された。理論上、力の大きさはエネルギーと共に変化するので、LEP 実験で精度よく測定された相互作用の大きさを、超対称性理論に従って非常に高いエネルギーまで変化させると 3 つの力が統一される可能性が示唆されている。図を図 2.6 に載せる。

超対称性理論ではボソンとフェルミオンの交換がおき、ボソンやフェルミオンとスピンの $1/2$ だけ異なるスーパーパートナーの存在を予言している。クォークやレプトンというフェルミオンに対してのスーパーパートナーとしてスクォーク (\tilde{q}) やスレプトン (\tilde{l}) というボソンの存在が予言され、ボソンのグルーオンに対してはフェルミオンのグルイーノ (\tilde{g}) が予言されている。

R-parity 保存則により、超対称性粒子は必ず対になって生成される。生成された粒子は崩壊し、最終的に超対称性粒子の中で最も質量の小さい LSP (Lightest Supersymmetric Particle) になる。この LSP の候補として考えられているのが最軽量ニュートラリーノ ($\tilde{\chi}_1^0$) であるが、この粒子は直接検出することはできない。しかし解析において消失横方向エネルギー E_T^{miss} として現れることになるため、ジェットと共にこの E_T^{miss} を指標に探索が行われる。以下に主な崩壊モードをあげる。

- Multi-jets + E_T^{miss} モード

$$\tilde{g} \rightarrow q\bar{q}\tilde{\chi}_1^0 \rightarrow jets + E_T^{miss}$$

$$\tilde{q} \rightarrow q\tilde{\chi}_1^0 \rightarrow jets + E_T^{miss}$$
- 同符号の 2 レプトンモード

$$2\tilde{g} \rightarrow 2(q\bar{q}\tilde{\chi}_i^\pm) \rightarrow 2(q\bar{q}W^\pm\tilde{\chi}_i^0) \rightarrow 2(jets + l^\pm + E_T^{miss})$$
- 3 レプトンモード

$$\tilde{\chi}_1^\pm\tilde{\chi}_2^0 \rightarrow l\nu\tilde{\chi}_1^0 \rightarrow ll\tilde{\chi}_1^0 \rightarrow 3l + E_T^{miss}$$

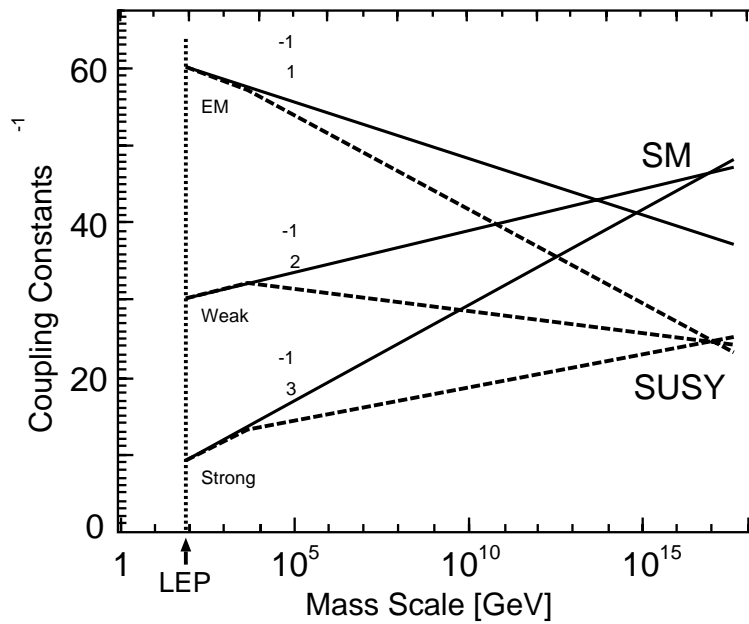


図 2.6: 超対称性理論による力の統一

2.2.4 その他の物理

ALTA5 実験では Higgs 粒子の発見、研究や超対称性という標準理論を超える新しい素粒子現象の探索以外に標準理論の検証、ボトムクォークの精密検証、QCD の精密検証などがある。標準理論の検証はトップクォークの質量や部分崩壊率の測定などにより行われ、ボトムクォークの精密研究はボトムクォークの稀崩壊現象を探索し標準理論を超える物理を探る手段として行われる。QCD の精密研究はクォークに内部構造がないかということの高い横運動量を持つジェット生成断面積の測定を通して行われる。

2.3 ATLAS 検出器

ATLAS 検出器 [2] は直径 22m、長さ 44m の円筒形の形をしており、総重量 7000t にものぼる大型汎用検出器である。ATLAS 実験は高いエネルギー領域での実験であり、イベントレートも高く放射線量も多いため検出器は高速かつ正確なデータ収集、データ処理、耐久性、放射線耐性が必要になってくる。ATLAS 検出器は内側から Inner Detector、カロリメータ、ミュオン・スペクトロメータから構成され、検出器の間にはマグネットシステムが設置される。検出器の全体図を図 2.7 に示す。また、ATLAS 検出器は以下に示すような要求を満たすように設計される。

- Inner Detector による高精度の飛跡検出
- 電磁カロリメータによる電子と光子の高精度測定
- ハドロンカロリメータによるジェットと消失横方向エネルギーの高精度測定
- ミュオン・スペクトロメータによるミュオンの運動量の高精度測定
- 広いラピディティ η^1 と完全な方位角のカバー
- 高レートのイベントをロスすることなく処理する
- 放射線耐性

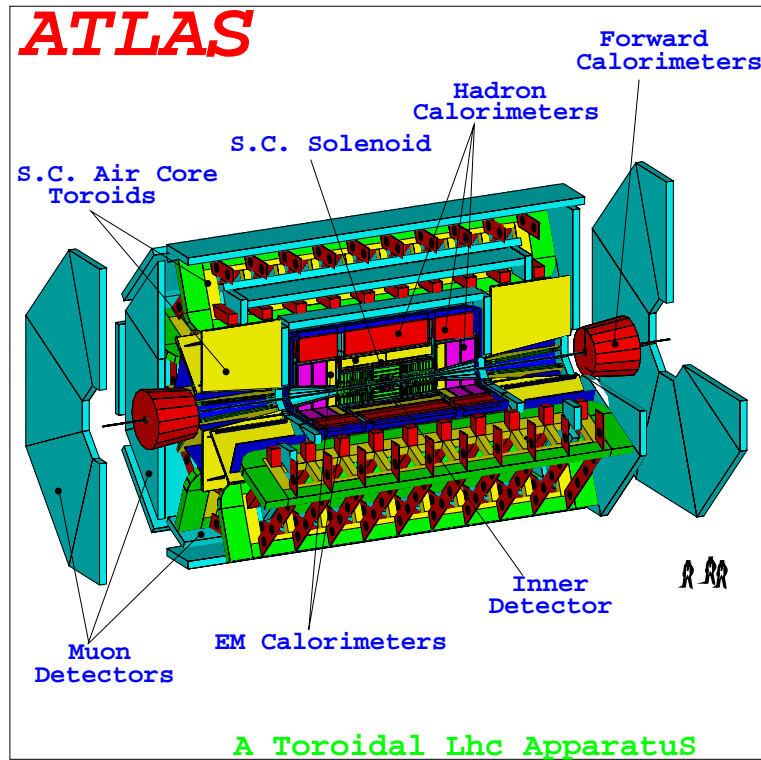


図 2.7: ATLAS 検出器

¹ η は擬ラピディティ (pseudo rapidity) と呼ばれ、ビーム軸とジェットのなす角を θ とすると $\eta = -\ln(\tan(\theta/2))$ で定義される。

2.3.1 Inner Detector

Inner Detector はビームの衝突点に最も近い場所に設置され、2Tの磁場をつくる超伝導ソレノイドの内部に位置する。主な機能は荷電粒子のトラッキングである。この検出器は内側から順に以下のような3種類の検出器で構成されている。図 2.8 参照

- ピクセル検出器
ひとつの要素が $50\mu\text{m} \times 300\mu\text{m}$ の半導体検出器で、高い位置分解能を持っている。
- セミコンダクタートラッカー (Semiconductor Tracker)
シリコンマイクロストリップと呼ばれる約 $80\mu\text{m}$ ピッチの細長い有感領域を持った半導体検出器である。
- 遷移輻射トラッカー (Transition Radiation Tracker)
径 4mm のストローチューブを束ねた検出器で、トラッキングの他に遷移輻射を利用した電子、光子の同定も行う。

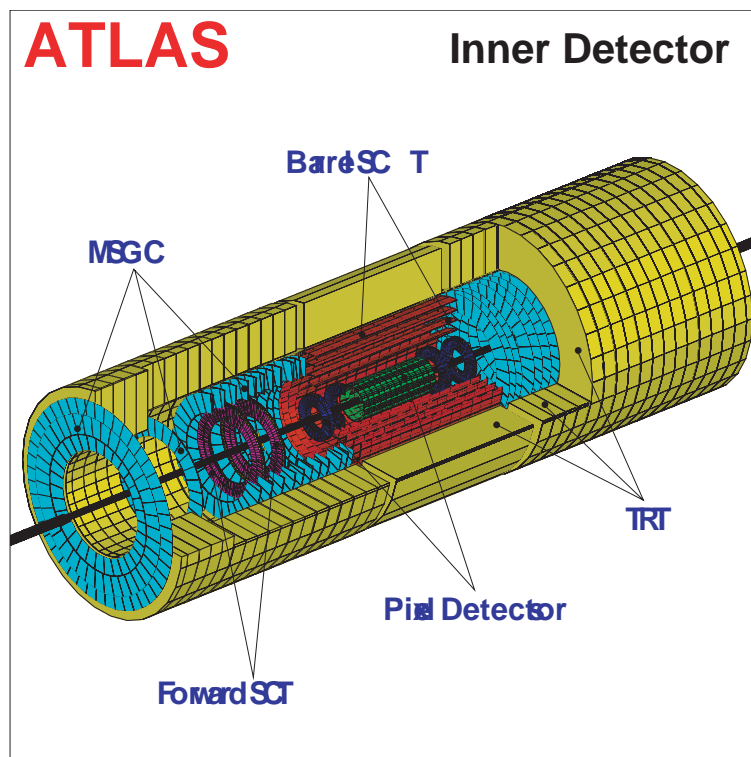


図 2.8: Inner Ditector

2.3.2 カロリメータ

カロリメータの役割は主に電子や γ 線、ジェットなどのエネルギー、位置、角度の測定である。ATLAS実験に使用されるカロリメータは大きく2つに分けられ計4種類あり、広い $|\eta|$ 領域をカバーし、領域によって目的にあわせて設置される。図2.9に構造を示し、以下に4種のカロリメータに関して簡単に説明する。

- 電磁カロリメータ
アコーディオン構造の鉛の吸収体と液体アルゴンからなり、放射線耐性に優れている。内側に設置されバレル/エンドキャップ領域をカバーする。電子と光子の同定に用いられる。
- ハドロンカロリメータバレル部は鉄の吸収体とタイル状のシンチレータからなるハドロンカロリメータが用いられ、放射線強度がより高いエンドキャップ部は、銅の吸収体と液体アルゴンからなるバドロンカロリメータが用いられる。さらに放射線強度の高いフォワード部は銅とタングステンの吸収体と液体アルゴンからなるハドロンカロリメータが用いられる。これらは電磁カロリメータの外側に設置され、ハドロンの同定、エネルギー測定、ジェットの再構成などを行う。

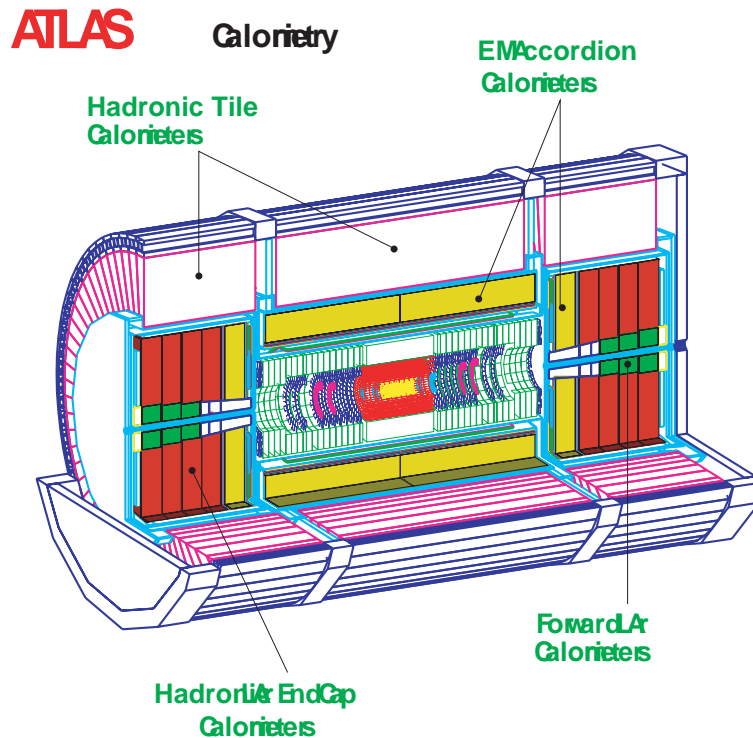


図 2.9: Calorimeter

2.3.3 ミューオン・スペクトロメータ

ATLAS 実験において重要なイベントの終状態にはほとんどの場合レプトンが含まれ、その中でもミュオンは物質の透過力が高いため内部にある検出器などの影響を受けずに外側で検出できる。そのためミュオン・スペクトロメータはもっとも外側に置かれている。また Inner Detector とは別に超伝導真空トロイダルマグネットを内包し、これによって ϕ 方向の磁場が生じ、この磁場によって曲げられたミュオンの曲率を測定することにより高精度に運動量を測定することができる。

ミュオン検出器は運動量の精密測定のための MDT(Monitored Drift Tube)、CSC(Cathode Strip Chamber) と、トリガーのための RPC(Resistive Plate Chamber)、TGC(Thin Gap Chamber) の計 4 種類からなる。これらの配置を図 2.10 に示し、またトリガー用検出器の配置を図 2.11 に載せ、以下にこれらの特徴を記述する。

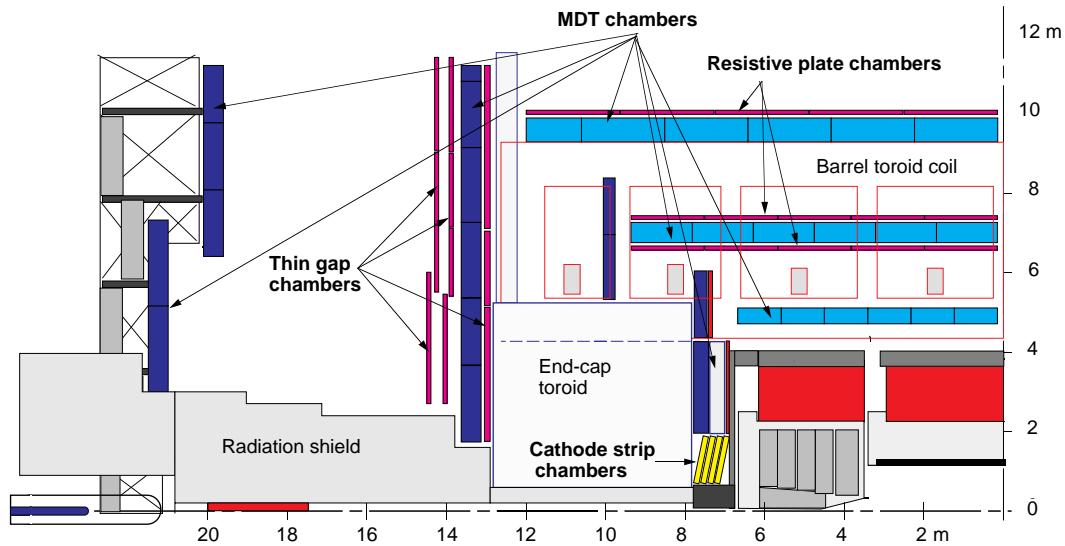


図 2.10: Muon Spectrometer $r - z$ 断面図

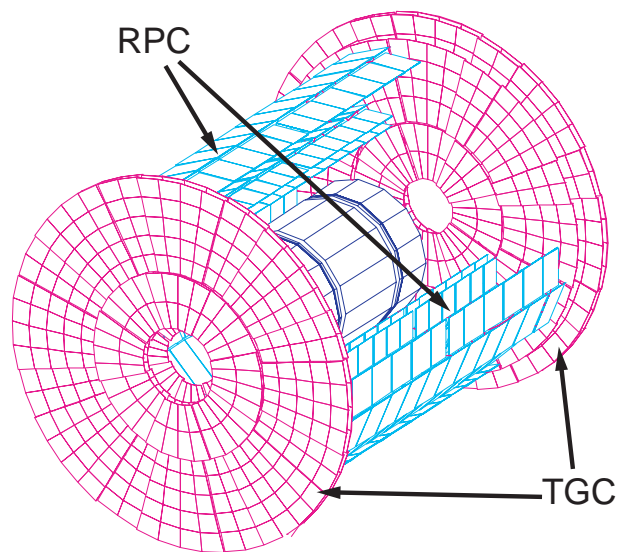


図 2.11: トリガー用 Muon Detector

- MDT(Monitored Drift Tube)

MDT は図 2.12 にあるようにパレル部、エンドキャップ部の広いラピディティ領域をカバーし、 r - z 方向成分を精密に測定する。このため位置とドリフト時間の線形性に優れており、最大ドリフト時間は $500ns$ 、位置分解能は $80\mu m$ となっている。構造はチューブ径 $30mm$ 、ワイヤー径 $50\mu m$ のドリフトチューブを積層したものであり、チューブ内は $Ar(91\%)N_2(4\%)CH_4(5\%)$ の混合ガスが 3 気圧で封入され、ワイヤーには $3270V$ の電圧がかけられている。図 2.12 に MDT の構造を示す。

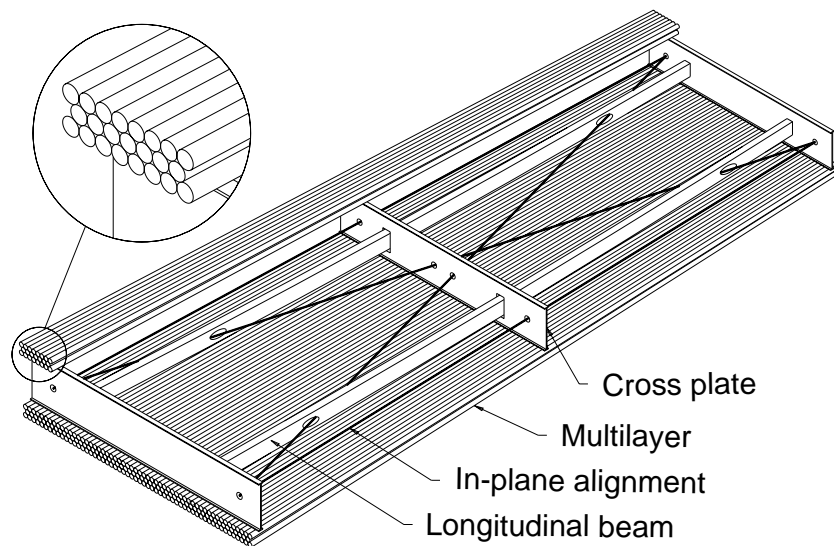


図 2.12: MDT の構造

- CSC(Cathode Strip Chamber)

CSC は放射線の多い高ラピディティ領域 $|\eta| \geq 2$ に置かれ、運動量精密測定を行うカソードストリップ読み出し用の MWPC(Multi-Wire Proportional Chamber) である。ドリフト時間は $30ns$ 以下であり、位置分解能は隣接するストリップ間の電荷の重心をとることにより $60\mu m$ になる。構造はワイヤー間隔が $2.54mm$ 、ストリップ間隔が $5.08mm$ でワイヤーとストリップが垂直に切っているのが特徴であり、封入するガスは $Ar(30\%)CO_2(50\%)CF_4(20\%)$ を用いてバックグラウンド中性子の感度を下げている。図 2.13 に CSC の構造を示す。

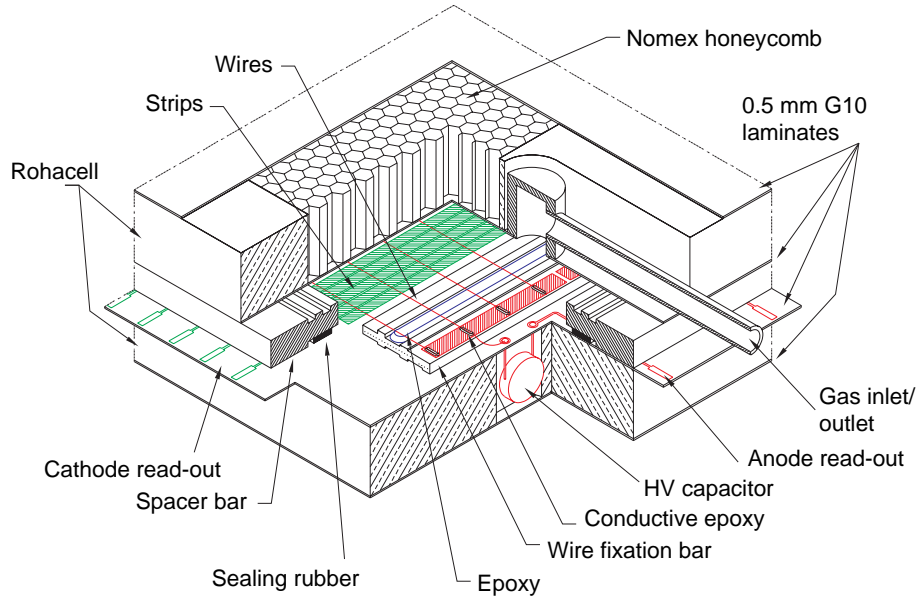


図 2.13: CSC の断面図

- RPC(Resistive Plate Chamber)

RPC はバレル部 ($-\eta-|1.1$) に設置され、 $r-z$ 方向、 $r-\phi$ 方向の運動量を測定しトリガー判定に用いられる。トリガー判定に用いられるため、時間分解能は LHC のバンチクロッシング周期の $24.95ns$ より短い $1.5ns$ となっている。構造はストリップを用いた検出器を 2 層に重ねた構造で、ストリップ間隔は $30.0\sim 39.5mm$ である。ガスが不燃性の $C_2H_2F_4(97\%)C_4H_{10}(3\%)$ を用いている。図 2.14 に RPC の構造を示す。

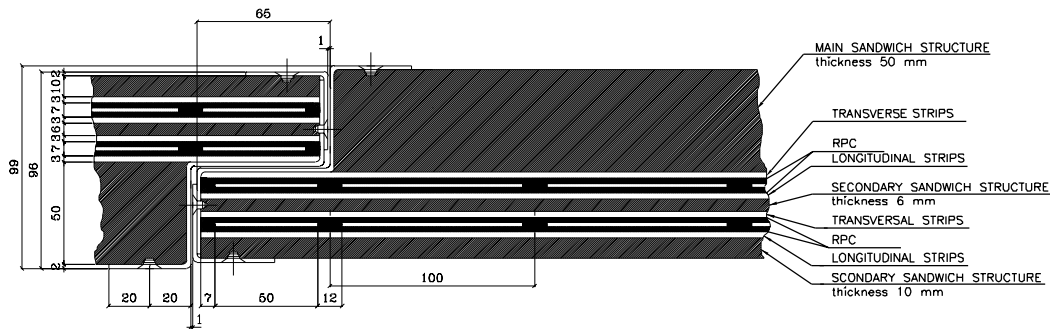


図 2.14: RPC の断面図

- TGC(Thin Gap Chamber)

TGCはエンドキャップ部の $1 < |\eta| < 2.7$ に設置され、RPCと同様 $r-z$ 方向、 $r-\phi$ 方向の運動量を測定しトリガー判定に用いられる。時間分解能は $25ns$ のゲート幅で99%以上の検出効率である。構造はワイヤー径が $50\mu m$ 、ストリップ間隔が $14.6 \sim 49.1mm$ のMWPCの一種であるが、ワイヤー同士の間隔 ($1.8mm$) がワイヤーとカソード面の間隔 ($1.4mm$) よりも狭くなっているのが特徴である。(図2.15参照) ガスは $CO_2(55\%)/n\text{-pentane}(45\%)$ を用いており、pentaneを混合することで紫外線を吸収し、放電を防いでいる。

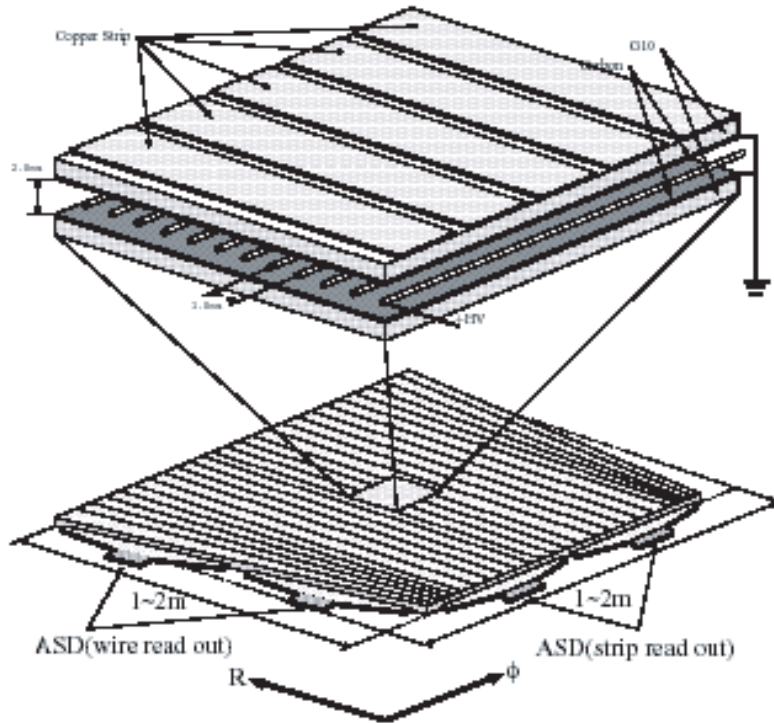


図 2.15: TGC の構造 [4]

2.3.4 マグネットシステム

ATLAS 検出器のマグネットシステムは大きく分けて 2 種類の超伝導マグネットから構成される。一つは Inner Detector を包み込んで設置されるセントラル・ソレノイドマグネットで、もう一方はカロリメータの外側に設置されるトロイダルマグネットである。このトロイダルマグネットはバレル部とエンドキャップ部で異なるマグネットが設置され、これらは共に 8 つのコイルがビーム軸に対して 8 回対称に配置されている。全体の大きさは長さ 26m、直径 20m におよぶ。図 2.16 にマグネットシステム全体図を載せる。図 2.17、図 2.18 に各ラピディティ η における積分磁場強度とトロイダルマグネットの $r-\phi$ 平面での磁束を示す。ここでトロイダルマグネットの磁場は不均一性を避けられないので r 方向成分も存在することがわかる。このためミュオンの運動量測定には ϕ 方向も考慮する必要がある。

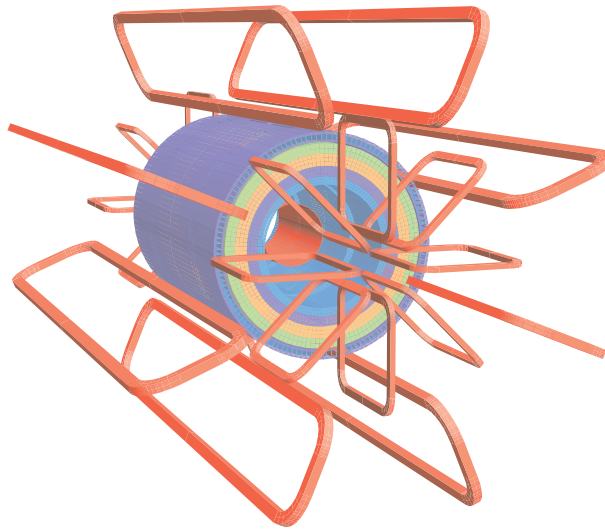


図 2.16: マグネットシステム

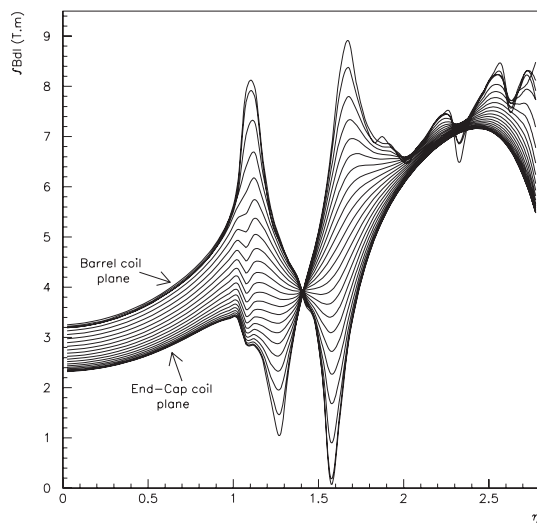


図 2.17: 積分磁場強度

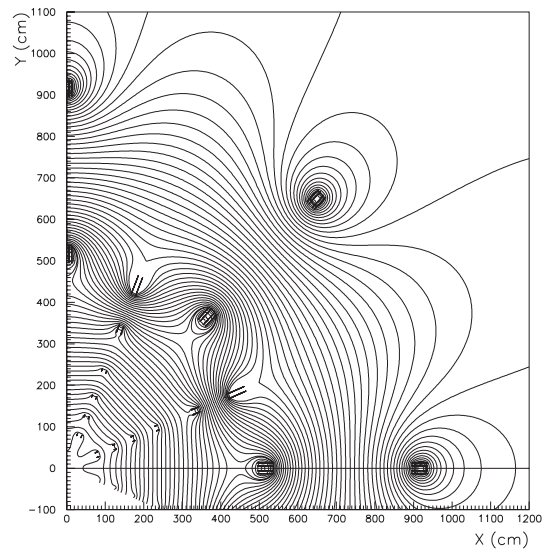


図 2.18: トロイダルマグネットの磁束 ($R-\phi$ 平面)

第3章 ATLAS トリガーシステム及びDAQシステム

ATLAS 実験では 40.08MHz でバンチクロッシングが起き、一回のバンチクロッシングで平均 23 回の陽子陽子衝突が起こると予測されており、これらから約 1GHz のレートでイベントが生じると見積もられている。しかし、これらすべてのイベントが必要とされる重要なイベントというわけではない。そこでこの膨大なイベントから重要なイベントを選定し、効率的に、なおかつ正確にデータを入手する必要がある。このため、段階的にレートを下げていき、最終的に 100Hz 程度までレートを落として記録する。ここではトリガーシステム及びDAQシステムの全体の流れを述べる。

3.1 トリガーシステム

図 3.1 に示すように、ATLAS 実験では 3 段階のトリガーレベルを設けている。その 3 段階とは順にレベル 1、レベル 2、イベントフィルタと呼ばれ、段階的に処理されてレートが下げられる。そして最終的には 100Hz までレートが落とされることとなる。以下にそれぞれのレベルに関して述べる。

- レベル 1 トリガー

レベル 1 トリガーはトリガー用カロリメータとトリガー用ミュオン検出器 (RPC、TGC) からの情報によってトリガー判定が行われる。カロリメータからはエネルギー情報が与えられ、ミュオン検出器からは位置情報と運動量の情報が与えられる。これらの情報は CTP(Central Trigger Processor) に集められてトリガー判定 (レベル 1 アクセプト;L1A) が行われ、TTC(Timing, Trigger and Control distribution system) を通して全ての検出器に送られる。このときインナーディテクターからの情報は、情報量が多く処理に時間がかかるためここでは使われない。カロリメータからの情報も位置精度を落として使われている。

ここでレベル 1 トリガー判定のレートは最大 75kHz であり、これによって 40.08MHz からイベントレートが 75kHz へと落とされる。このためレベル 1 トリガーの処理は衝突からフロントエンドのエレクトロニクスへレベル 1 トリガー判定を送るまで (この処理時間をレイテンシーと呼ぶ) を $2.0\mu s + 0.5\mu s$ 以内に行う必要がある。ここでレベル 1 アクセプトは最大 75kHz だが、各検出器の読み出し系は 100kHz まで耐えられなければならないことになっている。

またレベル 1 トリガーは他にも重要な役割を担っており、それはバンチ識別を確実に行うということである。 [3]

- レベル 2 トリガー

レベル 2 トリガーはレベル 1 トリガーの情報により選定された RoI(Region of Interest) と呼ばれる領域のみの情報を用いて行われる。RoI とは大きい運動量 (High pT) を持ったジェット、電子、ミュオンなどの粒子が検出された領域である。RoI のみの情報を用いることによって効率的な処理が可能になる。レベル 2 トリガーではレベル 1 トリガーでもちいたカロリメータ、ミュオン検出器以外にその他のミュオン検出器やインナーディテクターなどからの RoI 情報を加えて判定を行う。これによ

り測定の精度が上がり、より重要なイベントの選別が可能となる。レベル2トリガーのレイテンシーは最大で $10ms$ で、レートにすると約 $1kHz$ まで落とされることになる。

- イベントフィルタ

イベントフィルタでは全ての検出器の完全な情報を用いて行われるオンライン最後のトリガー判定である。ここでは通常はオフラインで使用しているアルゴリズムや手法をオンラインに導入している。イベントフィルタのレイテンシーは $1s$ で、レートにすると $100Hz$ まで落とされる。

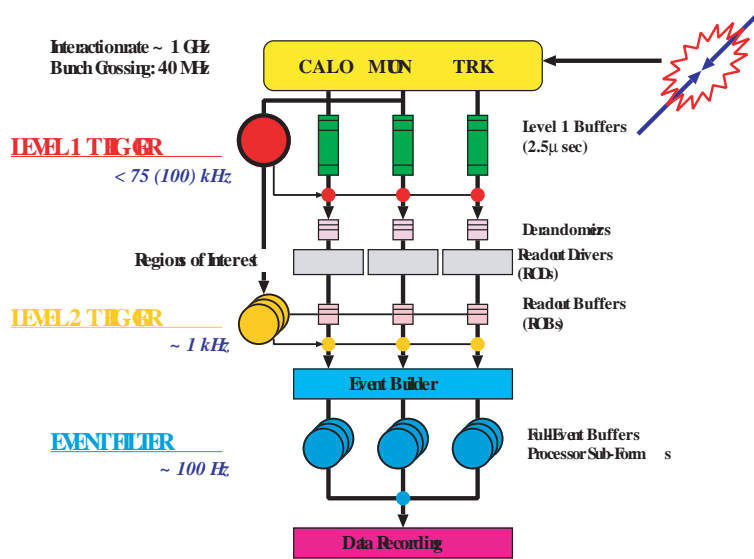


図 3.1: Trigger Levels

3.2 DAQ システム

図 3.1 のように DAQ システムはトリガー段階とともに異なっている。

まずレベル1トリガー判定時だが、レベル1アクセプトが与えられるまでそれぞれの検出器はすべてのデータをレベル1バッファと呼ばれるパイプラインメモリに保存しておき、レベル1アクセプト (L1A) が与えられると該当するデータのみをデランダムマイザーと呼ばれる一種のFIFOに送る。ここでレベル1バッファは少なくとも $2.5\mu\text{s}$ データを保持できる深さに作られている。デランダムマイザーは不規則に与えられるL1Aによって送られてきたデータを一時的に蓄え、データがROD(Read Out Driver)から読み出されるまで保持する。データはRODに送られる前に圧縮され、そこにバンチクロッシングカウンター、L1Aカウンターの値がそれぞれバンチクロッシングID、レベル1IDとして付加される。RODは複数のデランダムマイザーからデータを収集し、TTCから送られてくるバンチクロッシングナンバー、イベントナンバーと集められたデータのバンチクロッシングID、レベル1IDを比較し、同じイベントのデータをまとめ新たなデータフォーマットに変換する。RODでまとめられたデータはROB(Read Out Buffer)へと送られ、レベル2トリガーが与えられるまでそこで保持される。レベル2トリガーが与えられるとROBのデータはイベントビルダーを介してイベントフィルタへと送られる。イベントフィルタで処理を行った後、最終的にイベントレートは 100 Hz ほどになり、1イベント当たりのデータ容量は平均して 1 Mbyte となる。つまり最終的にデータは毎秒およそ 100 Mbyte 記録されることになる。

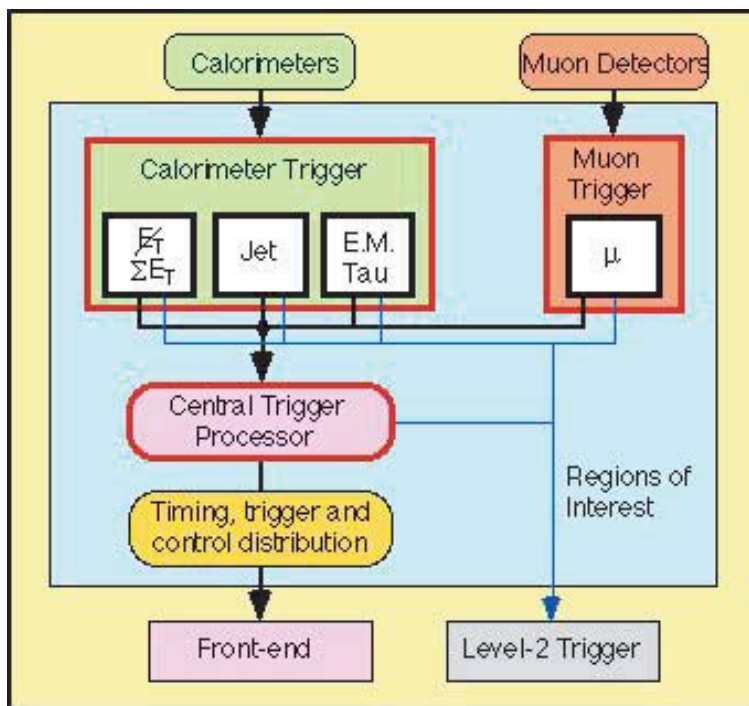


図 3.2: レベル 1 トリガーの流れ

3.3 レベル 1 ミューオントリガーシステム

図 3.2 に示すように、レベル 1 トリガーは大きく分けるとレベル 1 カロリメータトリガー、レベル 1 ミューオントリガー、CTP(Central Trigger Processor)、TTC(Timing, Trigger and Control distribution) から構成されている。トリガーの流れとしてはカロリメータ側からの電子、光子、ジェット、消失横運動量¹ (以下 pT) などの情報と、ミューオン側からのミューオンの Hihgt-pT 情報が独立に処理され、その情報が CTP へと送られトリガー判定が行われる。そこで L1A が出されると L1A 信号は TTC を経由してフロントエンドのエレクトロニクスに分配送信され、必要なデータの読み出し信号に使われる。このデータは RoI としてレベル 2 へと送られ、レベル 2 トリガー判定のための情報として使われる。

レベル 1 ミューオントリガーは、バレル部をカバーする RPC とエンドキャップ部をカバーする TGC から構成されている。レベル 1 ミューオントリガーの r-z 断面を図 3.3 に示す。

以下にレベル 1 ミューオントリガーの関連装置に関して述べる。

- MUCTPI(Muon Central Trigger Processor Interface)
これは CTP の前段階におかれ、RPC と TGC からの情報を処理し CTP へ送信する。RPC、TGC はセクタと呼ばれる単位ごとに pT の大きなミューオンの候補を送信してくる。MUCTPI はこれらの候補を収集して、境界部分でのダブルカウントが起きないように候補を選定し、その結果得られたミューオントラック候補を CTP へと送る。
- CTP(Central Trigger Processor)
レベル 1 カロリメータトリガーとレベル 1 ミューオントリガーからの情報を統合し、最終的なレベル 1 トリガー判定を行う。レベル 1 カロリメータトリガーとレベル 1 ミューオントリガーからは、それ

¹ ビーム軸に対して垂直方向の運動量のことである。

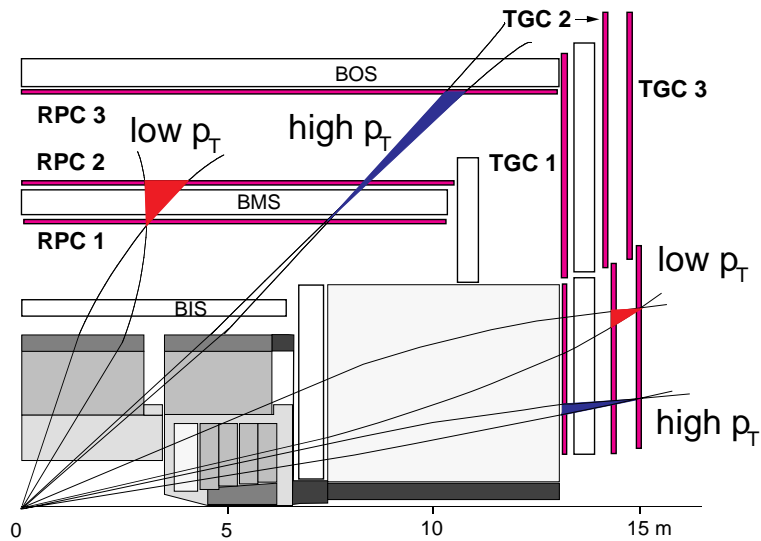


図 3.3: レベル 1 ミューオントリガー

ぞれ検出した粒子の中で高い運動量を持つものを、いくつかのレベルに分別した情報が送られてくる。CTP では入力情報が同一バンチにそろえられ、あらかじめ用意されているトリガー判定条件と比較することにより最終的なレベル 1 トリガー判定が行われ L1A が出力される。判定条件は最大 96 個設定することができる。出力された L1A は TTC へと送信される。

- TTC(Timing, Trigger and Control distribution)

TTC は ATLAS 検出器のフロントエンドエレクトロニクスの同期を正しくとるために信号を分配するシステムである。TTC は TTCvi [5] と TTCrx [6] からなり、TTCvi は VME² クレートに入る VME モジュールで LHC のクロックや CTP からの L1A などを受信し、これらをファンアウトするモジュールへ送信する。TTCrx はフロントエンドのエレクトロニクスに搭載され、TTCvi からの信号をエレクトロニクスへと分配する。図 3.4、図 3.5 に TTCvi Board と TTCrx Board の写真を載せる。以下表 3.1 に主な TTC の信号を記す。



図 3.4: TTCvi Board



図 3.5: TTCrx Board

²標準化されたバス規格である。汎用性が高く、通信速度が速い(約 50MB/s)。複数のマスターが共存できる非同期のバスで DMA 転送をサポートしている。[7]

LHC Clock	LHC のバンチクロッシングに同期した 40.08MHz のクロック
L1A	CTP から送られてくるレベル 1 アクセプト信号
BCR	Bunch Counter Reset
ECR	Event Counter Reset
BCID	Bunch Crossing ID
EVID	Event ID

表 3.1: TTC の主要信号

第4章 前後方エンドキャップ・ミュオントリガー・システム

この章ではエンドキャップ・ミュオントリガー・システムの構造、機能、設置環境について述べる。このシステムは形状の異なる数種の TGC と機能の異なる数種類のエレクトロニクスから構成されている。以下これらを順に説明していく。

4.1 TGC の構造

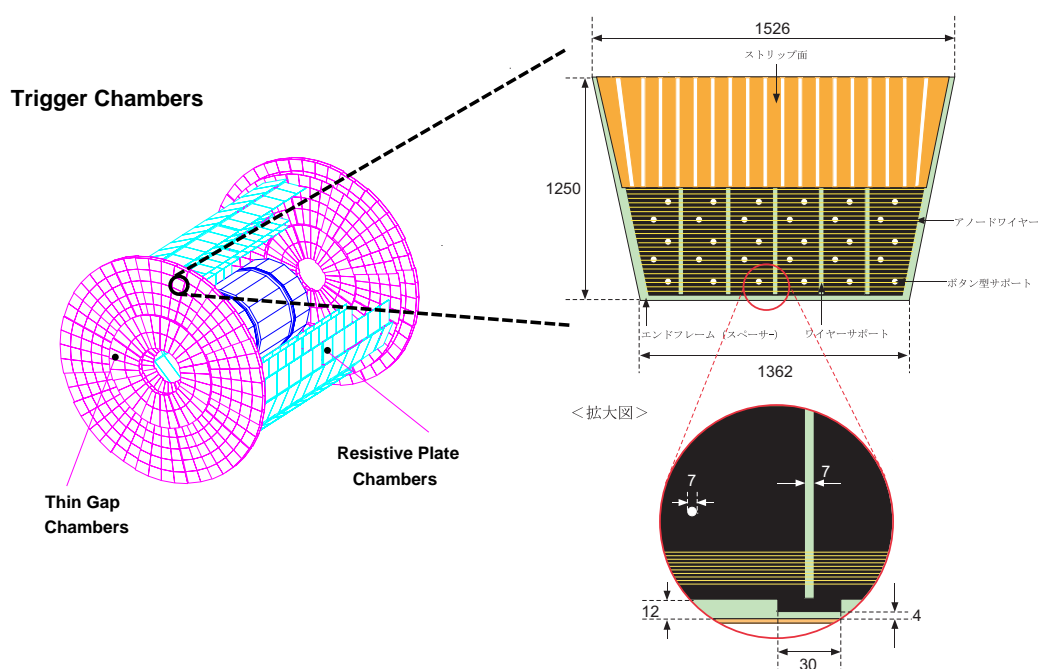


図 4.1: TGC の配置とその構造

第 2 章で述べたように TGC はエンドキャップ部の $1 < |\eta| < 2.7$ に設置される検出器で、その構造及び原理は MWPC (Multi-Wire Proportional Chamber) と同様である。ただし、TGC は一般の MWPC とは異なる特徴がある。それは狭い Wire と Wire の間隔 (1.8mm) より Wire と Strip の間隔 (1.4mm) がさらに狭いという点である。TGC の配置と構造を図 4.1 に示す。Wire と Wire の間隔が狭いのは電子のドリフト時間を短くし、バンチクロッシング間隔 (25ns) に対応できるようにするためである。Wire と Strip の間隔が狭いのは陽イオンのドリフト距離を短くし、粒子が高レートで入射してきても検出効率を落とさないようにするためである。

TGC1 台は、長さが 1m ~ 2m で台形状の形をしており、数種類が用意される。次に動作原理に関して述べる。TGC では Wire に 3.1kV の高電圧が印加され、チェンバー内部には CO_2 と n-pentane の混合ガスが封入

されている。ここで n-pentane が混合されているのは、ガス中の電離した電子と陽イオンが再結合した場合に紫外線が発生し、その紫外線によって再びガスの電離が生じ放電に至る現象を防ぐためである。n-pentane は紫外線を吸収する性質を持っているので選ばれた。この中を荷電粒子が通過すると通過経路に沿ってガスが電離され、電離された電子はアノード・カソード間の電場によってアノードへと向かう。このとき Wire 近傍では電場が大きく、電子は周辺のガスをさらに電離し、2 次電子が生成される。このような連鎖反応により電子が増幅し、アノードの Wire からシグナルとして読み出される。またカソード側は高抵抗のカーボンが塗布されており、同時にカソード側の Strip に誘起される電荷をシグナルとして読み出す。このように TGC はアノード Wire と Strip で 2 次元の位置読み出しが可能になっている。ここでアノード Wire は r 方向、Strip は ϕ 方向の位置読み出しにあたる。また Wire は 4 ~ 20 本が一組になり信号読み出しをする。TGC の主要パラメータを表 4.1 に示し、断面図を図 4.2 に示す。

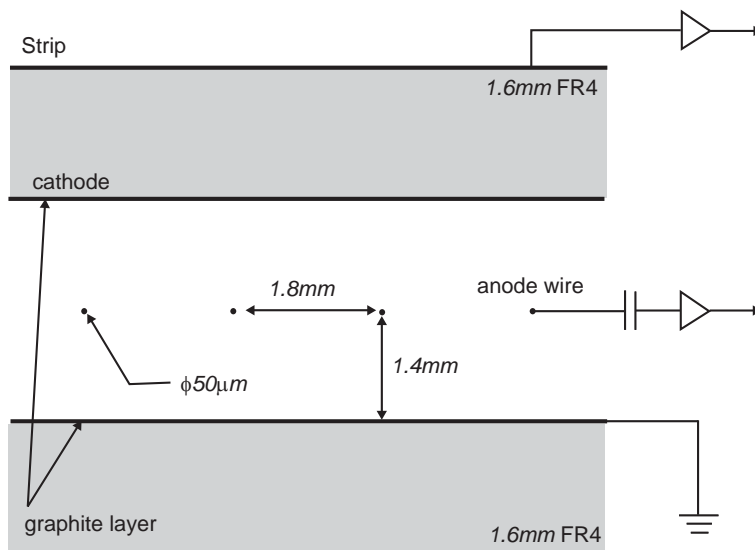


図 4.2: TGC の断面図

Wire・Strip 間隔	1.4mm	カーボン面抵抗	$\sim 1\text{M}\Omega$
Wire・Wire 間隔	1.8mm	電圧	3.1kV
Wire 径	50 μm	封入ガス	$\text{CO}_2(55\%), n - \text{C}_5\text{H}_{12}(45\%)$
Wire 張力	350gw	ガスゲイン	10^6

表 4.1: TGC の主要パラメータ

TGC は実際に使用する場合、単独ではなく 2 枚もしくは 3 枚を組み合わせて使用する。(図 4.3 参照) 2 枚のものは Doublet、3 枚のものは Triplet と呼ばれ、Triplet は Wire3 層、Strip2 層、Doublet は Wire2 層、Strip2 層から読み出しが行われる。ATLAS 実験に使用される TGC は約 3700 枚でチャンネル数は r 方向約 20 万、 ϕ 方向約 10 万の計 32 万チャンネルにもものぼる。

TGC は高レートで起こるイベントに対してロスすることなく検出できるように、Wire 間隔や Wire-Strip 間隔などが工夫されているわけだが、ここでその結果に関して述べる。

ATLAS 実験では 25ns で衝突が繰り返されるので、25ns 以内に TGC は信号を伝えなければならない。この信号の伝わる時間は、ミューオンが通過する位置や入射する角度によってドリフト時間が異なるため、さま

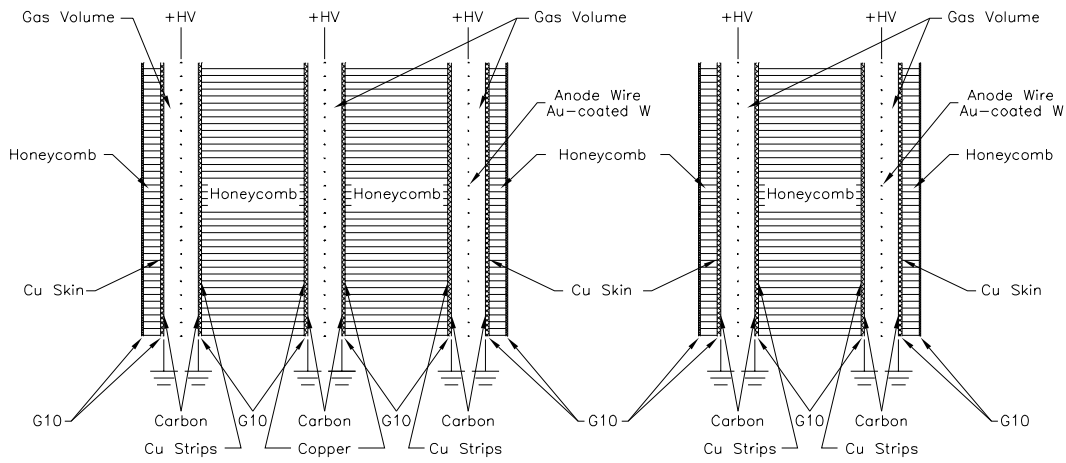


図 4.3: TGC Triplet、Doublet 構造

さまざまな角度からの入射に関して調べる必要がある。図 4.4 は、TGC に 3GeV の π を角度を変えて入射させた場合に得られた信号を到達時間に従って分布させたものである。図から分かるように、最も早く到達する信号は 115ns であり、ここから 25ns 以内の 140ns までに信号が到達すれば同じイベントで生成されたミュオンとして同バンチと同定できる。図から 115ns から 140ns までに到達した信号は全体の 99% となっており、TGC は同じイベントのほぼ全てのミュオンを 25ns 以内に検出できるといえる。

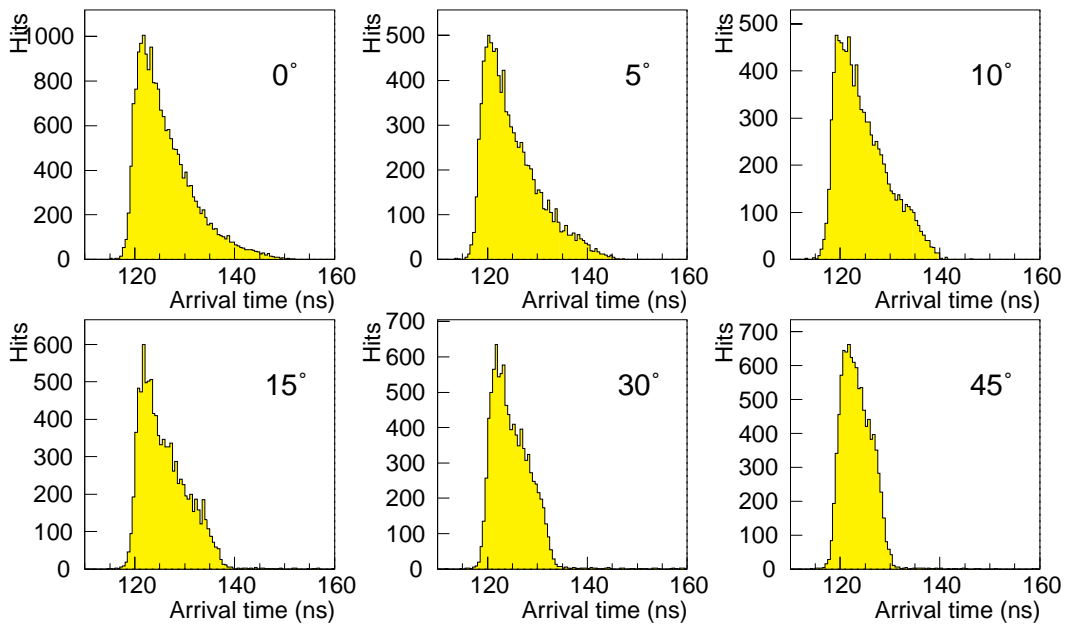


図 4.4: TGC のタイムジッター

4.2 TGCの配置

TGCはエンドキャップ部に円形状に配置される。(図4.5参照) この場合、不感領域ができないように重ね合わせられて配置され、最大でDoubletおよびTriplet4枚分のTGCが重なる場所がでてくる。

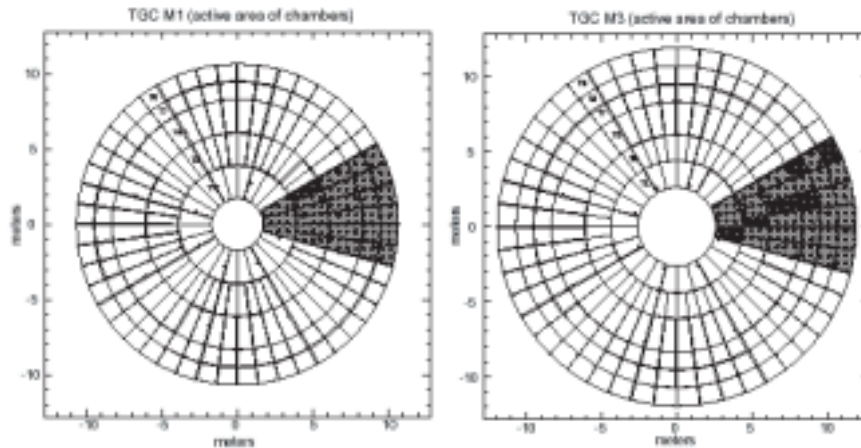


図 4.5: TGC r- ϕ 平面図

また、この円形状に配置されたものが5つ用意される。これは外側から順に、2枚構造のDoubletが2つ(M3、M2)、3枚構造のTripletが1つ(M1)でさらにトロイダルマグネットより内側に2層構造のEI(Endcap Inner)/FI(Forward Inner)が配置される。(図4.6参照) 一番外側のDoubletはピボットと呼ばれ、その内側のDoubletはミドルと呼ばれる。TGCは $1 < |\eta| < 2.7$ の領域をカバーしているが、この領域で $|\eta| < 1.9$ の領域をエンドキャップ、 $|\eta| > 1.9$ の領域をフォワードと呼ぶ。また、EI以外は ϕ 方向に対して100%の領域をカバーするが、EIは一部がトロイダルマグネットと交差する位置にあたるため、このを除いた70%の領域をカバーする。

4.3 トリガースキーム

ここではTGCのトリガースキームについて説明する。トリガースキームを説明するために、まずトリガーセクターという分割領域を解説する。まず、円形状に並べられたTGCを8等分したひとつをオクタンと呼ぶ。これを図4.7に示す。そして、エンドキャップ部($|\eta| < 1.9$)において1つのオクタンを ϕ 方向に6等分した領域、及びフォワード部($|\eta| > 1.9$)の1つのオクタンを ϕ 方向に3等分した領域をトリガーセクターと呼ぶ。このトリガーセクターの ϕ 方向の幅は1枚のTGCの幅と一致する。さらにトリガーセクターはサブセクターというセクターに分割される。サブセクターはエンドキャップ部ではr方向に37分割、 ϕ 方向に4分割されたものでフォワード部ではr方向に16分割、 ϕ 方向に4分割されたものである。1つのサブセクターは8Wireグループ、8Stripをカバーしており、トリガー処理の最小単位であり、1つのRoIでもある。

次にトリガースキームに関して説明する。衝突点で衝突が起きミュオンが生成されると、数層のTGCをミュオンが通過していき、その際に各層で信号が発生する。このとき、バックグラウンドと衝突によって生成されたミュオンを選別するために、各層での信号の同時性(コインシデンス)をもちいる。TGCは2枚(Doublet)または3枚(Triplet)構造になっているので、各層でコインシデンスをとることにより衝突点

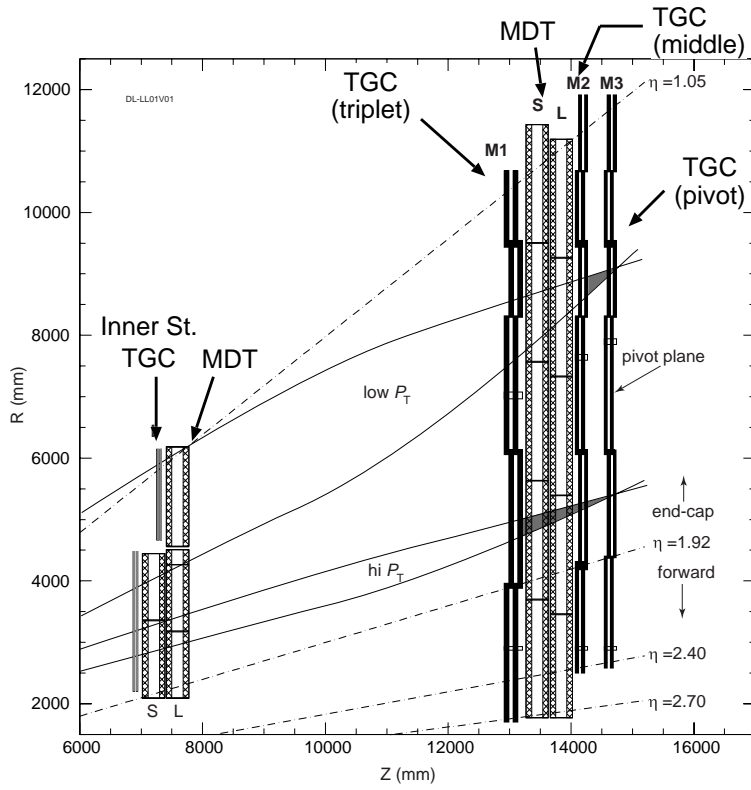


图 4.6: TGC r-z 平面图

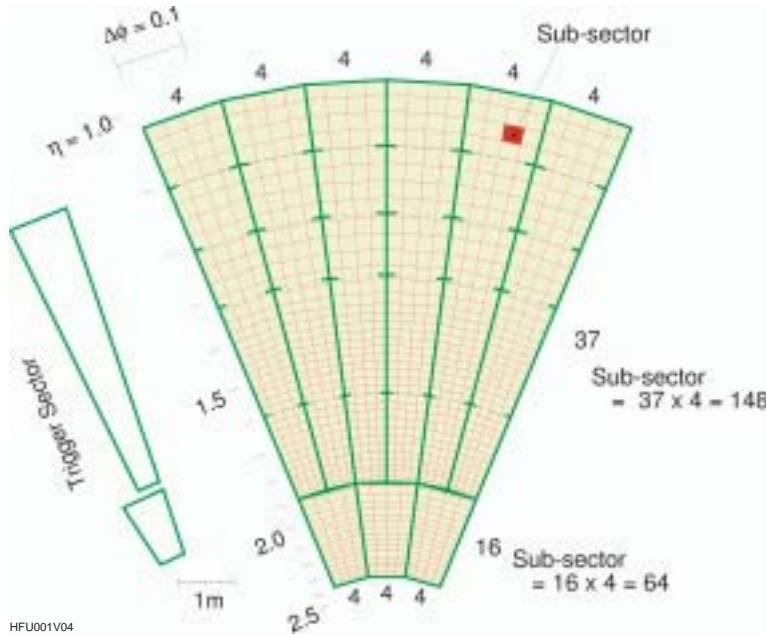


图 4.7: Sector

以外から飛来し、信号を発生させたミュオンを除去できる。具体的には、Doublet(ミドル、ピボット両方とも)では Wire、Strip とともに 4 層のうち 3 層に信号が発生した場合、ミュオンが通過したとする (3 out-of-4 コインシデンス)。Triplet では Wire が 2 out-of-3、Strip が 1 out-of-2 のコインシデンスがとられる。EI/FI では Wire、Strip とともに 1 out of 2 のコインシデンスがとられる。

このようにコインシデンスがとられ、ミュオンが通過したとみられる位置情報をもとに通過したミュオンの p_T が求められる。エンドキャップ・ミュオントリガー・システムでは Low- p_T と High- p_T という 2 つの閾値を持っている。閾値は Low- p_T は 6GeV 以上で、High- p_T は 20GeV 以上である。Low- p_T はふたつの Doublet(ミドル、ピボット) を使用して求められ、High- p_T はふたつの Doublet と Triplet を用いて求められる。 p_T はミュオンがトロイダルマグネットによって曲げられ、その曲がり具合によって p_T の大きさが測定される。曲がり具合が小さいほど大きな p_T を持つことになる。この曲がり具合は最も外側にあるピボット Doublet を基準に求められる。ピボットのある位置にミュオンが通過した信号が現れたとき、その位置とビーム衝突点を結ぶ直線を考える。この直線は Infinite Momentum Line と呼ばれ、トロイダルマグネットによって曲げられなかった無限運動量を持つミュオンの軌跡をあらわしている。しかし、実際はそのようなミュオンは存在しないので、仮想のトラックである。チェンバー上の r - ϕ 平面において、この無限トラックからのずれ Δr 、 $\Delta\phi$ によって p_T が測定される。ミドル Doublet と Triplet では、この δr 、 $\delta\phi$ の上限の値が定められている。この上限の値は、ミドル Doublet では Low- p_T 以上の運動量を持つミュオンが曲げられたときに通過できる範囲になっており、90%の確率でこの範囲を通過したミュオンは Low- p_T 判定が下される。同様に、Triplet では High- p_T 以上の運動量を持つミュオンが曲げられたときに通過できる範囲に設定されている。この場合も 90%の確率で High- p_T を持ったミュオンがここを通過し、High- p_T 判定が下される。EI/FI の情報は、衝突点以外から飛来したミュオンを除去するために用いられる。

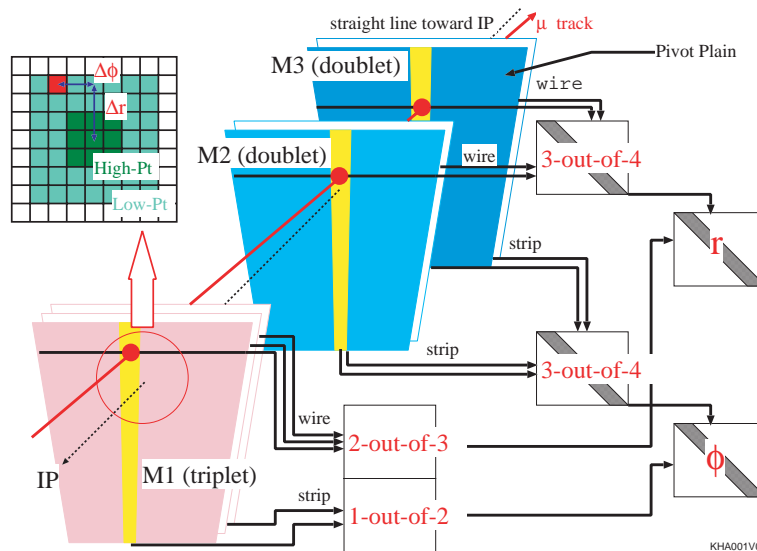


図 4.8: Trigger Scheme

4.4 エレクトロニクスシステム

TGC エレクトロニクスは 9 つのボード (モジュール) からなり、8 つの ASIC が使用される。ボード名とそのボードに搭載される ASIC 名を表 4.2 にまとめた。

これらのモジュールは、トリガー、リードアウト、コントロールという 3 つのパートに分けることができ

Board name	ASIC name
Amplifier Shaper Discriminator Board (ASD)	Amplifier Shaper Discriminator ASIC (ASD ASIC)
PS Board	Patch Panel ASIC (PP ASIC) Slave Board ASIC (SLB ASIC) JTAG Route Controller ASIC (JRC)
Service Patch Panel Board (SPP)	
High-pT Board (HPT)	High-pT ASIC (HPT ASIC) VME Protocol ASIC (VME ASIC)
Sector Logic Board (SL)	
Star Switch Board (SSW)	Star Switch rx ASIC (SSWrx ASIC) VME Protocol ASIC
ReadOut Driver Board (ROD)	
High-pT Star-switch Controller Board (HSC)	VME Protocol ASIC Primary Protocol Encoder ASIC (PPE ASIC)
Crate Control Interface Board (CCI)	

表 4.2: TGC エレクトロニクスのボードと ASIC の種類

る。各パートごとのモジュールの分類を表 4.3 に示し、図 4.9 に図示する。トリガーパートでは、データは ASD - PS Board - HPT - SL の順に流れていく。リードアウトパートでは、ASD - PS Board - SSW - ROD の順に流れていく。PS Board まではデータは共通で PS Board 上の SLB ASIC で 2 つに分かれる。コントロールパートは、DCS - CCI - HSC - SSW - PS Board、もしくは DCS - eLMB - PS Board とデータが流れていく。

トリガー	リードアウト	コントロール
HPT	SSW	SSW
SL	ROD	HSC CCI

表 4.3: パートごとのモジュールの分類表

3つのパートに関してさらに具体的に説明する。はじめに、チェンバーからの信号は ASD に入り、そこで信号の増幅、整形が行われ LVDS (Low Voltage Differential Signal)¹ レベルで PP ASIC に送られる。PP ASIC ではタイミングの調整、パンチ識別が行われ、次に SLB ASIC へと送られる。SLB ASIC に入ってきた信号はトリガー用とリードアウト用に分けられて用いられる。

ここからは、まずトリガーの流れを説明する。SLB ASIC に入ってきた信号では 6GeV 以上の pT を持つ信号が選り出す Low-pT 判定を受け、HPT へと送られる。ここまでは Doublet と Triplet は別々に処理される。HPT では Doublet と Triplet からの信号をもとに、20GeV 以上の pT を持つミューオンの信号を選り出す High-pT 判定を行い、光ファイバーを用いて信号は Sector Logic へと送られる。ここまでは r 方向 (Wire) と ϕ 方向 (Strip) の処理は独立に行われる。Sector Logic ではじめて $r-\phi$ のコインシデンスがとられ、 r 、 ϕ 双方の同じ位置に信号があったものが選り出される。この結果は MUCTPI に送られ、RPC の情報と共に最終的なミューオントリガーの判定が行われ、その結果が CTP へと送られる。

¹低電圧の作動式信号である。中心電圧が 1.2V で、電流駆動により 100 Ω の抵抗を介して作動電圧 400mV の信号を出力する。利点は、同相ノイズに強い、省電力、高速伝送などが上げられる

次にリードアウトの流れを説明する。SLB ASIC でリードアウト用のデータはレベル1バッファに記憶され、レベル1トリガー判定後、L1A が出されたデータのみデランダムマイザーに記録される。次にデランダムマイザーから Star Switch に送られ、そこでデータが圧縮されヘッダーやフッターがつけられる。その後 ROD に送られる。ROD では複数の SSW からのデータを受け取り、同じイベントのデータを選び出しフォーマットして ROB へ送る。

コントロールパートは、データの流に直接的に関わるのではなくトリガーやリードアウトに使われるモジュールの動作レジスタの設定や使用されるプログラマブルな LSI への書き込みを行う。流れは、データの流とは反対に実験室外の CCI から HSC を経由し SSW、HPT に送られる。PS Board へは SSW を通じてアクセスされる。ここから各モジュールごとの説明と各プロトタイプモジュールの説明をする。

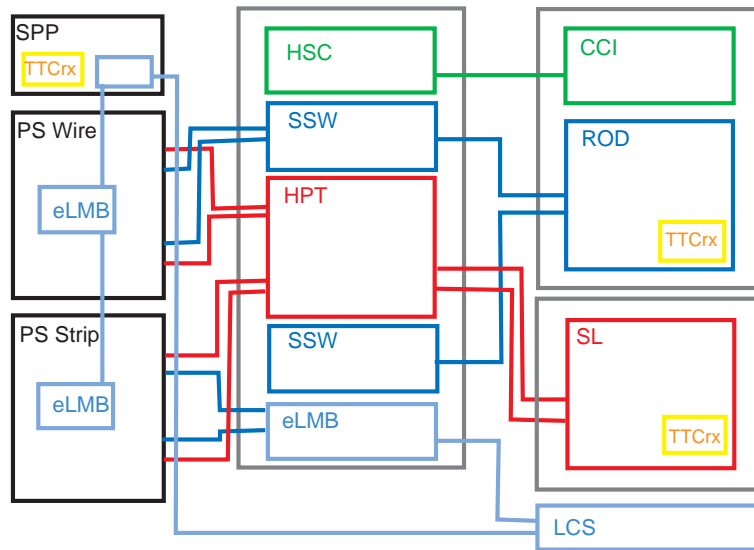


図 4.9: TGC エレクトロニクスのカテゴリ図

4.4.1 Amplifier Shaper Discriminator Board (ASD)

ASD は4チャンネルを持つ4つの ASD ASIC [8] が搭載され、1つのボードで16チャンネルを扱う。載せられる ASD ASIC はチェンバーからのアナログ信号を増幅し、ある閾値電圧を越えた信号を LVDS レベルのデジタル信号に変換する機能を持っている。また、回路系の診断やタイミング調整のためのテスト信号を出力する機能を持っており、このテスト信号のトリガーは PP ASIC から送られる。このボードはチェンバーの縁に直接設置され、閾値電源の設定や動作電源は PS ボードから供給される。

図 4.10、図 4.11 はそれぞれ ASD と ASD ASIC の写真である。ASD は既に完成している。

4.4.2 PS Board

PS Board には PP ASIC(8つ)、SLB ASIC(2つ)、JTAG Route Controller ASIC(1つ)、Detector Control System(1つ) が載せられる。PS Board では ASD からの LVDS レベルの信号を受け、PP ASIC、SLB ASIC で処理された後、再び LVDS レベルに変換されさらにシリアルライズして HPT、Star Switch へとデータを送る。以下、PS Board に搭載される各 ASIC の役割を説明していく。図 4.12 に PS Board のプロトタイプの写真を載せる。

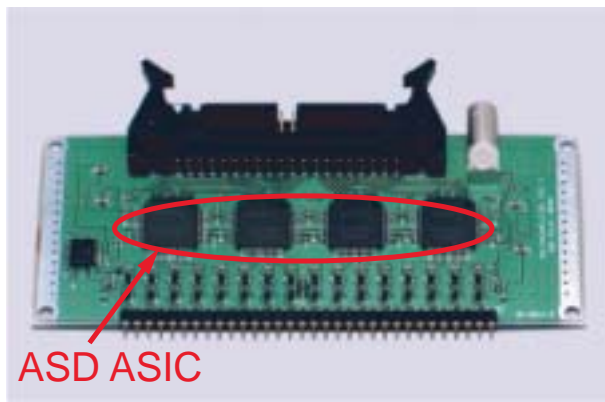


図 4.10: ASD の写真

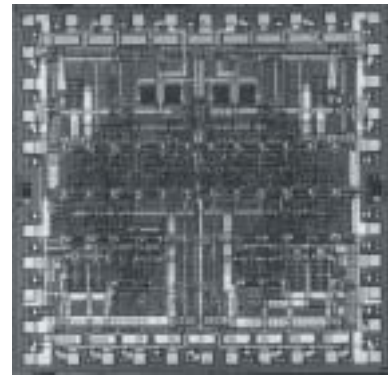


図 4.11: ASD ASIC のコアの写真

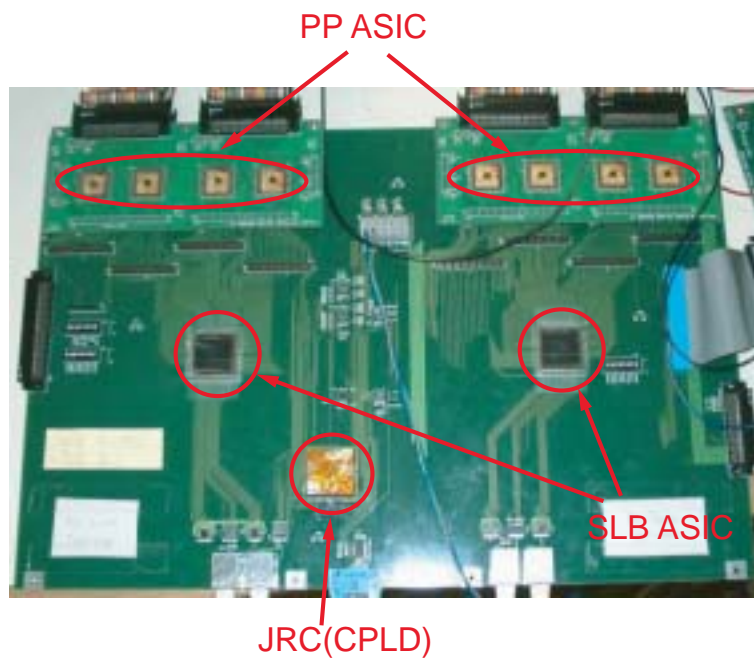


図 4.12: PS Board の写真

- Patch Panel ASIC (PP ASIC)

ASD からの LVDS 信号は、ミューオンが TGC を通過した位置や角度によって信号の到達時間が違うため、各チャンネルごとにタイミングが異なる。PP ASIC はまず、その非同期の LVDS 信号を Variable Delay 回路によって、0ns から 25ns の範囲でディレイをかけることによりタイミングの調整を行う。このディレイは 0.78ns 単位で調整が可能となっている。また、LVDS レベルの信号を受信するための、LVDS レシーバーも組み込まれている。次に、タイミング調整された信号は BCID(バンチクロッシング ID) 回路に入り、TTC から供給されているクロックと同期がとられバンチ識別が行われる。このあ

と、TGCの重なった部分のダブルカウントを防ぐため、OR ロジックを通して SLB ASIC に送られる。最新の PP ASIC は、1 つで 32 チャンネルの入出力を持つ。その他、PP ASIC には入力信号にマスクをかける機能や ASD ヘテスト信号のトリガーをかける機能などが備わっている。図 4.13 に PP ASIC のコアの写真を載せる。

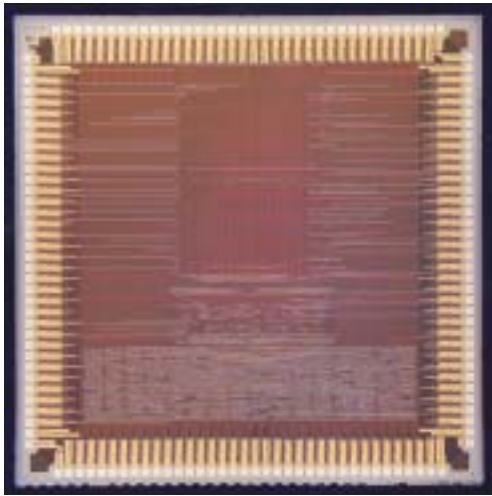


図 4.13: PP ASIC のコアの写真

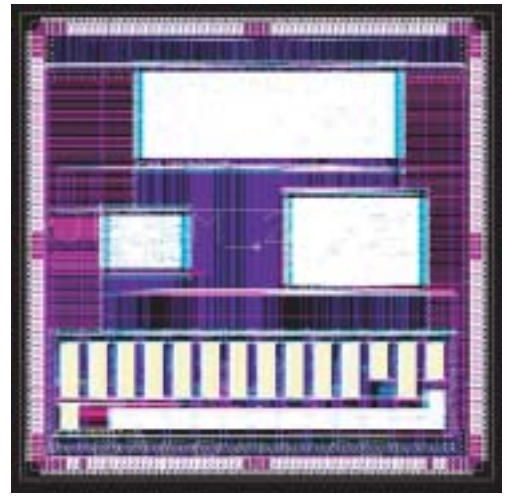


図 4.14: SLB ASIC のレイヤー図

- Slave Board ASIC (SLB ASIC)

SLB ASIC は 6GeV 以上の pT をもつミュオンの信号を選別する Low- pT 判定を行う。SLB ASIC の内部の機能は大きくトリガーパートとリードアウトパートという 2 つのパートから構成されている。まず、トリガーパートに関して述べる。SLB ASIC の段階では Doublet、Triplet と Wire、Strip は独立に処理されるため、トリガーパートは Doublet-Wire、Doublet-Strip、Triplet-Wire、Triplet-Strip、EI/FI 用の設定が必要となる。Doublet ではピボットとミドルの双方を用い、計 4 層を利用して 3 out-of 4 のコインシデンスをとる (図 4.15 参照)。このときピボットを基準にしてミドル上でのずれによって 6GeV 以上の pT 判定が行われる。Doublet では Wire も Strip も同様の処理になる。Triplet では Doublet と独立に処理されるため、ピボットの情報が無いので pT は測定されず位置のみが求める。また Doublet とは異なり Wire と Strip の処理が異なる。Wire は 2 out-of 3 のコインシデンスがとられ、Strip は 1 out-of 2 のコインシデンスがとられる (図 4.16、図 4.17 参照)。EI/FI はミュオンが通過したかどうかのヒット情報のみがとられる (図 4.18 参照)。

また、トリガーパートにはディレイ回路とマスク機能もつけられている。

次に、リードアウトパートに関して述べる。リードアウトパートはデータを保存しておく役目を担っており、その中身はレベル 1 バッファとデランダムマイザーからなる。まずデータはレベル 1 バッファという幅 212bit、深さ 128 段のシフトレジスタに蓄えられる。212bit の内訳は入力データ 160bit、トリガーパートの出力 40bit、パンチカウンター値 12bit となっている。このデータは CTP からの L1A (レベル 1 アクセプト) が与えられるまでの間保持され、L1A が与えられると該当するデータとその前後 1 パンチずつの 3 パンチ分がイベントカウンター 4bit を付け加えられてデランダムマイザーへと送られる。デランダムマイザーに送られると 3 つのパンチは別々にシリアル変換され、前後に Start Bit、Stop Bit が付け加えられて SSW に LVDS レベルで送られる。図 4.14 に SLB ASIC のレイヤー図を載せる。

- JTAG Route Controller (JRC)

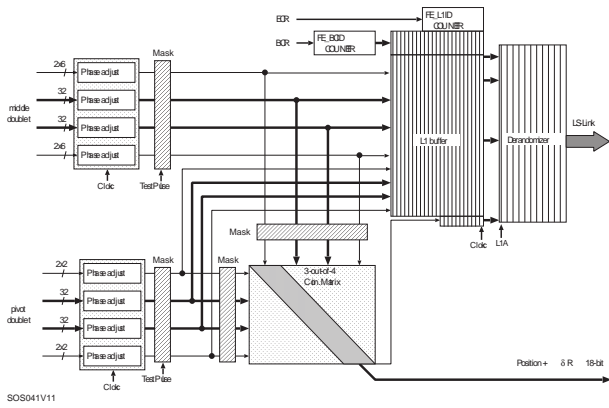


図 4.15: SLB 3 out-of 4 ブロック図

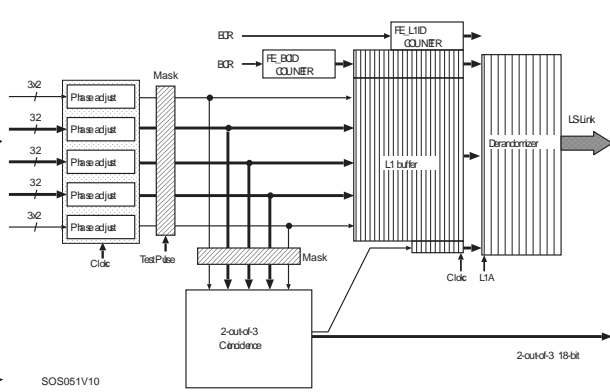


図 4.16: SLB 2 out-of 3 ブロック図

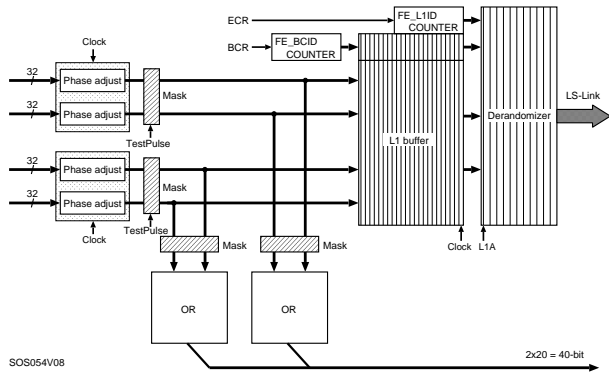


図 4.17: SLB 1 out-of 2 ブロック図

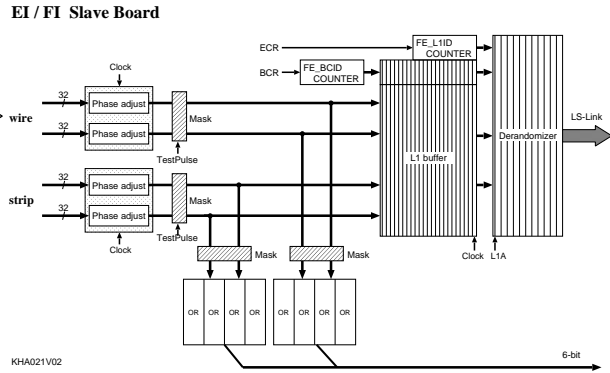


図 4.18: SLB EI/FI ブロック図

JRC は JTAG² によって PPASIC や SLB ASIC の設定を行う際に、そのルート进行调整する役割を持っている [9]。JRC は 2 つ JTAG 入力ポートと 7 つの JTAG 出力ポートを備えている。2 つの入力ポートのうち一方は SSW からで、もう一方は DCS(後述)からのものである。7 つの出力ポートは 4 つが PPASIC へのもので、2 つが SLB ASIC へのものである。図 4.19 のように、C1 JTAG が JRC 内のスイッチをコントロールすることで、Q1 ~ Q7 のポートのうち 1 つを選択して、D1 JTAG の信号をそのまま出力する。入力は D1 JTAG と D2 JTAG の 2 系統があるが、一方がアクセスしているとき他方がアクセスしようとした場合は、最初にアクセスしていたほうが優先度が高く、あとにアクセスしたほうはどこにも繋がらない状態になっている。図 4.20 に JRC ASIC のコアの写真を載せる。

²LSI の入出力端子にレジスタを設け、データの入出りをモニターする規格で、LSI のデバック用に開発されたものである。我々はこの機能を利用して、LSI の動作命令、機能設定を行っている。

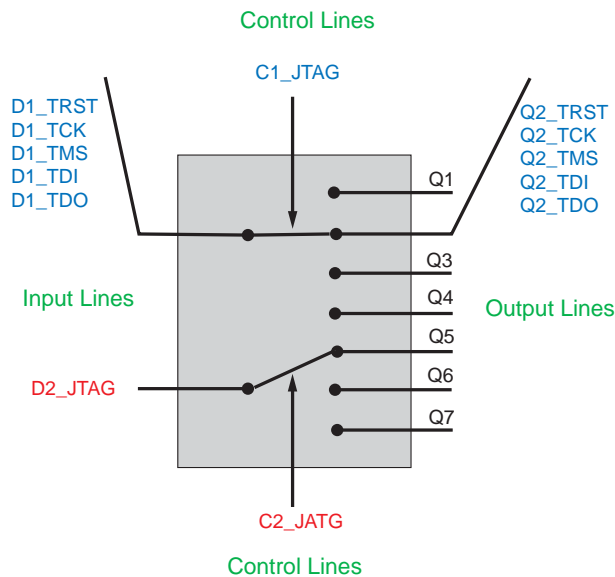


図 4.19: JRC の内部模式図

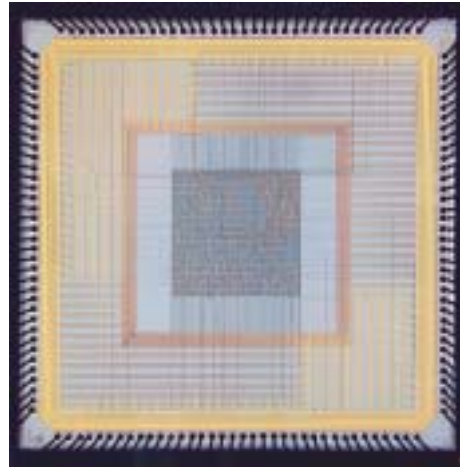


図 4.20: JRC ASIC のコアの写真

- embeded Local Monitor Box (eLMB)

eLMB は Detector Control System (DCS) の中で TGC を扱うサブシステムである。DCS は ATLAS 実験における検出器の監視、診断のためのシステムである。eLMB は温度モニターやエレクトロニクスの DC 電源の電流と電圧のモニター、TGC の内部ガスの流れや圧力などをモニターを行う。これらのモニターなどの eLMB の動作は CAN バスによって制御される。

ここで PS Board のプロトタイプに関して幾つか述べる。プロトタイプは 2001 年の秋に作られた。そのため、PP ASIC は 2001 年 6 月に submit された旧バージョンのものが搭載されている。旧バージョンは入出力が各 16 チャンネルとなっており、PS Board 上に 16 個搭載された。SLB は 2000 年 12 月に submit されたものを使用している。JRC はこのときまだ ASIC 化されていなかったため、CPLD³ によって実現されている。

4.4.3 Service Patch Panel Board (SPP)

SPP には TTCrx が載せられたメザニンボードが載せられ、TTC から LHC クロック、L1A、BCR、ECR などの信号を受け取り、それらの信号を PS Board に分配する。図 4.21 にプロトタイプの写真を載せる。

³Complex Programmable Logic Device の略 内部に回路構成用のメモリを内蔵し、自由に回路構成を行うことができる IC

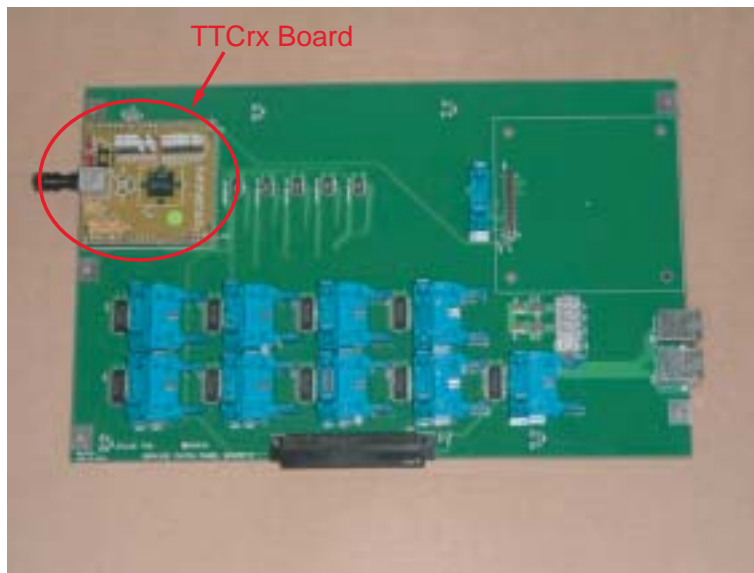


図 4.21: SPP の写真

4.4.4 High-pT Board (HPT)

HPT は 20GeV 以上の pT をもつミュオンを選別する High-pT 判定を行う。HPT はまず、SLB から送られてきた LVDS レベルのシリアライズされたデータを、パラレルのデータに変換する。HPT では、PS Board まで独立に処理されてきた Doublet と Triplet のデータを統合して High-pT 判定を行う。Triplet は 2 つの Doublet よりも内側で、しかも Doublet 同士の間隔よりも離れた位置に設置されているため、Triplet を用いることによりトロイダルマグネットによってあまり曲げられることがなかった大きな pT をもつミュオン信号を選別できる。HPT では Wire と Strip は独立に処理が行われる。データは G-Link Tx によってシリアライズされ、オプティカル信号に変換され、オプティカル・ファイバーによって 90~100m 離れた実験室外のカウンター・ルーム (USA15) にある SL に送信される。HPT はエンドキャップ領域 Wire 用とエンドキャップ領域 Strip 用のものと、フォワード領域用の計 3 種類作られる。どのタイプにも HPT ASIC は 4 つ搭載される。プロトタイプは Forward 領域のデータを処理するためのものである。HPT はひとつのボードに 4 つの HPT ASIC が搭載され、10 個の SLB からの入力を受けることができる。図 4.22、図 4.23 に HPT と HPT ASIC の写真を載せ、図 4.24、図 4.25 に HPT の Wire、Strip のブロック図を載せる。

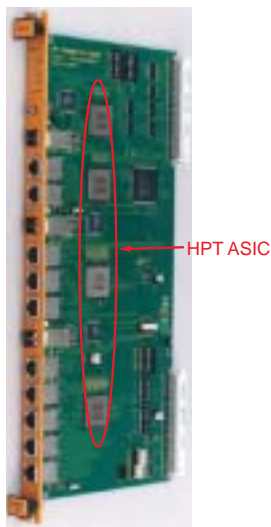


図 4.22: HPT の写真

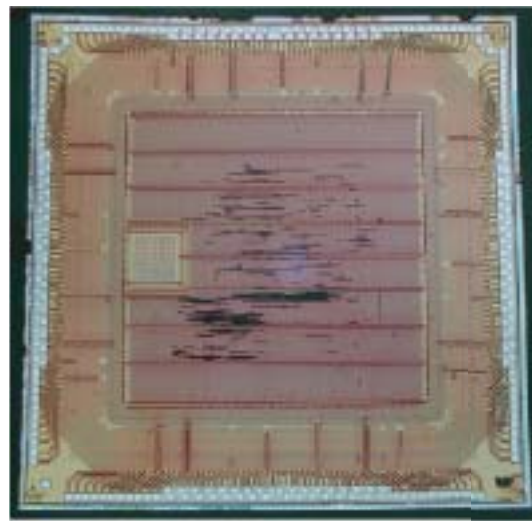


図 4.23: HPT ASIC のコアの写真

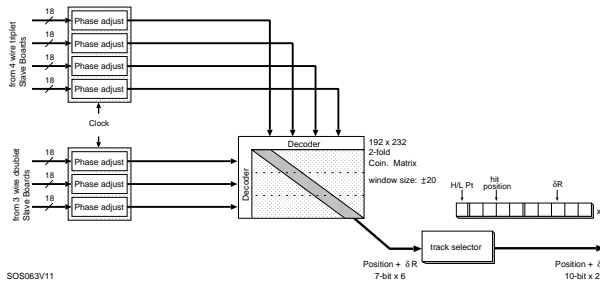


図 4.24: HPT Wire ブロック図

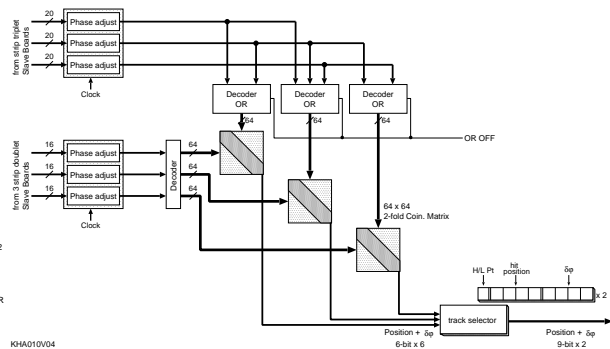


図 4.25: HPT Strip ブロック図

4.4.5 Sector Logic Board (SL)

SL は TGC エレクトロニクスシステムのトリガーのデータが最終的に集められるモジュールである。SL は Wire(r) と Strip(ϕ) のコインシデンスをとる。まず SL は、HPT から送られてきたシリアル化されているオプティカル信号を受け取り、電気信号に変換した後パラレル変換をする。SL には 6 段階の閾値が設けられていて、その情報は Look Up Table(LUT) に格納しておき、パイプライン処理によって入力されたデータと比較することによって判定を行う。6 段階のうち 3 段階は Low-pT 用で、残り 3 段階が High-pT 用となっている。ここではっきりと決まっている閾値は、Low-pT の 6GeV 以上と High-pT の 20GeV 以上という 2 つのみで、その他の細かい閾値の設定は、実験中に探索する粒子の種類や実験条件に応じて任意に書き換えることになる。LUT の採用はこのためである。このロジックは FPGA⁴ に書き込まれる。

図 4.26 に SL の写真を載せ、図 4.27 に SL のブロック図を載せる。

⁴Field Programmable Gate Array の略 CPLD 同様、内部に回路構成用のメモリを持ち、自由に回路構成を行うことができる。CPLD とは内部構造やプログラム素子が異なる。

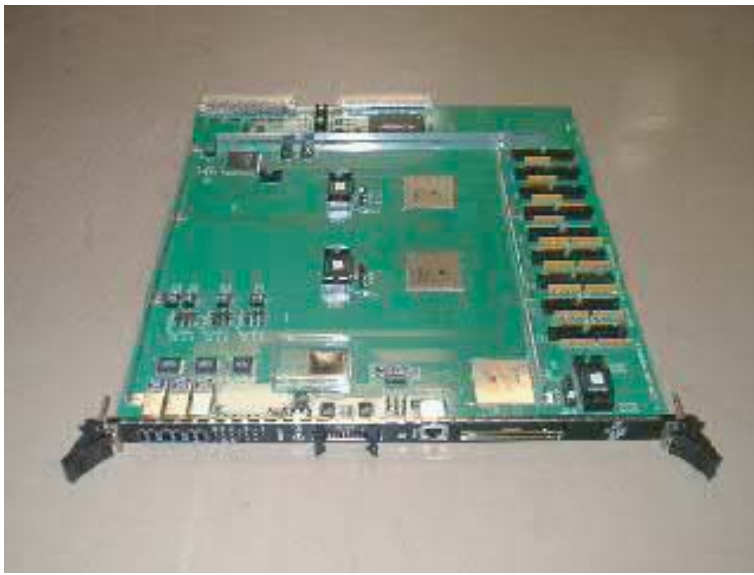


図 4.26: SL の写真

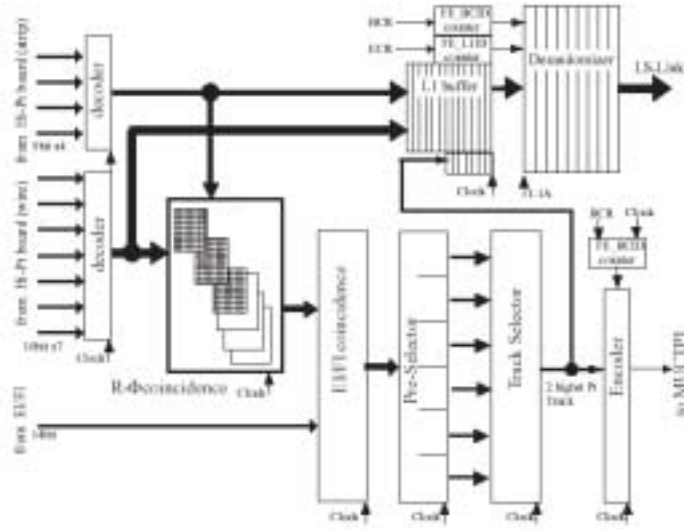


図 4.27: SL ブロック図

4.4.6 Star-Switch Board (SSW)

SSW はリードアウトにおけるデータ収集の中継のために導入され、ROD にデータを送る前にデータ量を減らすためデータ圧縮を行い、効率よい読み出しができるようにしている。データ圧縮は、まずデータを 8bit ごとに分け (これを cell という)、cell ごとにアドレスを付ける。次に、各 cell において値がゼロでないものだけ選び、cell アドレスと共に送る。SSW は最大 18 個の SLB のデータを受け取る。ボード上には SSWrx と SSWtx という IC が載せられる。SSW はまず、SLB からの LVDS レベルのシリアルライズされたデータを受け取り、それをパラレルのデータに変換する。そのデータは SSWrx(レシーバー) に送られ、データの圧縮が行われる。その後データは、SSWtx(トランスミッター) に送られ、フォーマットされる。フォーマットされたデータは G-Link Tx によってシリアルライズされ、オプティカル信号に変換されて 90~100m はなれた実験室の外にある ROD に送られる。

また SSW は、CCI - HSC 経由で送られてくる命令をもとに、PS Board 上の SLB ASIC、PP ASIC のレジスタの設定を行う。この際にプロトコルは JTAG が用いられ、信号は LVDS レベルで転送される。図 4.28 に SSW の写真を載せ、表 4.4 に SSWrx、SSWtx の機能を簡単にまとめる。



図 4.28: SSW の写真

IC 名	機能
SSWrx	データの圧縮 PS Board のコンフィギュレーション
SSWtx	データ送信

表 4.4: SSWrx、SSWtx の機能分類

4.4.7 Read Out Draiver Board (ROD)

ROD は TGC エレクトロニクスシステムの中でリードアウトのデータが最終的に集まるモジュールである。ROD は複数の SSW からシリアルライズされた圧縮データをオプティカル・ファイバーを通して受け取り、オプティカル信号を電気信号に変換した後並列の圧縮データに戻し、FIFO メモリに一時格納する。このデータをマイクロコンピュータ (SH-4) が読み出し、トリガー情報を元に同じイベントごとにまとめ、決められたフォーマットにしたがってヘッダー、フッターをつける。まとめられたデータは Slink という光信号を伝達するリンクモジュールによって ROB(Read Out Buffer) に送信される。また、ROB 以外に VME 経由でデータを取り出すことも可能である。

イベントの同定やヘッダー、フッターをつけるためには TTC からのトリガー情報が必要となるため、ROD には TTC_{Rx} が載せられたメザニンボードが搭載され、これにより TTC からの信号を受け取ることができるようになっている。

ROD は 100kHz でこれらの処理ができるように求められている。

図 4.29 に ROD の写真を載せる。



図 4.29: ROD の写真

4.4.8 High-pT Star-switch Contrller Board (HSC)

HSCはHPT、SSWと同じVMEクレート(HSCクレート)に載せられる、VMEマスターモジュールである。次節で述べるCCIとオプティカルケーブルで結ばれ、CCIからの命令を受け取る。命令を受け取ると、命令に対応した処理を行いその後CCIへ応答を返す。命令はHPTやSSWに対するもので、命令を受け取るとVMEバスを支配しスレーブモジュール(HPT、SSW)に対して命令を伝える。またVMEバス以外に独自のJTAGバスも使用されている。このためHSCには、CCIとの情報のエンコード、デコードのために2種類の機能が用意されている。ひとつはPPE(Primary Protocol Encorder)というJTAG用のもので、リセットやJTAGのコントロールを行う。もう一方はSPE(Secondary Protocol Encorder)というVME用のもので、VMEのコントロールを行う。

図4.30にHSCの写真を載せる。



図 4.30: HSC の写真

4.4.9 Crate Control Interface Board (CCI)

CCIはローカルホストからの命令を受け取り、命令専用レジスタに格納したあとHSCへと送信する。一方HSCからの応答は、応答専用レジスタに格納されローカルホストが読み出す。これら以外にも状態監視用のレジスタやVME優先割り込み用のレジスタが用意され、これらはVME経由でアクセスすることが可能である。

図4.31にCCIの写真を載せる。

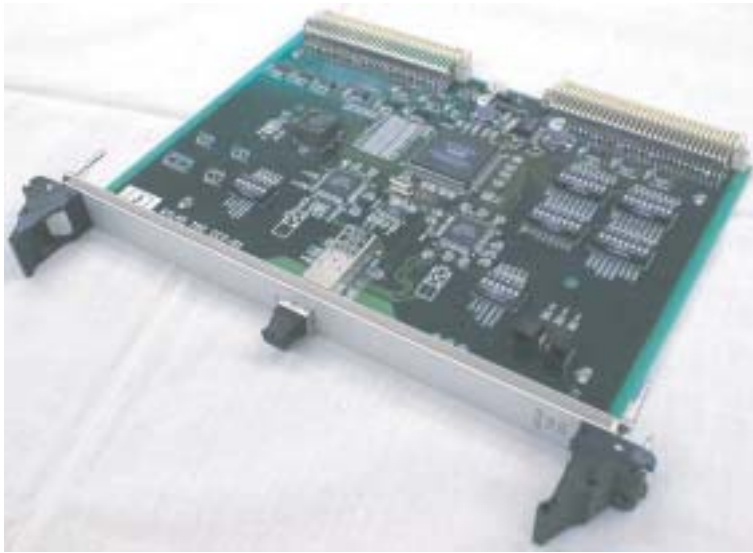
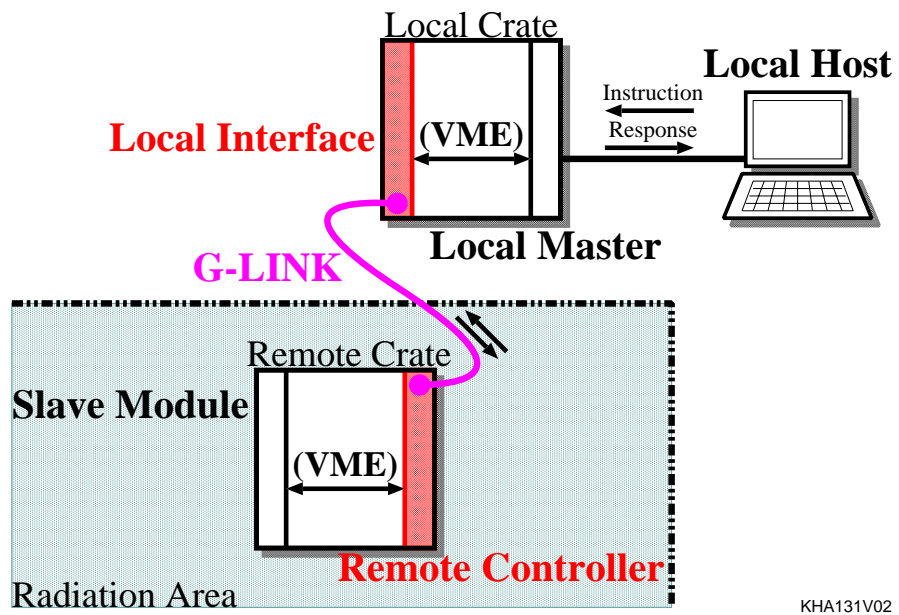


図 4.31: CCI の写真



KHA131V02

図 4.32: HSC-CCI の接続

4.5 設置環境

TGC エレクトロニクスは大きく 3 つの場所に分けられて設置される。まず 1 つは TGC 上、もしくは不感領域ができないように TGC が重ねられたときにできる場所に設置される。このスペースは Doublet もしくは Triplet4 枚分の厚さになり、衝突点から見て裏側にあたる。TGC に直接設置されるのは ASD で TGC の側面に直接信号を受け取るように設置される。TGC の裏には PS パックと呼ばれるものが設置され、PS パックの中には PS Board(PP、SLB、JRC、eLMB) と SPP (TTCrx) が設置される。PS パックは 1 つのオクタントに 3 台設置され、1 つの PS パックの中には PS ボードが 27 枚 (Doublet17 枚、Triplet10 枚) と SPP が 1 枚置かれる。

もう 1 つの場所は、TGC を支えるピックウィールという構造体の外縁に設置される。ここには HSC クレートと呼ばれる VME クレートがおかれ、そこには HSC、HPT、SSW が搭載される。HSC クレートは 1 つのオクタントに 3 台設置される。ここまでが UXA15 と呼ばれる実験ホール内である。

最後の場所は実験ホールから 90~100m 離れた外にある USA15 というコントロールルームに置かれる。ここには VME64x クレートが置かれ、CCI、SL、ROD が搭載される。

UXA15 内に設置されるモジュールは強い放射線環境下に置かれるため、放射線耐性が求められる。そのため主要な IC は、一般に放射線に強いとされる ASIC で作られている。図 4.33 に設置位置を示す。

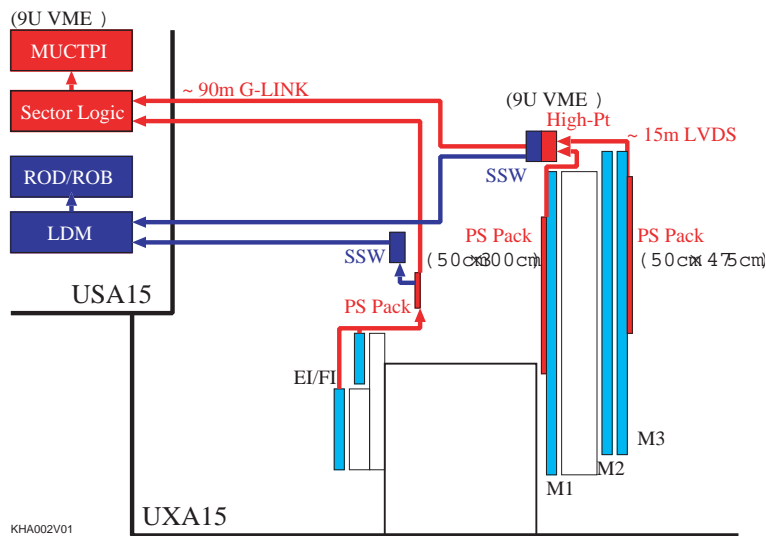


図 4.33: トリガーエレクトロニクスの配置

4.5.1 放射線の影響

実験室内に置かれる TGC エレクトロニクスはエンドキャップ部 (ビーム衝突点から離れている) とはいえ、高い放射線環境下に置かれる。放射線が半導体デバイスに与える影響としては TID、NIEL、SEE が考えられる。以下にこの 3 点に関して簡単に説明する。

- TID (Total Ionising Dose)

X 線、 γ 線によるイオン化の影響によるもので、積算的電離エネルギー損失を評価した量のことを指す。この現象は、入射粒子により半導体素子のゲート電極 - ウェル間にある酸化絶縁膜中で電離が引き起こされることにより起こる。これにより電源電流や入力電流の増加が引き起こされる。

- NIEL(Non Ionising Energy Loss)

入射粒子の非電離的過程によるエネルギー損失を積算的に評価した量を指す。半導体素子について考えると、1MeV 程度の中性子によって引き起こされる影響に置き換えられる。

- SEE(Single Event Effect)

単発の荷電粒子によって引き起こされるもので、粒子の通過により発生する電荷が大きい場合に回路内部に影響を与えるものである。この現象は積算吸収量には依存せず、照射時間が増すと発生確率が高くなるだけの偶発的なものである。また、この現象によって引き起こされる回路への影響は、メモリの誤動作というソフトエラーとデバイス自体の破壊にまで及ぶハードエラーとに分けられる。

ソフトエラーとしては SEU(Single Event Upset) という現象があり、これはメモリの記憶情報が反転される現象である。ハードエラーとしては SEL(Single Event Latch-up)、SEGR(Single Event Gate Rupture)、SEB(Single Event Burnout) がある。これらはメモリ以外の部分で発生し、電極間が導通状態になり大量の電流が流れることにより素子が損傷してしまう現象である。

ATLAS 実験における放射線レベルは、RHA(Radiation Hardness Assurance) グループ⁵ によって行われたシミュレーションによって得られている [10] [12]。RHA によって提示された、TGC エレクトロニクスが置かれる環境 (Triplet (Z,R)=1280~1290cm, 715~1180cm Doublet (Z,R)=1470~1480cm,680~1180cm) での放射線量は表 4.5 のようになっている [13]。SRL(Simulated Radiation Levels) とは検出器が設置される場所における放射線レベルのシミュレーション値である。TID、NIEL、SEE ごとに見積もられている。

	SRL _{TID} [Gy]	SRL _{NIEL} [/cm ²] (equivalent 1MeV neutron)	SRL _{SEE} [/cm ²] (E _{hadron} < 20MeV)
	10 年間の積算線量	10 年間のフルエンス	10 年間のフルエンス
Triplet	2.27	2.58 × 10 ¹⁰	6.54 × 10 ⁹
Doublet	2.49	1.42 × 10 ¹⁰	4.53 × 10 ⁹

表 4.5: シミュレーションによる放射線量の見積もり (RHA グループ)

表 4.5 で提示された値は不確定要素が多く、同時に安全係数 (SF) が用意されている。安全係数の種類とその値を表 4.6 に載せた [14]。

安全係数名	説明	TID	NEIL	SEE
SF _{sim}	SRL のシミュレーション誤差	3.5	5	5
SF _{ldr}	低線量率 (Low Dose Rate) での長時間照射の効果	5	1	1
SF _{lot}	チップ製造時のロット間で生じる違いの効果	4	2	2

表 4.6: 安全係数

これらの安全係数を考慮した耐放射線基準値 (RTC:Radiation Tolerance Criteria) は、SRL と安全係数をかけ合わせたものから導かれる。以上のことから、TGC エレクトロニクスにおける RTC は以下ようになる。

$$RTC_{TID} = (2.49) \times 3.5 \times 5 \times 4 \approx 2.0 \times 10^2 [\text{Gy}]$$

⁵ATLAS 実験において使用されるエレクトロニクスや従事する人々に対し、その安全性の確保を目的として活動している。また、LHC が 10 年間の運転する際の放射線量を見積もり、エレクトロニクスなどが安定して動作するような基準を設け、安全性を確かめている。

$$RTC_{NIEL} = (2.58 \times 10^{10}) \times 5 \times 1 \times 2 \approx 3.0 \times 10^{11} [/\text{cm}^2]$$

$$RTC_{SEE} = (6.54 \times 10^9) \times 5 \times 1 \times 2 \approx 7.0 \times 10^{10} [/\text{cm}^2]$$

ATLAS 実験において TGC エレクトロニクスは、上記の基準値の放射線を受けても正常に動作することが確認されて初めて使用が認められる。TID に関して、TGC エレクトロニクスに課せられている基準値は約 200Gy であるが、この値は耐放射線用ではない ASIC や FPGA でも耐えられる程度の値となっている。一方、SEE の一つである SEU に関しては絶対的な基準がなく、シミュレーションによるフルエンスが与えられているだけである。よって SEU に対する基準値は、システムの構成をもとに判断されるべきもので、TGC エレクトロニクスの設計では SEU による影響と対策を考慮する必要がある。

ここからは TGC エレクトロニクスのデバイスが、前述した放射線環境下でどの程度の頻度で SEU を被るかを簡単に見積もり、TGC エレクトロニクスの対策を述べる。

SEU の発生頻度 (SEU_p) は、ハドロンフルエンス Φ と半導体デバイスの SEU 断面積 σ が分かれば、以下の式のように計算できる。この式で T は LHC の稼働時間 10 年 $\approx 10^8 \text{sec}$ が入る。

$$SEU_p = \frac{\Phi \sigma}{T}$$

SEU 断面積はデバイスの種類に大きく依存するが、20MeV 以上のハドロンに対してはその断面積が一定となり、およそ $10^{-15} \sim 10^{-13} [\text{cm}^2/\text{bit}]$ であることが分かっている [11]。SEU の影響が問題となるのは LSI 内のパラメータ設定用のレジスタであり、例えば SLB ASIC の場合は 1 個につき 674 ビット存在する。ATLAS 検出器全体で SLB ASIC は 2880 個使用されるため、SEU の発生確率は、

$$SEU_p = \frac{10^{-14} [\text{cm}^2/\text{bit}] \times 7 \times 10^{10} [/\text{cm}^2/10\text{year}]}{10^8 [\text{sec}]} \times 674 [\text{bit}] \times 2880 \approx 1 [SEU/\text{day}]$$

となる。しかしこの見積もりは、1 オーダー程度の誤差を含んでいる可能性がある。

この見積もりから、TGC エレクトロニクスで使用される LSI の設定レジスタには、SEU 対策を講じる必要があることがわかる。そこで TGC エレクトロニクスでは、ASIC の設定レジスタは多数決論理回路 (Voting Logic) を実装することで SEU による誤動作を防止している。多数決論理回路は図 4.34 に示すような構造で、1 ビットの情報を記憶するのに 3 つのレジスタを使用し、それらの多数決をとるようになっている。つまり、情報が反転するためには、少なくとも 2 つのレジスタにおいて SEU が発生する必要がある。1 つのレジスタが反転する確率を P とすると、多数決論理回路内の 2 つのレジスタが反転する確率は $P \times P \times {}_3C_2 = 6P^2$ となる。SEU 断面積 σ は $10^{-15} \sim 10^{-13} [\text{cm}^2/\text{bit}]$ であり、 $P \propto \sigma$ だと考えると、多数決論理回路全体の断面積はその 2 乗となり、SEU の発生確率は非常に小さなものとなる。またこの多数決論理回路には、多数決が割れたことを検出する回路も搭載され、その出力は ASIC 内全てのレジスタに関して OR がとられ、読み出すことができる。この情報を定期的に監視することで、SEU による被害を排除することが可能となる。

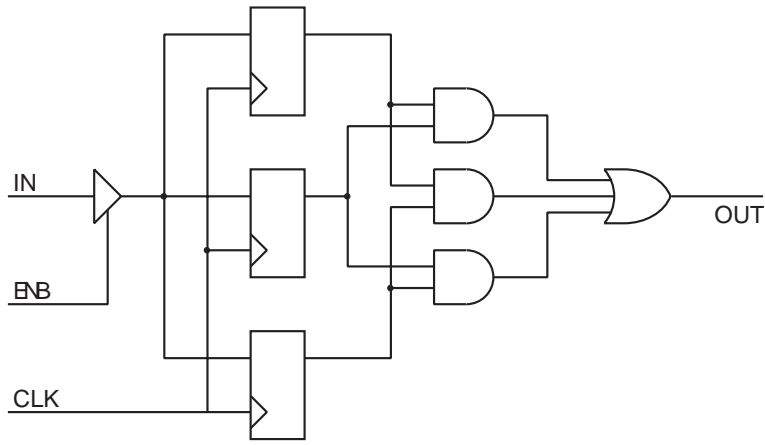


图 4.34: 多数决逻辑回路 (Voting Logic)

第5章 統合テスト

TGC エレクトロニクス・グループが開発を行ってきた全ての ASIC が完成段階をむかえ、それぞれが載せられるモジュールのプロトタイプが作られてきた。またそれらの ASIC やモジュールの単体テストも行われ、バグの修正が行われてきた [17] [18]。次の段階として考えられるテストが本実験のセットアップに近い状態で行う統合テストである。このテストの主な目的は単体のデバックが終わった各モジュールが相互に接続しても正確に動作することの確認と接続の際のリンクの確立、ディレイ設定によるデータラッチのタイミング調整、接続状態で要求されたレイテンシーを満たすかどうかチェックすることなどがあげられる。

またこのテストを効率よく行うためには、テストするための環境を構築するが必要となる。統合テスト環境のために重要になってくるのがソフトウェアである。このソフトウェアは、将来行われるチェンバーを実際に接続したビームテストや量産時のモジュールのテストなどにも使われる予定で、完成を目前にして今後更に必要なものとなり、重要性を増してくる。

統合テストシステム構築と統合テストは、昨年からはまり大別すると 2 回に分けられる。この章では統合テストの流れ、必要になるソフトウェアに関して述べる。

また、我々はこの統合テストをスライステスト (Slice Test) と呼んでいる。

5.1 テスト方法

テストはそれぞれのモジュールを接続して入出力をチェックすることにより正確に動作するか、また求められているレイテンシーを満たしているかということを行う。動作の正当性確認のためにはハードウェアのみの出力結果を検討するだけでは信頼性にかけるうえに、入力したデータがどう処理されて出力されるかを毎回確認するには手間がかかりすぎる。そのためトリガーシミュレーションソフトを開発しスライステストに活用した。トリガーシミュレーションに関しては次節で詳しく述べる。

トリガーシミュレーションによって得られた入力データ (テストベクタ) は PPG (Pulse Pattern Generator) という汎用のモジュールを用いて PS Board に入力する。(PPG に関しては次章のセットアップの説明部分で詳しく述べる)

出力は VME 経由で読み出し可能なモジュールによって読み出され、ファイルに出力される。そのデータをトリガーシミュレーションによって得られた出力と比較することによりハードウェアの動作の正当性を確認する。ここでトリガーシミュレーションに関しても正しいという保障はないため、シミュレーションとハードウェアからの出力がお互いに一致することによって双方の正当性を確認するという方法をとっている。そのため不一致が生じた場合は、互いのロジックの確認を行い、どちらに問題があるか調べ、バグをなくしていかなければならない。レイテンシーの測定はあるテストパターンを入力し、オシロスコープまたはロジックアナライザにより入力側と出力側の遅延を測ることによって行う。

5.2 トリガーシミュレーション

トリガーシミュレーションの主な目的は

1. スライステストで使用するテストベクタを生成する。

2. 各エレクトロニクスモジュールの動作、設計の妥当性を検証する。

3. SL の $R-\phi$ Coincidence Map を作る。

4. トリガー効率の評価

である [16]。

トリガーシミュレーションは大きく次の 3 つの段階に分けられる。

- Geant3 DICE

Geant3 は KeV ~ TeV 領域での高エネルギー粒子と検出器との相互作用をシミュレートし、検出器の性能評価を行うものである。

DICE は Geant3 によって陽子衝突から TGC に到着するまでのシミュレーションを行うものである。このときの TGC を通過した位置情報をトリガーシミュレーションに用いる。

- Digitize

DICE の出力は座標になっているため、これを TGC の Wire Group、Strip Number に変換する必要がある。この作業を行うのが Digitize である。

- TIME (Trigger Level 1 Muon Endcap)

Digitize で得られたデータをもとに Patch Panel、Slave Board、High-pT、Sector Logic のそれぞれのインプットとアウトプットを出力する。このデータは、モジュールの入出力信号以外にバンチクロッシング ID や位置、pT などの情報も付け加えられている。

トリガーシミュレーションの流れをまとめると図 5.1 のようになる。

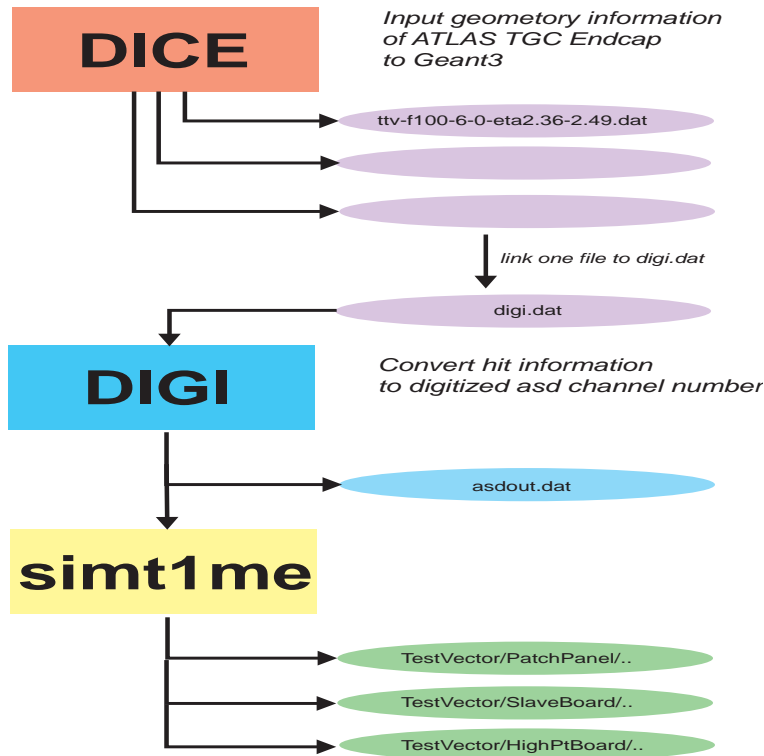


図 5.1: トリガーシミュレーションの流れ

5.3 Software

ここでは、統合テストに必要なソフトウェアに関して簡単に述べる。

5.3.1 テストベクタ変換ソフトウェア

トリガーシミュレーションから出力されたデータは、信号以外の付加情報も付け加えられており、データ信号も PP ASIC の bit 順序と異なるため、そのままの状態ではテストベクタとして PPG 経由で PP ASIC に入力することができない。そのため、信号情報だけを抜き出し、bit の順序を入れ替えるソフトウェアが必要となる。種類としては Wire Doublet 用、Wire Triplet 用、Strip Doublet 用、Strip Triplet 用の 4 種類が求められ、その他にこれらをまとめて 1 つのファイルにするものが必要となる。

5.3.2 Property Control ソフトウェア

VME に接続されるモジュールに関しては、VME 経由でモジュールの設定レジスタにアクセスできるようになっている。これにより、モジュールの初期設定やテストに応じた設定ができるようになっている。単体テストの段階では、各モジュールの開発担当者が独自にプログラムを開発し設定を行っていたが、接続テストでは全てのモジュールを 1 台の PC から同じように操作できることが望ましい。そのため、各モジュールの設定プログラムを同じ形式で記述し、それらを統合して操作しやすいように整備しなければならない。接続テスト環境ではこれらのモジュール設定用のプログラムを、Property Control プログラムと称し、まとめた。

5.3.3 Run Control ソフトウェア

接続テストでは、データの入出力をコントロールし、得られたデータが正しいものか比較しなければならない。そのためのコントロールソフトウェアが必要となる。このソフトウェアは大きく 3 つのパートから成り立つ必要がある。まず、最初にデータを受け取るモジュールをコントロールし、テストベクタを入力するパートが必要となる。次に、最後にデータが出力されるモジュールをコントロールし、得られたデータを保存するパートである。最後は、得られたデータとシミュレーションによって得られたデータとの比較を行い、その結果を出力するパートである。このソフトウェアは、Property Control ソフトウェアの上位につくり、モジュールのコントロールなどは Property Control のプログラムを使用することができるよう実装した。

5.3.4 周辺ソフトウェア

VME に接続されていないモジュール (PS Board) にも設定レジスタは設けられている。これらは JTAG を用いて設定することが可能である。この設定は、CCI-HSC-SSW と経由して PS Board へデータを送り行われる。現在のところ、SSW が完全に完成していないためこの方法を使うことができない。そのため、PT4 という FPGA(Field Programmable Gate Array) が搭載されたプログラマブルな VME 汎用モジュールを使用し、PT4 から PS Board にアクセスして設定を行うことにした。したがって、そのためのソフトウェアの開発も必要となる。

また、全てのモジュールを接続する前に段階的に接続するモジュールを増やしていった際に、ほとんどのモジュールは VME バスにデータを出力する機能を持っていないため、リンクのケーブルを受けデータ

を読み出す必要性が出てくる。この場合も PT4 を用いることによってこれを可能にした。PT4 はメザニンカードによって様々な信号レベルのデータを受け取ることができる。また FPGA で FIFO を作成することにより、受け取ったデータの保存が可能となる。このデータは VME 経由で読み出すことにより、直接 VME へのデータ出力機能を持たないモジュールの出力データも読み出すことができる。そのための PT4 のプログラムも必要となる。

周辺ソフトウェアの一覧を表 5.1 に載せる。

ソフトウェア	機能
PS Board レジスタ設定プログラム	PP ASIC、SLB ASIC のレジスタ設定
FIFO プログラム (LVDS 用)	FPGA で FIFO を実装し、FIFO から VME 経由でデータをファイルに出力する
FIFO プログラム (Optical 用)	FPGA で FIFO を実装し、FIFO から VME 経由でデータをファイルに出力する
PS Board 出力比較プログラム	出力データのフォーマットを変換し、シミュレーション出力と比較する
HPT 出力比較プログラム	出力データのフォーマットを変換し、シミュレーション出力と比較する
SL 出力比較プログラム	出力データのフォーマットを変換し、シミュレーション出力と比較する

表 5.1: 周辺ソフトウェア一覧

第6章 第1回統合テスト

2001年初めにPP ASIC、SLB ASIC、HPT ASIC がほぼ完成し、これらが載せられるPS Board、HPT が2001年秋にかけて作られた。そのため2001年9月からテスト環境の構築が始まり、テストのためのソフトウェア開発が始められた。第1回の統合テスト(以下、スライステスト)ではトリガーパートの主にr方向(Wire)のみでPP ASICからHPTまでの流れの部分を行った。また、独立してコントロールパートであるCCI-HSCの接続テストも行われた[18]。

6.1 セットアップ

セットアップ図を6.1に示し、

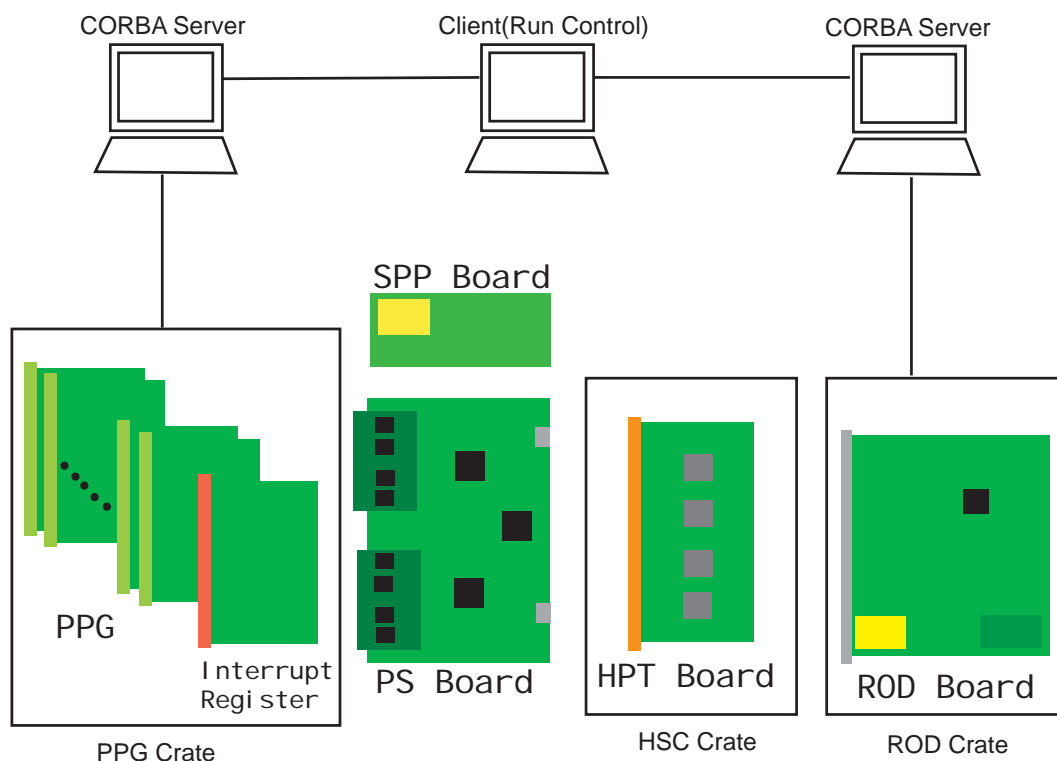


図 6.1: 第1回スライステスト セットアップ図

- PPG(Pulse Pattern Generator)
PPGは幅32bit、深さ64kのメモリを持ち、32チャンネルの出力を持つVMEモジュールである。このメモリはユーザーが自由にデータを記述でき、また必要とするデータの長さを256の倍数で最大65536まで指定できる。さらにこのデータを繰り返し出力させるリピートモードと一度だけ出力させ

るシングルモードが選択できる。また出力はメザニンボードにより LVDS レベルと TTL レベルを選択することが可能である。

テストのためにはチェンバーからの信号が必要になるが現段階でのエレクトロニクスのテストでは実際のチェンバーを用いる必要は無いため、代わりにこの PPG を用いた。今回のテストでは PPG を 9 台使用した。そのうち 1 台は TTC の代用として ECR、BCR、L1A などの入力用に使い、残りの 8 台はデータ入力用に使用した。このテストでは Bit3 を経由して PC との通信を行った。

- Interrupt Register

PPG はトリガー信号を受け取らない限り動作しないためトリガー信号を送る必要がある。トリガーをかけるには 2 つの方法があり、ひとつはトリガー用の設定レジスタによるものである。そのレジスタに書き込むことによってトリガーをかけることができる。もうひとつの方法は外部からトリガー信号を入力する方法である。前者の方法を利用すると 9 台すべて同時に書き込むことができないため Interrupt Register を利用し、外部から 9 台同時にトリガー信号を入力する方法をとった。Interrupt Register の信号は fan-out ボードにより PPG に分配される。

- PS Board(Wire)

今回のバージョンでは、16 チャンネルの PP ASIC が Doublet 側 4 つ、Triplet 側 4 つの計 8 つ搭載されている。また Doublet 用 Triplet 用の SLB ASIC が 2 つと JRC が 1 つ搭載されている。

PP ASIC は 2001 年 6 月に submit されたものである。このバージョンではデフォルトで入力にマスクがかかる使用になっており、このマスクをはずすためのプログラムとして汎用モジュールである PT4 を用いた dShake というプログラムを使用した。SLB ASIC は 2000 年 12 月に submit されたものである。JRC はこの段階でまだ ASIC 化されていなかったため CPLD で実現されている。

- LVDS Serializer/Deserialiser [19]

PS Board からのデータ転送は、LVDS Serializer によって 1 つの Chip につき 10bit のパラレル・データがシリアル・データに変換されて、LVDS レベルで転送される。一方、受信側の HPT では LVDS Deserialiser により再びパラレル・データに戻される。この転送ではカテゴリ 5 ケーブルを使用し、本実験では約 15m であるが今回のテストでは 7m のものを使用した。

- SPP Board

TTCrx の代わりに簡単なボードを製作しクロック、L1A、各リセットなどを入力できるようにした。ここから PS Board へとこれらの信号を分配した。

- HPT

Forward 用に作られたボードを用いた。HPT ASIC は年月に submit されたものを使用している。またこのモジュールも VME クレートにおさまる VME モジュールである。ただしこのモジュールは Bit3 による通信ではなく HSC-CCI を用いて通信を行う。

- CCI/HSC

この 2 つのモジュールはリモートコントロールシステムであり、今回は HPT の設定のために使用された。

- G-Link Tx/Rx [20]

ATLAS 検出器に取り付けられる HPT と実験室外のカウンター・ルーム (USA15) に設置される SL とのデータ転送は、G-Link ベースの光信号が用いられる。Tx 側では 16bit(data)+1bit(flag) の計 17bit のパラレル・データがシリアル・データに変換され、O/E コンバータ¹ で電気信号が光信号に変換され

¹Optical/Electro-Optical Converter 今回は Infinion 社製のものを使用した [21]

て転送される。Rx 側では、受信した光信号を O/E コンバータで電気信号に変換し、シリアル・データを再びパラレル・データに戻す。本実験では約 90m のオプティカル・ファイバーが使用されるが、今回のテストでは 10m のものを使用した。

- ROD

本来 HPT Board からの信号は SL に送られるが、Wire 側のためのテストのため SL に入力しても意味をなさないのと同じオプティカル・ファイバーで接続できかつデータの読み出しが可能な ROD に接続した。このモジュールも VME モジュールであり Bit3 を用いて通信を行った。

このテストでは VME モジュールである PPG、HPT、ROD は VME クレータに接続され、PPG と ROD の入るクレータはそれぞれ Bit3 を経由して PC に接続されている。HPT は HSC と同じくクレータに搭載され、ROD クレータにおさめられる CCI から HSC を経由して PC との通信を行う。このとき CCI は Bit3 を経由して PC と通信を行う。Bit3 は VME のマスターとなり、おなじクレータ上にあるモジュールと通信をする。PC は PCI バス上に取り付けられたモジュールを経由して Bit3 と通信を行う。また VME のドライバは vmehb-2.2.7 を使用し、ライブラリは vmelib-1.4 を使用した [22]。

6.2 CORBA(Common Object Request Broker Architecture)

ここでまず、第 1 回スライテストにおいて PC 間の通信手段として使用した分散オブジェクト技術である CORBA² に関して述べる。CORBA を用いることによって、遠隔地にあるオブジェクトを相手の機種、OS や動作環境、開発環境を意識することなく、あたかもそのオブジェクトが使用している PC 内に存在するかのように使用することが可能となる。

現在 C 言語、C++、Smalltalk、COBOL、Ada、Java などの言語に対応している。

このテストでは各クレータごとに PC が接続され、その PC が接続されているクレータに搭載されているモジュールをコントロールしているため、複数の PC が必要になる。複数の PC を 1 台の PC からコントロールしたいためこの CORBA を導入した。

以下 CORBA に関して簡単に説明する。CORBA は大きく ORB(Object Request Broker) というミドルウェアを中心として、CORBA サービス、CORBA ファシリティ、ドメインインタフェースというオブジェクト群と、これらを利用するアプリケーション・オブジェクトによって構成されている。それぞれは以下のような内容となっている。

- CORBA サービス

CORBA の基礎的な機能を提供するオブジェクト群である。ネーミングサービス、セキュリティサービス、トランザクションサービスなど、現在 15 種類のサービスが標準として定義されている。

- CORBA ファシリティ

アプリケーションレベルにおける共通機能を提供するものである。データベースへのアクセス、ドキュメント制御、ファイル情報の印刷機能などが提供されている。

- ドメインインタフェース

ビジネスアプリケーションレベルを単位とするサービス提供を行う。業種ごとに特化したサービスも提供されている。

- アプリケーションオブジェクト

サーバオブジェクトとクライアントオブジェクトから構成されるオブジェクト群と、これらが ORB に接続するためのインタフェース部から構成されている。

²CORBA は OMG(Object Management Group) という団体が標準化を行っている。 [23]

- ORB

コンピュータネットワークと CORBA オブジェクト間に存在し、ORB 同士のやり取りにより CORBA オブジェクトの相互的な動作を実現している。クライアントオブジェクト側からはスタブや動的起動インタフェース (スタブを使用しない場合) が起動され、サーバオブジェクト側からはスケルトンや動的スケルトンインタフェース (スケルトンを使用しない場合) が起動される。

ここでスタブとは、他のオブジェクトに対してのメッセージを送信するためのコード部のことである。またスケルトンとは、他のオブジェクトからのメッセージを受信するためのコード部のことである。

これらの関連図を図 6.2 に示す。

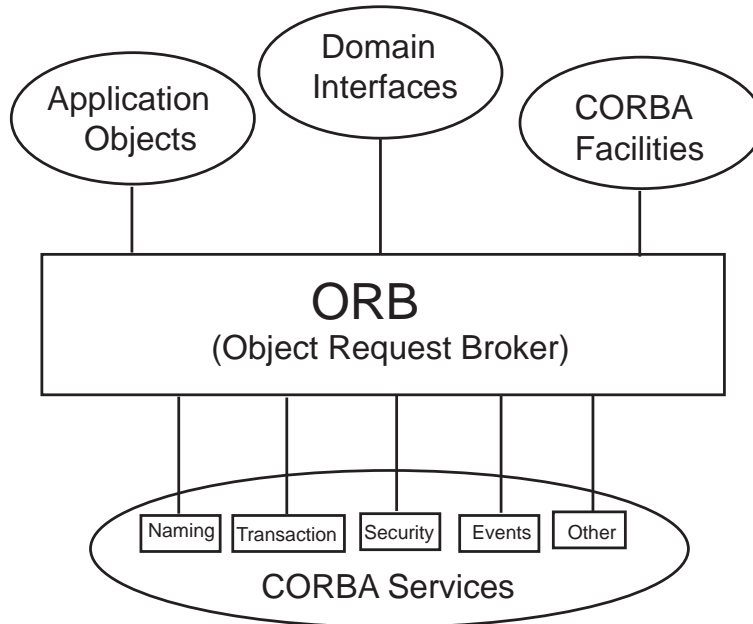


図 6.2: ORB を中心とした CORBA の関連図

ORB を中心としたこれらを用いた実際の要求と応答の流れに関して述べる。まず、クライアントプログラムはネーミングサービスにより CORBA 共通のオブジェクトリファレンスを取得し、これをもとに IDL (Interface Definition Language) コンパイラによって生成されたスタブ内の代理オブジェクトと関連付けをする。次にサーバオブジェクトを呼び出すと、代理オブジェクトがこれを受けてリクエストオブジェクトを生成し、ORB に引き渡す。ORB は引き渡されたリクエストオブジェクトをデータ転送用フォーマットに変換 (マーシャリング) してサーバ側に転送する。サーバ側の ORB は受信したデータ転送用のフォーマットからもとのリクエストオブジェクトを復元する (アンマーシャリング)。これをオブジェクトアダプタが受け、スケルトン内の処理を起動し目的のオブジェクトを探し、クライアント側からの情報を引き渡す。戻り値などの応答が存在する場合は、逆の流れによってクライアントオブジェクトへと値が返される。

このような流れによって、クライアントオブジェクトは離れたサーバオブジェクトとの要求や応答が可能になる。この流れを図 6.3 に図示した。これらの一連の流れは、ユーザーが意識しなくてもよいレベルで行われているため、ユーザーはサーバ側にあるオブジェクトをクライアント側でそのまま使用することができる [24]。

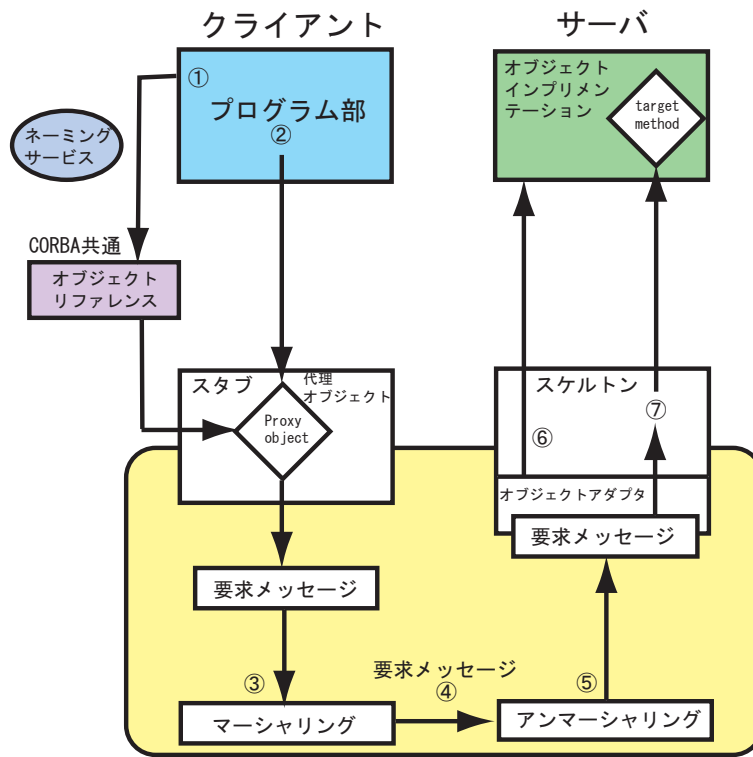


図 6.3: ORB を中心とした CORBA の要求、応答の流れ

6.3 Software System

第 1 回スライステストのオペレーションに関するソフトウェアは C++ で記述し、各モジュールに関する情報は XML (eXtensible Markup Language) によって記述しデータベースとしてまとめた。また PC 間の通信は前節で説明した分散オブジェクト技術である CORBA を用いて行った。

まずオペレーションのソフトウェアに関して述べる。スライステスト用のオペレーションソフトウェアシステムで必要な機能はいくつかある。それは

- ・各モジュールの設定のコントロール
- ・Input のコントロール
- ・Output のコントロール
- ・実際に得られたデータとシミュレーションとの比較

などである。はじめの 1 つはモジュールコントロールシステム (Property Control) とし、残りの 3 つをまとめてランコントロールシステムとして設計を行った。

6.3.1 Property Control

まず各モジュールの設定のコントロールに関するモジュールコントロールシステムだが、各モジュールには様々な設定レジスタがあり、それらのレジスタを操作することによってモジュールの設定をすることができる。これらの設定レジスタは VME を介して PC からのアクセスができるように設計されている。そのためモジュールコントロールに必要な関数は VME を経由したレジスタへの Read、Write である。この関数を作るにあたって、直接 VME に Read、Write アクセスするプログラムは vmlib-1.4 にある Read、Write の

関数を使用した。vmlib は VME とつながった PC にインストールされ、つながっている VME クレート上のモジュールに対してアクセスすることができる。vmlib の Read、Write 関数にはモジュールによって異なる Address Type、Data Type 及び Base Address を指定しなければならない、モジュール上のレジスタに対してはレジスタの Address、Data を指定しなければならない。そのため各モジュールごとにクラスを作成し、この Address と Data という引数を vmlib の Read、Write 関数に渡すようにした。各モジュールクラスは一つのモジュールを一つのボードというクラスとして実装し、モジュールの中にあるレジスタをチップクラスとして実装することにした。ボードクラスは複数のチップを持ち、動的にチップオブジェクトを生成できる。これらのクラスは各モジュールによって設定レジスタが異なるため、各々に関して記述しては記述量が増え拡張性も悪くなる。そのため、まずボードとチップの基本クラスを作成し、その中でボードにおいてはチップを純粹仮想関数として定義し、仮想関数として Initialize と Configure を定義した。またチップにおいては Read、Write を純粹仮想関数とし、Initialize、Configure を仮想関数として定義した。各モジュールのコントロールソフトウェアはこれらの基本クラスを継承して実装される。この関係図を 6.4 に示す。

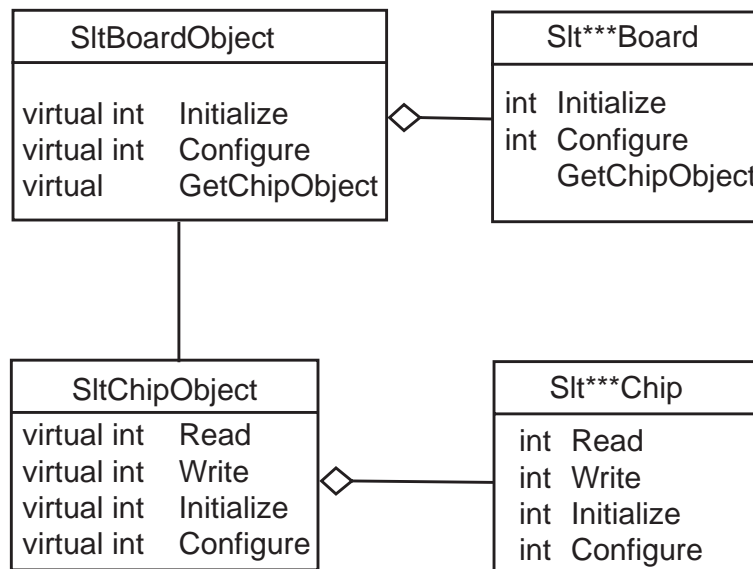


図 6.4: モジュールコントロール・ソフトウェアの関係図

具体的にまず各モジュールのチップクラスに関して述べる。各モジュールには様々な設定レジスタがあり、関連のあるいくつかのレジスタごとにチップクラスとして実装されている。ひとつのチップクラスの中には Initialize、Configure、Read、Write という関数が存在する。Initialize 及び Configure はいくつかの Write 関数を使用することによって成り立っている。Write 関数はレジスタ名、値、長さを受け取り、switch 文によって構成されレジスタ名を受け取ることにより分岐し、レジスタの Address と与えられた値を vmlib の Write 関数へと受け渡す。Read 関数は Write 関数と同様にレジスタ名を受け取ることにより分岐し、そのレジスタの Address を vmlib の Read 関数に渡し、帰ってきた値を返すようにできている。

次にボードクラスだが、これらは Initialize、Configure、チップオブジェクトの生成から成り立っている。Initialize、Configure は各チップの Write 関数を用いて行われる。

このように各モジュールのボードクラス、チップクラスを作成したわけだがこれらをそれぞれコンパイルして実行ファイルを作ると使用においても管理においても扱いに困るのでこれら呼び出すクラスを作り、実行ファイルをひとつにまとめることにした。また操作しやすいように GUI を作成することにした。GUI は QT ライブラリ [25] を使用することにより作成した。これらを作るにあたり各モジュールをデータベー

ス化が必要になり、XML を使用してデータベースを作成した。記述例を図 6.5 に載せる。

```
<Board xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation = "SLTXMLBoard.xsd"
BoardID = "1024" BoardName = "PPG"
IconName = "/home/komatsu/tgcework/CML/src/GUI/Resources/ppgicon.png"
Background = "/home/komatsu/tgcework/CML/src/GUI/Resources/gray_stone.png">

  <Chip ChipID="0" ChipName="PPGModule"
  IconName = "/home/komatsu/tgcework/CML/src/GUI/Resources/ppgicon.png">
  <Property PropertyID="0" PropertyName="Reset"
  Writable="true" Readable="false" Data="" />
  </Chip>

</Board>
```

図 6.5: 各モジュールごとの XML ファイルの記述例

データベース化に伴ない各モジュールへの ID づけ、モジュール中のチップの ID づけ、またチップ中の設定レジスタへの ID づけを行った。データベースにはこれらの ID とレジスタ名、そのレジスタへの Read、Write 可、不可情報を記述した。これらによりひとつの実行ファイルによって複数のモジュールのコントロールが可能になった。しかしこの場合プログラムがおかれている PC がつながっている VME クレート上のモジュールに対してのみアクセスが可能であり、ほかの PC につながっている VME モジュールに対してはアクセスできない。そこで前節でふれた CORBA を用いることによりリモートのアクセスを可能にした。VME につながっている PC ごとに CORBA サーバをたてることにより、ほかの VME につながっていない PC からクライアントとしてリモートでアクセスができるようになる。この場合 CORBA サーバをあらかじめ立てておけば、ユーザーは CORBA を意識することなく、また自ら使用したい PC にアクセスすることなく自動で接続され、他の PC 上のリソースが使用している PC 上にあるかのように扱うことができる。このように CORBA を利用するためと直接 VME につながっている PC からローカルにモジュールにアクセスできるように選択できるように、vmelib と各モジュールのプログラムの間に CORBA アクセス用とローカルアクセス用のプログラムを導入した (図 6.6 参照)。これらをまとめるとモジュール設定をしたい場合はサーバ名、モジュールの XML ファイル名、BaseAddress、VME のアクセスタイプを指定してプログラムを実行することにより指定したモジュールに対する GUI があらわれるようにした。Initialize、Configure、Read、Write は GUI のボタンや GUI 上で値を書き込むことにより操作ができる。Property Control の GUI を図 6.7 に示す。

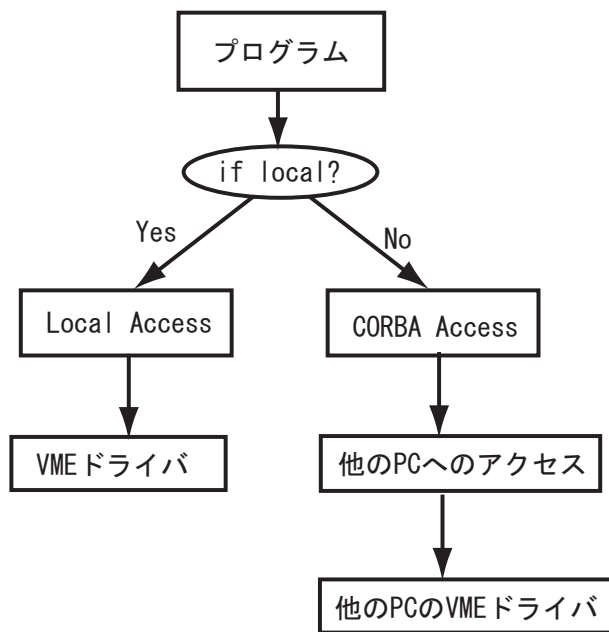


図 6.6: アクセスの選択の流れ

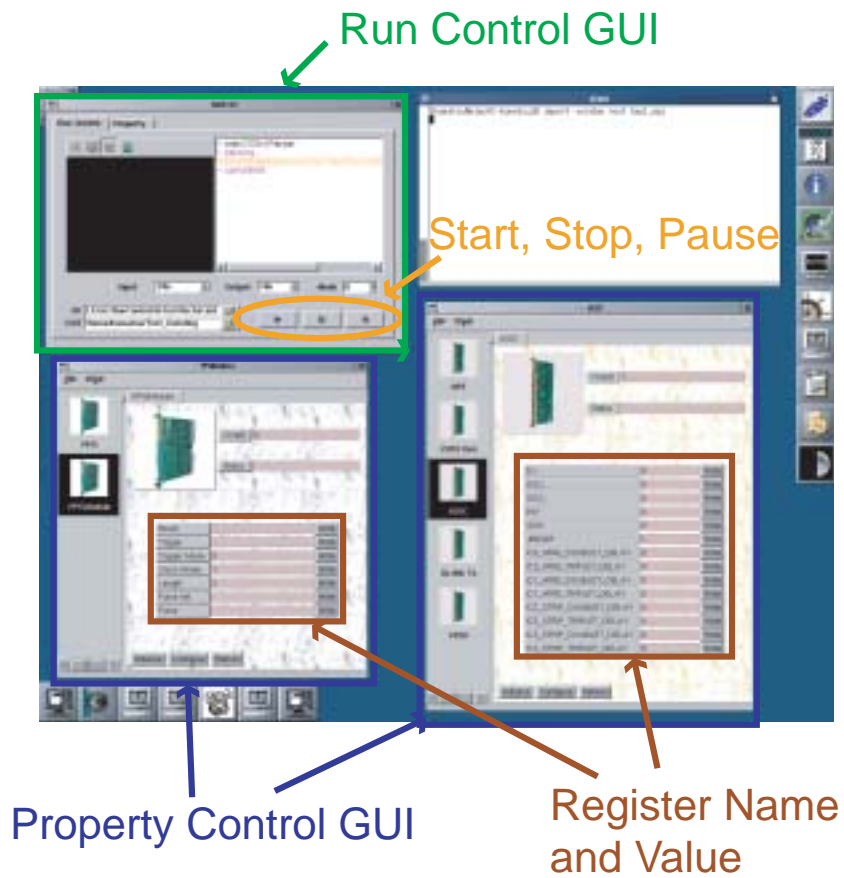


図 6.7: 第 1 回スライステスト GUI

6.3.2 Run Control

ランコントロールは大きく3つのパートから構成した。まずひとつはテストベクタのインプットの管理である。内容は9台のPPGへのテストベクタの書き込み、メモリの深さの設定、リピートモードかシングルモードの選択などPPGの設定などと、Interrupt RegisterによってPPGへトリガーをかけることがあげられる。これらはPPGとInterrupt Registerのモジュールコントロールのプログラムを利用することによって実現される。またこれらのソフトウェアシステムとは独立に、シミュレーションのテストベクタをPPGへの入力用に変換するプログラムも必要となり作成した。

次に必要なパートはモジュールからのアウトプットの管理である。今回はRODを用いてデータの収集を行った。RODには32bit幅、深さ32kのFIFOメモリが搭載されており、このメモリはVME経由で読み出すことが可能である。読み出されたデータは動作検証のためファイルに保存する必要がある。これらのことからアウトプットの管理にはRODのモジュールコントロールのプログラムを利用し、RODの設定を行う事と、データをファイルに保存する機能を記述した。

第3のパートは動作検証のパートで、モジュールからのアウトプットとシミュレーションによって予測されているアウトプットとの比較を行い、正確に動作しているかの検証をおこなうというものである。ここではトリガー・シミュレーションのアウトプットのファイルとモジュールのアウトプットファイルはお互いのフォーマットの違いがあるため変換を行わなければならない。またモジュールのアウトプットデータはデータをとるタイミングによってデータの先頭の位置が変わるため、データの頭だしの機能も必要となる。

これら3つのパートによってランコントロールは成り立っている。ランコントロールプログラムのクラスダイアグラムを6.8に示す。またランコントロールもプロパティコントロール同様にQTライブラリによってGUIを作り操作しやすいようにした。このGUIではインプットのテストベクタファイルの選択、アウトプットファイルの選択をしてスタートボタンを押せば動き出すというようにとてもシンプルで扱い易いものとなった。Run ControlのGUIを図6.7に示す。

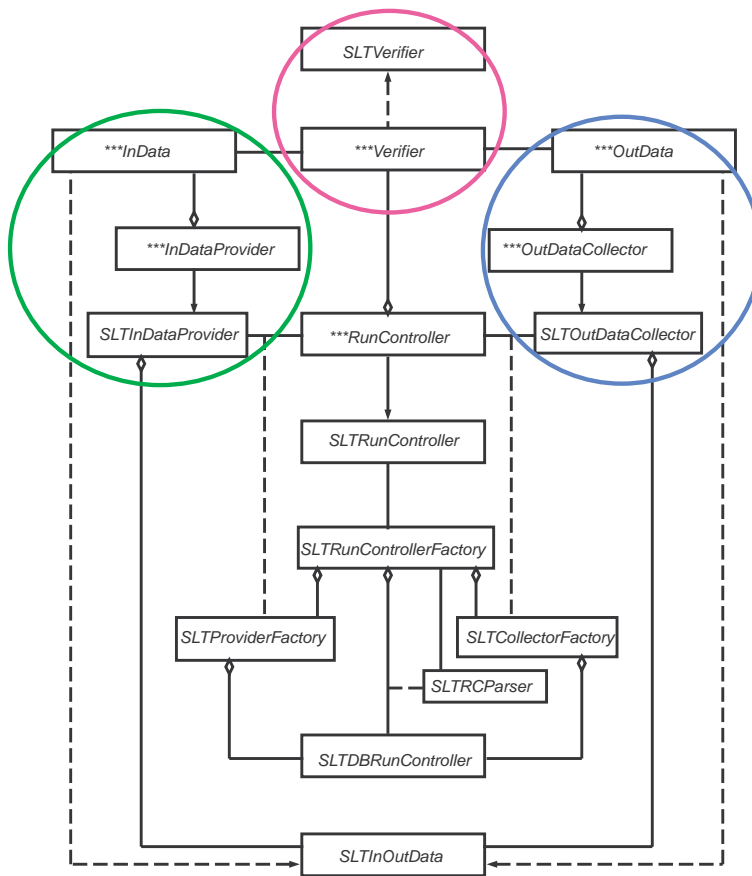


図 6.8: Run Control Diagram

また、データベース・プログラムと GUI プログラム、モジュールのプログラム同士の関係を図 6.9 に示す。
 ランコントロールのデータベース用 XML ファイルは、プロパティコントロールで使ったモジュールの XML ファイルで必要なファイルを呼び出すようにした。ランコントロール用の XML ファイルの記述例を 図 6.10 に載せる。

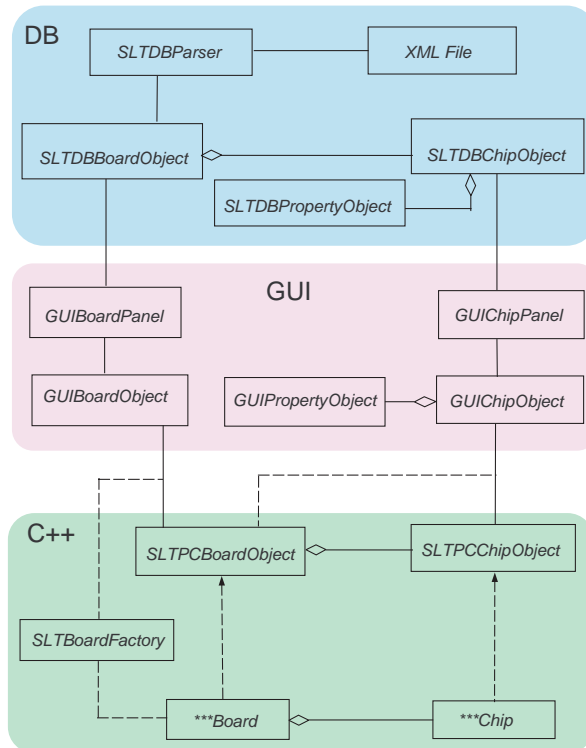


図 6.9: Data Base Diagram

```

<RunController xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation = "SLXMLRController.xsd"
  ID = "0" Name = "TestController">

  <Provider ID="0" Name="Provider">
    <Board XMLFile="SLTINTRBoard.xml" ServerID="5" BaseAddress="8000" AccessMode="2D"
      Initialize="true" Configure="0"/>
  </Provider>

  <Collector ID="0" Name="Collector">
    <Board XMLFile="SLTRODBoard.xml" ServerID="1" BaseAddress="15000000" AccessMode="10D"
      Initialize="true" Configure="0"/>
  </Collector>

  <Verifier ID="0" Name="XoVerifier"/>

</RunController>

```

図 6.10: ランコントロール用 XML ファイルの記述例

6.4 テスト結果



図 6.11: 第 1 回スライステスト セットアップ写真

第 1 回のスライステストではトリガーパートのなかの Wire パートの接続テストを行った。実際のセットアップの状況を図 6.11 に示す。本来トリガーパートは Wire パートと Strip パートからなり SL で $R-\phi$ のコインシデンスがとられるが、Wire パートのためのテストのため出力は HPT の出力となり、HPT のシミュレーション結果と出力を比較した。HPT からのデータはオプティカル・ファイバーによって出力されるため、オプティカル・ファイバーのデータ受信ができる ROD へ出力した。ここで本来なら SL へと出力されるのだが、今回は Wire パートのみなので SL をつないでも $r-\phi$ コインシデンスをとることができず、意味をなさないということとまた SL は VME バスにデータを出力する機能を持っていないため、VME バスにデータを出力できる ROD を用いた。テストは、はじめから全てのモジュールを接続して行うのではなく、PP ASIC から順に確認しながらモジュールを増やしていくという方法で行った。まずはじめに、PPG から PP ASIC へデータが出力されているか、PP ASIC の出力を調べてみた。PP ASIC の出力は PS Board 上のデバックピンにより確かめた。ここで今回使われている PP ASIC は、電源を投入した際の初期設定として入力データにマスクがかかる仕様となっており、そのままデータを流してもデータは出力されない。そのため、PT4 を用いた dShake というプログラムによってマスクをはずしてから、データを入力した。結果は正常に出力されていることが確かめられた。続いて SLB ASIC の出力を、LVDS シリアライザーから PT4 で取得して SLB ASIC のシミュレーション結果と比較した。LVDS シリアライザーから出力は HPT への入力にあわせてパッチが当てられているため、そのままでは bit の入れ替えがありシミュレーションとは一致しない。パッチを図 6.12、図 6.13 に示す。

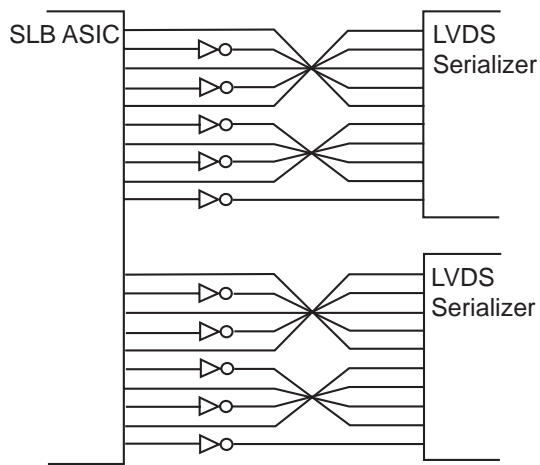


図 6.12: PS Board Wire Doublet Patch

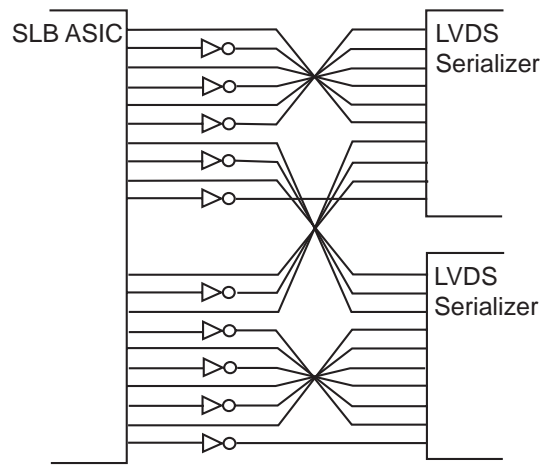


図 6.13: PS Board Wire Triplet Patch

そのため、比較プログラムに bit 入れ替えを追加しシミュレーションの bit アサインと同一にした。この結果、SLB ASIC の出力とシミュレーションの出力は一致した。次に今回のスライステストに必要なモジュールをすべてつないでのテストを行った。ここで HPT からの出力は SL の入力にあわせて bit アサインが行われており、HPT Board の出力で SL に必要のないものは送信しないようになっている。そのため、シミュレーションのフォーマットと異なるうえに bit 数も少なくなる。(表 6.5、図 6.14 参照) よって比較プログラムでは、シミュレーションの出力において HPT から SL に送信されない bit をマスクすることにした。また、HPT の出力はシミュレーションのフォーマットにしたがって変換を施した。この結果、両者が一致することが確かめられた。これにより、Wire パートの Logic の正当性、及び LVDS Serializer/Deserializer、G-Link のリンクが正常に動作していることが確かめられた。

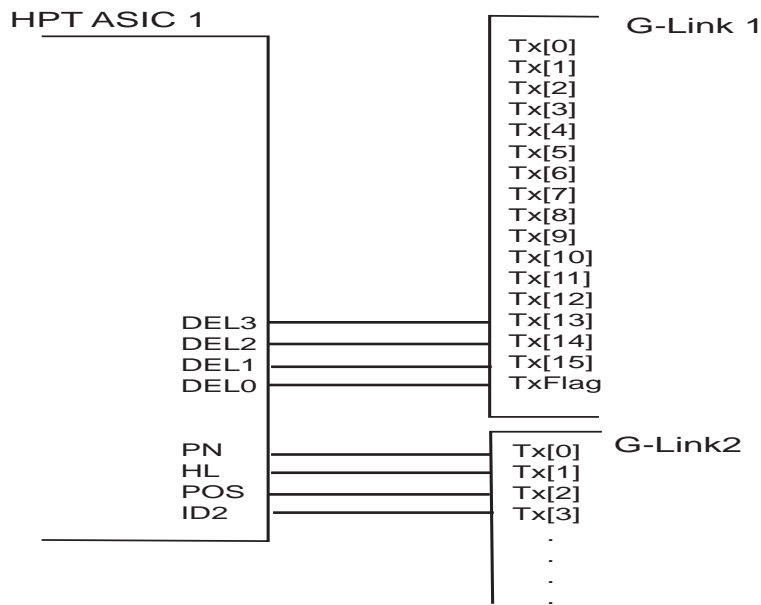


図 6.14: HPT ASIC - G-Link 間の接続と HPT Board のアウトプット (Wire 側の第一候補の出力)

0	SGN	PT2	PT1	PT0	POS4	POS3	POS2	POS1	POS0
---	-----	-----	-----	-----	------	------	------	------	------

表 6.1: SLB Strip Doublet フォーマット

0	PT0	PT1	PT2	SGN	POS0	POS1	POS2	POS3	POS4
---	-----	-----	-----	-----	------	------	------	------	------

表 6.2: SLB Wire Doublet ハードウェア出力

0	HIT	POS4	POS3	POS2	POS1	POS0
HIT	POS4	POS3	0	POS2	POS1	POS0
HIT	POS4	POS3	POS2	POS1	POS0	

表 6.3: SLB Strip Triplet フォーマット

0	POS0	POS1	POS2	POS3	POS4	HIT
POS0	POS1	POS2	0	POS3	POS4	HIT
POS0	POS1	POS2	POS3	POS4	HIT	

表 6.4: SLB Wire Triplet ハードウェア出力

HL	ID0	ID1	ID2	POS	PN	DEL0	DEL1	DEL2	DEL3
----	-----	-----	-----	-----	----	------	------	------	------

表 6.5: HPT の出力フォーマット [26]

6.5 第1回統合テストのまとめ

第一回スライステストでは、まずソフトウェアシステムを一から構築することから始まった。まず、各モジュールごとに設定レジスタの設定を行う Property Control を作り、それを利用した Run Control のプログラムを作成した。これらのデバックが終わり実用化段階に入ると、テストベクタの編集プログラムを作り、各モジュールの出力をシミュレーションと比較するプログラムの作成を行った。これにより、スライステストのソフトウェアシステムを構築することができた。このシステムを用いスライステストが行い、トリガーパートの Wire 側の HPT の出力までの動作検証を行った。入射する粒子数は1つから3つまで行い、それぞれ正確に動作していることがシミュレーションとの比較によって明らかになった。結果を表 6.6 に載せた。しかし、Wire 側のみではトリガーパートのスライステストが全て終わり、問題なく動作するとは言えない。Strip 側の PS Board が完成次第すぐにテストにとりかけられるようにシステムの拡張をしなければならない。特に行わなければならないのは、テストベクタ編集プログラムの Strip への拡張、Strip 側が増えることによる入力チャンネルの増加に対しての Run Control の入力部分の改良、SL を使用することによってデータ読み出しのために使われるモジュールプログラムの作成などがあげられる。また、リードアウトパートのモジュールの設定プログラムを作成し、Property Control に追加し、TGC エレクトロニクスのモジュール全てをコントロールできるようにしなければならない。さらに、今回のソフトウェアシステムは我々のグループ

の完全オリジナルのシステムであるため今後は ATLAS 検出器の統一されたソフトウェアシステム (ATLAS Online software) への組み込みも必要となってくる。そのため、次回のスライステストでは、ATLAS Online software への組み込みも含めた新たなソフトウェアシステムを構築して行うことが必要となる。

入射トラック数	サンプル数	エラー数 T
1	4992	0
2	4992	0
3	3678	0

表 6.6: 第 1 回スライステストの最終結果

第7章 第2回統合テスト

第2回統合テスト(以下、スライステスト)では前回のソフトウェアを改良することにした。大きな改良点はALIAS Online グループが提供しているソフトウェアシステムの枠組みに前回のソフトウェアを導入するというものである。これは前回のソフトウェアがALIAS Online Software への拡張性がよいように作られていたということと、ALIAS Online Software が提供する様々なサービスを利用できるようにするために行った。また将来的には組み込みの必要があるため、早くからALIAS Online Software を導入しておくという理由もある。

さらに改良点としてあげられるのは、TTCvi、TTCrx の導入である。そのため、TTCvi をコントロールするためのプログラムを記述し、テストシステムの中へ組み込んだ。TTC を導入することによって、さらに本実験に近い状態でのテストが可能になる。

また今回のテストでは、Strip 用 PS Board を1台加え、Wire と Strip の両方のデータを扱うことにし、これによって SL を導入してのテストが可能になり、SL での $R-\phi$ コインシデンスまでのテストを行った。以上のことから、第2回スライステストでは、トリガーパートのモジュール全てを使用した統合テストが可能となり、そのテストを行った。

7.1 セットアップ

- PPG(Pulse Pattern Generator)
今回は PPG を計 15 台使用した。8 台を Wire 側のデータ入力用として使用し、6 台を Strip 側のデータ入力用に使用した。残り 1 台は前回のテストで TTCrx の擬似信号を出すために使われていたものである。今回は TTCvi、TTCrx を用いたため使用されないが、予備のため残しておいた。
- PS Board(Wire & Strip)
Wire 側の PS Board は、前回使用したものを用了。Strip 側は新たに Strip 用の PS Board を製作した。使用した PP ASIC と SLB ASIC は Wire 側で使用しているものと同じバージョンのものを使用した。
- LVDS Serializer/Deserialiser
Chip、カテゴリー 5 ケーブル共に前回と同じものを使用した。
- SPP Board
今回は TTCrx を実装し、TTCvi から送られてきたクロックを PS Board へと分配した。
- HPT Board
前回と同じ Forward 用に作られたボードを用了。
- CCI/HSC Board
前回と同じものを使用した。
- G-Link Tx/Rx
Chip、オプティカル・ファイバー共に前回と同じものを使用した。

- SL
Wire 側と Strip 側の入力を受け $R-\phi$ のコインシデンスをとる。2001 年 6 月に作られたプロトタイプを使用した。
- TOM
32bit 幅、深さ 64k のメモリを搭載した VME モジュールである。SL からの出力は VME 経由で読み出すことができないため、SL の出力を TOM に送り VME 経由でデータを読み出すために使用した。
- TTC
今回は本実験同様、すべてのモジュールに TTC を用いてクロックを分配できるようにした。また、各種信号も VME 経由で TTCvi のレジスタ設定を行うことにより出力が可能で、それを TTCrx によって受信することができる。今回は 40.08MHz のクロックを TTCvi に入力し、TTCvi から出力されるクロックを Fanout Board に入力してすべてのモジュールに分配した。特に SPP Board ではボードに TTCrx Board を載せ、TTCvi から送られてくるクロックを受信して PS Board にクロックを分配した。

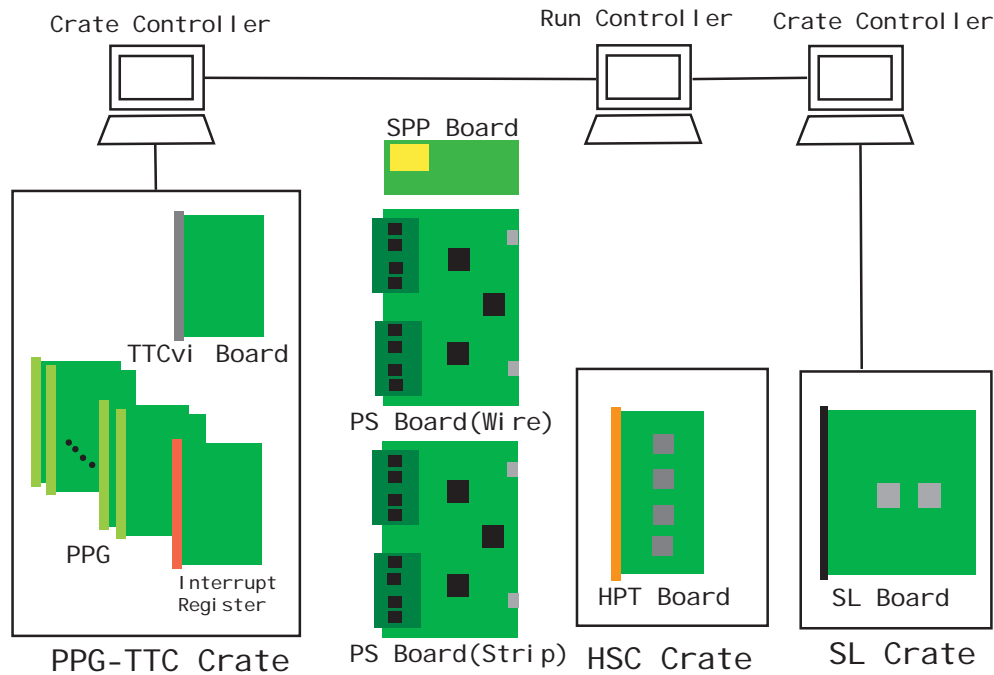


図 7.1: 第 2 回スライステスト セットアップ図

7.2 ATLAS Online Software

ATLAS Online Software は ATLAS 実験における検出器の制御、及び DAQ などのために用意されたソフトウェアである。この中身は core components と trigger/DAQ/detector integration components に分けられる。core components は以下に示す 5 つから構成されている [27]。

- Configuration Database(ConfDB)
DAQ システムの莫大な数のパラメータを記述するもので、ランニングモードやステータスに関するハードウェアやソフトウェアのコンポーネントが記述される。ATLAS DAQ の設計において、ConfDB の内容によってパラメータ化され、できるだけ柔軟性をもつことが重要になってくる。ConfDB は XML で記述されている。
- Message Reporting System(MRS)
分散された環境の中で、他のコンポーネントにソフトウェアのコンポーネントがメッセージを送ることを可能にするシステムである。MRS はメッセージの転送、フィルタリング、ルーティングを行う。
- Information Service(IS)
ソフトウェア・コンポーネント間の情報交換を可能にする。これにより様々な部分からの多数の情報が分類され、非同期あるいはオンデマンドに利用することが可能になる。
- Process Manager(PMG)
ソフトウェア・コンポーネントの基礎的なジョブをコントロールする。根本的なオペレーティングシステムと独立したコンポーネントのスタート、ストップ、モニタリングなどが可能になる。
- Run Control(RC)
DAQ サブシステム、バックエンド・ソフトウェアおよび外部システムのオペレーションを調整をし、データ収集をコントロールする。また、オペレーションの変更やデータ収集の変更のために、ユーザ・インタフェースを持っている。さらに、コマンドやステータス、制御情報を交換するために、DAQ サブシステムやバックエンド・ソフトウェアとのソフトウェア・インタフェースも持っている。

次に trigger/DAQ/detector integration components を簡単に説明する。

- Resource Manager(RM)
自由に共有できないハードウェアとソフトウェアの資源を割り振り、いくつかのグループが妨害しあわず並列に動作することを可能にする。
- Partition Manager
いくつかのパーティションの同時動作を許すために RM を拡張する。
- Test Manager(TM)
ハードウェアとソフトウェアの個々のテストを行うためのものである。
- Diagnostics Package(DS)
問題を分析し、個別のコンポーネントのステータスの機能確認を行う。また、全システム中でコントロールを行い、分析し、自動もしくはオペレーター援助モードでシステムの機能の異なるフェイズで問題を解決する。
- Integrated Graphical User Interface(IGUI)
ディテクターのコンフィグレーション、トリガー・レート、バッファの占有率、サブシステムの状態など、オンラインでデータ収集状況やコントロールをモニターする。

- Online Bookkeeper
DAQ システムによって記録されたデータを保管するもので、ラン毎に情報を記録する。また、情報を検索、更新するための多くのインタフェースを提供する。
- Event Dump
イベントの保全、構造の確認のため、データフローの中からサンプルを取り出す。

ALIAS Online Software は大まかに以上のような構成になっている。

この中で、今回のスライステスト・ソフトウェアの組み込みに必要なものは ConfDB の記述である。ConfDB の中に Detector 情報やスライステストで使用する TGC エレクトロニクスのパラメータを記述しなければならない。また、図 7.2 にあるように、Crate Controller というものの記述も必要となる。Crate Controller とは、クレートごとに必要でクレートに搭載されるモジュールをコントロールするためのプログラムである。ConfDB に記述されたパラメータはインフォメーション・サーバーを介し、Crate Controller へと送られクレートに搭載されたモジュールのコントロールに使用される。

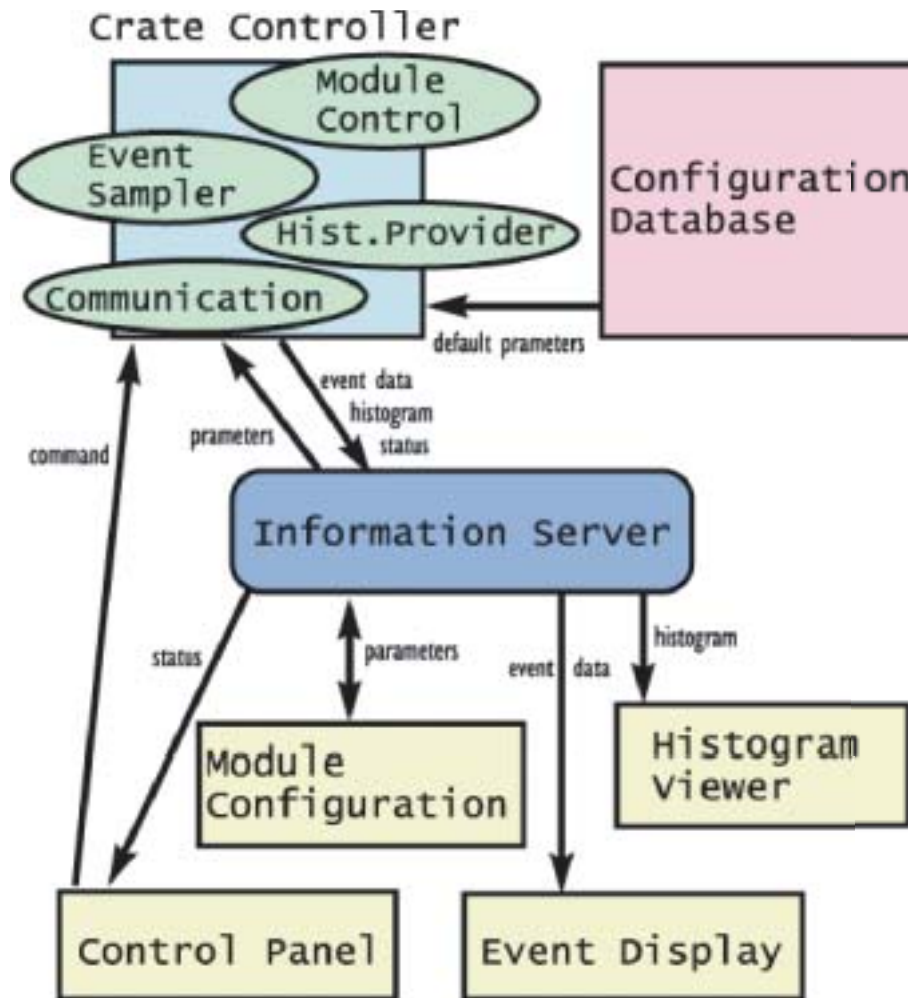


図 7.2: ATLAS Online Software relation

7.3 Software System

第2回スライステストのソフトウェア開発は、まず TTCvi をコントロールするソフトウェアの開発と SL からのデータを VME 経由で読み出す TOM をコントロールするソフトウェアの開発を行った。これらのソフトウェアは、第一回スライステストとの Property Control で使用したソフトウェアと同じ構造で作成した。続いて行ったのが、前述したように ATLAS Online Software の枠組みで記述することである。ここで、前回使用したオリジナルのソフトウェアにおいて、モジュールのコントロールを行うプログラムの引数の型が、図 7.3 に示すように ATLAS Online Software で要求されているパラメータの型と一致していたため、拡張は比較的容易に行うことができた。これによりモジュールごとに記述したプログラムの変更なしに、ConfDB の XML ファイルを記述するだけでよいことになる。ただし、Crate Controller は新たに記述する必要がある。

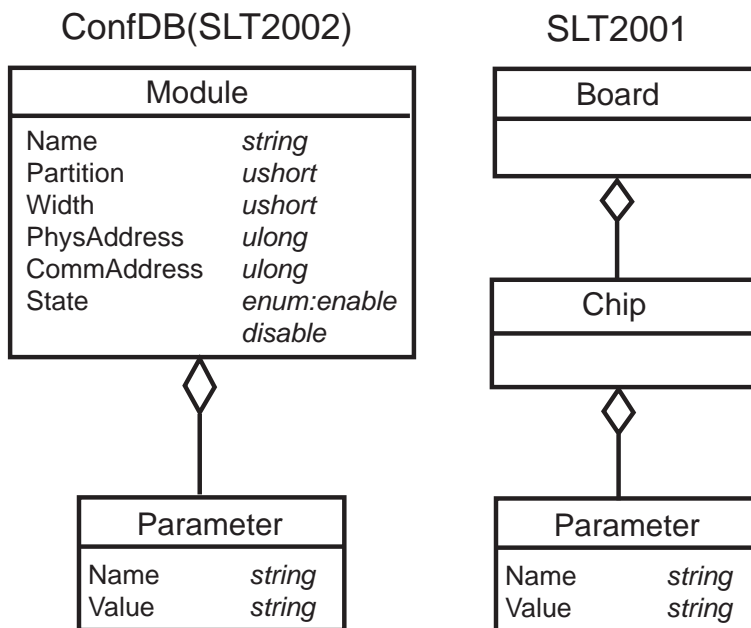


図 7.3: ソフトウェアの引数の比較

7.3.1 Configure Database

まず ConfDB の XML ファイル記述に関して述べる。スライステストに必要な XML ファイルは、大きく分けると 4 種類に分類できる。それは

- ・パーティションに関する XML ファイル (図 7.4 参照)
- ・構成するハードウェアに関する XML ファイル (図 7.5 参照)
- ・使用するソフトウェアに関する XML ファイル (図 7.6 参照)
- ・各モジュールのパラメータに関する XML ファイル (モジュールごとに存在する) (図 7.7 参照)

である。

```

<obj class="RunControlApplication" id="RootCtrl">
<attr name="CommTimeout" type="s32">30</attr>
<attr name="CreationType" type="Enumeration, Default, DAQ_Setup,
  DAQ_Shutdown, SOR, EOR, Supervised" num="1">Supervised</attr>
<attr name="Name" type="string">RootCtrl</attr>
<attr name="Parameters" type="string">-c RootCtrl -P RunParams</attr>
(snip)
<rel name="children" num="2">RunControlApplication "RODCtrl_01"
  RunControlApplication "PPG_Ctrl01"</rel>
<rel name="SWObject">SW_Object "rc-root-controller"</rel>
<rel name="RunsOn">Workstation "MyWorkstation"</rel>
<rel name="NeedsEnvironment" num="0"></rel>
<rel name="InitializationDependsFrom" num="2">RunControlApplication "
  RODCtrl_01" RunControlApplication "PPG_Ctrl01"</rel>
<rel name="ShutdownDependsFrom" num="0"></rel>
<rel name="HasParameters" num="0"></rel>
</obj>

```

コントローラの名前
 プログラムに渡す引数
 childrenが2つある
 SWオブジェクトの名前
 プログラムを起動するPC

図 7.4: パーティションに関する XML ファイルの記述例

```

<obj class="Workstation" id="PPGWorkstation">
<attr name="OsType" type="enum">linux</attr>
<attr name="Name" type="string">elec05.kek.jp</attr>
</obj>

<obj class="Workstation" id="RODWorkstation">
<attr name="OsType" type="enum">linux</attr>
<attr name="Name" type="string">elec01</attr>
</obj>

<obj class="Workstation" id="MyWorkstation">
<attr name="OsType" type="enum">linux</attr>
<attr name="Name" type="string">elecref</attr>
</obj>

```

PPG Ctrl. 用WSとしてelec05を使用
 ROD Ctrl. 用WSとしてelec01を使用
 Root Ctrl. 用WSとしてelecrefを使用

図 7.5: ハードウェアに関する XML ファイルの記述例

```

<obj class="Program" id="rc-root-controller on linux">
<attr name="OsType" type="enum">linux</attr>
<attr name="ExecutableFile" type="string">rc_root_ctrl</attr>
<attr name="DefaultParameters" type="string">"</attr>
<attr name="DefaultPriority" type="s32">0</attr>
<attr name="DefaultPrivileges" type="string">"</attr>
<rel name="NeedsEnvironment" num="0"></rel>
<rel name="DescribedBy">SW_Object "rc-root-controller"</rel>
</obj>

<obj class="SW_Object" id="rc-root-controller">
<attr name="Name" type="string">rc-root-controller</attr>
<attr name="DefaultParameters" type="string">"</attr>
<attr name="DefaultPriority" type="s32">0</attr>
<attr name="DefaultPrivileges" type="string">"</attr>
<rel name="NeedsExclusive" num="0"></rel>
<rel name="NeedsEnvironment" num="0"></rel>
<rel name="NeedsShared" num="0"></rel>
<rel name="ImplementedBy" num="1">Program "rc-root-controller on linux"</rel>
</obj>

```

Root Ctrl. の実体(実行形式)
 "rc-root-controller"のLinuxでの実装
 SWオブジェクトの名前
 実体は"rc-root-controller on linux"にある

図 7.6: ソフトウェアに関する XML ファイルの記述例

<code><obj class="parameter" id="SETTING"></code>	レジスタ名
<code><attr "Name" type="string">"Csr/Setting+r+w"</attr></code>	Read, Writeの許可指定
<code><attr "Value" type="string">"3"</attr></code>	レジスタの設定値
<code></obj></code>	

図 7.7: モジュールに関する XML ファイルの記述例

まず 1 番目のファイルに関してだが、このテストのパーティション名を `slt_trg`(SliceTest Trigger) として作成した。このファイルにはスライステストに必要な残りの 3 種類のファイルを統括するファイルにあたるため、必要な XML ファイルに関して記述してある。

2 番目のハードウェアに関するファイルには、使用するモジュール、クレートなどが記述されている。

3 番目のソフトウェアに関するファイルには、コントローラなど必要なソフトウェアに関して記述されており、どのような OS で使用されるかなども記述される。

最後のモジュールのパラメータに関するファイルには図 7.3 の左側に書かれているようなモジュールのアドレスやレジスタの名前、レジスタの設定値などが記述される。今回は Property Control によってモジュールの初期設定を行ったが、今回は以下に述べる Crate Controller を立ち上げる段階で XML ファイルに記述された設定値が設定されるようになっている。

7.3.2 Crate Controller

次に Crate Controller に関して述べる。今回のテストでも必要になるクレートは、インプットのデータをコントロールする PPG が搭載されるクレートと HSC、HPT が搭載されるクレート、アウトプットデータを収集する SL、ROD が搭載されるクレートの 3 台である。しかし、前回のテスト同様 HSC クレートは PC とは直接接続されず、CCI 経由で HPT のコントロールが行われる。そのため Crate Controller のプログラムが必要となるのは 2 台のクレートに対してである。Crate Controller のダイアグラムを図 7.8 に示す。

図から分かるように、各モジュールの設定に関するプログラムは `SLTBoardObject` を呼び出すことにより、前回のテストに使用したプログラムを使用することができるようにした。

また `TgcModuleBase` という仮想関数を用意し、各モジュールに対する記述を行った。ここでの記述は前回使用したモジュール設定プログラムを使用するためのパラメータ名の変換が記述されているのみで新たにモジュールのコントロールを記述したわけではない。

さらに今回は `TgcCrateBase` という仮想関数を用意した。これにより Crate Controller に対する記述を統一し、このプログラムを継承することにより複数ある Crate Controller の動作を、効率よくまた拡張性を高く実装することを可能にした。

Crate Controller のコンポーネントは Load、Initialize、Configure、Startaction などがあり、Load で Information Server から情報を取得し、Initialize で必要なインスタンスの生成を行い、その後 Configure によって XML ファイルに記述されている各モジュールの設定値を実際に設定する。続いて Startaction により、データの入出力などを行っている。

起動するにあたっては、前回同様クレートに直接接続されている PC 以外に Run Control 用の PC を用いて行う。この場合も ATLAS Online Software では CORBA を用いて PC 間の通信を行っているが、前回のように Server をユーザーが立てるのではなく、起動すると PMG が立ち上がり自動的に ATLAS Online Software の各種の Server を立ち上げ、処理を行ってくれる。操作に関しては ATLAS Online Software から供給されている GUI を用いることができる。この GUI は Java を用いて記述されている。GUI を図 7.9 に載せる。図の

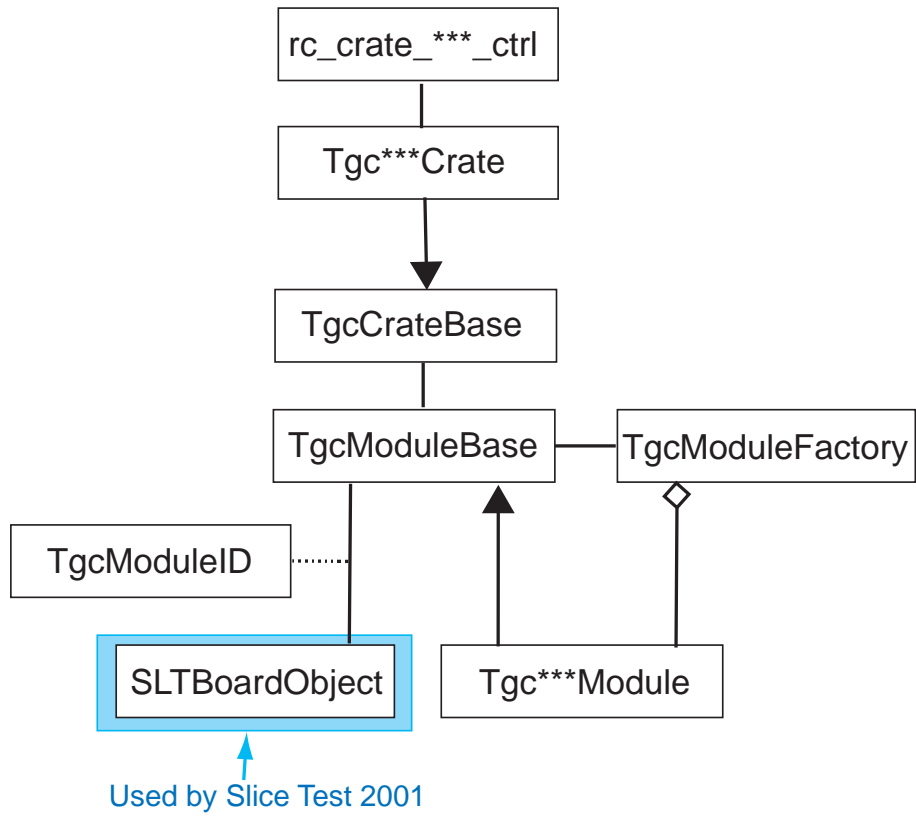


图 7.8: Crate Controller Diagram

左側に BOOT、Initialize、Configure、Start などのボタンがあるように、これらによって Run Control を行う。

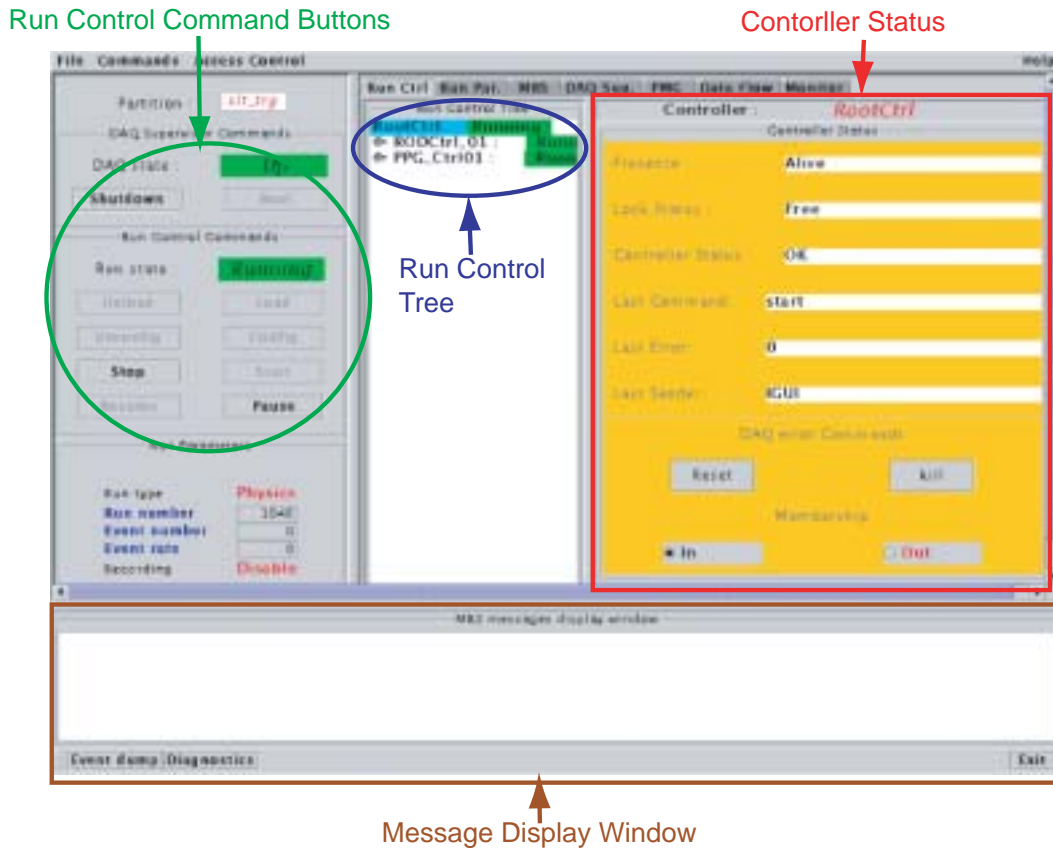


図 7.9: Run Contrller GUI

7.4 テスト結果

テストは一度に全体を繋げるのではなく、PP ASIC から順に確認しながら行った。前述のソフトウェアシステムにおいて、インプットデータのコントロールによりテストベクタを PPG から PP ASIC へと入力した。PP ASIC の段階では PPG から同期して信号が入ってくるため、PP ASIC の動作に関して特に問題は無いのだが、今回使用されている PP ASIC は、前回のテストと同様のもので、電源投入時の初期設定で入力に対してマスクがかかる仕様となっており、初期設定のままではデータは出力されない。そのため設定をし直し、マスクが完全にはずれ信号を流す状態になっているか重要になる。マスクをはずすプログラムは、当初前回使用した PT4 を用いた dSake というプログラムによって行っていたが、マスク以外の PP ASIC の設定に対して拡張性を持たせるため、新たに C++ で記述しなおした。しかし、本来なら CCI から HSC を経由し SSW から PS Board へ命令を出し、PP ASIC の設定を行うのだが、SSW がまだ完成に至っていないため今回のプログラムも PT4 を用いたものとなった。このプログラムによりマスクが完全にはずれることを確認した。なお、確認は PS Board 上に設けられたデバック用の PP ASIC からの出力ピンをオシロスコープで見ることにより行った。またこのプログラムは柔軟性があり、ディレイの設定なども容易に行うことができるように記述した。これで PP ASIC までのデータの流れが確認された。

次に SLB ASIC までの確認を行った。確認方法としては、SLB ASIC のアウトプットは LVDS レベルに変

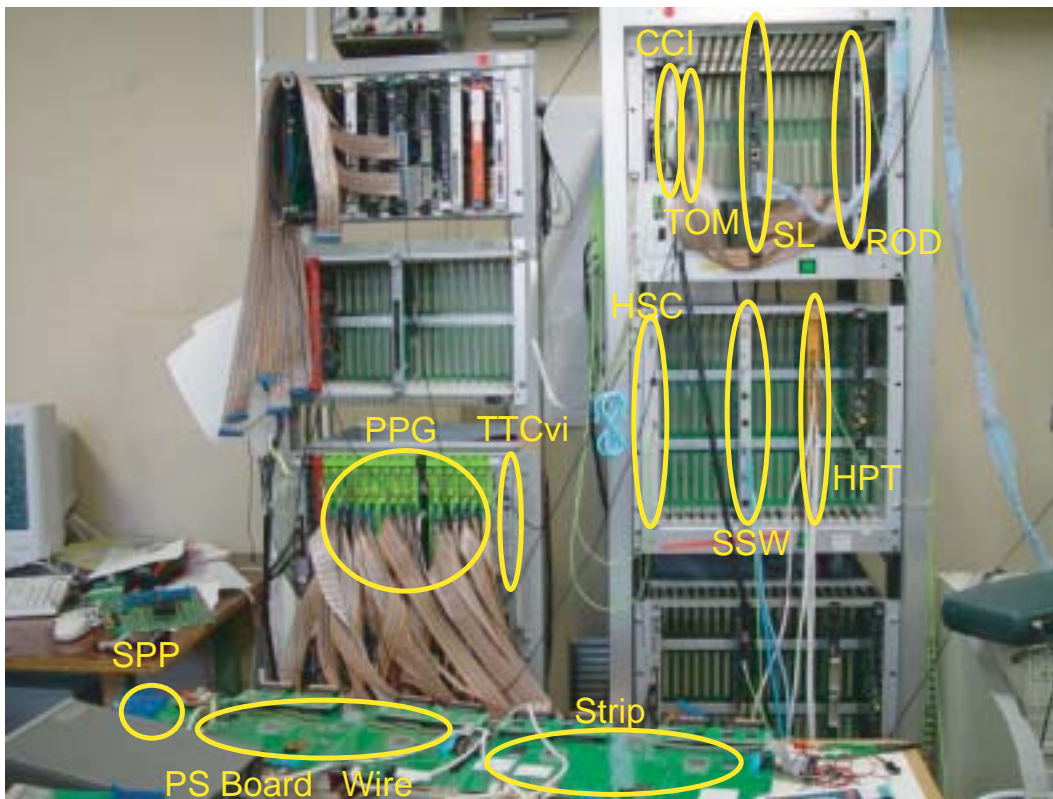


図 7.10: 第 2 回スライステスト セットアップ写真

換されてカテゴリ 5 のケーブルを通して出力されるため、その出力を PT4 で実装された FIFO で受信し、シミュレーションと比較するという方法を用いた。しかし、PS Board 上で HPT の入力にあわせパッチが当ててあるため、出力とシミュレーションをそのまま比較できなかったため、出力を変換して比較するプログラムを作成し比較を行った。まず前回のテストで確認されている Wire 側の比較を行った。これは出力とシミュレーションが問題なく一致することが確認された。続いて新たに作成された Strip 側の比較を同様の方法で行った。Strip 側の PS Board の出力は図 7.11 のようになっている。

その結果は出力とシミュレーションが一致しないものとなった。この結果から、一致しない原因に規則性があるかどうか調べた。これにより一致しない部分は、Strip Doublet の部分に見られることがわかった。SLB ASIC のトリガーパートの Strip Doublet のアウトプットフォーマットは 10bit 単位で表 7.1 のようになっている。

先頭の 0 は Low-pT をあらず 0 で固定されている。SGN は曲がり具合がプラス方向かマイナス方向かを

0	SGN	PT2	PT1	PT0	POS4	POS3	POS2	POS1	POS0
---	-----	-----	-----	-----	------	------	------	------	------

表 7.1: SLB Strip Doublet フォーマット

あらずのに使われている。PT は無限運動量トラックからのずれ具合をあらわし、POS は位置情報をあらわす。詳しく調べた結果、出力全てがシミュレーションに対して POS では -1、PT では +1 になっていることが分かった。図 7.12 は、無限運動量トラックが通過したと仮定した場合の模式図である。Middle Doublet と Pivot Doublet の位置情報が違うため POS と PT が異なることがわかる。これが起きた原因は Strip Doublet の

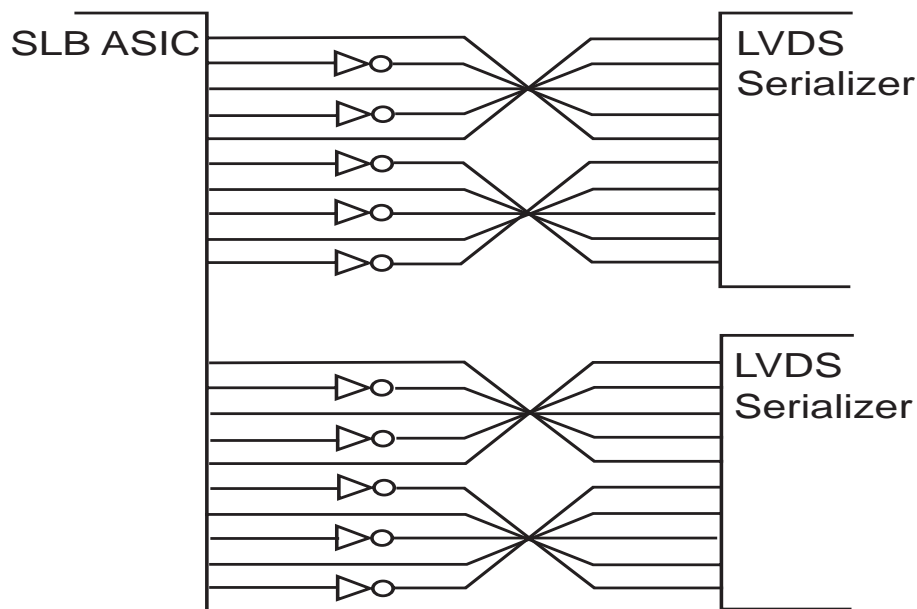


図 7.11: PS Board(Strip) の出力

ロジック作成の際に Wire Doublet との違いがチャンネル数のみであると考えられており、Wire Doublet のロジックをチャンネル数を変えて開発されてきたのが原因であった。具体的には Wire Doublet と Strip Doublet ではピボットとモデルの位置の対応が異なることによって起きていると考えられる [17] [16]。

しかし、このことは以前から知られていたことで、シミュレーションはハードウェアにあわせるように修正されているはずだった。しかしシミュレーションのプログラムを調べたところ、修正部分に変更されていることが分かり、ハードウェアにあわせるように変更した。ここで Strip Triplet は SLB ASIC では位置情報のみを処理するため、ピボットの情報を必要としないため、Strip Doublet のようなバグはなかったと考えられる。Strip Doublet のシミュレーションの変更後、出力とシミュレーションを比較したところ、すべて一致することが確かめられた。

次の段階として、HPT までの接続、比較を行った。HPT のアウトプットは光信号で出力される。そのため PT4 を使い、メザニンカードを光信号を受信するものに換え、SLB ASIC のときと同様に FIFO を作成しデータを収集した。収集したデータをシミュレーションと比較することにより確認を行った。ここで、HPT に搭載されている HPT ASIC はシミュレーションで対応しているバージョンよりも古いもので、ID 付けが異なっている。そのためこの部分は比較プログラムによって出力データを変換してから比較を行った。この結果、HPT も Wire、Strip 共に問題なく一致することが確かめられた。

最後に SL を接続するトリガーパート全てのスライステストを行った。SL からのアウトプットは同じレートに搭載された TOM に送り、TOM から VME 経由でファイルへと出力した。このファイルと SL のシミュレーション結果を比較した。出力結果を比較したところ、シミュレーションにおいてヒットなしのイベントに関してはハードウェアからの出力もヒットなしという結果が得られたが、シミュレーションにおいてヒットありのイベントに関してはハードウェアからヒットが出力されるものとされないものがあった。この割合はシミュレーションでヒットありと出るもののおよそ 3 割であった。この 3 割のハードウェアから出力されたデータは、シミュレーション結果と一致した。ここでまず考えられるのは、SL におけるデータラッチのタイミングのずれがあげられる。そのため、クロックのディレイを変えて試してみた。しかし、

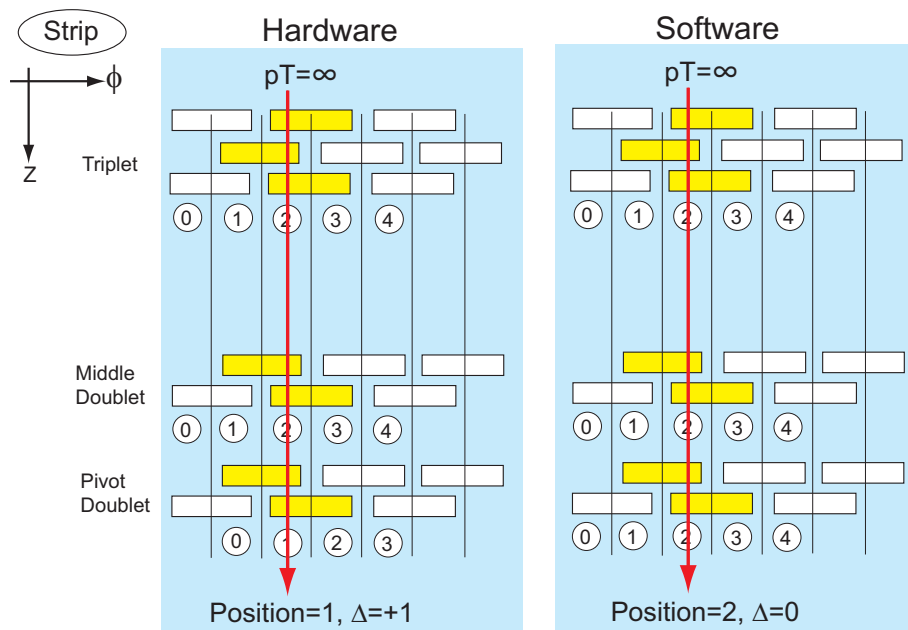


図 7.12: Strip の位置

結果は変化しなかった。出力されるイベントも毎回同じものであった。さらに SL に用意されているデバク用の出力ピンで、HPT からのデータがきちんと入力されているかを調べた。その結果 HPT からのデータは正しく SL に入力されていることが確認された。これらのことから、データラッチのタイミングによっての不一致とは考えにくくなった。そのため次に考えられるのが、SL 内のロジックに問題がないかということである。しかし、SL は単体テストで問題は生じていなかったため考えにくい。さらに考えられるのは、シミュレーションに問題がないかということである。しかし、SL の単体テストはシミュレーションによって得られた SL のインプットとアウトプットを用いて行われたためこれも考えにくい。そこで、SL の入力と出力をもう一度詳しく解析することにした。その結果明らかになったのは HPT ASIC のバグが原因であるということであった。今回の HPT に使われている HPT ASIC は、古いバージョンのためトラックセクタにバグがある。トラックセクタでは、ひとつの ASIC につき最大 2 つのトラックを pT の大きい順に選ぶ。High-pT トラックが 2 つあるときは 2 つとも選ばれるのだが、トラックが 1 つしかなかった場合 2 つめのトラックをどう選ぶかというところにバグが存在した。旧バージョンでは、High-pT が 1 つだけ存在した場合、2 つめのトラックはこの High-pT を除外して残りの Low-pT を選ぶということができない。2 つめのトラックも同じ High-pT のトラックを選んでしまい、2 本同じトラックが選ばれるのである。そのため、今回のテストにおいて HPT からのアウトプットには、同じトラックが 2 本入っているものが存在することが明らかになった。このようなデータが SL に入ってきた場合、SL の処理は同じトラックが 2 本検出されるということはないので、そのような場合はトラックが存在しなかったというように処理を行うようになっていた。このため SL ではシミュレーションでヒットありの結果がでていたものも、ヒットなしとして出力していた。ここで今回の HPT ASIC にあわせて SL のプログラムを変更し、2 本の同じトラックがあった場合は 1 本目のトラックを選んで処理をするようにした。変更後再度テストを行ったところ、今まで SL の出力においてヒットが出ていなかったイベントに関してヒットが現れた。この結果をシミュレーションの結果と比較したところ、すべて一致することが確認された。

これでトリガーパートのモジュールは全て接続され、その結果がシミュレーションと一致し、正当性が確

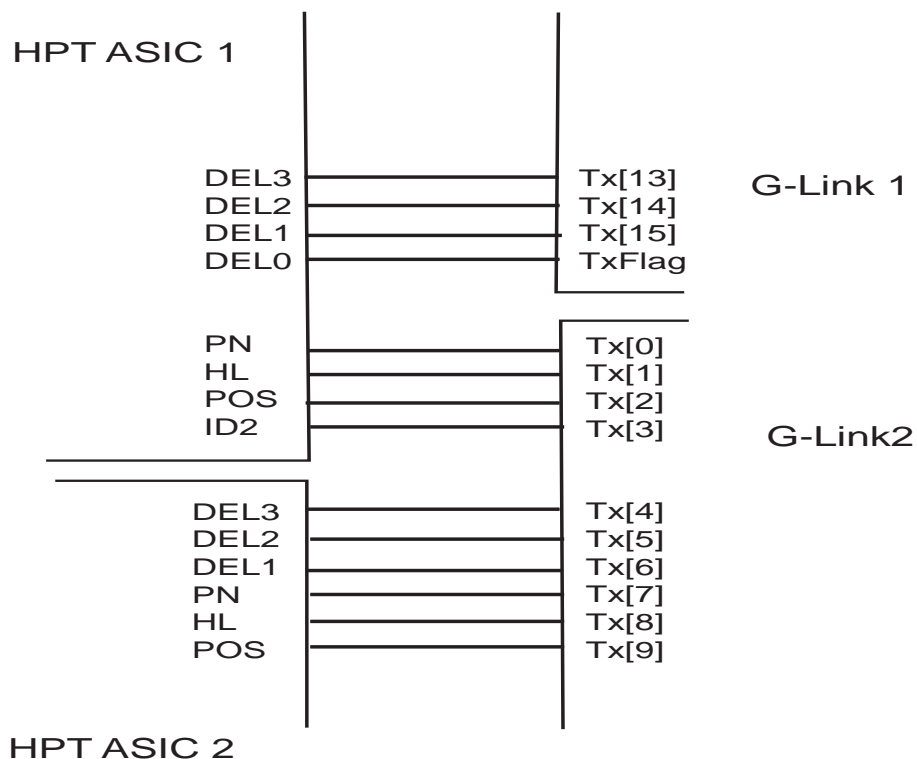


図 7.13: HPT ASIC - G-Link 間の接続と HPT のアウトプット (Wire 側の第一候補の出力と Strip 側の第一候補の出力)

認された。

次にレイテンシー測定の結果に関して述べる。レベル 1 トリガーは、40.08MHz のイベントレートから 100kHz までレートを下げ、トリガー判定を行う。そのためレベル 1 トリガーに要求されているレイテンシーは 1250ns である。今回のレイテンシー測定では、トリガーパートのエレクトロニクス全てを接続し、PP ASIC のインプットから SL のアウトプットまでの処理時間を、オシロスコープを用いて測定した。また、実際の実験時と今回のスライステストではケーブル長が異なるため、各 ASIC、リンク関係の IC の処理時間も個別に測定した。結果を表 7.2 に示す。

表の Requirement は要求されている値である。実測値においてケーブル、ファイバーに関しては、5ns/m としてケーブル長から求めたものである。また、ASD 以前に関しては要求値を用いた。表を見てわかるように、ASIC に関しては要求値よりも早く処理できることがわかる。LVDS や Glink などのリンク関係に関しては、ラッチのタイミングによる効果などで数 ns 遅くなったと考えられる。個々の値から全体を通して見てみると、TGC エレクトロニクス全体のレイテンシーは 1205ns となり、目標の 1250ns を満たしていることが確認された。

Part	Observation(ns)	Requirement(ns)
TOF to TGC	75	75
TGC	25	25
ASD	10	25
Cable	50	50
PP	43	50
SLB	49	75
LVDS(Tx,Rx)	83	75
Cable	75	75
HPT	55	75
Glink(Tx,Rx)	105	75
Fiber	450	450
SL	160	175
Cable	25	25
TOTAL	1205	1250

表 7.2: レイテンシーの実測値と要求値の比較

7.5 第2回統合テストのまとめ

第2回のスライステストでは、前回のシステムを ATLAS Online Software へ組み込むことから始まった。この組み込み作業において、前回の Property Control は変更することなしに組み込むことができたため、この点では比較的スムーズに行われた。ConfDB の XML ファイルの記述は、前回我々が作った仕様とは異なっていたため改めて記述することになったが、前回のファイルに設定用のレジスタに関してまとめられていたためそれほど困難な作業ではなかった。Crate Controller のプログラムは新たに記述することになったのだが、データの入出力部分に関しては殆ど変更する必要がなかったためその点ではスムーズに行うことができた。その他の比較プログラムも前回の Wire 側のものがあったため、Strip 側は多少の変更程度で済ませることができた。総合してソフトウェアシステムの構築は比較的容易にスムーズに行うことができた。

この新しいシステムを用いたトリガーパート全てのスライステストでは、Wire 側、Strip 側双方が問題なく動作することが確認され、SL での $r\text{-}\phi$ コインシデンスも正確に動作していることが確認された。入射粒子数は1つの場合と2つの場合で行った。結果を表 7.3 に載せる。また、トリガーシステムのレイテンシーも要求されている値を満たしていることが確認された。

今後は、入射粒子数を増やしてのテスト、およびバックグラウンドがある場合のテストを行う必要がある。また SSW が完成次第、リードアウトパートのテストができるように SSW の Property Control プログラムを整備しなければならない。さらに、SSW による PS Board 上の ASIC の設定などのコントロールパートのテストもできるようにしなければならない。

入射トラック数	サンプル数	エラー数 T
1	10000	0
2	10000	0

表 7.3: 第2回スライステストの最終結果

第8章 まとめ

ATLAS 実験では 40.08MHz という高レートでイベントが発生し、レベル1 トリガーのエレクトロニクスは高い処理能力と正確なトリガー処理が必要とされる。我々のグループは、このレベル1 トリガーの一部である TGC エレクトロニクスの研究開発を行ってきた。まずは ASIC の開発がはじめられ、その開発も終盤にさしかかり、数種類の ASIC は試作、試験を繰り返さずすべての ASIC が完成もしくは完成目前のものとなってきたため、これらの ASIC を搭載したモジュールのプロトタイプが試作され、単体テストが行われてきた。次の段階として、これらのプロトタイプモジュールを接続しての統合テストの段階に入ってきた。そのため、統合テストのソフトウェアシステムの構築が必要となり、それを用いての統合テストを行った。現在のところこの統合テストは2回に分けて行われ、ソフトウェアシステムもより充実したものへと改良を加えられてきた。このソフトウェアシステムは、将来量産が始まったときのテスト体系にも応用される可能性や、実際にチェンバーを接続してのビームテストにも使用される可能性があり、今後益々重要な役割を担っていくものと予想される。

第1回の統合テストでは、我々のグループ独自のソフトウェアシステムを構築し、トリガーパートの Wire 側のテストを行った。ソフトウェアシステムは Property Control、Run Control、テストベクタ変換プログラム、モジュールごとの比較プログラム、PS Board の設定プログラム、PT4 による読み出しプログラムから構成されている。これらを用いることによって簡単に効率よく統合テストができるようになった。統合テストの結果は、Wire 側の HPT までのハードウェア出力とシミュレーション結果が一致し互いの正当性を確認できた。また、LVDS Serializer/Deserializer、Glink Tx/Rx などの IC のリンク手法を確立し問題なく動作することが確認された。

第2回の統合テストでは、第1回で作られたシステムを ATLAS 実験共通の ATLAS Online Software への導入をはかり、将来 ATLAS 実験全体への TGC エレクトロニクスの組み込みの足がかりとなった。ATLAS Online Software への組み込み内容は、主に前回の Property Control を組み込むというものだったが双方のプログラムの引数が同一だったため、比較的容易に行うことができた。前回の Run Control にあたる部分は、Crate Controller として新たに記述することになったが共通部分も多くこれもスムーズに行うことができた。その他の周辺プログラムは前回と同様に使用した。また、Strip 側のプログラムを追加し、より充実したものとなった。統合テストはトリガーパート全てのモジュールを接続してのテストとなり、トリガーパート全体を通しての動作を確認することができた。結果は Wire、Strip 共に正確に動作し、SL での r - ϕ コインシデンスまで問題なく動作していることが、シミュレーション結果と比較することにより明らかになった。また同時に、トリガーシミュレーションも正確にシミュレートされていることが明らかになった。

次に重要になるのがレイテンシーである。正確に動作しながらもなおかつ高速に処理しなければならない。レイテンシーは、オシロスコープを使用して入出力の遅延時間により測定した。結果は要求されている 1250ns を上回る 1205ns で処理できていることが分かった。しかし、今回の測定では PP ASIC や SLB ASIC、HPT ASIC のディレイをゼロとして測定したため完全に満足できる値ではない。これらのディレイは設置場所によるケーブル長の違いから生じるディレイを調整するためにあるものなので全てのモジュールのディレイ値をゼロで使用することはないからである。しかし、今回の測定では LVDS Serializer/Deserializer、Glink Tx/Rx の処理時間が、ケーブルによるディレイなどでラッチのタイミングがずれた可能性があり、スベックよりも遅い値になっていたため、これらがうまくラッチできれば ASIC で設定するディレイを吸収できる可

能性もある。

今後のトリガーパートの統合テストは、より多くの異なるトラックをテストする必要がある。チェンバーが重なった部分でも正確に動作するかや、様々な位置にあるチェンバーに対して問題なく動作するかを検証しなければならない。

また、SSW が完成次第リードアウトパートを加えての統合テストをしなければならない。このとき TTC からの L1A を受けたときレベル 1 バッファからデランダムマイザーへと選択されたものだけが確実に送られるかなどが重要になってくる。また、ROD が 100kHz でデータを処理できるかということも重要になってくる。

さらに、CCI、HSC、SSW などのコントロールパートの動作もテストしなければならない。現在のところ CCI、HSC 間はテストされているが SSW は完成に至っていないため、SSW から PS Board へのアクセスなどが重要になってくる。

付録A ROD(Read Out Driver)の開発

統合テスト環境の構築とトリガーパートの統合テストと同時に、リードアウトパートに属する ROD の開発も行ってきた。ここでは ROD の開発について述べる。

A.1 ROD の役割

ROD は TGC エレクトロニクスシステムの中でリードアウトのデータが最終的に集まるモジュールで、USA15 内に設置される。ROD は 1 つのオクタントに 1 台設置され、最大 13 台の SSW から圧縮されたデータが転送されてくる。ROD は転送されてきたデータを TTC からの情報を元にイベントごとにまとめ、ヘッダー、トレーラーに必要な情報を付加して ROB(Read Out Buffer) に送信する。このとき VME 経由でデータの読み出しを行うことができ、イベントのモニターやヒストグラミングなどに利用される。

ROD は 100kHz でデータ処理を行わなければならない。これらのことから ROD に求められるのは正確なデータ処理と速度である。

A.2 基盤実装

ROD は 9U VME64x ボードで 13 個のデータ信号用入力ポートを持ち、それらは実験室内 (UXA15) に設置される SSW からのシリアル信号を受信し、それぞれ独立の FIFO メモリにデータを蓄える。また、1 個のトリガー情報信号 (TTC 信号) 用入力ポートを持ち、受信された信号情報は 2 つの FIFO メモリに蓄える。最大 13 個の FIFO メモリの蓄えられたデータは、ボード内のマイクロ・コンピュータに読み出され、2 つの FIFO メモリに蓄えられたトリガー情報を元に処理が行われ、出力用の FIFO メモリに蓄えられる。出力用 FIFO メモリに蓄えられたデータは、S-Link という光信号を伝達するモジュール経由で ROB に転送される。また、VME 経由で読み出すことも可能である。ROD の制御は VME を用いて行う。

以下に ROD 上の主なパートおよびパーツに関して述べる。

- データ入力ポート
データ入力ポートは Agilent HDMP-1032 Transmitter(G-Link) でシリアライズされた光信号を受信し、このデータを電気信号に変換した後 Agilent HDMP-1034 Receiver(G-Link) でデシリアライズし、FIFO メモリに蓄える。FIFO メモリは幅 36bit、深さ 16k のものを用いた。G-Link からのデータは 16bit+1bit のデータ幅だが、2 回のデータ転送によって 32bit 幅にして FIFO メモリに蓄える。
- トリガー情報入力ポート
トリガー情報入力ポートは、TTCrx Board を ROD 上に搭載することにより情報を取得し、データ幅 36bit、深さ 4k の 2 つの FIFO メモリに情報を格納する。情報の内容は、表 A.1 のようになっている。
- データ出力ポート
データ出力ポートはデータ幅 33bit、深さ 16k の FIFO メモリが置かれ、S-Link 出力モード時は自動的にデータが転送される。

35..24	23..0
BCID[11..0]	EVID[23..0]

35..32	31..24	23..0
none	EventType[7..0]	EventOrbit[23..0]

表 A.1: TTCrx トリガー情報

- VME 用データ出力ポート
VME 用データ出力ポートはデータ幅 32bit、深さ 64k の FIFO メモリが置かれ、VME からのリード命令によって読み出しが行われる。
- SH-4 [28]
HITACHI 社製のマイクロコンピュータで、ROD の中心となるパーツである。表 A.2 に SH4 の主な特徴をまとめた。

LSI	周波数 167MHz、電圧 1.8V(内部) 3.3V(IO)
CPU	32bit データバス、RISC タイプ命令セット
FPU	単精度、倍精度、IEEE754 に準拠
MMU (メモリ管理ユニット)	4G バイトアドレス空間
キャッシュ	命令 8k、オペランド 16k(8k を RAM として使用可)
BSC (バスステートコントローラ)	外部メモリアccessをサポート
DMAC (ダイレクトメモリアccessコントローラ)	4 チャンネル物理アドレスをサポート

表 A.2: SH-4 の特徴

- Dual-Port RAM
ROD 上の SH-4 と VME との通信に用いられる、データ幅 32bit、深さ 32k の RAM である。
- Control Status Register(CSR)
G-Link、S-Link、TTC、FIFO、SH-4 などのモニターを行うレジスタである。SH-4 から VME 経路でアクセスが可能で、設定を行うことができる。
- メモリ
64bit 幅、128M の SDRAM と 32bit 幅 2MB の FlashRAM が搭載されている。SDRAM には SH-4 のプログラムが格納され、FlashRAM にはボードの電源を投入した際のブート・プログラムが格納されている。ブート・プログラムは、SH-4 のプログラムであり、VME 経路で送信されてくる SH-4 のデータ処理プログラムを SDRAM に格納するものである。FlashRAM は専用ツールで JTAG を用いることによって書き込みを行う。
- ROD_BUSY 出力
入力ポートの FIFO メモリが、ほぼすべて書き込まれた場合に上位モジュールに対して ROD_BUSY 信号を出力する。

以上が主なパート、パーツであるが、これらの接続を模式的にあらわしたのが図 A.1 である。また ROD ボードの作成、ハードウェアの設定は三井造船システム技術株式会社 [29] に依頼した。

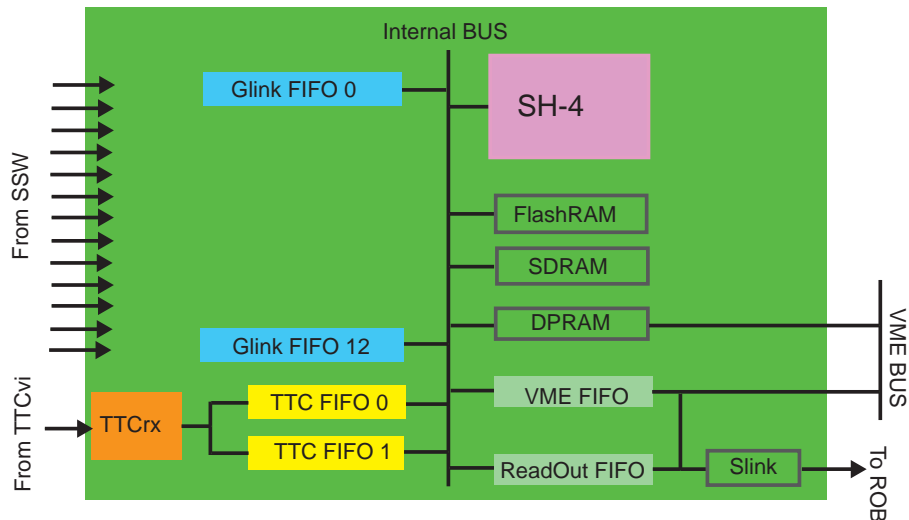


図 A.1: ROD の構成図

A.3 ソフトウェアの開発

ROD に必要なソフトウェアは大きく以下の 5 つに分類できる。

- ・ ROD の CSR に VME 経由でアクセスするためのプログラム
- ・ 出力用 FIFO から VME 経由でデータを読み込むプログラム
- ・ 電源投入直後の SH-4 プログラム
- ・ データ処理用の SH-4 プログラムを ROD に送るための VME アクセスプログラム
- ・ データ処理用の SH-4 プログラム

これら 5 種類のプログラム開発を行ったわけだが、この中で最も重要なものはデータ処理用の SH-4 プログラムである。ここから順に各プログラムに関して説明する。

- ROD の CSR に VME 経由でアクセスするためのプログラム

このプログラムでは CSR の読み出し、書き込みの両方ができるようにした。これによって、現在の状態のモニターができ、また用途に合わせた設定が可能である。プログラムは、あらかじめ CSR の各レジスタのアドレスを登録しておき、ユーザが CSR のアドレスを知らなくとも設定が行えるようにした。
- 出力用 FIFO から VME 経由でデータを読み込むプログラム

VME 経由で ROD の VME 読み出し用 FIFO のアドレスを参照し、FIFO 中のデータをモニターに表示させるプログラムである。読み出したい長さを指定できるようにした。
- 電源投入直後の SH-4 プログラム

SH-4 の設定で、電源投入直後に FlashRAM の内容を読み出して、その中のプログラムを実行するようにしておいた。これは、SH-4 内にもメモリはあるが (オペランドキャッシュを内蔵 RAM に設定した場合)、メモリ量が少ないためプログラムを格納しておくことができないためである。FlashRAM の

中の SH-4 用の初期プログラムは、Dual-Port RAM に VME からプログラムが送信されてきたらそのプログラムを SDRAM にコピーするというものにした。ここで FlashRAM と SDRAM を使用している理由は、それぞれのメモリの特徴をいかしてのことである。Flash メモリの特徴は、不揮発性で電源を切っても中に保存された情報は消えることがないが、データの書き換え可能な回数が少ないということである。一方 SDRAM の特徴は、多くのデータ書き換えに耐えることができるが、電源を切るとデータが消えてしまうということである。現在はデバックのため何度もプログラムを書き換える必要があり、FlashRAM だけでは耐えられない可能性があったため SDRAM を導入し、その都度プログラムを SDRAM に格納するようにした。

- データ処理用の SH-4 プログラムをボ ROD に送るための VME アクセスプログラム
バイナリコードにした SH-4 のプログラムを PC から VME 経由で ROD に送信するプログラムである。ROD 上の Dual-Port RAM のアドレスを指定しておき、そこにプログラムを書き込むようにした。
- データ処理用の SH-4 プログラム
このプログラムが ROD の役割であるイベントビルドを担うプログラムである。このプログラムは、VME 経由で命令を受け取り、命令に従ってスタート、ストップなどの動作を行うようにした。スタート命令がくると、まず SH-4 から CSR にアクセスし、初期設定を行い、その後、CSR の FIFO のステータス情報を読み、データが入ってきている場合は FIFO からデータの読み出しを行うようにした。FIFO が FULL になった場合は入力にマスクをかけデータを FIFO に書き込めないようにした。各 FIFO から読み込まれたデータは SH-4 内でひとつにまとめ、VME 出力用の FIFO に書き込むようにした。ここで VME-FIFO に書き込んでいるのはデバックのためである。本実験の場合は必要がない限り、ReadOut-FIFO にデータを書き込み、S-Link 経由で ROB に送信される。また、速度向上のため内蔵キャッシュを使用し、頻繁に使用するプログラムは SDRAM に何度もアクセスすることの無いようにした。

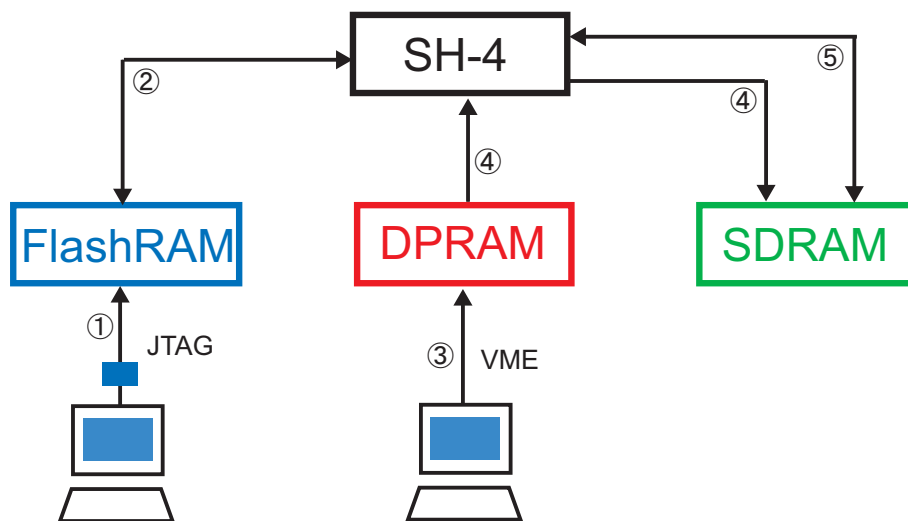


図 A.2: ROD の SH-4 に対するプログラムの流れ

A.4 性能測定とソフトウェア改良

完成したソフトウェアを用いて性能評価を行った。擬似データを PT3 [30] という汎用モジュールを使って作成し、G-Link の出力を持つ SL Tester というモジュールにデータを送り、そこから ROD にデータを送信した。

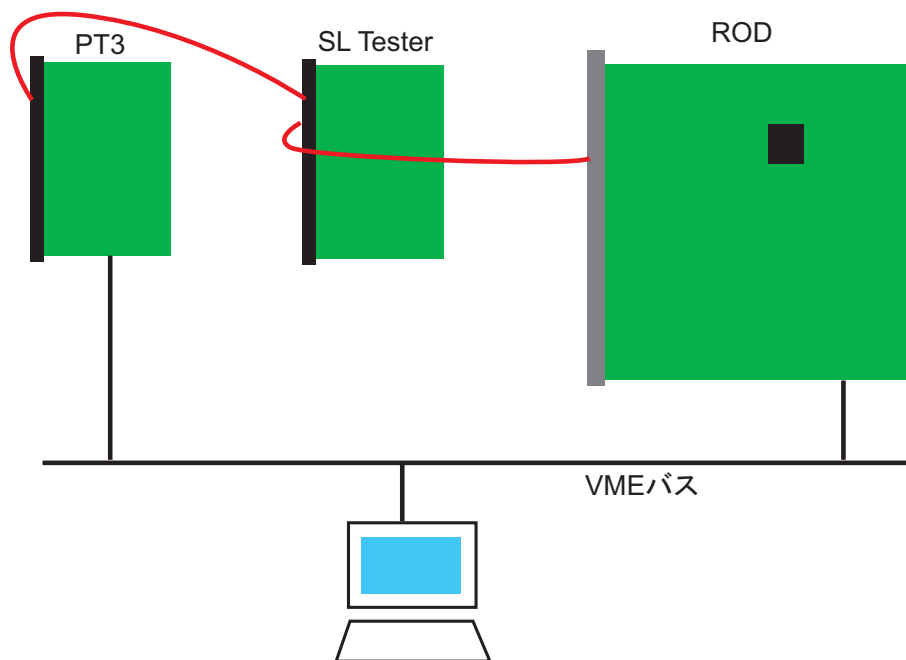


図 A.3: ROD 性能評価試験のセットアップ

このテストでは、ROD が 100kHz(10 μ s) でデータ処理が行えるかということを調べた。処理時間の測定は、CSR の ROD_BUSY 信号を制御するレジスタにアクセスし、スタート時に強制的に ROD_BUSY 信号を High にし、ストップ時に Low 状態に戻すことにより、その信号をオシロスコープで調べ、処理している時間を測定するという方法をとった。まずこの方法で、データ入力 FIFO から 32bit¹ データを読み込み (R)、SH-4 内でデータを確認し VME-FIFO にデータを書き込む (W) という一連の動作の処理速度を測定した。同様に、読み込みのみと書き込みのみ、ROD_BUSY の ON/OFF のみについても測定を行った。その結果を表 A.3 にまとめた。

機能	測定時間
32bit R/W + ROD_BUSY On/Off	680ns
32bit R + ROD_BUSY On/Off	528ns
32bit W + ROD_BUSY On/Off	492ns
ROD_BUSY On/Off	200ns

表 A.3: ROD の処理速度測定値

この結果から、入力 FIFO からのデータ読み込み、出力 FIFO へのデータの書き込み共に 300ns 前後かかっていることがわかった (ROD_BUSY の On/Off 時間を差し引いたもの)。また、この結果と ROD_BUSY

¹バス幅を 32bit にしているため、1 回のアクセスですむ

の On/Off にかかる時間を考慮すると、CPU からボードのバスにアクセスするのにおよそ 200ns かかっていることがわかる。これから、SH-4 でプログラムの処理にかかる時間よりも、バスアクセスが ROD の処理速度に大きく影響していると言える。SSW から ROD に入ってくるデータは 1 チャンネルにつきヘッダーとトレーラーだけでも 32bit × 4 だけある。入力ポートが 13 チャンネルあるので、これだけで 52 回バスアクセスをしなければならず、単純にバスアクセス時間をかけて処理速度を見積もるとすでに 10 μ s を超えてしまう。このため、バスアクセスの回数をできるだけ減らし、またアクセス速度を向上させる必要がある。まず、アクセス回数を減らす方法を考えてみた。SH-4 には DMA(Direct Memory Access) 機能がサポートされているので、この機能を利用を考えた。これにより CPU を介さずに 1 度のバスアクセスでメモリに対して最大 32byte のデータを扱うことができる。DMA を利用して、データを内蔵 RAM として使用できるキャッシュに読み込み、SH-4 内でデータ処理を行えないか考察した。その結果、DMA は外部メモリ間では可能だが、SH-4 のキャッシュを内蔵 RAM として設定しても、内蔵 RAM には DMA が使用できないことがわかった。入力 FIFO から出力 FIFO へ直接送信することはできても、データのチェックができないのでは複数のポートからのデータを、同じイベントごとにまとめることができないので使用することはできない。次にバスのアクセス速度の向上を考えた。現在のアクセス状況をロジックアナライザーで調べたものを図 A.4 に載せる。FIFO へのアクセスは 40.08MHz のバスクロック 4 回で行われている。また次にアクセスするまで 3 バスクロックかかっている。このアクセスサイクルを少なくする方法を考えた。現在 4 クロックかかっているのは、FIFO にアクセスしてリードイネーブル (RE) を出すまでに CSR へ FIFO のステータスを送るなどのハードウェアで記述された動作が入っているためである。そのためこのクロックを少なくするのは難しい。また考えられるのは、バスクロックをあげることである。しかし、バスクロックを上げた場合 ROD 上の FPGA などのロジックがついてこれなくなり、誤動作を起こす可能性が出てくるため今の状態では難しい。

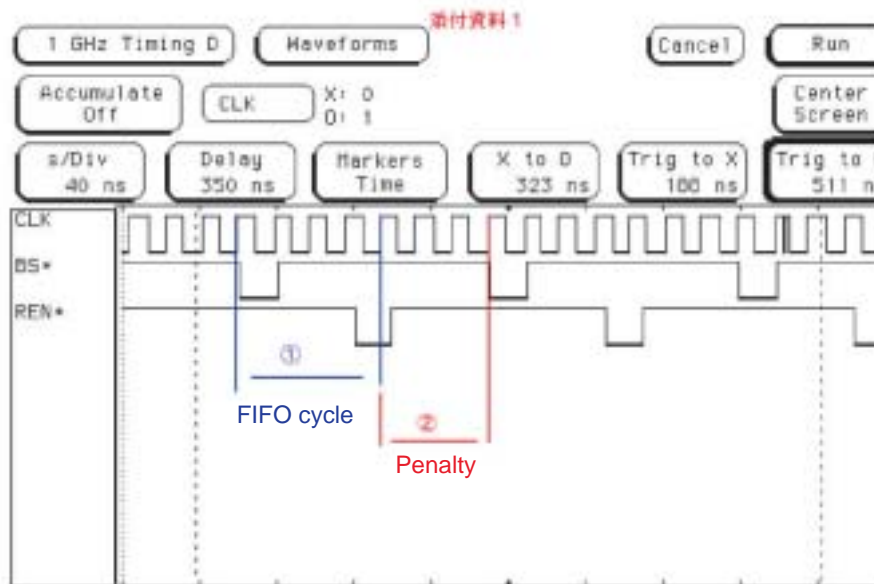


図 A.4: CPU から FIFO へのアクセスサイクル

A.5 改良点

前節の性能評価からわかるように、現在の ROD はこのままでは要求されている処理速度を満たすことができない。そこで4つの改良点を考えてみた。

- 入力ポート数の変更
現在の ROD のデザインのまま処理速度を満たすためには、単純に入力するポート数を少なくすることがあげられる。ROD1 台で処理するものを3台に分散させ、13の入力ポートの1/3だけ使用すれば要求値を満たせるものと考えられる。
- マイクロ・コンピュータの変更
現在使用している SH-4 は、内蔵 RAM に DMA を適用できないので内蔵 RAM に DMA が適用できるマイクロ・コンピュータに変更し、バスアクセス回数を減らす。CPU から内蔵 RAM へのアクセススピードは、バスアクセスより速いので大幅な速度の向上が見込まれる。
- FPGA のプログラムの変更
データを FIFO に入れる前に、SSW 以上の圧縮をかけてヘッダーなどの情報を減らすことによってデータ量が少なくなり、バスアクセス回数を減らすことができる。しかし、圧縮したデータをどの段階で戻すかが問題になる。
- SH-4 プログラムの改良
バスアクセス速度向上やアクセス回数を減らすこと以外に、SH-4 のプログラムの高速化も必要になる。現在のプログラムでは、VME からの命令を受けるために SH-4 に Dual-Port RAM をポーリングさせているため無駄な時間を生んでいる。この部分に割り込み処理を行わせることで余計な CPU の負荷を軽減させることができると考えられる。

A.6 ROD のまとめ

レベル1トリガーのリードアウトパートは、100kHz でデータの処理を行うことが要求されている。ROD は、TGC エレクトロニクスのリードアウトパートの最後のモジュールにあたり、大量のデータを受け取り、イベントビルドを行わなければならない。効率よくこの処理を行うため、マイクロ・コンピュータを用いる方法を採用し、プロトタイプを作成した。そして作成されたモジュールを用い、SH-4 のプログラム開発を行い処理能力を評価した。この結果、現在の状況では、要求された処理速度を満たすことができていないがわかった。しかし、前節で提案した改良を施せば 100kHz でのデータ処理が可能になると考えられる。

関連図書

- [1] CERN/LHCC/99-15, "ATLAS Detector and Physics Performance Technical Design Report Volume2", ATLAS TDR 15, 25 May 1999
- [2] CERN/LHCC/99-14, "ATLAS Detector and Physics Performance Technical Design Report Volume1", ATLAS TDR 14, 25 May 1999
- [3] CERN/LHCC/98-14, "ATLAS First Level Trigger Technical Design Report", ATLAS TDR 12, 30 June 1998
- [4] 香取 勇一, 修士学位論文"アトラス実験ミュオントリガーシステム用 IC の開発", Jan 2001
- [5] Ph.Farthouat, P.Gällnö, " TTC - VMEbus INTERFACE TTCvi - Mk2", RD12 Project, CERN EP - ATE, Rev 1.6, May 2000
- [6] J.Christiansen, A.Marchioro, P.Moreira and T.Toifl, " TTCrx Reference Manual -A Timing, Trigger and Control Receiver ASIC for LHC Detectors-", CERN - EP/MIC, Geneva Switzerland, Version 3.6, Jun 2002
- [7] VME Member, "VMEbus アーキテクチャマニュアル Revision C.1", CQ 出版,
- [8] O.Sasaki, " ATLAS Thin Gap Chamber Amplifier-Shaper-Discriminator ICs and ASD Boards", ATLAS TGC Collaboration KEK, Revision 1, Feb 1, 1999
- [9] T.Takemoto, " JRC(JTAG Route Controller) Data Sheet", ATLAS TGC Electronics Group ICEPP, Version 1.1, Sep 5, 2002
- [10] Radiation Hardness Assurance (RHA), <http://atlas.web.cern.ch/Atlas/GROUPS/FRONTEND/radhard.htm>
- [11] M.Huhtien,F.Faccio, " Computational method to estimate Single Event Upset rates in accelerator environment", CERN, <http://atlas.web.cern.ch/Atlas/GROUPS/FRONTEND/WWW/RAD/RadWebPage/Tutorial/seu.pdf>
- [12] M.Dentane, " ATLAS Policy on Radiation Tolerant Electronics", ATLAS Project Document, ATC-TE-QA-0001, 2000
- [13] "Radiation Levels Simulated for Various ATLAS Locations - Version18", Appendix 1 of ATLAS Project Document. ATC-TE-QA-0001, November 2000
- [14] "Safety Factors on Simulated Radiation Levels ", export from Appendix 1 of ATLAS Project Document. ATC-TE-QA-0001, 21 July 00
- [15] Joe Fabula and Howard Bogrow, "Total Ionizing Dose Performance of SRAM-based FPGAs and supporting PROM's", 2001 MAPLD International Conference, September 2000, http://klabs.org/richcontent/MAPLD00/Papers/Session_C/C2_Fabula_P.pdf
- [16] 戸塚 真義, 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", Jan 2002

- [17] 田中 賢一, 修士学位論文”ATLAS 実験ミュオントリガーシステム用 LSI の開発”, Jan 2002
- [18] 中村 佳央, 修士学位論文”ATLAS 実験ミュオントリガー用リモートコントロールシステムの開発”, Jan 2002
- [19] Texas Instruments Corporation, <http://www.tij.co.jp>
- [20] Agilent Technologies Inc, <http://www.agilent.com>
- [21] Infineon Technologies, <http://www.infinition.com>
- [22] N.Kruszynska, ” VME bridge driver vmehb for Linux”,
- [23] OMG CORBA, <http://www.corba.org>
- [24] 小泉 修, ”図解でわかる 分散オブジェクト技術のすべて”, 日本実業出版社, Oct 2001
- [25] Trolltech Inc, <http://www.trolltech.com/product/pt>
- [26] H.Kano, ” High-pT Chip (HITCHI-GA Version2) Data Sheet”, ATLAS TGC Electronics Group ICEPP, Revision 2.0, Aug 28, 2002
- [27] Bob Jones, ” A Proporsal for Converting the ATLAS DAQ Back-end Sub-system into an Open Source Project”, CERN EP/atd, 4th Feb 2000
- [28] HITACHI, <http://www.hitachi.co.jp>
- [29] 三井造船システム技術株式会社, <http://www.msr.co.jp>
- [30] 西田 昌平, 修士学位論文”ATLAS TGC エレクトロニクス読み出し系の開発”, Feb 2000
- [31] 石田 康明, 修士学位論文”Application of Novel technologies for Development of ATLAS End-Cap Muon Trigger Electronics System at the Large Hadron Collider”, Jan 2002
- [32] 一宮 亮, 修士学位論文”ATLAS 実験前後方ミュオントリガシステム用 Sector Logic の開発”, Feb 2001
- [33] Altera Corporation, <http://www.altera.co.jp>
- [34] Xilinx Corporation, <http://www.xilinx.co.jp>
- [35] CERN-LHCC-2002-34, ”8th Workshop on Electronics for LHC Experiments”, LHCC-G-014, 11 Oct 2002

謝辞

本研究を通して私は多くのことを学ぶことができ、様々な経験を得ることができました。本研究の機会を与えて頂き、適切な指導、助言を下された指導教官福永力助教授^aに深く感謝致します。本研究を通して多くの御指導、御助言を頂いた佐々木修氏^bに深く感謝致します。技術面から研究を支えて下さり、御指導頂いた池野正弘氏^bに深く感謝致します。本研究のソフトウェア開発において多くの御指導、御助言を頂いた前野忠嗣氏^dに深く感謝致します。TGC エレクトロニクスグループにおいて様々な御指導、御助言を頂いた、坂本宏氏^c、倉重久弥氏^d、新井康夫氏^b、蓮子和巳氏^{c2}、狩野博之氏^{c3}にも深く感謝致します。また、様々な面で御指導頂いた近藤敬比古氏^b、岩崎博行氏^b、田中秀治氏^b、小林富雄氏^c、石野雅也氏^c、竹下徹氏^e、長谷川庸司氏^e、石井恒次氏^d、越智敦彦氏^d他 ATLAS 日本グループの皆様にも深く感謝致します。ソフトウェアに関して多大な御指導、御協力を頂いた安芳次氏^b、仲吉一男氏^bに深く感謝致します。TGC エレクトロニクスグループに所属した当初御指導頂いた先輩方である石田康明氏、田中賢一氏、戸塚真義氏、溝内健太郎氏、中村佳央氏に深く感謝致します。共に研究を行ってきた一宮亮氏^d、辻伸介氏^f、竹本亨史氏^c、松本悠氏^c、里山典彦氏^e、片岡洋介氏^c、澁谷和弘氏^c、作間孝雄氏^g、高田徳之氏^eにも深く感謝致します。様々な面で御協力して頂いた TGC グループの 大下英敏氏^e、中川義徳氏^e、坂東隆哲氏^c、奥村和恵氏^d、杉本拓也氏^d、美馬孝行氏^d、荒滝陽二氏^dに深く感謝致します。また ROD の開発で御世話になった三井造船システム技術株式会社の皆様にも感謝致します。

以上、全ての方に心より感謝したいと思います。上記の方々の御協力がなければ私の研究生生活は成り立たなかったと思います。

また陰ながら様々な面で私の研究生生活を支えてくれた多くの友人、そして家族にこの場を借りて感謝したいと思います。

所属

東京都立大学 理学研究科^a

高エネルギー加速器研究機構 (KEK)^b

東京大学 素粒子物理国際研究センター (ICEPP)^c

神戸大学 自然科学研究科^d

信州大学 工学系研究科 (理学系)^e

京都大学 理学研究科^f

東京農工大学 工学研究科^g

²現在、理化学研究所研究員

³現在、高エネルギー加速器研究機構 (KEK) 研究員