

修士学位論文  
ATLAS実験前後方部レベル1  
ミューオントリガーシステムの動作検証

物理学専攻 粒子物理学研究室  
098s110s  
吹田 航一

神戸大学大学院理学研究科博士課程前期課程

平成23年2月12日

## 概要

スイスの CERN 研究所の地下に設けられた LHC 加速器が 2008 年 9 月に稼働し始めた。途中、加速器内でのヘリウム漏れの事故によりビーム周回の中断があったが、2009 年 11 月以降、加速器は再稼働し、2010 年 12 月の冬季シャットダウンまで問題なく稼働している。

LHC のビーム衝突点に設置された ATLAS 検出器を用いた ATLAS 実験は、LHC の稼働と同時にデータテイキングを始めた。ATLAS 検出器を構成する各検出器はどれも高い稼働率を誇り、十分なパフォーマンスを提供している。

実験において重要な LVL1 ミューオントリガーを担うミューオン検出器 TGC は、神戸大学が ATLAS 実験開始当初から関わってきた研究開発事項である。その TGC は 2010 年の実験において Trigger Efficiency の観点から見て高いパフォーマンスを提供している。

私は今回、ハードウェアの観点から LVL1 TGC ミューオントリガーのパフォーマンスの評価を行った。本論文はその研究についてまとめたものである。

本研究を進めるにあたり、その評価をするためのツールの作成、ツールの評価も同時に行った。

本研究の解析の結果、ハードウェアの観点からみた LVL1 TGC ミューオントリガーは約 99.9 % の稼働率を示した。これは Trigger Efficiency 等の観点からみた稼働率とコンシステントな結果である。

よって TGC は現在の状態を保ったまま、2011 年のデータテイキングに臨むことが可能だと結論づけられた。

# 目次

第 1 章	序論	1
第 2 章	ATLAS 実験	4
2.1	LHC 加速器	4
2.1.1	LHC 加速器のパラメータと現状	5
2.1.2	衝突点に設置された検出器	7
2.2	ATLAS 実験の目指す物理	7
2.2.1	Higgs 粒子の探索	7
2.2.2	超対称性粒子の探索	11
2.2.3	その他の物理	11
2.3	ATLAS 検出器の構成	12
2.3.1	ATLAS 検出器に要求される設計	12
2.3.2	ATLAS の座標系	13
2.3.3	マグネットシステム	14
2.3.4	内部飛跡検出器	16
2.3.5	カロリメータ	19
2.3.6	ミュオンスペクトロメータ	25
2.4	トリガーシステムおよび DAQ システム	29
2.4.1	レベル 1 トリガー (LVL1)	30
2.4.2	レベル 2 トリガー (LVL2)	31
2.4.3	イベントフィルター (EF)	32
2.5	ATLAS 実験の現状	32
2.5.1	陽子陽子衝突のデータ収集状況	32
2.5.2	検出器のパフォーマンス	33
第 3 章	前後方部レベル 1 ミュオントリガーシステム (TGC システム)	35
3.1	ミュオントリガーの重要性	35
3.2	ミュオン検出器 TGC	35
3.2.1	TGC チェンバーの構造	36
3.2.2	TGC の動作原理	37
3.2.3	TGC Wheel の構造	38
3.2.4	TGC の配置	38
3.3	トリガーのシステム	41
3.4	TGC エレクトロニクス	43
3.4.1	システム全体	43

3.4.2	Amplifier Shaper Discriminator Board(ASD)	46
3.4.3	Patch Panel ASIC(PP)	46
3.4.4	Slave Board ASIC(SLB)	47
3.4.5	High-Pt Board(HPT)	47
3.4.6	Sector Logic(SL)	48
3.4.7	Star Switch(SSW)	49
3.4.8	Read Out Driver Board(ROD)	49
3.5	TGC の現状	50
<b>第 4 章</b>	<b>オフラインデータ処理システム</b>	<b>52</b>
4.1	ATLAS 実験におけるデータの形式	52
4.1.1	グリッドシステム	52
4.1.2	RAW Data	52
4.1.3	Event Summary Data(ESD)	53
4.1.4	Analysis Object Data(AOD)	53
4.2	オフラインデコーディングの流れ	53
4.2.1	Raw Data Object(RDO)	53
4.2.2	Prep Raw Data(PRD)	53
4.2.3	シミュレーションにおけるデータ生成の流れ	54
4.2.4	ミュオンスペクトロメータのデータデコードの流れ	54
4.3	TGC オフラインデコードソフトウェアの概要	55
4.3.1	MuonCnv/MuonTGC_CnvTools	56
4.3.2	MuonCommissioning/MuonCommAlgs	58
4.3.3	MuonConditions/MuonCablings/TGCcabling12	58
4.3.4	MuonReconstruction/MuonRecExample	58
4.3.5	MuonReconstruction/MuonRecEvent/MuonRDO	59
4.3.6	MuonReconstruction/MuonRecEvent/MuonPrepRawData 及び Muon- TrigCoinData	65
4.3.7	MuonSimulation/MuonDigitization/TGC_Digitizaiton	66
4.4	TrigMuonCoinHierarchy	66
4.4.1	TrigMuonCoinHierarchy の概要	66
4.4.2	TrigMuonCoinHierarchy の改良	70
4.4.3	TrigMuonCoinHierarchy に関するまとめ	76
<b>第 5 章</b>	<b>実データを用いた TGC システムの動作検証</b>	<b>78</b>
5.1	TrigT1TGC を用いた動作検証の手法	78
5.1.1	Trigger/TrigT1/TrigT1TGC の概要	79
5.1.2	実データによる TrigT1TGC の動作	81
5.1.3	TrigT1TGC に実データのヒット情報を入力する方法	83
5.1.4	RDO with TrigT1TGC と RDO with StDec の入手方法	83
5.2	TGC システムの性能評価	85
5.2.1	データサンプル	85

5.2.2	解析環境 . . . . .	86
5.2.3	hitRDO(DoubletWire) . . . . .	86
5.2.4	hitRDO(DoubletStrip) . . . . .	87
5.2.5	hitRDO(TripletWire) . . . . .	88
5.2.6	hitRDO(TripletStrip) . . . . .	88
5.2.7	trackletRDO(DoubletWire) . . . . .	88
5.2.8	trackletRDO(DoubletStrip) . . . . .	89
5.2.9	trackletRDO(TripletWire) . . . . .	90
5.2.10	trackletRDO(TripletStrip) . . . . .	94
5.2.11	highptRDO(Wire) . . . . .	95
5.2.12	highptRDO(Strip) . . . . .	97
5.2.13	slRDO . . . . .	98
5.2.14	TGC システム全体としての性能 . . . . .	99
5.2.15	TrigMuonCoinHierarchy を用いた TrigT1TGC による実データ解析 の性能評価 . . . . .	99
5.2.16	TrigT1TGC を用いた TGC システムの動作検証のまとめ . . . . .	100
<b>第 6 章 まとめ</b>		<b>102</b>
<b>付 録 A TGC のナンバリングの補足</b>		<b>103</b>
A.1	RoI ナンバリング . . . . .	103
A.2	1/12 セクターナンバリング . . . . .	104
A.3	1/8 セクターナンバリング . . . . .	104
A.4	トリガーセクター ID . . . . .	105
<b>付 録 B 各種変数のまとめ</b>		<b>106</b>
B.1	Identifier . . . . .	106
B.1.1	OfflineID . . . . .	106
B.1.2	ReadoutID . . . . .	107
B.1.3	OnlineID . . . . .	108
B.2	MuonRDO . . . . .	109
B.2.1	hitRDO . . . . .	109
B.2.2	trackletRDO . . . . .	110
B.2.3	highptRDO . . . . .	111
B.2.4	slRDO . . . . .	112
B.3	MuonPRD . . . . .	113
B.3.1	hitPRD . . . . .	113
B.3.2	trackletPRD . . . . .	114
B.3.3	highptPRD . . . . .	115
B.3.4	slPRD . . . . .	116

<b>付録 C</b>	<b>各種パッケージのクラス階層構造図</b>	<b>117</b>
C.1	MuonCnv/MuonTGC_CnvTools . . . . .	117
C.2	MuonCommissioning/MuonCommAlgs . . . . .	118
C.3	MuonSimulation/MuonDigitization/TGC_Digitizaiton . . . . .	118
C.4	MuonConditions/MuonCablings/TGCCabling12 . . . . .	119
C.5	MuonReconstruction/MuonRecExample . . . . .	120
C.6	MuonReconstruction/MuonRecEvent/MuonRDO . . . . .	121
C.7	MuonReconstruction/MuonRecEvent/MuonPrepRawData . . . . .	122
C.8	MuonReconstruction/MuonRecEvent/MuonTrigCoinData . . . . .	122
<b>付録 D</b>	<b>各種データフォーマット</b>	<b>123</b>
D.1	SLB のデータフォーマット . . . . .	123

# 目 次

2.1	LHC 加速器の外観 [1] . . . . .	4
2.2	陽子陽子 7TeV 衝突における LHC の積分ルミノシティの遷移 [2] . . . . .	6
2.3	陽子陽子 7TeV 衝突における LHC の最高ルミノシティの遷移 [2] . . . . .	6
2.4	CMS 検出器 [3] . . . . .	7
2.5	ALICE 検出器 [4] . . . . .	7
2.6	LHCb 検出器 [5] . . . . .	7
2.7	標準模型と超対称性粒子 . . . . .	8
2.8	Higgs 粒子生成のファインマン・ダイアグラム [6] . . . . .	8
2.9	Higgs 粒子の質量と生成断面積の関係 [7] . . . . .	8
2.10	Higgs 粒子の質量と崩壊分岐比の関係 [7] . . . . .	10
2.11	ATLAS 実験における Higgs 粒子の発見能力 [8] . . . . .	11
2.12	ATLAS 検出器 [9] . . . . .	12
2.13	ATLAS で用いられる座標系 . . . . .	13
2.14	ATLAS 検出器のマグネットシステム [9] . . . . .	14
2.15	超電導ソレノイド磁石 [9] . . . . .	15
2.16	バレル部超電導トロイド磁石 [9] . . . . .	15
2.17	前後方部超電導トロイド磁石 [9] . . . . .	15
2.18	x-y 平面における磁場 ( $z=10.5\text{m}$ )[10] . . . . .	15
2.19	トロイド積分磁場強度と $\eta$ の関係 [10] . . . . .	15
2.20	内部飛跡検出器の全体像 [9] . . . . .	16
2.21	内部飛跡検出器の r-z 平面の断面図 [9] . . . . .	16
2.22	ピクセル型半導体検出器のピクセルモジュールの構造と写真 [9] . . . . .	17
2.23	ピクセル型半導体検出器の写真 (Endcap 部)[1] . . . . .	17
2.24	前後方部用 SCT モジュールの写真と構造 [9] . . . . .	18
2.25	バレル部用 SCT モジュールの写真と構造 [9] . . . . .	18
2.26	バレル部 SCT の写真 [1] . . . . .	18
2.27	バレル部 TRT の写真 [9] . . . . .	19
2.28	前後方部 TRT の写真 [9] . . . . .	19
2.29	カロリメータの全体像 [9] . . . . .	20
2.30	カロリメータ毎の吸収長 [9] . . . . .	20
2.31	バレル部 LAr 電磁カロリメータの写真 [9] . . . . .	21
2.32	バレル部 LAr 電磁カロリメータの構造スケッチ [9] . . . . .	21
2.33	前後方部 LAr 電磁カロリメータの写真 [9] . . . . .	21
2.34	バレル部 Tile ハドロンカロリメータ (Tile) の構造図 [9] . . . . .	22

2.35	バレル部 Tile ハドロンカロリメータ (Tile) の写真 [1]	22
2.36	エンドキャップ部 LAr ハドロンカロリメータ (HEC) の断面図 [9]	23
2.37	エンドキャップ部 LAr ハドロンカロリメータ (HEC) の写真 [9]	23
2.38	フォワード部 LAr ハドロンカロリメータ (FCal) の配置図 [9]	24
2.39	フォワード部 LAr ハドロンカロリメータ (FCal) の構造図 [9]	24
2.40	フォワード部 LAr ハドロンカロリメータ (FCal) の写真 [9]	24
2.41	ミューオンスペクトロメータの全体像 [9]	25
2.42	ミューオンスペクトロメータの z-y 平面の断面図から見た配置図 [9]	26
2.43	MDT の断面図 [9]	26
2.44	MDT チェンバーの構造 [9]	26
2.45	MDT の写真 (Aside)[1]	27
2.46	CSC の構造 [9]	28
2.47	CSC の写真 (茶色の部分)[1]	28
2.48	RPC の構造 [9]	28
2.49	RPC の写真 (インストールの様子)[1]	28
2.50	ATLAS 実験のトリガーシステムおよび DAQ システム図 [13]	29
2.51	ATLAS 実験のトリガーシステムとその実行場所 [14]	30
2.52	レベル 1 トリガー処理の流れ [13]	31
2.53	陽子陽子 7TeV 衝突における積分ルミノシティの遷移 [15]	33
2.54	陽子陽子 7TeV 衝突のイベントディスプレイ [1]	34
3.1	TGC の配置と構造図 [16]	36
3.2	TGC の断面図 [9]	36
3.3	ワイヤーとワイヤーサポート [16]	36
3.4	TGC の写真 (Big Wheel)[1]	38
3.5	TGC の Triplet の構造 (左) と Doublet の構造 (右)[9]	39
3.6	TGC の r-z 断面における全体像 [13]	39
3.7	TGC の 1/12 セクターの構造 (チェンバー単位)[17]	40
3.8	TGC のフォワードチェンバーとバックワードチェンバーの区別 [17]	41
3.9	TGC の r- $\phi$ 断面における Wheel 図 (M3)[17]	41
3.10	$p_T$ 測定の原理[16]	42
3.11	TGC のトリガー判定の方法 [16]	42
3.12	TGC エレクトロニクス [16]	43
3.13	トリガー系とリードアウト系のデータの流れ図 [16]	45
3.14	TGC エレクトロニクスの配置 (r-z 断面)[13]	46
3.15	ASD Board の写真 [16]	48
3.16	PS Board の写真 [16]	48
3.17	HPT Board 写真 [16]	49
3.18	SL の写真 [16]	49
3.19	SSW の写真 [16]	50
3.20	ROD の写真 [19]	50
3.21	TGC のトリガー効率の遷移 [18]	51



4.1	実データとシミュレーションのデータフロー . . . . .	54
4.2	シミュレーションのデータフロー [11] . . . . .	55
4.3	RAW Data から PRD までのデコーディングの流れ [12] . . . . .	56
4.4	MuonCnv/MuonTGC_CnvTools で使用されているデータ形式及び変数 . . . . .	57
4.5	TGC のヒットとトリガーの階層構造図 . . . . .	67
4.6	Trigger/TrigAnalysis/TrigMuonCoinHierarchy のクラス階層構造 [12] . . . . .	68
4.7	SLB(Doublet) のブロック図と mask2[13] . . . . .	70
4.8	TrigMuonCoinHierarchy を用いた各階層のトリガーセクター毎の対応状況 (run167844、mask2 を考慮した場合) . . . . .	77
5.1	TrigMuonCoinHierarchy と TrigT1TGC を用いた TGC トリガーシステム の検証のイメージ図 . . . . .	79
5.2	TrigT1TGC を MC で走らせるときのイメージ図 . . . . .	79
5.3	Trigger/TrigT1/TrigT1TGC のクラス階層構造 [12] . . . . .	80
5.4	TrigT1TGC を実データで走らせるときのイメージ図 . . . . .	81
5.5	TrigT1TGC を用いた TGC システムの動作検証の流れ . . . . .	82
5.6	TrackletDoubletWire における delta 値の対応関係 . . . . .	89
5.7	TrackletDoubletStrip における delta 値の対応関係 . . . . .	90
5.8	TrackletTripletWire における rphi 値の対応関係 . . . . .	91
5.9	TrackletTripletWire で対応がとれない Cside sector=6 の SLB の TGC 上 での位置 . . . . .	93
5.10	TrackletTripletStrip における seg 値の対応関係 . . . . .	94
5.11	HighptWire における sub 値の対応関係 . . . . .	96
5.12	HighptWire における実機から出力される delta 値 (Aside でシミュレーシ ョンと一致しなかったトリガー情報) . . . . .	97
A.1	RoI のナンバリング . . . . .	103
A.2	1/12 セクターにおける phi 番号 . . . . .	104
A.3	1/8 セクターにおける octant 番号 . . . . .	104
A.4	トリガーセクター ID のナンバリング . . . . .	105
C.1	MuonCnv/MuonTGC_CnvTools のクラス階層構造 [12] . . . . .	117
C.2	MuonCommissioning/MuonCommAlgs のクラス階層構造 [12] . . . . .	118
C.3	MuonSimulation/MuonDigitization/TGC_Digitization のクラス階層構造 [12] . . . . .	118
C.4	MuonConditions/MuonCablings/TGCCabling12 のクラス階層構造 [12] . . . . .	119
C.5	MuonReconstruction/MuonRecExample のクラス階層構造 [12] . . . . .	120
C.6	MuonReconstruction/MuonRecEvent/MuonRDO のクラス階層構造 [12] . . . . .	121
C.7	MuonReconstruction/MuonRecEvent/MuonPrepRawData のクラス階層構 造 [12] . . . . .	122
C.8	MuonReconstruction/MuonRecEvent/MuonTrigCoinData のクラス階層構 造 [12] . . . . .	122

D.1 SLB のデータフォーマット . . . . .	123
------------------------------	-----

# 表 目 次

2.1	LHC 加速器の主要パラメータ	5
2.2	LHC 加速器の 2010 年 10 月 31 日現在の主なパラメータ	6
2.3	陽子陽子衝突のデータ収集状況	32
2.4	各検出器の稼働状況 [2]	34
3.1	TGC のパラメータ	37
4.1	mask2 ファイルのフォーマット	72
4.2	TrigMuonCoinHierarchy を用いた run167844 の対応結果 (mask2 を考慮しない場合)	75
4.3	TrigMuonCoinHierarchy を用いた run167844 の対応結果 (mask2 を考慮した場合)	75
5.1	hitRDO(DoubletWire) の対応関係	86
5.2	hitRDO(DoubletWire) の対応関係 (データフォーマット間で情報が一致しないイベントを除外した場合)	87
5.3	hitRDO(DoubletStrip) の対応関係	87
5.4	hitRDO(TripletWire) の対応関係	88
5.5	hitRDO(TripletStrip) の対応関係	88
5.6	trackletRDO(DoubletWire) の対応関係	88
5.7	trackletRDO(DoubletStrip) の対応関係	89
5.8	trackletRDO(TripletWire) の対応関係	90
5.9	trackletRDO(TripletWire) の対応関係 (ノイジーチャンネルを除外した場合)	91
5.10	trackletRDO(TripletWire) の対応関係 (ノイジーチャンネルと疑わしい SLB を除外した場合)	92
5.11	trackletRDO(TripletStrip) の対応関係	94
5.12	trackletRDO(TripletStrip) の対応関係 (seg の反転を考慮した場合)	95
5.13	highptRDO(Wire) の対応関係	95
5.14	highptRDO(Wire) の対応関係 (TrackletTripletWire でケーブリングスワップにより対応がとれなかったイベントを除外した場合)	96
5.15	highptRDO(Strip) の対応関係	98
5.16	slRDO の対応関係	98
5.17	slRDO の対応関係 (SL における muplus 値のバグを除外した場合)	98
B.1	OfflineID の変数一覧	106
B.2	ReadoutID の変数一覧	107

---

B.3	OnlineID の変数一覧 . . . . .	108
B.4	hitRDO の変数一覧 . . . . .	109
B.5	trackletRDO の変数一覧 . . . . .	110
B.6	highptRDO の変数一覧 . . . . .	111
B.7	slRDO の変数一覧 . . . . .	112
B.8	hitPRD の変数一覧 . . . . .	113
B.9	trackletPRD の変数一覧 . . . . .	114
B.10	highptPRD の変数一覧 . . . . .	115
B.11	slPRD の変数一覧 . . . . .	116

# 第1章 序論

2008年9月、約10年間の準備期間を経てCERN研究所に設置されたLHC加速器が稼働し始めた。途中、トラブルでシャットダウンを余儀なくされたが、2009年11月に再稼働した。それから2010年まではトラブルなく順調にビームの衝突が実現し、2011年1月現在、冬季シャットダウンのためビーム衝突実験は一時中断され、2月からの再稼働に向けてコミッショニング中である。また2011年のビーム衝突で記録された重心系陽子陽子衝突エネルギー7TeVは2011年1月現在、世界記録である。

LHC内のビーム衝突点に設置された検出器の1つであるATLAS検出器を用いたATLAS実験もLHCの稼働開始と同時にデータテイキングが開始された。2010年冬現在、ATLAS検出器内に設置された各検出器は高いパフォーマンスを実験に提供している。またATLAS実験グループとして、2010年以降公式論文が数件発表され、実験が順調に進んでいることを物語っている。

実験ではミューオンを精度よく検出することが求められる。なぜならば、ミューオンは実験で得られる膨大なデータを、それが興味ある物理イベントを含むか否かを選別する作業(イベントトリガー)に用いられるからである。ミューオンによって成されるトリガーをミューオントリガーと呼び、ATLAS実験では3段階のトリガーによってデータを選別するシステムの中の第1段階のトリガー(LVL1トリガー)で用いられる。陽子同士を衝突させるLHC加速器の中で最も多い反応が終状態に複数のジェット(粒子の多重発生現象)が観測される事象(QCDジェット事象)である。しかし新しい物理、例えば標準模型Higgs粒子の崩壊モードの中に終状態にレプトン対を含むイベントがある。終状態がジェットのイベントが豊富に存在する中で、この崩壊モードはHiggs粒子の探索モードの中で最もQCDバックグラウンドとの選別がしやすい。そのためミューオンがどの程度効率よく観測されているかは実験の中で重要なパラメータとなる。

このミューオントリガーで用いられる検出器の1つがTGCミューオン検出器である。TGCは検出器の前後方に飛来してきたミューオンを用いて、ミューオントリガーを出す。ATLAS実験に計画段階から参加してきた神戸大学は、2000年から2004年にかけてKEKにて約500枚のTGCを生産した。そして2002年から2005年の間に、神戸大学にてその全数の性能検査を行い、随時CERNへ発送した。2005年から2007年の間にCERNにてTGCのセクターの組み立てを行い、2008年にATLAS検出器への据え付けを完了した。この間、TGCからの情報を処理するハードウェアの開発及びそのTGCへの据え付け、それらを制御するソフトウェアの開発を行ってきた。そして実験が始まった現在も、TGCの動作チェック、実験状況に応じたハードウェア、ソフトウェアの改良、TGCからデータのクオリティ評価、加えて10年後に計画されているLHCのアップグレードに合わせたTGCの制御回路の開発を中心に研究活動が行われている。私は、中でもTGCから出力されるトリガー情報の性能評価、及びそれらを制御するソフトウェアのデバッグ等

を中心に研究を行ってきた。

前述のとおりミュオントリガーは ATLAS 実験において鍵となる存在である。ミュオントリガーの性能が落ちれば、それは後のデータ解析にまで伝播し、実験結果の信頼度を落としかねない。そのため常時システムの性能は評価され、一定に保たれるように管理されていなければならない。また以下のような理由から、それはツール化されていることが望ましい。

- TGC のチャンネル数が膨大である  
TGC には合計約 32 万の読み出しチャンネルが搭載されている。これを人の手でもって常時チェックすることは不可能である。
- 迅速性、緊急性が求められる  
実験はほぼ 24 時間休みなしでデータテイキングが行われ、かつグローバルなランにおいては TGC のコミショニングのみに多くの時間を割くことはできない。また問題の発生が示唆された場合には迅速かつ限られた時間で原因の追究及び解決が求められる。
- 複数人間が評価する可能性がある  
例えば TGC シフターのように複数人間が評価することになれば、その手法は共通でなければいけない。

私は今回、TGC 内の制御回路の観点から TGC のトリガーシステムを動作検証するツールを開発した。これは実験で TGC を通過したミュオンのイベントをトリガー用の制御回路がどのようにトリガー処理を行うかをシミュレートし、そのトリガー結果が実験中に制御回路自身が行っているトリガー処理の結果と一致しているかどうかを調べるツールである。また同時に実データの制御回路間での情報の伝達が正しく行われているかを調べるツールの改良に取り組んだ。そして 2010 年の実データで解析を行い、2010 年のデータテイキングを終えた ATLAS 検出器の TGC システムが、現状のまま 2011 年のデータテイキングに臨むことができるかどうかの判断をした。

検証により制御回路内のバグ、シミュレーションソフトウェアのバグ等も発見され、それらの諸問題を解決した 2011 年 1 月現在、TGC のトリガーシステムは約 99.9 % の稼働率で稼働していると、本研究で見積もられた。これより、TGC のトリガーシステム及び、TGC 本体において現状、問題はなく、この状態のまま 2011 年 2 月以降のデータテイキングに臨むことができると結論づけられた。

本論文では私の研究内容を中心に、以下の流れでまとめる。

- 第 2 章 ATLAS 実験  
LHC 加速器及び ATLAS 実験の概要について述べ、ATLAS 実験の現状を報告する。
- 第 3 章 前後方部レベル 1 ミュオントリガーシステム (TGC システム)  
本論文の主題となる TGC システムについてまとめる。始めに TGC の構造などを記述し、その後 TGC システムの概要、及び TGC システムで用いられる主なハードウェアについて説明する。

---

- 第4章 オフラインデータ処理システム

TGCで検出されたミュオン、及びTGCシステムについてオフラインで解析、評価等を行うオフラインデータ処理について述べる。始めにATLAS実験におけるデータの取り扱い方について説明し、その後TGCに関わるオフラインソフトウェアの概要を述べる。最後に現在、オフラインでTGCシステムの評価に用いられているTrigMuonCoinHierarchyパッケージについて説明し、私がパッケージに施した改良内容、そして2010年最新のデータを用いたTGCシステムの評価をTrigMuonCoinHierarchyパッケージを用いて行った結果について述べる。

- 第5章 実データを用いたTGCシステムの動作検証

本論文の主題となる部分である。この章ではこれまでシミュレーションでTGCシステムを再現するために用いられてきたTrigT1TGCパッケージを、実データを使って動かすことで、TGCの現状をシミュレーションと比較することを試み、TGCシステムの現状の性能を評価したことについてまとめる。まずTrigT1TGCパッケージの概要、及び実データを用いた動作について説明する。最後に2010年の実データとTrigT1TGCパッケージを用いてTGCシステムの現状を定量的に見積もる。

- 第6章 まとめ

この章では、本論文のまとめ及び結論について述べる。

## 第2章 ATLAS実験

ATLAS実験はCERN研究所のLHC加速器を用いて行われている実験で、世界37カ国の研究機関や大学から約3,000人の研究者が参加している。約10年間の準備期間を経て、2009年の冬から本格的に稼働を始めた。2010年冬季現在のLHCビームの重心系エネルギーは7TeV(3.5TeV+3.5TeV)で、これは世界記録である。また2013年には重心系エネルギー14TeVを達成するためにLHCの修理が約1年間に渡って行われる予定である。

この章では始めにLHC加速器について説明し、加速器の衝突点に設置されたATLAS検出器を用いたATLAS実験について説明する。

### 2.1 LHC加速器

LHC(Large Hadron Collider)は、スイスとフランスの国境にあるCERN(欧州合同原子核共同研究機構)研究所の地下約100mに設置された陽子陽子衝突型加速器である。LHC加速器の外観を図2.1に示す。

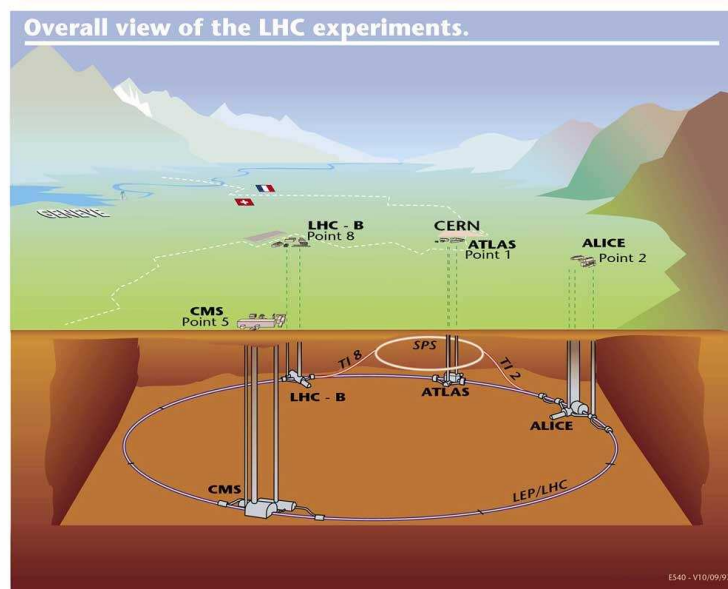


図 2.1: LHC 加速器の外観 [1]



## 2.1.1 LHC 加速器のパラメータと現状

この加速器は周長約 27km の世界最大の加速器である。LHC は 2008 年 9 月 10 日に最初のビームの周回に成功した。しかし同年 9 月 19 日にヘリウム漏れにより稼働停止を余儀なくされた。これは超電導マグネット間をつなぐ超電導ケーブルの接合の不備により、そこに抵抗領域が発生したことが原因である。結果、発生した熱がヘリウム容器に穴をあけ、そこから大量の液体ヘリウムが漏れたことにより二次災害が発生し、大きなトラブルを引き起こした。その後、原因の追究と復旧作業が行われ、2009 年 11 月 LHC は再び稼働を開始し、以降、現在に至るまで順調に稼働している。LHC の主なパラメータを、表 2.1 に示す。

Parameter	Design Value
主リング周長	26.66km
重心系エネルギー (陽子+陽子)	7.0TeV+7.0TeV
ルミノシティ	$10^{34} \text{cm}^{-2} \text{sec}^{-1}$
ルミノシティ寿命	10 時間
入射エネルギー	450GeV
衝突頻度	40.08MHz
バンチ間隔	24.95nsec
1 バンチあたりの陽子数	$10^{11}$ 個
バンチの長さ	75mm
バンチ数	2835 個
バンチ衝突当たりの陽子衝突	23
衝突点のビーム半径	$16 \mu\text{m}$
衝突角度	$200 \mu\text{rad}$

表 2.1: LHC 加速器の主要パラメータ

ここに 2010 年現在までの LHC 加速器の運転状況を示す。

まず、図 2.2 の緑色のヒストグラムが日毎に増加する LHC の積分ルミノシティを表している。データ量が増えた要因として LHC 内のバンチ数増加やビームをより絞ったことが挙げられる。結果最高ルミノシティが順調に増加していったためである。7TeV 衝突における最高ルミノシティの遷移の様子を図 2.3 に示す。

2010 年冬季シャットダウンまでに LHC は積分ルミノシティにして約  $48 \text{pb}^{-1}$  だけのデータ量を供給している。また 2010 年 10 月 31 日現在の LHC の主なパラメータを表 2.2 に示す。

LHC は 2010 年 10 月 31 日で、陽子陽子ビーム衝突の実験を終了し、それから冬季シャットダウンまでは重イオンを用いた衝突実験を行った。2011 年は 2 月から陽子陽子ビーム衝突の実験を再開する。

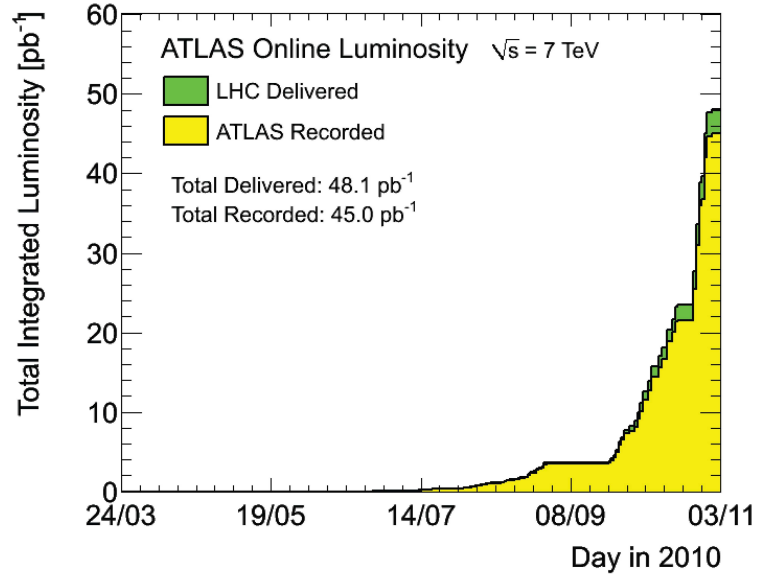


図 2.2: 陽子陽子 7TeV 衝突における LHC の積分ルミノシティの遷移 [2]

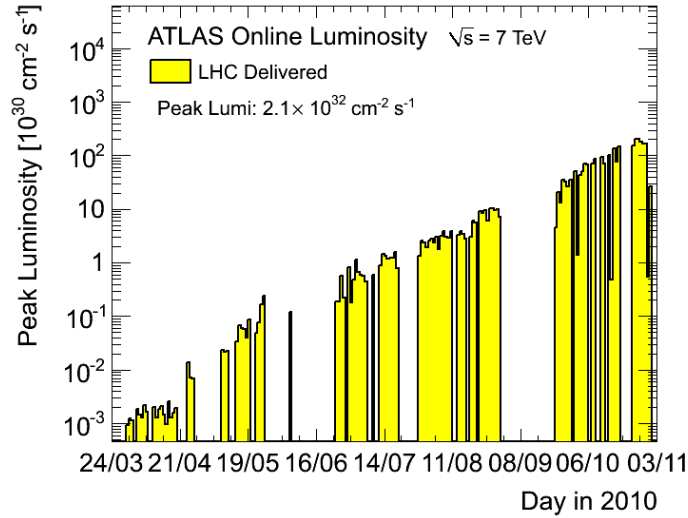


図 2.3: 陽子陽子 7TeV 衝突における LHC の最高ルミノシティの遷移 [2]

Parameter	Value
重心系エネルギー (陽子+陽子)	3.5TeV+3.5TeV
ルミノシティ	$2 \times 10^{32} \text{cm}^{-2} \text{sec}^{-1}$
積分ルミノシティ	48pb <sup>-1</sup>
バンチ数	368 個

表 2.2: LHC 加速器の 2010 年 10 月 31 日現在の主なパラメータ

### 2.1.2 衝突点に設置された検出器

LHC には図 2.1 に示した通り 4 つのビーム衝突点があり、それぞれに検出器が設置されている。後述する大型汎用検出器 ATLAS(A Troidal LHC ApparatuS)、ATLAS より小型の汎用検出器である CMS(the Compact Muon Solenoid(図 2.4))、重イオン衝突実験用検出器の ALICE(A Large Ion Collider Experiment(図 2.5))、B-Physics に特化した検出器 LHCb(図 2.6) が設置されている。ATLAS 検出器については次節以降で詳細を述べる。

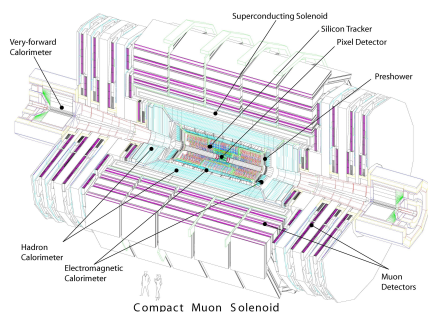


図 2.4: CMS 検出器 [3]

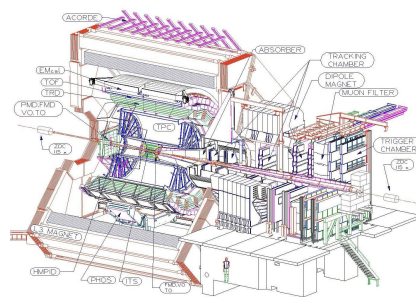


図 2.5: ALICE 検出器 [4]

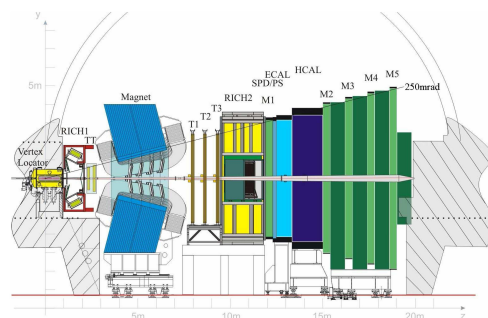


図 2.6: LHCb 検出器 [5]

## 2.2 ATLAS 実験の目指す物理

ATLAS 実験において発見が期待される物理の主なものが標準模型 Higgs 粒子と超対称性粒子である。本節では、主に Higgs 粒子について説明する。

### 2.2.1 Higgs 粒子の探索

#### 標準模型 Higgs 粒子

現在の素粒子物理学の基礎を成しているのが標準模型である。標準模型に現れる粒子の一覧は図 2.7 の左の粒子であり、クォーク (黄色)、レプトン (赤色)、力を媒介するゲージ

ボゾン (緑色) と、質量の起源となる Higgs 粒子 (青色) から成る。Higgs 粒子はゲージボゾンとフェルミオンに質量を与える粒子である。

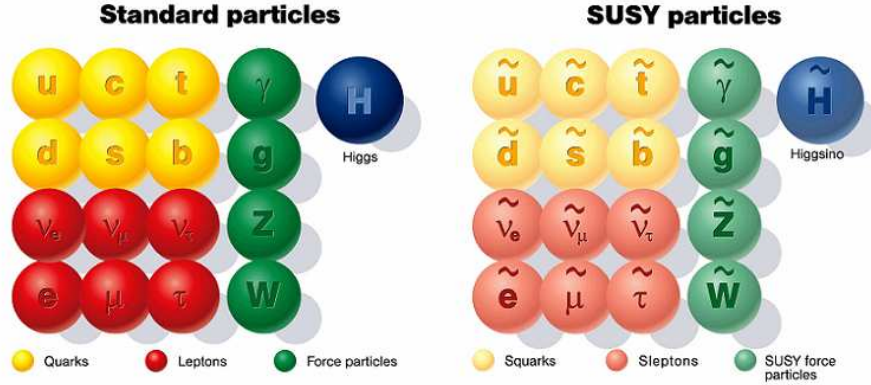


図 2.7: 標準模型と超対称性粒子

Higgs 粒子はその存在が予言されているものの、現在までにその存在が実験的に確認されていない。ATLAS 実験ではこの Higgs 粒子の発見が期待されている。

**Higgs 粒子の生成過程**

Higgs 粒子は重い粒子と結合しやすいため、LHC では主に以下の 4 つの生成過程が考えられる。それぞれのファインマン・ダイアグラムを図 2.8 に、生成断面積と質量の関係を図 2.9 に示す。

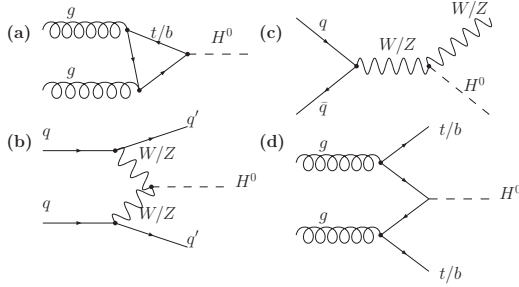


図 2.8: Higgs 粒子生成のファインマン・ダイアグラム [6]

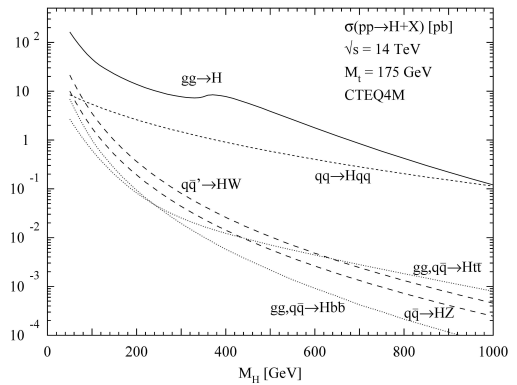


図 2.9: Higgs 粒子の質量と生成断面積の関係 [7]

以下、それぞれの生成過程を断面積が大きい順に解説する。

1.  $gg \rightarrow tt \rightarrow H^0$  (gluon fusion)

トップクォークのループを介した過程で、最も断面積が大きい。その反面、Higgs 粒

子が崩壊して出来る粒子以外に大きな横運動量  $p_T$  を持つ粒子がなく、バックグラウンドとの選別が非常に難しい。

2.  $qq \rightarrow qqWW \rightarrow qqH^0$  (VBF:vector boson fusion)  
クォークから放出されたゲージボゾンから Higgs 粒子が生成される。断面積も比較的大きく、反跳したクォークに起因する大きな  $p_T$  を持つジェットが 2 本観測される特徴があり、イベントの選別が比較的行きやすい。
3.  $q\bar{q} \rightarrow (W/Z)H^0$  (W/Z associate production)  
クォークの対消滅で生成されたゲージボゾンから、更に Higgs 粒子が放射される過程。終状態にゲージボゾン (W/Z) が観測される特徴がある。
4.  $gg \rightarrow t\bar{t}H^0$  (top associate production)  
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。断面積は小さいが、特徴のあるトップクォーク対を終状態に含んでいる。

### Higgs 粒子の崩壊モードと探索

Higgs 粒子の探索には以下の崩壊モードが有効と期待されている。それぞれ簡単に説明する。

1.  $H^0 \rightarrow \gamma\gamma$  ( $m_H < 150\text{GeV}$ )  
この質量領域では、 $b\bar{b}$ 、 $c\bar{c}$ 、 $\tau^+\tau^-$  が支配的であるが、陽子陽子衝突から引き起こされる QCD ジェットバックグラウンドと区別することが難しい。そこで不変質量  $m_{\gamma\gamma}$  分布を求めると、Higgs 粒子の質量が鋭いピークとして存在する。これは稀崩壊ではあるが  $H^0 \rightarrow \gamma\gamma$  の観測である。エネルギー及び角度分解能の優れた電磁カロリメータが必要となる。
2.  $H^0 \rightarrow \tau^+\tau^-$  ( $m_H < 150\text{GeV}$ )  
Higgs 粒子が軽い場合、発見に有効とされているのがこのチャンネルである。 $\gamma\gamma$  よりも崩壊確率が高く、W/Z fusion の生成過程を考えることでバックグラウンドと区別することができる。この場合、Higgs のピークはバックグラウンドである Z のピークのテールに現れる。 $\tau$  の崩壊にはニュートリノが含まれるので、 $E_T^{miss}$  の精度が重要になる。 $\tau$  の同定には  $\tau^- \rightarrow \pi^-\nu\pi^0$  や  $\tau^- \rightarrow \pi^-\pi^+\pi^-\nu\pi^0$  のようなハドロニックな崩壊モードが主に用いられる。 $\tau$  のハドロニックな崩壊の特徴は 1 本または 3 本の細いジェットの存在であり、それをトリガーに用いる。
3.  $H^0 \rightarrow ZZ^* \rightarrow l^+l^-l^+l^-$  ( $120 < m_H < 180\text{GeV}$ )  
このモードは、最も綺麗なピークが得られるモードの 1 つである。1 つのレプトン対に対しては、不変質量  $m_Z$  に等しい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、検出器には運動量、エネルギーに対する高い分解能が求められる。  
バックグラウンドとしては、 $ZZ^*$ 、 $Z\gamma^*$ 、 $t\bar{t}$ 、 $Zb\bar{b}$  がある。このうち  $ZZ^*$ 、 $Z\gamma^*$  は生成断面積もそれほど大きくなく、質量分布がピークにならない。 $t\bar{t}$ 、 $Zb\bar{b}$  はそれぞれ

レプトン対が、 $Z$  起源または  $Z^*$  起源であるという条件をつけることによって取り除くことができる。

4.  $H^0 \rightarrow WW^* \rightarrow l^+l^-\nu\nu$  ( $150 < m_H < 180\text{GeV}$ )

170GeV 付近では  $W$  への崩壊が支配的となる。終状態に  $\nu$  を含むため直接 Higgs 粒子の質量の再構成をすることはできないため、 $\nu$  に起因する消失横方向エネルギー  $E_T^{miss}$  を精密に測定することが必要になる。Higgs 粒子の探索には有効なモードである。

5.  $H^0 \rightarrow ZZ \rightarrow l^+l^-l^+l^-$  ( $180 < m_H < 800\text{GeV}$ )

このモードが最も綺麗なピークを得られる。2 組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ることができるため、信頼性の高いモードである。ただし、Higgs 粒子の質量が大きくなるにつれて崩壊幅が大きくなるため有効性が落ちる。

6.  $H^0 \rightarrow ZZ \rightarrow l^+l^-\nu\nu$  ( $400\text{GeV} < m_H$ )

この領域では、このモードの方が  $H^0 \rightarrow ZZ^* \rightarrow l^+l^-l^+l^-$  よりも分岐比が約 6 倍も高い。 $\nu\nu$  の不変質量は再構成することが出来ないため、これに起因する消失横方向エネルギー  $E_T^{miss}$  を精密に測定することが必要になる。

7.  $H^0 \rightarrow WW \rightarrow l\nu jj$ ,  $H^0 \rightarrow ZZ \rightarrow lljj$  ( $600\text{GeV} < m_H$ )

この領域ではこれらのモードが  $H^0 \rightarrow ZZ^* \rightarrow l^+l^-l^+l^-$  に比べて、 $H^0 \rightarrow WW \rightarrow l\nu jj$  は約 150 倍、 $H^0 \rightarrow ZZ \rightarrow lljj$  は約 20 倍の分岐比を持つ。これらのモードでは、バックグラウンドと区別するために Higgs 粒子が  $W/Z$  fusion によって生成された場合を考える。この過程では、散乱角前方にクォークによる 2 つのジェットが存在するのが特徴的で、このジェットを指標とすることでバックグラウンドを排除することができる。

以上の崩壊分岐比と Higgs の質量との関係を図 2.10 に示す。

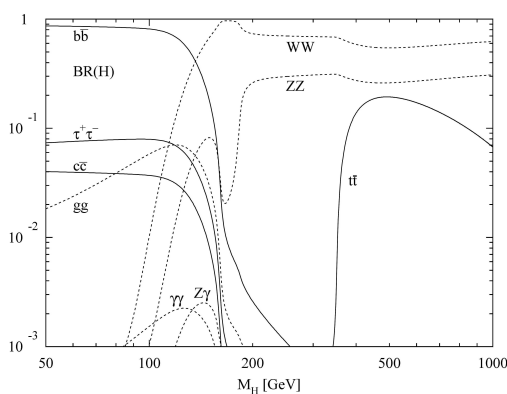


図 2.10: Higgs 粒子の質量と崩壊分岐比の関係 [7]

## ATLAS 実験における Higgs 粒子の発見能力

ATLAS 実験における Higgs 粒子の発見の可能性を図 2.11 に示す。

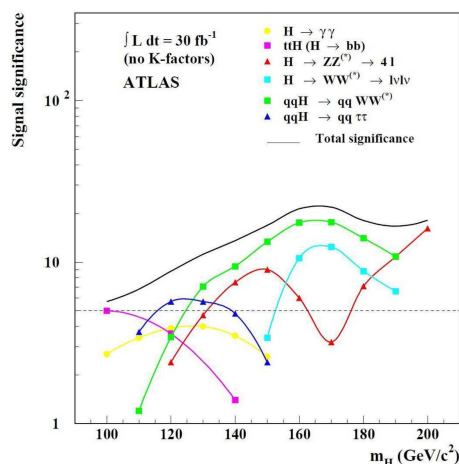


図 2.11: ATLAS 実験における Higgs 粒子の発見能力 [8]

図 2.11 は  $30\text{fb}^{-1}$  のデータが溜まった時の ATLAS における Higgs 粒子の発見能力を示している。図の横軸は Higgs 粒子の質量を、縦軸はシグナル量をバックグラウンド量の平方根で割った値を示している。この値が 5 の値を超えれば、その事象が  $5\sigma$  の確からしさと確認できたとすることができる。そのため図 2.11 より、 $30\text{fb}^{-1}$  だけの積分ルミノシティのデータがあれば、Higgs 粒子が  $5\sigma$  の確からしさと発見できると予想されている。

図 2.11 によれば、Higgs 粒子の質量が低い領域では、 $H^0 \rightarrow WW^*$ 、 $H^0 \rightarrow ZZ^*$  などの事象が発見可能である。

### 2.2.2 超対称性粒子の探索

#### 超対称性粒子 (SUSY)

超対称性理論では、標準模型のそれぞれ粒子とスピンの異なるスーパーパートナーと呼ばれる超対称性粒子の存在が予言されている。図 2.7 の右に、標準模型粒子と対を成す形で並べられているのが超対称性粒子である。

超対称性粒子にはクォークの超対称性粒子であるスクォーク、レプトンの超対称性粒子であるスレプトン、グルーオンの超対称性粒子であるグルイーノなどがある。もしもこの理論が正しければ、LHC では強い相互作用をするスクォークやグルイーノ対が大量に生成され、それらの発見が期待されている。

### 2.2.3 その他の物理

ATLAS 実験の目的は上で述べた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 の精密検証などもある。

標準模型の検証はボトム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の生成断面積の測定を通して行われる。

## 2.3 ATLAS 検出器の構成

ATLAS 検出器は、円筒形の巨大な汎用検出器であり、その直径は約 25m、長さ約 44m、総重量が約 7,000t である。検出器の全体像を図 2.12 に示す。

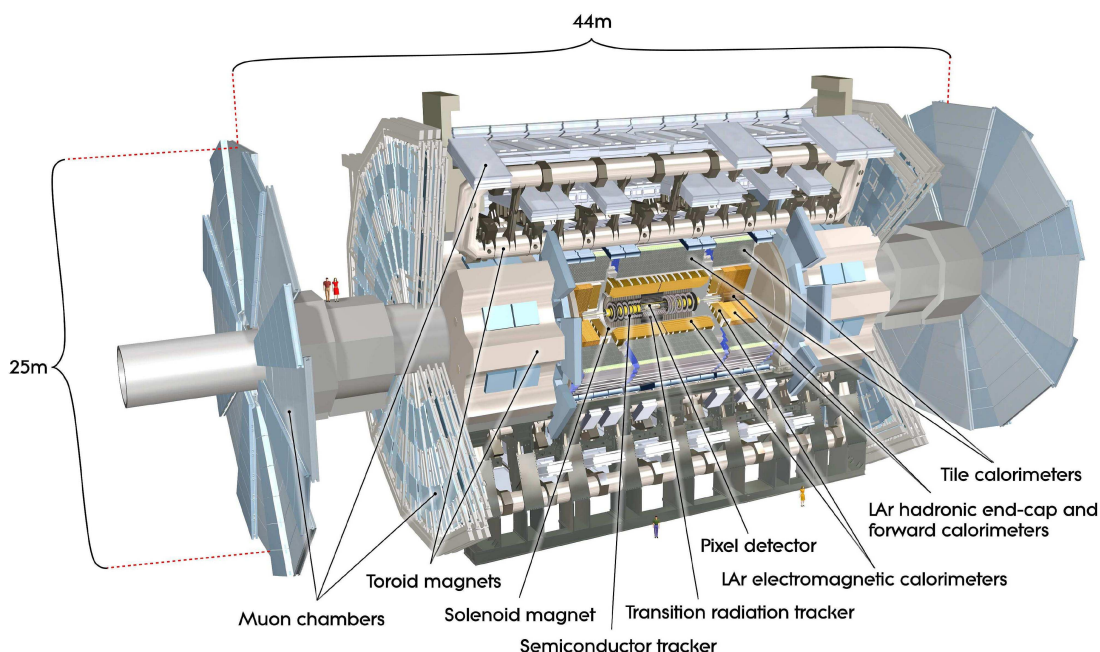


図 2.12: ATLAS 検出器 [9]

検出器は内側から、内部飛跡検出器、カロリメータ、ミューオンスペクトロメータで構成され、検出器の間にはマグネットが設置されている。以降、ATLAS 検出器に要求される設計を述べた後、ATLAS の座標系と、マグネット、ATLAS 検出器を構成する各検出器についてまとめる。

### 2.3.1 ATLAS 検出器に要求される設計

LHC の高ルミノシティにおいて、光子、電子、ミューオン、ジェット、 $E_T^{miss}$  等の信号を高速かつ正確に処理できるように、ATLAS 検出器は次のような要求を満たすように設計されている。

- 広範囲の擬ラピディティ $\eta$ (後述) と完全な方位角のカバー



- 広範囲の擬ラピディティと全方位角に対する感知領域を持つ電磁カロリメータによる電子とフォトンのエネルギーと位置の高精度測定と、ハドロンカロリメータによる高精度なジェットと横方向消失エネルギー  $E_T^{miss}$  の測定
- ミューオンスペクトロメータによる高精度のミューオンの運動量測定
- 内部飛跡検出器による荷電粒子の運動量測定とバーテックスの精密検出
- 高頻度のイベントを逃すことなく処理するシステム
- 大量のバックグラウンドに対する耐放射性

### 2.3.2 ATLAS の座標系

ATLAS 実験で用いられる座標系を図 2.13 に示す。

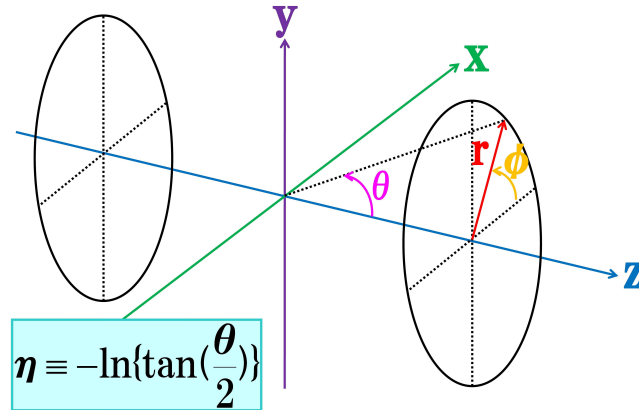


図 2.13: ATLAS で用いられる座標系

原点は衝突点にとられる。LHC のビーム軸方向に  $z$  軸をとり、それと直交する形で、地面と水平方向には LHC リングの内側に向かって  $x$  軸を、垂直な方向には天頂方向に向かって  $y$  軸がとられる。このとき検出器の両サイドをそれぞれ、 $z$  が正の側を *Aside*、 $z$  が負の側を *Cside* と呼ぶ。また円筒型の検出器に対して、円筒座標系がとられ、図 2.13 のように  $r$ 、 $\theta$ 、 $\phi$  がとられる。本実験では、 $\theta$  と  $\phi$  はそれぞれ、 $0 \leq \theta \leq \pi$ 、 $0 \leq \phi < 2\pi$  の範囲をとる。もう 1 つ重要なパラメータとして、擬ラピディティ (pseudo rapidity) があり、これは  $\eta \equiv -\ln(\tan(\theta/2))$  で定義される。 $\eta$  を用いることによって、衝突点由来の飛来粒子の頻度が、広い範囲に渡り  $\eta$  の値に依らず一様になる。 $\eta$  がとる範囲は、 $-\infty < \eta < +\infty$  である。円筒の筒の部分にあたる  $|\eta| \leq 1$  の領域をバレル部、円筒の両サイドにあたる  $|\eta| > 1$  の領域をエンドキャップ部と呼ぶ。さらにエンドキャップ部は、 $1 < |\eta| < 1.9$  の領域を狭義のエンドキャップ部、 $|\eta| > 1.9$  をフォワード部と呼ぶことがある。以降、本論文ではエンドキャップはこの狭義のエンドキャップのことを表し、広義のエンドキャップを前後方部と呼ぶことにする。

### 2.3.3 マグネットシステム

ATLAS のマグネットシステムは 3 種類の超電導磁石から成る。ATLAS のマグネットの構成を図 2.14 に示す。

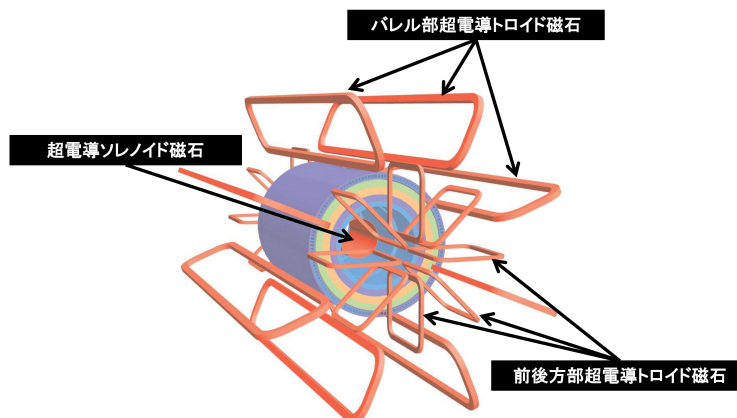


図 2.14: ATLAS 検出器のマグネットシステム [9]

図 2.14 の内側からビーム軸に沿った磁場を発生させるソレノイド磁石と、 $\phi$  方向の磁場を発生させるバレル及び前後方部のトロイド磁石である。以下に 3 種のマグネットについて述べる。

#### 超電導ソレノイド磁石

超電導ソレノイド磁石は内部飛跡検出器とバレル部電磁カロリメータの間に設置され、ビーム軸に沿った 2T の磁束密度を内部飛跡検出器に提供する。コイル部分は  $z$  方向の長さ 5.3m、直径 2.4m、径方向の厚さ 45mm の円筒形をしている。この薄さは電磁カロリメータの内側の物質量を最低限にして、カロリメータの分解能を損なわないようにするというコンセプトのもとで開発された。超電導ソレノイド磁石の写真を図 2.15 に示す。

#### バレル部超電導トロイド磁石

トロイド磁石は、バレル部と前後方部に分かれている。両方の磁石で  $0 \leq \eta \leq 2.7$  の領域をカバーするトロイダル磁場を発生させる。バレル部はハドロンカロリメータ及び、前後方部トロイド磁石を囲うように  $\phi$  に対する 8 回対称で設置されている。提供できる磁束密度は平均で 0.5T である。大きさは、長さ 25.3m、内径 9.4m、外径 20.1m である。バレル部超電導トロイド磁石の写真を図 2.16 に示す。

前後方部超電導トロイド磁石

前後方部トロイド磁石は Aside、Cside のそれぞれで、最も内側にある前後方部ミュオンスペクトロメータの外側に設置されている。前後方部のトロイドもバレル部のトロイドと同じく 8 回対称になっている。提供する平均の磁束密度の大きさは 1.0T である。磁石の大きさは、長さが 5.0m、内径 1.65m、外径 10.7m である。前後方部トロイド磁石の写真を図 2.17 に示す。



図 2.15: 超電導ソレノイド磁石 [9]

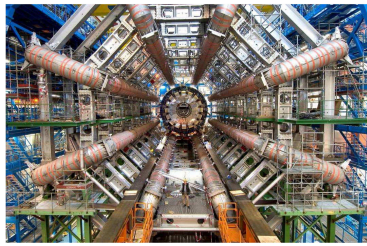


図 2.16: バレル部超電導トロイド磁石 [9]

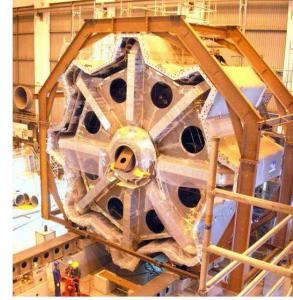


図 2.17: 前後方部超電導トロイド磁石 [9]

理想的には磁場は  $\phi$  方向成分のみを持つべきであるが、図 2.18 と図 2.19 に示すように、境界部の磁場は複雑で  $\phi$  方向だけでなく、 $r$  方向にも成分を持っていることがわかる。また、境界部以外でも強度は低いものの  $r$  方向成分の発生が避けられない。このため、ミュオンは  $\phi$  方向にもわずかながら曲げられることになり、ATLAS のミュオンスペクトロメータでは  $\phi$  方向の曲がりも測定できるように設計されている。

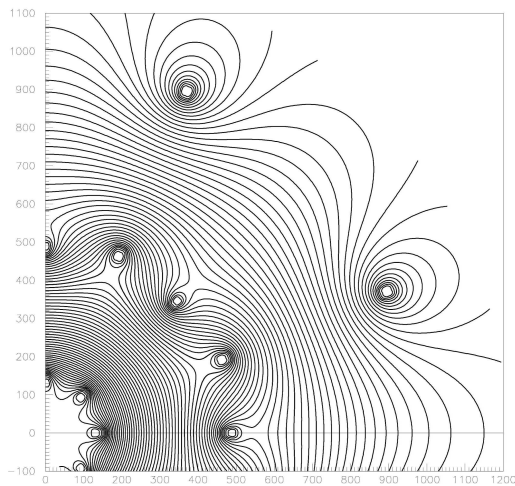


図 2.18: x-y 平面における磁場 ( $z=10.5\text{m}$ ) [10]

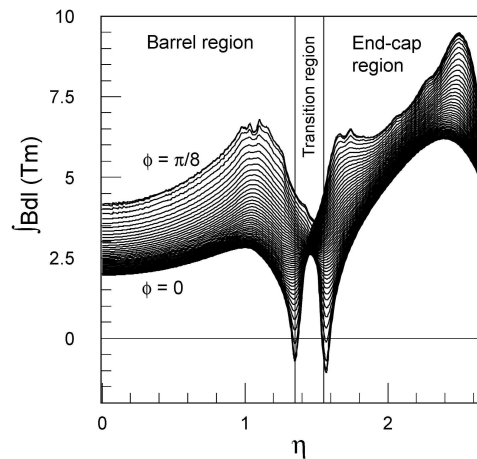


図 2.19: トロイド積分磁場強度と  $\eta$  の関係 [10]

### 2.3.4 内部飛跡検出器

内部飛跡検出器は超電導ソレノイド磁石の内側の領域、すなわちビームラインに最も隣接する検出器である。内部飛跡検出器の主な役割は、荷電粒子の軌跡を再構成し、磁場中での曲がり具合を知ることで、粒子の  $p_T$  や電荷を測定することである。内部飛跡検出器の全体像を図 2.20 に示す。

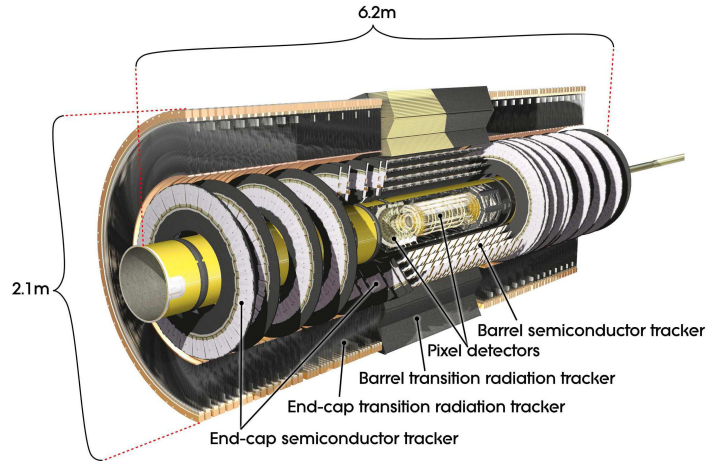


図 2.20: 内部飛跡検出器の全体像 [9]

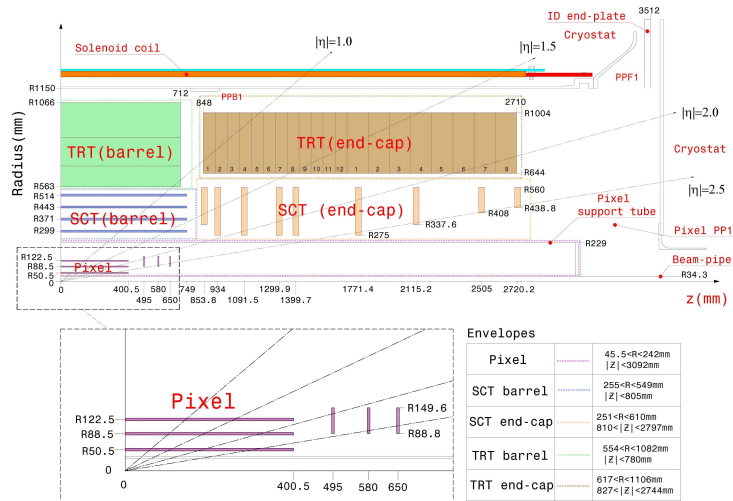


図 2.21: 内部飛跡検出器の r-z 平面の断面図 [9]

図 2.21 は内部飛跡検出器を r-z 方向の断面図として図式したもので、内側からピクセル型半導体検出器 (Pixel)、シリコンストリップ検出器 (SCT)、遷移輻射飛跡検出器 (TRT) の 3 種類から構成されている。これら 3 つの検出器で  $\phi$  方向全て、及び  $|\eta| < 2.5$  の領域

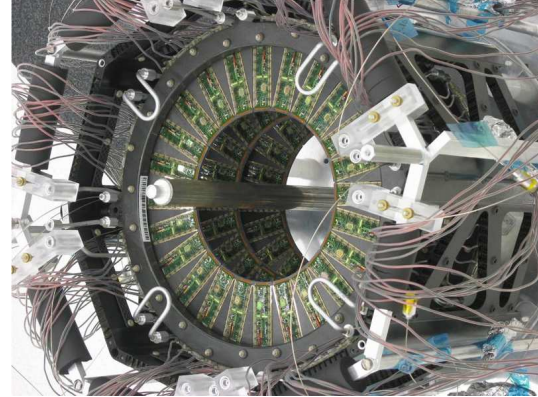
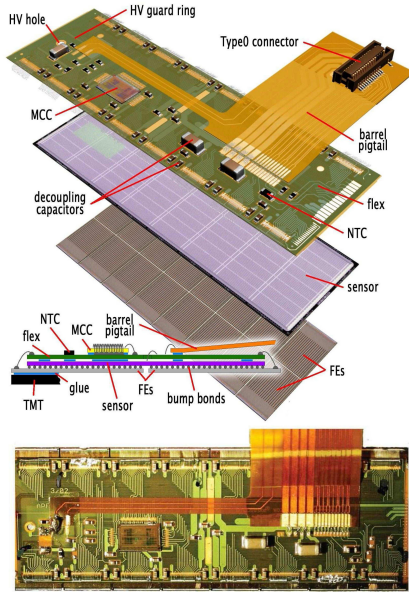


図 2.23: ピクセル型半導体検出器の写真 (Endcap 部)[1]

図 2.22: ピクセル型半導体検出器のピクセルモジュールの構成と写真 [9]

をカバーしている。

以下、順に 3 つの検出器について概要を述べる。

### ピクセル型半導体検出器 (Pixel)

Pixel は衝突点からもっとも近い領域に設置される検出器である。 $50\mu\text{m} \times 400\mu\text{m}$  または、 $50\mu\text{m} \times 600\mu\text{m}$  を 1 つのピクセルとし、1 つのモジュールは 47,268 個のピクセルから成る。図 2.22 にピクセルモジュールの構成と写真を示す。

モジュールで 2 次元の位置測定が可能で、バレル部 3 層、前後方部が両サイド 3 層ずつで構成され、そのピクセルの総数は 8,000 万にも及ぶ。位置分解能は  $r-\phi$  方向で  $10\mu\text{m}$ 、 $z$  方向で  $115\mu\text{m}$  と高い性能を示している。これを Endcap 領域に搭載した Pixel 検出器の写真を図 2.23 に示す。

### シリコンストリップ検出器 (Semi Conductor Tracker, SCT)

1 つの SCT モジュールは 6~7cm 各の長方形シリコンウエハ 2 枚からなり、それぞれのウエハには  $80\mu\text{m}$  幅の読み出し用アルミニウムストリップが敷き詰められており、1 枚あたり 768 本のストリップがある。1 つのモジュール内で 2 枚のウエハをずらして設置することで 2 次元の位置測定が可能となっている。SCT 前後方部用モジュールの写真とその構成を図 2.24 に示す。写真は左から、前後方部の outer、middle、inner に設置されている。SCT バレル部用モジュールの写真と構成については図 2.25 に示す。

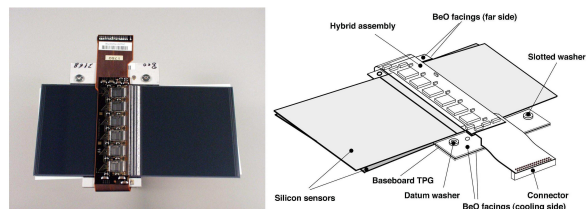
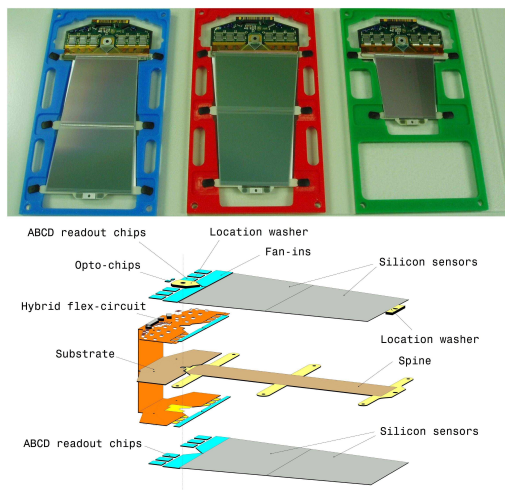


図 2.25: バレル部用 SCT モジュールの写真と構造 [9]

図 2.24: 前後方部用 SCT モジュールの写真と構造 [9]

バレル部ではモジュールを円筒形の筒面上に 4 層並べ、前後方部では円盤状に 9 層ずつ両サイドに設置されている。モジュールの総数は 4,088 であり、読み出しチャンネルは全部で約 600 万に及ぶ。位置分解能は  $r$ - $\phi$  方向で  $17\mu\text{m}$ 、 $z$  方向で  $580\mu\text{m}$  である。モジュールを実際に搭載した写真を図 2.26 に示す。

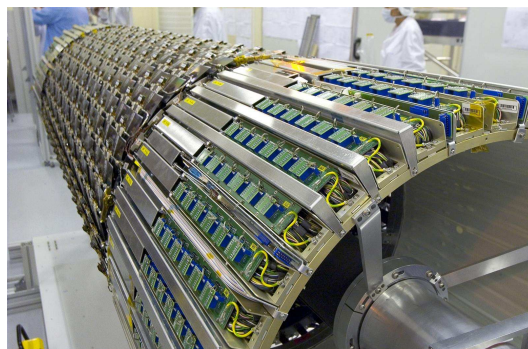


図 2.26: バレル部 SCT の写真 [1]

### 遷移輻射飛跡検出器 (Transition Radiation Tracker, TRT)

SCT の外の領域のトラッキングに用いられるのが TRT である。TRT はストローチューブ検出器を重ね、その隙間をポリプロピレンファイバーで埋めた構成である。ストローチューブは直径 4mm、長さ 40~150cm の筒の中に  $30\mu\text{m}$  の金メッキタングステンワイヤーを貼り、キセノン 70%、二酸化炭素 27%、四フッ化メタン 3% の混合ガスを封入した構造になっている。荷電粒子が通過した際に放射される X 線を検出する。この X 線は誘電率

### 2.3. ATLAS 検出器の構成

の異なる境界部を荷電粒子が通過するとき生じる遷移輻射に起因するものである。TRT のバレル部の写真を図 2.27 に、前後方部の写真を図 2.28 に示す。

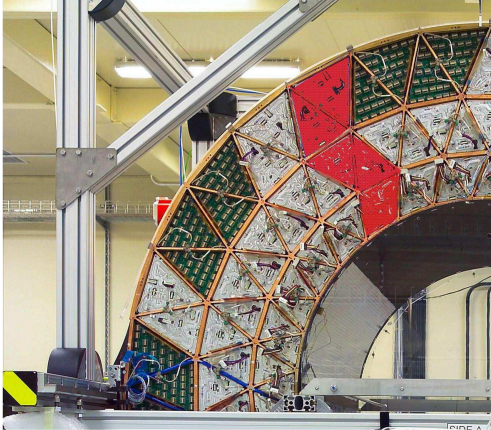


図 2.27: バレル部 TRT の写真 [9]

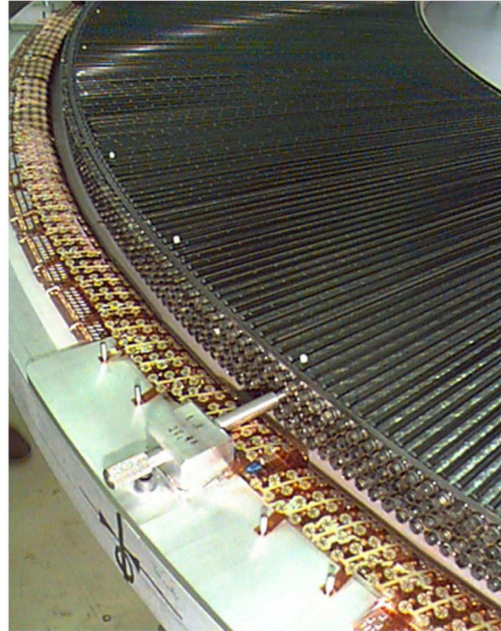


図 2.28: 前後方部 TRT の写真 [9]

バレル部にはビーム軸に平行なストローチューブ 52,544 本が 73 層に、前後方部では両サイドにそれぞれ径方向に並行な 122,880 本が 160 層に並べられている。位置分解能は  $130\mu\text{m}$  と低い、多層構造を用いることにより平均してトラッキングに用いる点が 35 点得られるため精度良く飛跡を得ることができる。

#### 2.3.5 カロリメータ

カロリメータの役割は電子や  $\gamma$ 、ジェットなどのエネルギー、位置の測定である。ATLAS 検出器のカロリメータは大きく分けて電磁カロリメータとハドロンカロリメータがある。電磁カロリメータは電子と  $\gamma$  の情報を取得し、ハドロンカロリメータはジェットの情報を得る。カロリメータの全体像を図 2.29 に示す。

図 2.29 に示すように ATLAS のカロリメータは電磁カロリメータ (EM) と、バレル部 Tile ハドロンカロリメータ (Tile)、エンドキャップ部 LAr ハドロンカロリメータ (HEC)、フォワード部 LAr ハドロンカロリメータ (FCal) から成る。 $\eta$  毎のカロリメータの吸収長をカロリメータ毎に色分けしたものを図 2.30 に示す。

以下、それぞれのカロリメータについてまとめる。

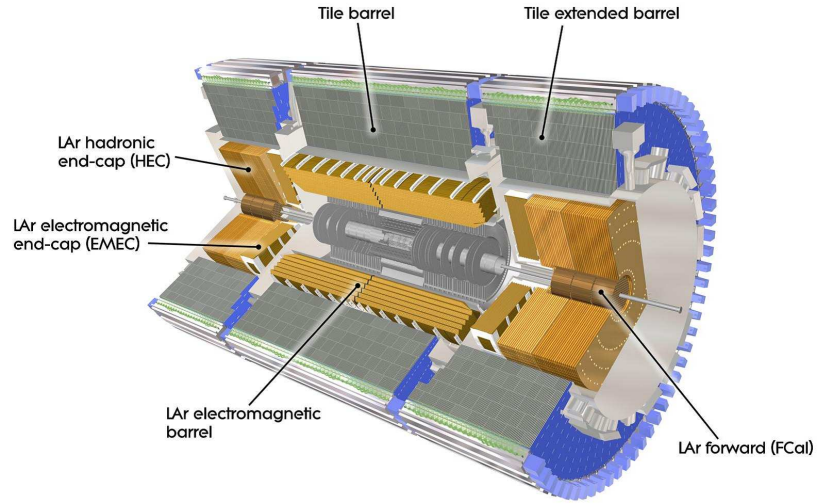


図 2.29: カロリメータの全体像 [9]

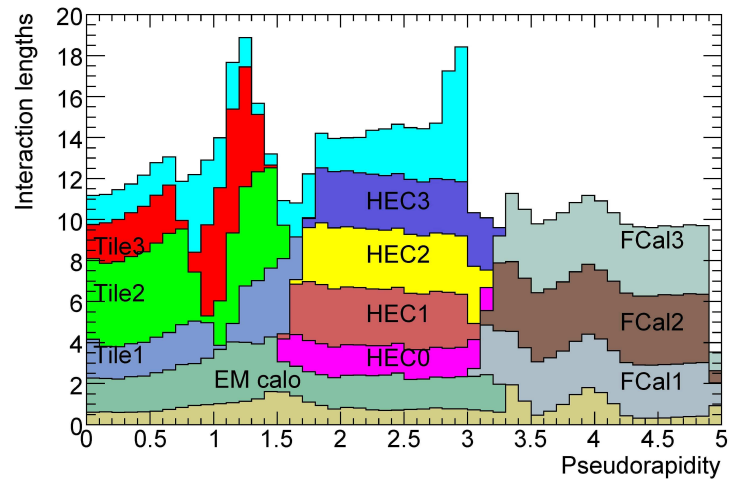


図 2.30: カロリメータ毎の吸収長 [9]



LAr 電磁カロリメータ (EM)

電磁カロリメータは、 $|\eta| < 1.5$  を担当するバレル部と、 $1.4 < |\eta| < 3.2$  を担当する前後方部に分かれている。両者ともに構造的には同じで、鉛の吸収体と放射線耐性に優れる液体アルゴン (LAr) から成るサンプリングカロリメータを用いている。特徴は吸収体がアコーディオン構造をしている点である。これは  $\phi$  方向の不感領域を無くするためである。バレル部の電磁カロリメータの写真を図 2.31、構造のスケッチを図 2.32 に、前後方部の写真を図 2.33 に示す。写真、スケッチの中でギザギザになっているのが、アコーディオン構造である。

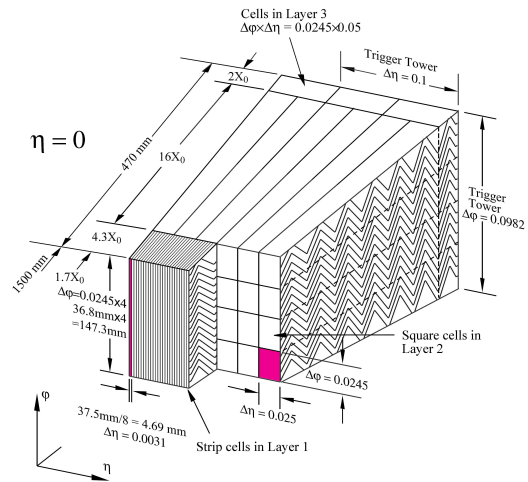
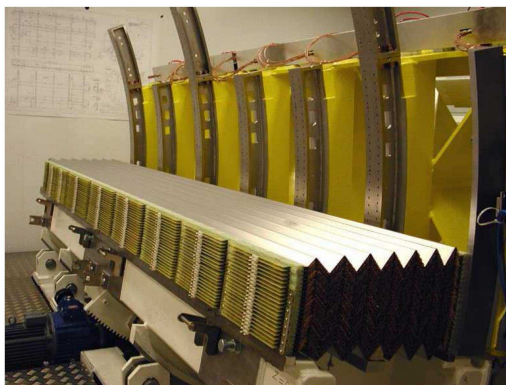


図 2.31: バレル部 LAr 電磁カロリメータの写真 [9]

図 2.32: バレル部 LAr 電磁カロリメータの構造スケッチ [9]



図 2.33: 前後方部 LAr 電磁カロリメータの写真 [9]

図 2.32 に示すように電磁カロリメータは 3 層構造をとっていて Layer1、Layer2、Layer3 と呼ぶ。それぞれ、

- Layer1

$\eta$ 、 $\phi$  方向の超精密測定により、 $\pi^0 \rightarrow \gamma\gamma$  による  $\pi^0$  の同定が可能。

- Layer2  
 $\eta$ 、 $\phi$  ともに精密測定が可能で、主にエネルギーの測定を行う。
- Layer3  
ハドロンシャワーと電磁シャワーの区別を行う。

となっている。全体のエネルギー分解能は、電子のエネルギーを  $E[\text{GeV}]$  として次の式で表される。

$$\frac{\Delta E}{E} = \frac{10.1 \pm 0.4 \%}{\sqrt{E[\text{GeV}]}} \oplus (0.2 \pm 0.1) \% \quad (2.1)$$

第1項はシャワー中の電子の数  $N$  の統計的ゆらぎによるサンプリング項、第2項はキャリブレーション精度や温度のゆらぎによる定数項である。すなわち  $E=100\text{GeV}$  の時には約1%の分解能で測定可能である。

### バレル部 Tile ハドロンカロリメータ (Tile)

ハドロンカロリメータで  $|\eta| < 1.7$  を担当しているのがバレル部 Tile ハドロンカロリメータで Tile と呼ばれる。図 2.34 の Tile の構造に示すように、Tile は吸収体の鉄とタイル型シンチレータを交互に重ね合わせた構造のカロリメータである。鉄とシンチレータの体積比はおよそ 4.7:1 である。また Tile の写真を図 2.35 に示す。

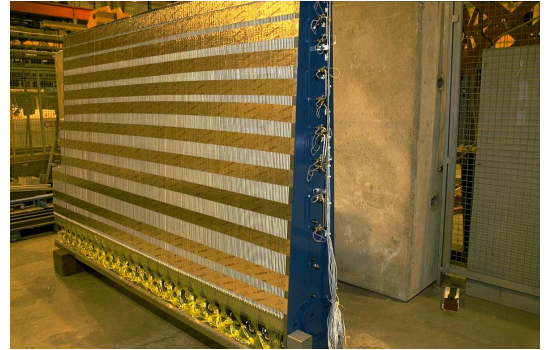
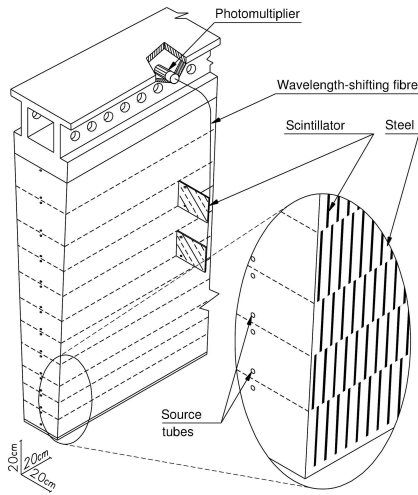


図 2.35: バレル部 Tile ハドロンカロリメータ (Tile) の写真 [1]

図 2.34: バレル部 Tile ハドロンカロリメータ (Tile) の構造図 [9]

エネルギー分解能は次の式で表される。

$$\frac{\Delta E}{E} = \frac{56.4 \pm 0.4 \%}{\sqrt{E[\text{GeV}]}} \oplus (5.5 \pm 0.1) \% \quad (2.2)$$

### エンドキャップ部 LAr ハドロンカロリメータ (HEC)

HEC はハドロンカロリメータの中でも  $1.5 < |\eta| < 3.2$  の領域を担当している。この領域は放射線強度が高いため LAr が使われている。吸収体には銅が使われている。HEC の断面図を図 2.36 に示す。また HEC の写真を図 2.37 に示す。

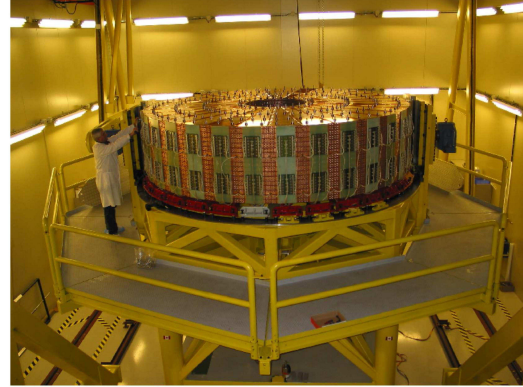
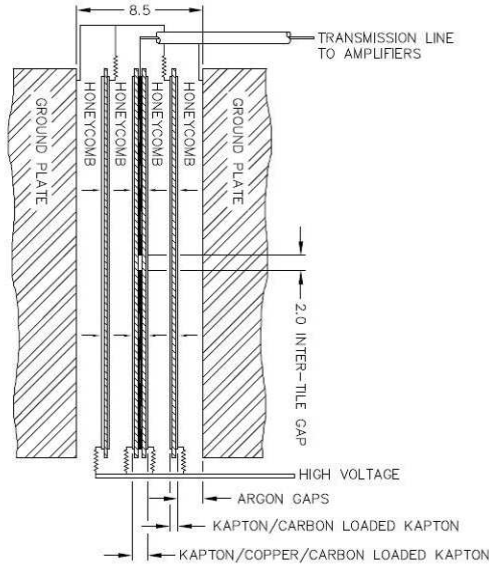


図 2.37: エンドキャップ部 LAr ハドロンカロリメータ (HEC) の写真 [9]

図 2.36: エンドキャップ部 LAr ハドロンカロリメータ (HEC) の断面図 [9]

エネルギー分解能は次の式で表される。

$$\frac{\Delta E}{E} = \frac{70.6 \pm 1.5 \%}{\sqrt{E[\text{GeV}]}} \oplus (5.8 \pm 0.2) \% \quad (2.3)$$

### フォワード部 LAr ハドロンカロリメータ (FCal)

$3.1 < |\eta| < 4.9$  の領域をカバーしているのが FCal である。FCal は最も高い放射線強度に対応するために特殊な構造のカロリメータを用いている。図 2.38 のように、第 1 層に電磁カロリメータ、第 2、3 層にハドロンカロリメータが用いられている。第 3 層の後方には、銅の合金製のシールドプラグが設置され、後方のミュオンスペクトロメータにおけるバックグラウンドを減らす役割を持っている。

カロリメータ部分は、図 2.39 に示すように、銅板にあけた穴にチューブとロッドを通した構造になっている。チューブとロッドはそれぞれ、FCal1 では銅と銅、FCal2 と FCal3 では銅とタングステンが用いられている。ロッドの周りには高放射線耐性のプラスチックファイバーが巻かれており、チューブとの間には  $300\mu\text{m}$  の隙間があくようになっていて、この部分を LAr で満たしている。モジュールの写真を図 2.40 に示す。

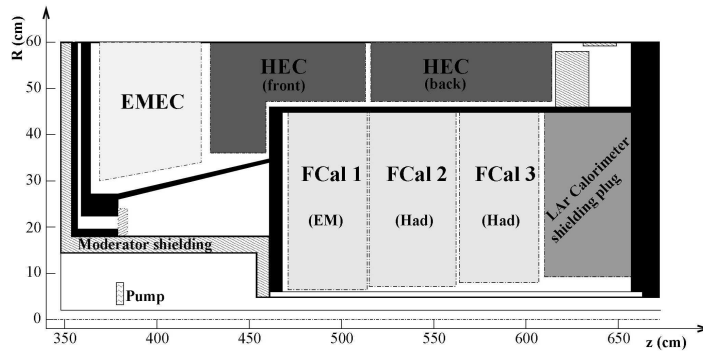


図 2.38: フォワード部 LAr ハドロンカロリメータ (FCal) の配置図 [9]

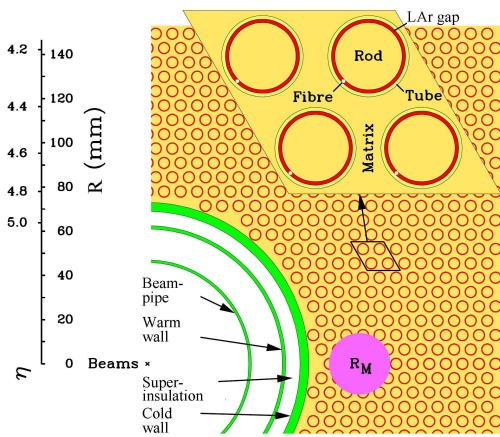


図 2.39: フォワード部 LAr ハドロンカロリメータ (FCal) の構造図 [9]



図 2.40: フォワード部 LAr ハドロンカロリメータ (FCal) の写真 [9]

エネルギー分解能は次の式で表される。

$$\frac{\Delta E}{E} = \frac{94.2 \pm 1.6 \%}{\sqrt{E[\text{GeV}]}} \oplus (7.5 \pm 0.4) \% \quad (2.4)$$

### 2.3.6 ミューオンスペクトロメータ

ミューオンスペクトロメータは ATLAS 検出器の一番外側に設置されていて、ミューオンの測定を目的としている。ミューオンは長寿命の粒子であり、物質の透過力が高いため、この位置まで到達するのである。ミューオンスペクトロメータはトロイダル磁場によるミューオンの曲がりを用いて運動量の測定を行う Monitored Drift Tube(MDT) と Cathode Strip Chamber(CSC)、イベントトリガーとしての役割を持つ Resistive Plate Chamber(RPC) と Thin Gap Chamber(TGC) の4つの検出器から構成されている。二つのトリガー用検出器は高い時間分解能で正確なトリガーを発行することができる。ミューオンスペクトロメータの全体像を図 2.41 に示す。

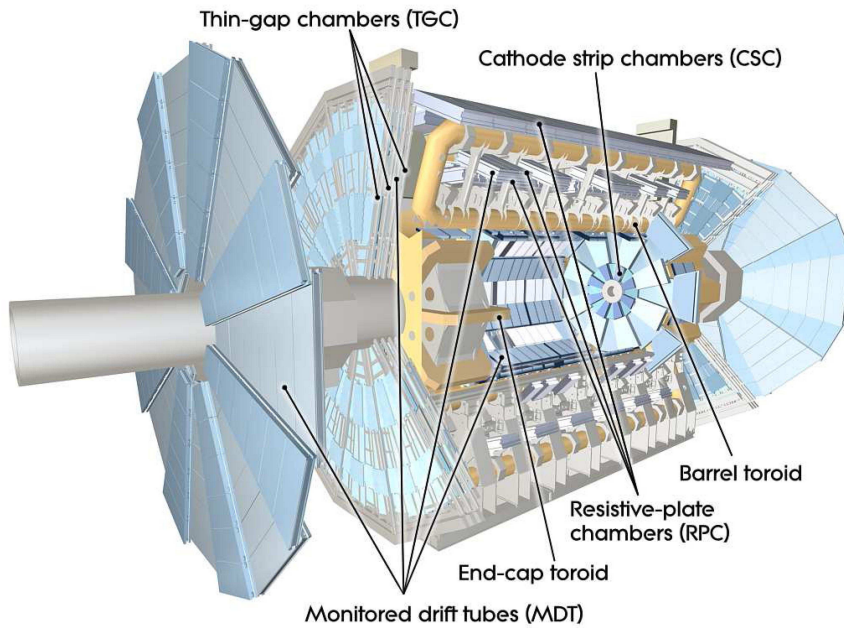


図 2.41: ミューオンスペクトロメータの全体像 [9]

またミューオン検出器の並びを z-y 平面の断面図として見たものを図 2.42 に示す。

以下、各ミューオン検出器について説明する。ただし TGC に関しては次章で詳細を述べるため、ここでは省略する。

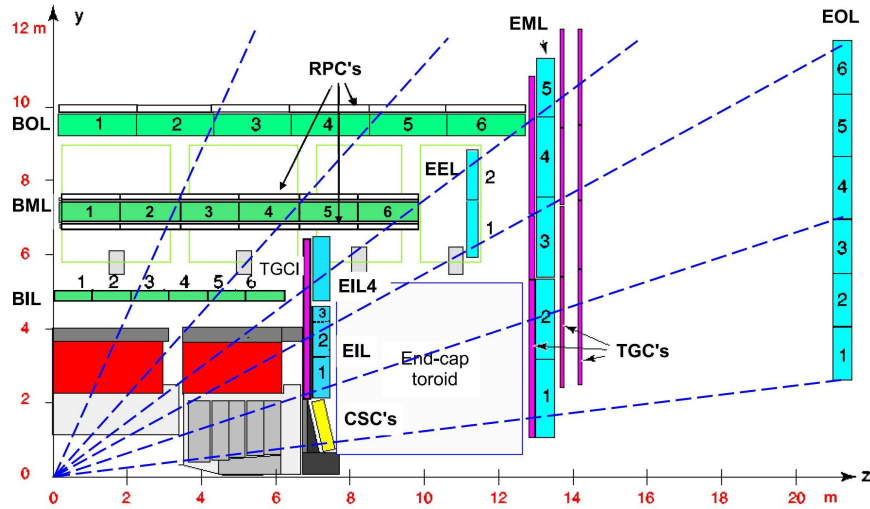


図 2.42: ミューオンスペクトロメータの z-y 平面の断面図から見た配置図 [9]

### Monitored Drift Tube(MDT)

MDT はバレル部と前後方部の広い領域に多数設置され、それぞれ  $r-\phi$  平面、 $r-z$  平面におけるミューオンの位置精密測定を担っている。MDT 一本の構造を図 2.43 に、チューブを並べたチェンバーを図 2.44 に示す。

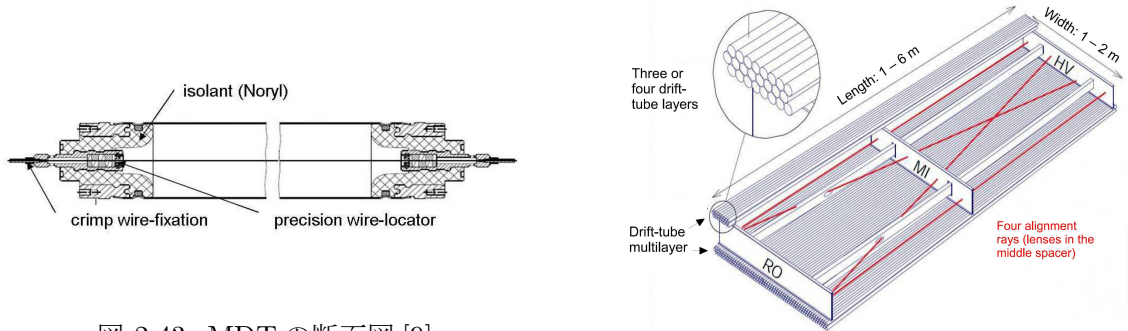


図 2.43: MDT の断面図 [9]

図 2.44: MDT チェンバーの構造 [9]

MDT は直径約 30mm のカソードチューブに直径  $50\mu\text{m}$  のアノードワイヤーを張った構造である。チューブはアルミニウム、ワイヤーはタンゲステン (97%) とレニウム (3%) から成る。バレル部、前後方部にそれぞれ 3 ステーションずつ設置されている。荷電粒子の通過と、それによって生じる電子雪崩に起因するパルスの発生の時間差によってチューブ内の粒子の通過位置測定を行う。図 2.44 にある赤い線は、設置後の歪みを監視するためのビームである。Aside に設置された MDT の写真を図 2.45 に示す。

総チャンネル数は約 37 万チャンネルに及ぶ。ガスはアルゴン (97%) と二酸化炭素 (3

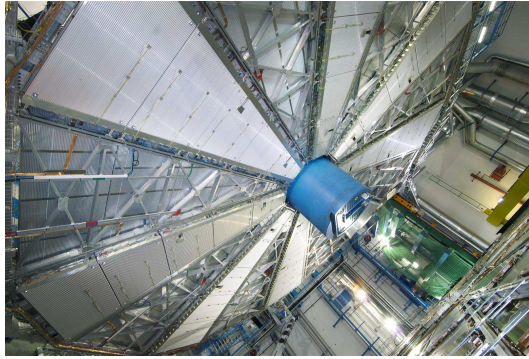


図 2.45: MDT の写真 (Aside)[1]

%) の混合ガスで、その気圧は 3 気圧と高く設定されている。ガスゲインは  $2 \times 10^4$  である。ドリフト時間は最大で 700nsec である。位置分解能はチューブの径方向に 1 本当たり  $50\mu\text{m}$  である。カウントレートの上限は  $150\text{Hz}/\text{cm}^2$  である。

#### Cathode Strip Chamber(CSC)

CSC は衝突点に近く、さらに  $\eta$  の大きいフォワード部の測定に特化している。担当している範囲は  $2.1 < |\eta| < 2.7$  である。この領域では MDT が多大なバックグラウンドによってうまく働かないため CSC が用いられている。CSC は Multi Wire Proportional Chamber(MWPC) であり、その構造を図 2.46 に示す。

CSC は並行に並ぶ陰極ストリップで挟まれたガスギャップに、ストリップと垂直になるように陽極ワイヤーを張った構造になっている。ワイヤーは MDT と同じくタングステン (97%) とレニウム (3%) から成る。ワイヤー間隔は 2.5mm、ストリップ間隔は 5.3mm と 5.6mm の 2 種類がある。ガスはアルゴン (80%) と二酸化炭素 (20%) の混合ガスで、ガスゲインは  $6 \times 10^4$  である。ドリフト時間は 30nsec 以下である。CSC の写真を図 2.47 に示す。中央部の茶色のチェンバーが CSC である。周りの銀色の細いチューブが並んでいるチェンバーは前述の MDT である。

位置分解能は  $45\mu\text{m}$  となっている。カウントレートの上限は  $1\text{kHz}/\text{cm}^2$  である。

#### Resistive Plate Chamber(RPC)

RPC は  $|\eta| < 1.05$  のバレル部のイベントトリガーを主な目的としている。

位置情報は直交したストリップによる  $\phi$ - $z$  方向の 2 次元読み出しである。総読み出しチャンネル数は約 37 万チャンネルに及ぶ。RPC は計 3 ステーションから構成されている。RPC の構造を図 2.48 に示す。高抵抗ベークライトの平行ストリップ間のガスギャップ (2.0mm) に高電圧 (9,800V) をかける構造である。ガスはイソテトラフルオルエタン (94.7%)、イソブタン (5%)、六フッ化硫黄 (0.3%) の混合ガスが用いられている。RPC の写真を図 2.49 に示す。この写真は ATLAS 検出器に RPC をインストールする時の写真で、手前の白いチェンバーが RPC で、これから ATLAS 検出器の側面部に据え付けられようとしている。

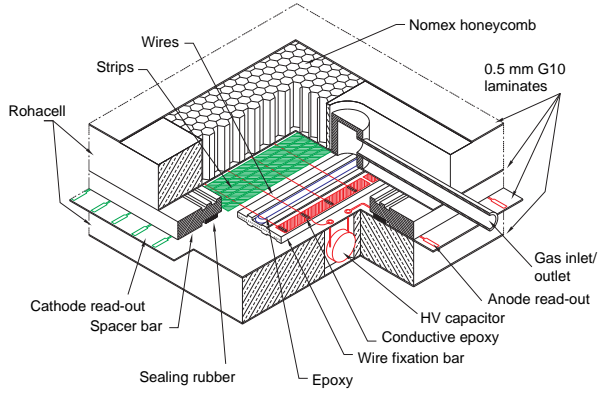


図 2.46: CSC の構造 [9]



図 2.47: CSC の写真 (茶色の部分)[1]

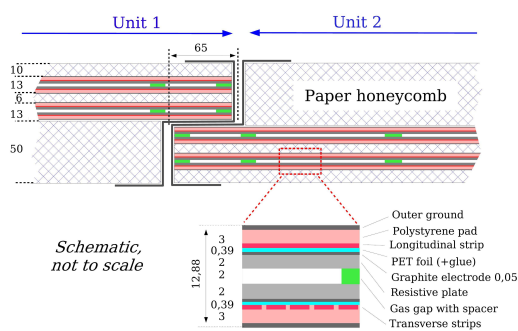


図 2.48: RPC の構造 [9]

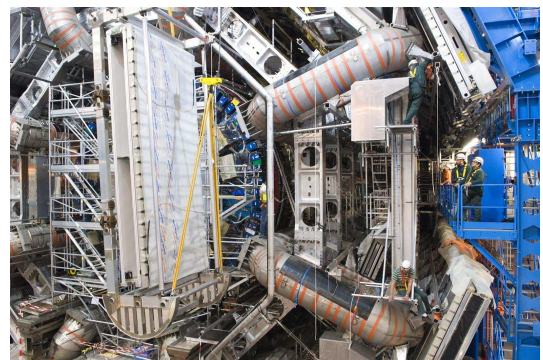


図 2.49: RPC の写真 (インストールの様子)[1]



## 2.4 トリガーシステムおよび DAQ システム

LHC の最高ミノシティ時には、40.08MHz のビーム衝突毎に平均で 23 個の衝突が起こり、イベントレートは約 1GHz になる。1 イベントを記録するために必要なデータ量は約 1.5MByte である。そのため記憶装置、計算機資源の制限から実験中の全てのデータを記録することはできず、物理的に重要なイベントと膨大なバックグラウンドを効率よく選り出すことが必要である。ATLAS ではこの条件を満足するために 3 段階のトリガーシステムを設けて、段階的にトリガーレートを落としていく。順にレベル 1 トリガー (LVL1)、レベル 2 トリガー (LVL2)、イベントフィルター (EF、レベル 3 トリガー) と呼ばれ、そのトリガーレートは始めの約 1GHz から順に約 75kHz、約 3.5Hz、約 200Hz となっている。またレベル 2 トリガー以降をまとめてハイレベルトリガー (HLT) と呼ぶこともある。トリガーの流れと DAQ システムの様子を図 2.50 に示す。また各トリガーが行われる物理的

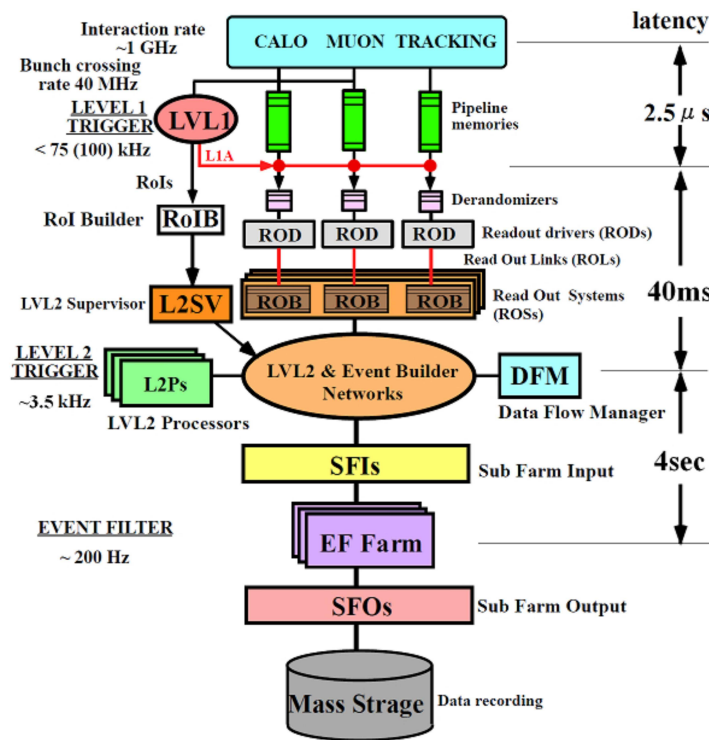


図 2.50: ATLAS 実験のトリガーシステムおよび DAQ システム図 [13]

な場所を示した図を図 2.51 に示す。

図 2.51 の通り、LVL1 は ATLAS 検出器に設置された回路、および検出器傍にある実験室のクレート群によってオンラインで行われる。LVL1 判定が下された情報は光ファイバーを伝わり地上に送られ、LVL2 の処理が行われる。LVL2 以降は計算機による処理である。EF を通過し残ったイベント情報は光ファイバーを通過して、オフライン解析用の計算機センター (グリッド) に保存される。ここに保存されたデータを用いて、各研究者は実データ解析を行うことができる。グリッドについては第 4 章で説明する。

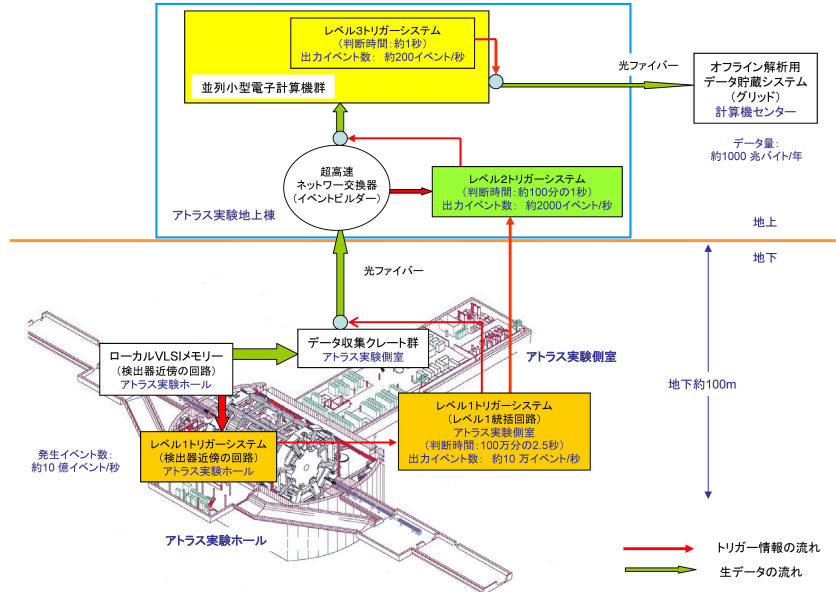


図 2.51: ATLAS 実験のトリガーシステムとその実行場所 [14]

以下、イベント情報の流れも合わせて説明しながら、各段階のトリガーについて説明する。

### 2.4.1 レベル1トリガー (LVL1)

LVL1は40.08MHzのバンチ衝突から発生する大量のイベントを75kHzに落とすためのトリガーである。

図2.52に示すように、LVL1はカロリメータ、トリガー用ミューオンチェンバー (TGC、RPC)、Muon Trigger to CTP Interface (MuCTPI)、Central Trigger Processor (CTP)、Timing Trigger and Control distribution system (TTC) から構成されている。カロリメータからは、 $e/\gamma$ 、 $E_T^{miss}$ 、 $\tau$ 、ジェットのエネルギー等の情報がCTPに送られる。TGCとRPCからは高い $p_T$ を持つミューオンの情報がCTPに送られる。各検出器から出た全ての情報は、各検出器のデータ処理システムに備えられたLVL1 Bufferと呼ばれるパイプラインメモリに一時的に保存される。そしてTGC、RPC、カロリメータの情報によりレベル1トリガーの判定がなされ、Level1 Accept (L1A) 信号が発行される。このL1A信号はTTCシステムによって各検出器に配られる。L1A信号は衝突の発生から $2.5\mu\text{sec}$ 以内に発行されることになっており、パイプラインメモリはその間の情報を全て一時保存できるようになっている。各システムはL1A信号を受け取ると同時にLVL1 Bufferの内容はデランダムマイザ (derandomizer) に送られる。デランダムマイザは不規則にやってくるデータをRead Out Driver (ROD) に読み出されるまで保持する働きを担う。その後LVL1 Bufferから該当イベントの情報をデータの圧縮や整形の処理を行い所定のフォーマットにしてRODに送信する。このとき各データには、何番目のバンチ衝突であるかを示すBunch Crossing

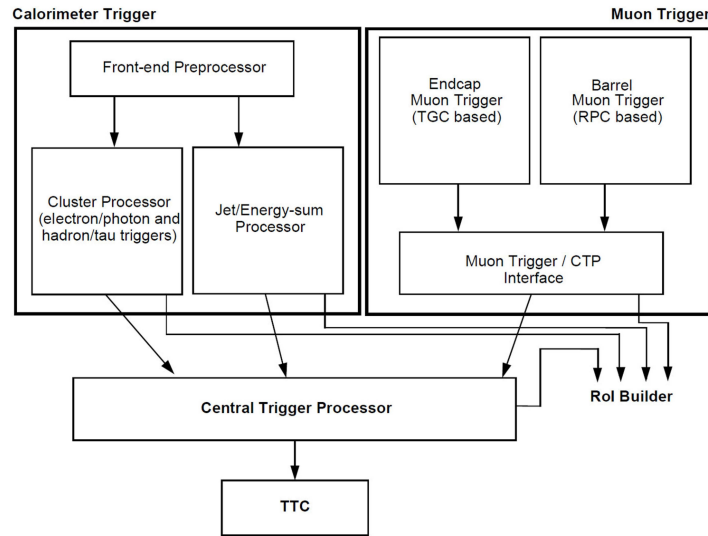


図 2.52: レベル 1 トリガー処理の流れ [13]

ID(BCID) や何番目の L1A 信号によってアクセプトされたデータであるかを示す L1AID のタグ情報が付けられる。ROD で各検出器毎にまとめられたデータは S-Link と呼ばれる規格の Read Out Link(ROL) を通して Read Out Buffer(ROB) へと送られる。ROB を複数持つ 1 つのシステムを Read Out System(ROS) と呼ぶ。

#### 2.4.2 レベル 2 トリガー (LVL2)

LVL2 はイベントレートを 75kHz から 3.5kHz 程度に落とすトリガーである。カロリメータ、MDT からの情報、内部飛跡検出器からの完全な位置情報に基づいて、より精度を上げて処理を行うが、効率を良くするために LVL1 の情報により選定された Region of Interest(RoI) と呼ばれる領域のみの情報を用いてトリガー判定が行われる。RoI とは大きな運動量を持ったジェット、電子、ミューオンなどが検出された領域である。

複数の LVL1 の結果が RoI Builder(RoIB) に渡され、そこで RoI が決定される。RoI は他の LVL1 の情報と共に LVL2 Super Visor(L2SV) に渡される。L2SV は受け取った RoI 及び LVL1 の情報から LVL2 を判定するため、負荷分散アルゴリズムに従い、いくつかの LVL2 Processor(L2P) をそのイベントのトリガーの判定に割り当てる。割り当てられた L2P は、LVL2 ネットワークを介して必要なイベント情報を ROS から受け取り、LVL2 判定を行う。その結果 LVL2 Accept 信号は L2SV に戻され Data Flow Manager(DFM) に渡される。

LVL2 は 3.5kHz でトリガーを出し、L2P には最大で 500 台程度の PC が使われる予定で、それらで平行に LVL2 判定の処理を行う。1 イベントにかかる処理時間は 40msec 以下とされている。また LVL2 判定はその精度をソフトウェアで自由に換えられるため、精度を落とせば 10msec より高速に LVL2 判定を行うこともできる。

LVL2 Accept 信号が DFM に送られると、DFM は負荷分散アルゴリズムに従ってデータを受け取る Sub Farm Input buffer(SFI) を割り当てる。SFI はイベントビルダーネットワーク (物理的には LVL2 ネットワークと同じもの) を介して ROS からデータを受け取り、フォーマットに従ってイベント構築を行う。構築されたイベントは EF に送られるために SFI 内のバッファに保持される。

### 2.4.3 イベントフィルター (EF)

EF によりイベントレートは 200Hz にまで落とされる。EF では全検出器の完全な情報を用いてトリガー判定が行われている。

EF は全体で 1,600 台程度の PC で構成される大規模なプロセッサ・ファームであり、幾つかの独立した EF sub farm から構成されている。EF sub farm はスイッチング・ネットワークを介して、1 つ又は複数の SFI に繋がっている。EF Sub farm では EF Data flow control program(EFD) というプログラムが走っており、SFI から完全なイベントデータを受け取り、トリガー判定を行う。あるイベントのトリガー判定が終わると、EFD は SFI から次のイベントデータを取って来て処理を続ける。EF 全体としてのトリガーレートは 200Hz であるが、各 EF sub farm で並列に処理が行われるため 1 イベントにかかる処理時間は 4sec 以下と定められている。

EF 判定のためにデータは SFI より EF sub farm に取り込まれる。EF Accept 信号が出ると、EFD はトリガー判定のために生成された情報に、イベントの完全な生データを付加し、Sub Farm Output buffer(SFO) へとデータを送る。そして、SFO から Disk に記録される。1 イベントで発生するデータ量は 1.5MByte 程度と見積もられており、最終的に 1 秒当たり 300MByte のデータが記録されていく。

## 2.5 ATLAS 実験の現状

本節ではこれまでに ATLAS 実験で記録された陽子陽子衝突データの収集状況についてまとめた後、ATLAS の各検出器のパフォーマンスの現状を述べる。

### 2.5.1 陽子陽子衝突のデータ収集状況

2010 年冬季シャットダウンまでの陽子陽子衝突データの収集状況についてまとめたものを表 2.3 に示す。

Parameter	Value
LHC 重心系エネルギー	3.5TeV+3.5TeV
LHC 最高ルミノシティ	$2.1 \times 10^{32} \text{cm}^{-2} \text{sec}^{-1}$
総衝突回数	$3.2 \times 10^{12}$ 回
積分ルミノシティ	$45 \text{pb}^{-1}$

表 2.3: 陽子陽子衝突のデータ収集状況

この 7TeV ランの間にとれたデータ量 (積分ルミノシティ) の移り変わりを示したものが図 2.53 である。LHC のオペレーションの向上により、日毎にデータ量が増加しているのが伺える。また LHC の安定したビーム (Stable Beam、黄色のヒストグラム) の、およそ全てを ATLAS は記録することができている (濃い灰色のヒストグラム)。

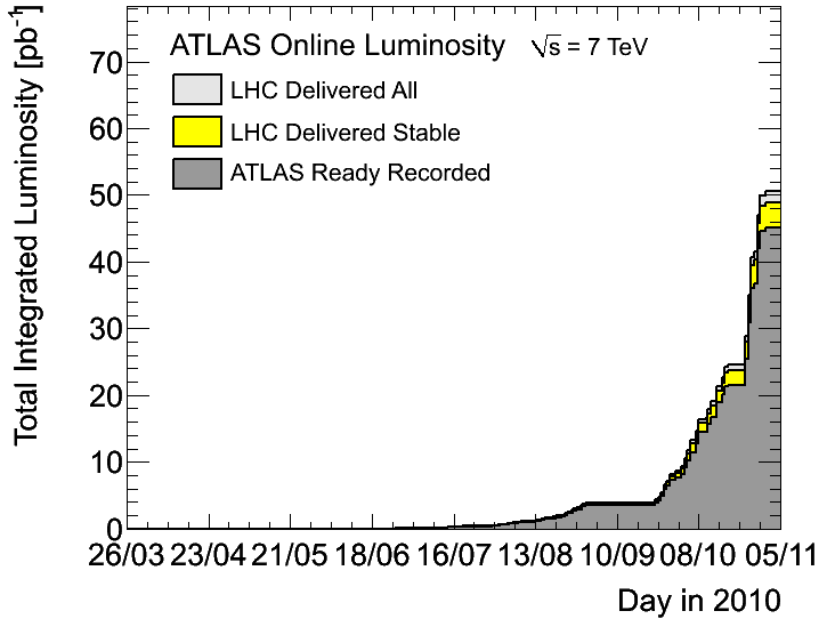


図 2.53: 陽子陽子 7TeV 衝突における積分ルミノシティの遷移 [15]

陽子陽子衝突のイベントディスプレイを図 2.54 に添付する。このイベントは 7TeV 衝突の run において、 $Z$  粒子が 2 つの  $\mu$  粒子に崩壊した事象である。赤い 2 本の直線が再構成された  $\mu$  粒子の軌跡を表していて、衝突点から紙面左方向に走っている  $\mu$  粒子が、3 つの紫色の検出器でもって検出されていることが伺える。この紫色の検出器は次章で述べる TGC を示していて、イベントによって現れた  $\mu$  粒子を TGC が正しく検出していることがわかる。

### 2.5.2 検出器のパフォーマンス

次に現在の各検出器の稼働率を表 2.4 にまとめる。このパフォーマンスは重心系エネルギー 7TeV のデータテイキングで得られた全てのデータ (積分ルミノシティ  $45\text{pb}^{-1}$ ) をもとに算出されたものである。LAr のパフォーマンスが悪いが改善される予定であり、各検出器ともに高いパフォーマンスを提供している。

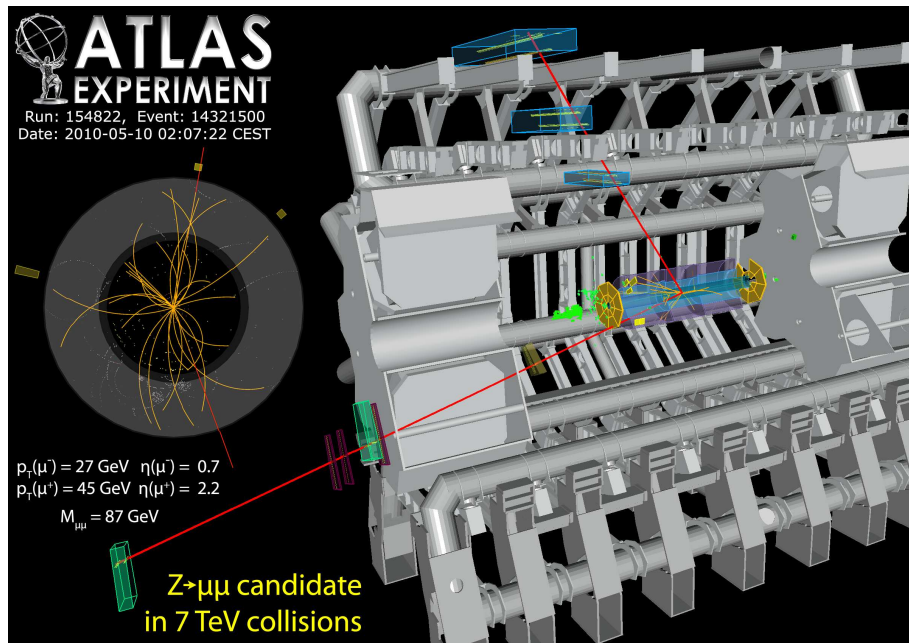


図 2.54: 陽子陽子 7TeV 衝突のイベントディスプレイ [1]

検出器	総チャンネル数	稼働率
Pixel	80M	99.1 %
SCT	6.3M	99.9 %
TRT	350k	100 %
EM	170k	90.7 %
Tile	9800	100 %
HEC	5600	96.6 %
FCal	3500	97.8 %
LVL1 Calo Trigger	7160	99.9 %
MDT	350k	99.9 %
CSC	31k	96.2 %
RPC	370k	99.8 %
TGC	320k	99.8 %
LVL1 Muon RPC Trigger	370k	99.5 %
LVL1 Muon TGC Trigger	320k	100 %

表 2.4: 各検出器の稼働状況 [2]

## 第3章 前後方部レベル1 ミューオントリガーシステム (TGC システム)

本章では私が性能評価を行った TGC のミューオントリガーシステムについてまとめる。始めにミューオントリガーの重要性を述べた後に、ATLAS 検出器前後方部に設置されたミューオントリガー検出器である TGC についてまとめる。その後、いかにトリガー判定を下すかの手法と、それに使われるハードウェアについて説明する。

### 3.1 ミューオントリガーの重要性

陽子同士を衝突させる LHC 加速器では、反応の終状態に複数のジェットが観測されるイベントが大多数を占める。ジェットというのはクォークやグルーオンが束になって発生すること (QCD ジェット事象) で、このようなイベントは ATLAS 実験が目指す新物理探索においては、そのほとんどがバックグラウンドとなる。しかし、第2章で述べたような Higgs 粒子や超対称性粒子に代表される新物理にはその相互作用の終状態として  $\mu$  粒子を代表とするレプトン対が含まれる崩壊モードが存在する。この崩壊モードは Higgs 粒子の探索モードの中で最も QCD ジェット事象バックグラウンドとの区別がしやすい。そのためミューオンをどの程度効率よく観測できているかは実験の中で重要なパラメータとなるわけである。

またミューオンというのは陽子同士の反応後に観測される粒子の中では、物質の透過力が高く、長寿命のため非常に安定している。そしてミューオンを観測するには他の多くの粒子はすでに内部の検出器等によって止められている。よってミューオンは、バックグラウンドとなる他の粒子が少ないため、大変観測しやすい粒子なのである。

以上の理由から ATLAS 実験においてミューオントリガーは大変重要な作業である。

### 3.2 ミューオン検出器 TGC

実験中のミューオントリガーは、第2章で説明した RPC と本章で詳細を述べる TGC が担う。TGC は検出器の前後方部に飛来してきたミューオンを用いて、ミューオントリガーを出す。第1章で述べた通り、神戸大学は実験の開発段階から TGC に関する研究開発を中心に ATLAS 実験に貢献してきた。

TGC は Thin Gap Chamber の略である。TGC の配置場所とその構造を図 3.1 に示す。図 3.1 の両サイドにある赤い部分が TGC であり、 $r$  方向と  $\phi$  方向で区切られた一つ一つがチェンバーである。

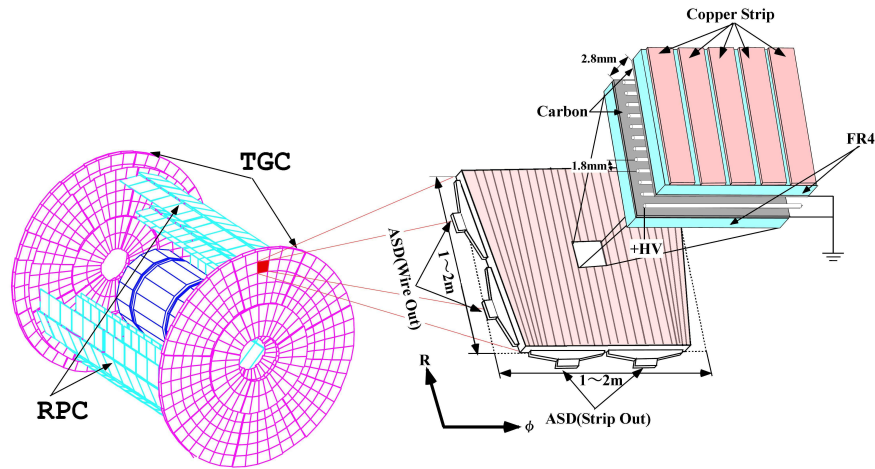


図 3.1: TGC の配置と構造図 [16]

この節ではその TGC チェンバーの構造と動作原理を説明し、それらを組み合わせた総称としての TGC(TGC Wheel) について述べる。

### 3.2.1 TGC チェンバーの構造

全てのチェンバーを組み合わせたときに円形を成すために、各チェンバーは台形になっている。1枚のチェンバーを拡大したものが図 3.1 の右の図である。Wheel に取り付けられたチェンバーは場所によって大きさが異なるが、構造は全く同じである。TGC チェンバーの断面図を図 3.2 に、TGC の各種パラメータを表 3.1 に示す。

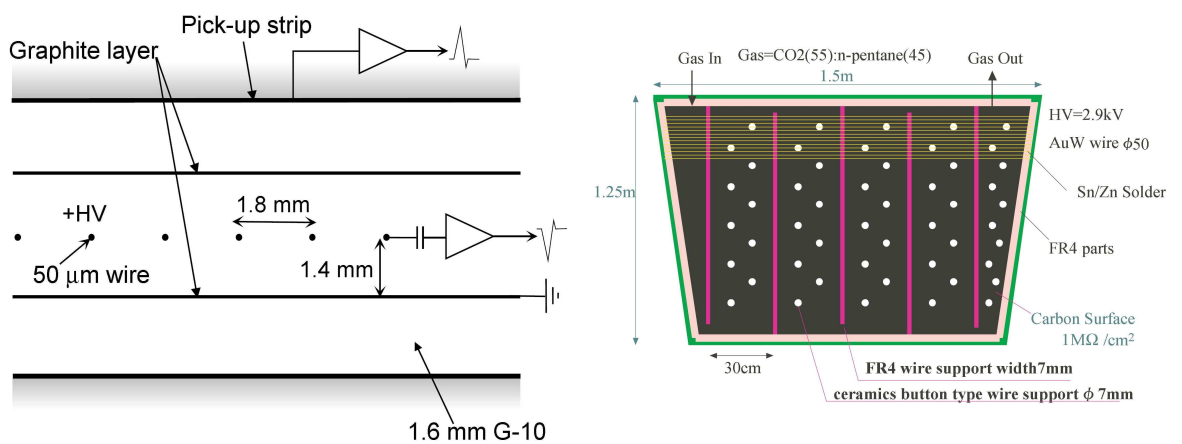


図 3.2: TGC の断面図 [9]

図 3.3: ワイヤーとワイヤーサポート [16]



### 3.2. ミューオン検出器 TGC

Parameter	Design Value
ガスギャップ	2.8±0.1mm
ワイヤー間の幅	1.8±0.05mm
ワイヤーの直径	50μm
ワイヤー・カソード間の距離	1.4mm
アノード電圧	2900±100V
ガス混合	CO <sub>2</sub> (55 %)/n-pentane(45 %)
ガス増幅率	~10 <sup>5</sup>
大きさ (1枚当たり)	1~2m×1~2m
重さ (ユニット当たり)	50kg~70kg
TGC 総数	約 3,600 枚
総読み出しチャンネル数	約 32×10 <sup>4</sup>

表 3.1: TGC のパラメータ

TGC は高エネルギー実験でよく使われる Multi Wire Proportional Chamber(MWPC) 型のチェンバーの一種であり、その構造は図 3.2 に示すように 2 枚の平行陰極板に等間隔に並んだ陽極線ワイヤーが配置されている。ただし、TGC の特徴は図 3.2 の通りワイヤ面とカソード間の間隔が 1.4mm と、ワイヤー間の間隔 1.8mm よりも狭くなっているところである。ワイヤーの間隔が狭いのは電子のドリフト時間を短くし、LHC の 25nsec という速いバンチクロッシング間隔に対応できるようにするためである。ワイヤーとストリップ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。

アノードとして直径 50μm の金メッキ加工されたタングステンワイヤーが φ 方向に平行に張られている。カソードにはガラス・エポキシ板に表面抵抗が約 1MΩ のカーボンを塗布している。ガラス・エポキシ板を挟んだ反対の面には、1 面を 32 分割した扇型の銅のストリップがワイヤーに直交する形で並んでいる。ワイヤーには約 30cm 毎にワイヤーサポートが取り付けられている。ワイヤーサポートはワイヤーのたるみを防ぐためだけでなく、ガスの流路の形成と TGC の歪みを防ぐ役割を持つ。ワイヤーは 6~20 本 (幅にして 10.8mm~36mm) をまとめて 1 つのチャンネルとして読み出す。ワイヤーとワイヤーサポートの配置図を図 3.3 に示す。ワイヤーは r 方向の読み出しに用いられている。

ストリップは各チェンバーに 32 本並べられており、各ストリップがエンドキャップ領域では 4mrad、フォワード領域では 8mrad に相当する幅を持っている。ストリップは、各々 1 つが 1 つのチャンネルとして読み出され、φ 方向の位置の検出に使われる。

使われているガスは CO<sub>2</sub>/n-pentane が 55/45 の比で混合されたガスである。このガスは紫外線を吸収し放電を起こしにくくするクエンチ効果という特性がある。

#### 3.2.2 TGC の動作原理

ガスギャップ中を荷電粒子が通過すると、ガス分子が電離され電子と陽イオンが生じる。この電子を一次電子という。次に電場に従って、一次電子はアノードワイヤー側へ、陽イ

オンはカソードストリップの方へ引き寄せられ移動する。このときワイヤーの近傍には高電場がかかるため、一次電子はガス中の他の分子を電離させるのに十分なエネルギーを得ることができる。結果、ガス分子は電離され、新たに電子と陽イオンができる。ここでできた電子が再びガス分子を電離させる。ガス中ではこれが雪崩増幅的に繰り返される。この現象はタウンゼント型電子雪崩と呼ばれる。以上の過程で発生した電子群が信号として読み出される。同時にカソード面では、塗布された高抵抗カーボン面に電荷が誘起され、外側のストリップにも電荷が誘起され信号として読み出される仕組みとなっている。

### 3.2.3 TGC Wheel の構造

TGC Wheel は図 3.1 の赤い部分であり、全体で ATLAS 検出器の前後方部に相当する  $1 < |\eta| < 2.7$  の領域をカバーする。 $|\eta| < 1.9$  の領域をエンドキャップと呼び、 $|\eta| > 1.9$  の領域をフォワードと呼ぶ。

この赤い部分を写真で見たものが、図 3.4 である。前述した台形の TGC チェンバーが並べられることにより、円筒形の円部分を成している。

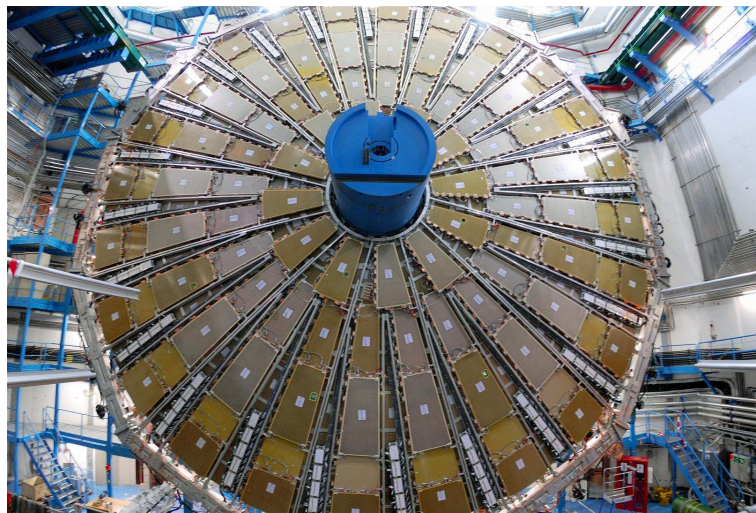


図 3.4: TGC の写真 (Big Wheel)[1]

### 3.2.4 TGC の配置

本節では ATLAS 検出器内での TGC の配置について説明する。

#### r-z 平面における配置

実際の実験では図 3.5 のように TGC は、2 層、3 層と重ねた構造にして用いられる。一般的に 2 層の構造の方を Doublet、3 層の構造の方を Triplet と呼ぶ。Triplet は 2 層目にストリップがなく、3 層のワイヤー面と 2 層のストリップ面によって読み出しが行われる。

### 3.2. ミューオン検出器 TGC

Doublet は 2 層のワイヤー面と、2 層のストリップ面によって読み出しが行われる。TGC は現在で Triplet が 1 つと Doublet が 2 つの計 7 層がトリガー判定に用いられている。7 層の並びを図式したものが図 3.6 で、これは  $r$ - $z$  平面の 1/4 断面図である。図中で M1 と名付けられた層は Triplet で、他にも TGC1 などと呼ばれることもある。M2、M3 と名付けられた層は Doublet で、それぞれ TGC2、TGC3、又は Middle、Pivot と呼ばれる。多層にすることで、各層のコインシデンスを取ってバックグラウンドによるフェイク信号の影響を減らすだけでなく、ワイヤーサポートによる不感領域の影響も減らすことが出来る利点がある。

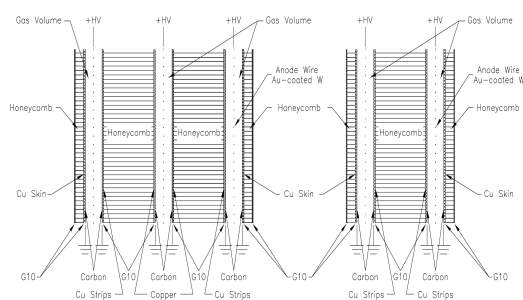


図 3.5: TGC の Triplet の構造 (左) と Doublet の構造 (右)[9]

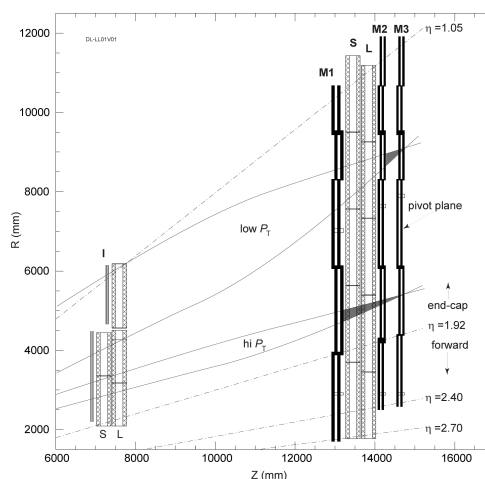


図 3.6: TGC の  $r$ - $z$  断面における全体像 [13]

TGC には Triplet よりもさらに内側に Inner Station と呼ばれる TGC が設置されている。図 3.6 でいう一番左の I と付けられた TGC である。Inner Station の中でも  $r$  の大きい方に設置されたものを Endcap Inner(EI)、小さい方に設置されたものを Forward Inner(FI) と呼ぶ。ただし現在この TGC はトリガーには用いられてはおらず、Inner Station に関しては説明を省略する。

#### $r$ - $\phi$ 断面における配置

図 3.7 は  $r$ - $\phi$  平面の 1/12 セクターの構造を示している。

左から M1 の Aside、Cside、M2 の Aside、Cside、M3 の Aside、Cside を表している。 $r$ - $\phi$  で区切られた台形の部分が 1 枚のチェンバーを表していて、図中の T7 や T2 というのはチェンバーを識別するための名前前で、同じ名前のチェンバーはその大きさが全く同じということである。M1 はエンドキャップ部は内側から  $r$  の増加方向に T3、T6、T7、T8 の 4 枚から、フォワード部は T1 によって構成されている。同様に M2 のエンドキャップ部は T4、T6、T7、T8、T9 の 5 枚、フォワード部は T2 によって、M3 のエンドキャップ部は T5、T6、T7、T8、T9 の 5 枚、フォワード部は T2 から構成されている。1/12 セクターはエンドキャップ部に  $\phi$  方向に 4 枚、フォワード部は 2 枚のチェンバーから構成されている。また F1 や B2 というのは取り付けられているチェンバーがフォワードタイプ、バック

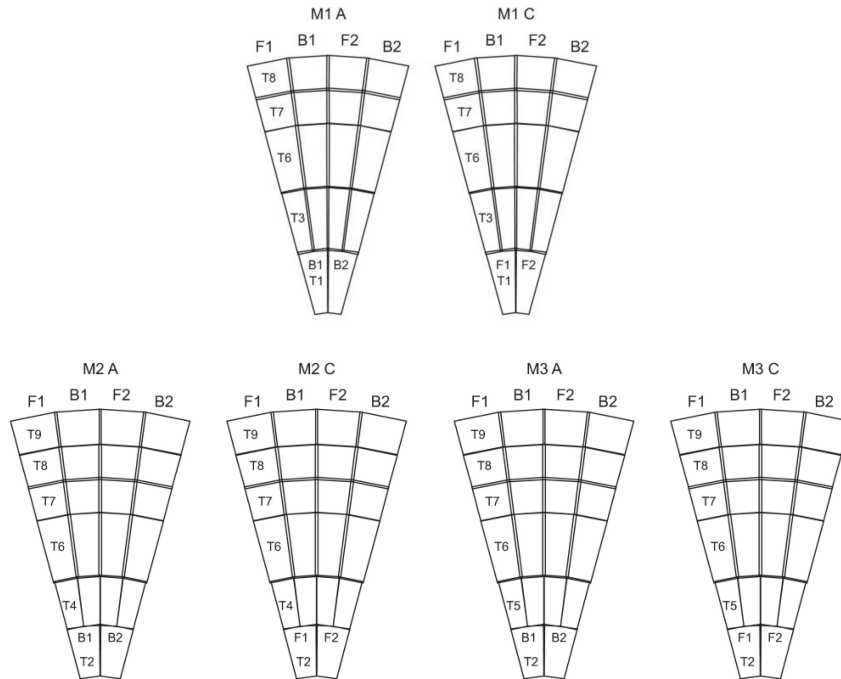


図 3.7: TGC の 1/12 セクターの構造 (チェンバー単位)[17]

ワードタイプのどちらかを示している。この区別は図 3.8 のようになっている。ASD の取り付けられている場所が異なるためストリップの読み出しチャンネルの番号付けも逆になっている。エンドキャップ部は順番にフォワード、バックワードと互い違いに並べられている。フォワード部は Aside であれば全てバックワードの向きで、Cside であればフォワードの向きで取り付けられている。

1/12 セクターを  $\phi$  方向に 12 だけ並べると円形をした 1 つのレイヤーに成る。一般的に Wheel と呼ばれる。この表現を用いて M1、M2、M3 の 7 層それぞれを Big Wheel と呼ぶことがある。それと対応して Inner Station は Small Wheel と呼ばれる。LVL1 に関する部分では 1/12 セクター (図 3.9 の桃色の部分) はさらに分割され、図 3.9 で示すように、エンドキャップ部を  $\phi$  方向に 4 等分、フォワード部を 2 等分した領域をトリガーセクターと呼ぶ。図 3.9 の水色で示した部分である。トリガーセクターはさらにエンドキャップ部で r 方向に 37 分割、 $\phi$  方向に 4 分割、フォワード部では r 方向に 16 分割、 $\phi$  方向に 4 分割され、この領域をサブセクターと呼び図 3.9 の黄色の部分で表される。サブセクターはワイヤグループ 8 つと、ストリップ 8 つに対応しており、これらはトリガーの最小単位となっており、1 つの Region of Interest (RoI) でもある。RoI のナンバリングの方法については図 A.1 を参照されたい。

ATLAS 検出器に搭載された TGC の総数は約 3,700 枚で、読み出しチャンネル数は r 方向で約 22 万、 $\phi$  方向で約 10 万に及ぶ。

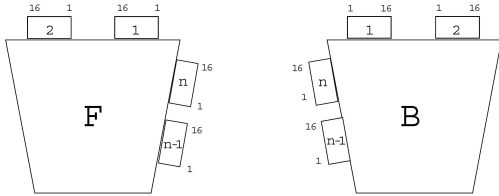


図 3.8: TGC のフォワードチェンバーとバックワードチェンバーの区別 [17]

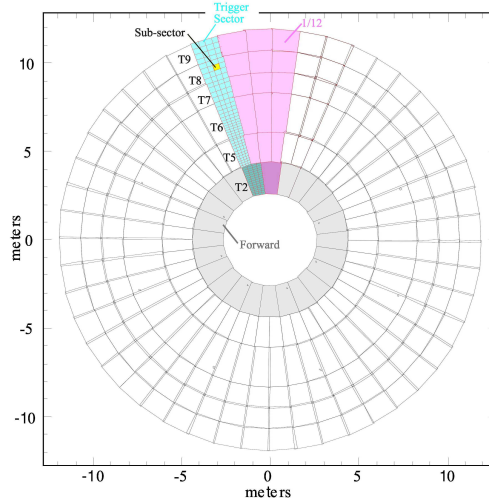


図 3.9: TGC の  $r$ - $\phi$  断面における Wheel 図 (M3)[17]

### 3.3 トリガーのシステム

本節ではミューオントリガーの仕組みをまとめる。

TGCをミューオンが通過したとする。図3.10に示すように、まずM3でのミューオンの通過位置のビームの衝突点を結ぶ直線を考える。この直線は無限運動量のミューオン(磁場によって曲げられない)が飛来した場合の軌跡に対応する。Doublet及びTripletでミューオンが通過した位置と無限運動量のミューオンが通る位置の差を、 $r$ 方向、 $\phi$ 方向それぞれで検出し、この値を $\Delta R$ 、 $\Delta\phi$ とする。トロイダル磁場が理想的な $\phi$ 方向成分のみの磁場であれば、 $\Delta\phi = 0$ となるが、実際には図2.18と図2.19に示したように $r$ 方向にも存在するので、 $\phi$ 方向の変位も考慮しなければいけない。

ミューオントリガーシステムでは $\Delta R$ と $\Delta\phi$ の値を用いてLow- $p_T$ とHigh- $p_T$ と呼ばれる2種類のコインシデンス処理が行われる。このコインシデンス処理のイメージ図を図3.11に示す。

まずLow- $p_T$ コインシデンスの判定が行われる。これはM2、M3の2つのDoubletを使用して行われる。ワイヤー、ストリップ共に計4層中3層以上(3 out-of 4)でウインドウ内にヒットがあるか否かで判定を下す2stationコインシデンスである。Tripletにおいても、ワイヤーで計3層中2層以上(2 out-of 3)、ストリップで計2層中1層以上(1 out-of 2)のヒットがウインドウ内にあるか否かで判定を下す、実質的には3stationコインシデンスの要求である。これがHigh- $p_T$ にコインシデンスである。ウインドウというのは、M2及びM1ではそれぞれ $\Delta R$ と $\Delta\phi$ の上限が設定されており、図3.10に示す通り $\Delta R$ と $\Delta\phi$ で形成される空間のことで、それぞれLow- $p_T$ ウインドウ、High- $p_T$ ウインドウと呼ぶ。ワイヤーとストリップはそれぞれ独立にコインシデンス判定が下され、次節で説明するSector Logicにおいて、初めて $r$ - $\phi$ コインシデンスがとられる。

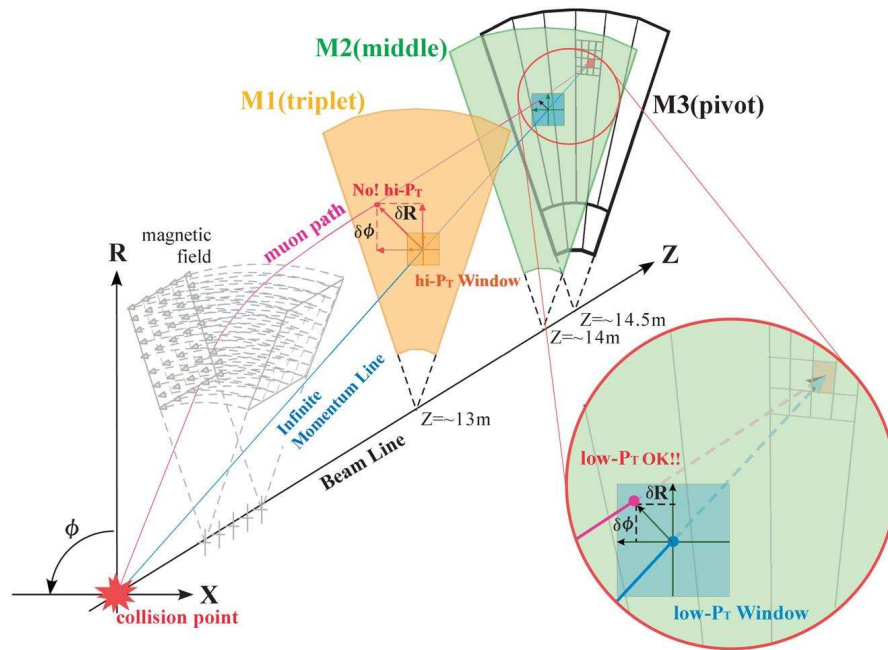


図 3.10:  $p_T$ 測定の原理[16]

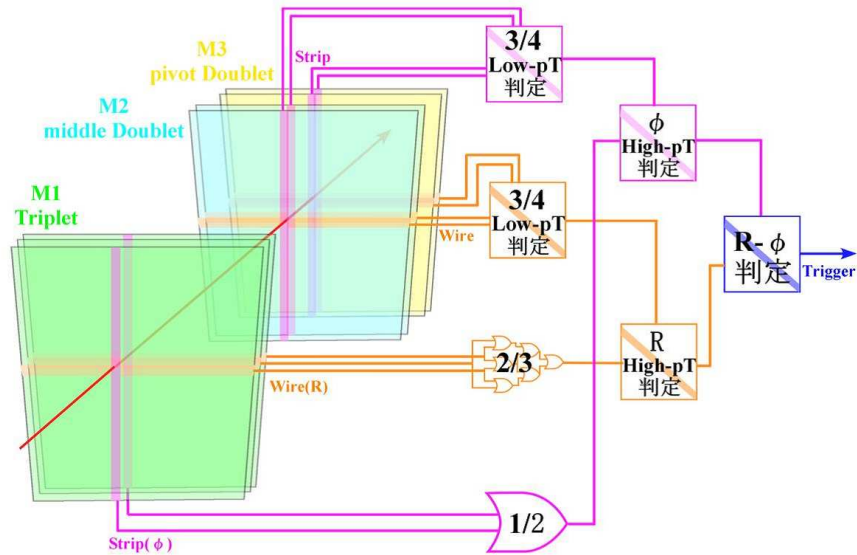


図 3.11: TGC のトリガー判定の方法 [16]

### 3.4 TGC エレクトロニクス

本節では TGC からの情報を処理するハードウェアについて説明する。始めにシステム全体のデータの流れを示した後、本論文で主に関わる Trigger 系のモジュールを中心にまとめる。

#### 3.4.1 システム全体

TGC エレクトロニクスの全体像を図 3.12 に示す。

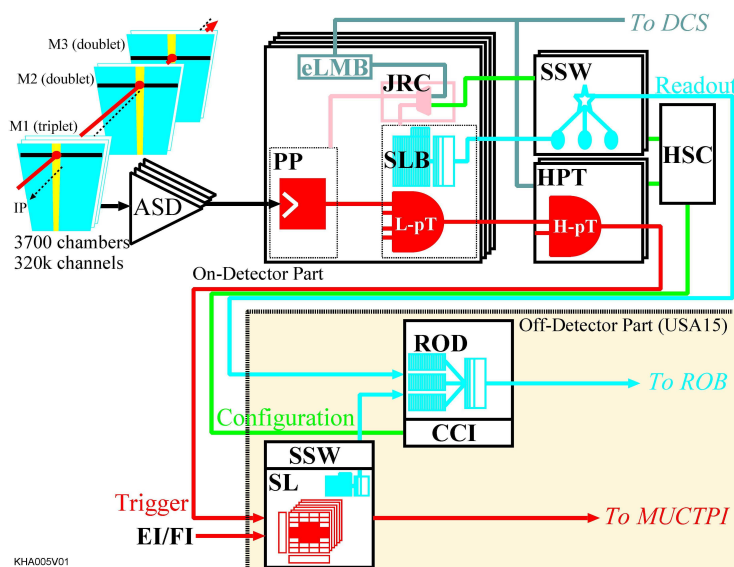


図 3.12: TGC エレクトロニクス [16]

TGC におけるデータの流れには以下の 3 種類がある。

- トリガー系  
赤のライン。25nsec 毎に流れるトリガー判定の為の情報。
- リードアウト系  
青のライン。L1A が来ると読み出される TGC 各層でのミュオンヒットチャンネル情報。
- コントロール系  
緑のライン。TGC エレクトロニクスの各モジュールをコントロールするための情報。

以下でそれぞれの流れについて説明する。

### トリガー系

図3.12の赤のラインで示される流れである。トリガー系の流れは、Amplifier Shaper Discriminator Board(ASD)、Patch Panel ASIC(PP ASIC)、Slave Board ASIC(SLB ASIC)、High-Pt Board(HPT)、Sector Logic(SL)の順に信号が処理される。TGCから出力されるアナログ信号はASDに送られ、信号の増幅、整形、デジタル化が行われ、LVDS信号でPPに送られる。PPでは各チャンネルへ粒子が到達するまでの飛行時間であるTime Of Flight(TOF)やケーブル遅延などから生じるタイミングのずれを調整し、バンチ識別を行う。SLBではPPからの信号を受け取り、2つのDoubletからの信号をもとにコインシデンス処理が行われ、Low- $p_T$ トリガーの判定が下される。またTripletからの信号についてもコインシデンス処理が行われる。HPTではSLBで測定したDoublet及びTripletのコインシデンス処理の結果をもとに、High- $p_T$ コインシデンス情報が生成される。SLではそれまで独立に行われていたワイヤー、ストリップの情報を一つに統合し、コインシデンス処理が行われる。そしてTGCのトリガー系の最終的な情報として、 $p_T$ が大きき2つのトラックをセクター毎に選別する。SLの結果はMuon CTP Interface(MuCTPI)に送られ、バレル部のトリガーを担っているRPCの情報を合わせて、ミューオンの最終的なトリガー判定が下される。

### リードアウト系

図3.12の青のラインで示される流れである。PPから送られてきたデータは、SLBの中にあるLevel1 Bufferに蓄積され、L1A信号を受けたデータのみがデランダムイザを通じて、次のStaw Switch(SSW)に送られる。SSWではデータの圧縮が行われる。またトリガー情報はSLに搭載されたSLBから同様にSSWに送られる。変換されたデータはReadOut Driver(ROD)に送られ、RODでは通常9個のSSWからの情報を収集し、それらのデータとTiming Trigger Control(TTC)から送られてくる情報との整合性を確認し、データが正しいものであれば、要求されるフォーマットにデータを変換して、ReadOut Driver(ROD)に送られる。

トリガー系とリードアウト系の流れをまとめたものを図3.13に示す。

### コントロール系

図3.12の緑のラインで示される流れである。コントロール系はATLAS実験では各検出器の制御と監視を統一的な方法で行うためにDetector Control System(DCS)が使用される。DCSにはプロセッサであるembedded Local Monitor Box(eLMB)、Analog-Digital Converter(ADC)やDigital-Analog Converter(DAC)が搭載され、PS Board上に設置される。HPT/SSWが搭載されるVMEクレートは実験室外のCrate Control Interface(CCI)からHigh-Pt Star Switch Controller(HSC)を介してコントロールされ、PS Board上のPP/SLBの設定はSSWから行う。また、TGCエレクトロニクスではDCSは温度状態の管理や供給電源の監視、さらに通常SSWから行われるPP/SLBの設定も、バックアッ



### 3.4. TGC エレクトロニクス

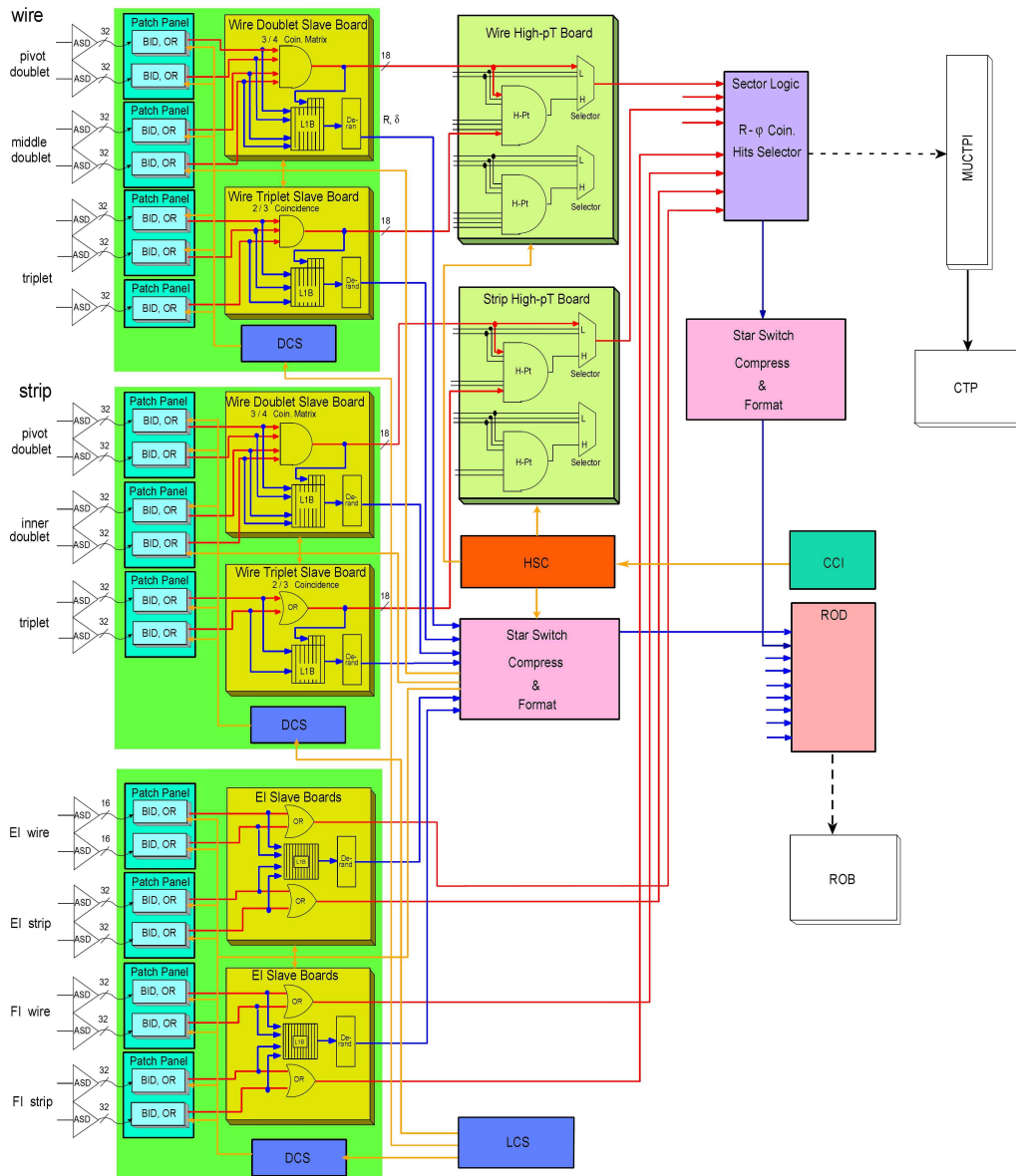


図 3.13: トリガー系とリードアウト系のデータの流れ図 [16]

プのために eLMB から行えるようになっている。PS Board 上には JTAG プロトコルの PP/SLB への経路選択を行うために、JTAG Route Controller(JRC) が設置されている。

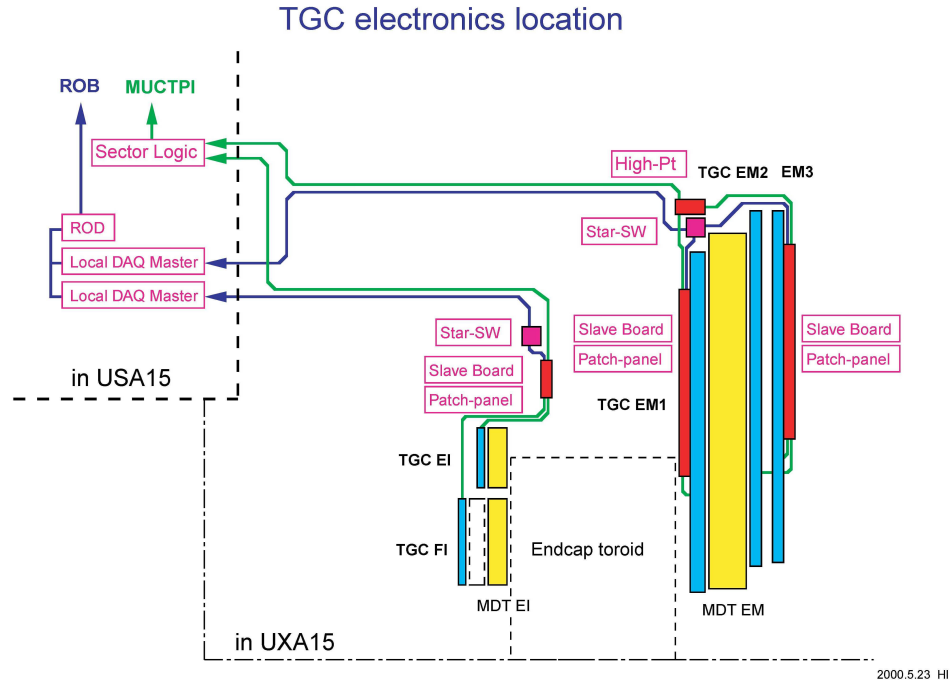


図 3.14: TGC エレクトロニクスの配置 (r-z 断面)[13]

### 3.4.2 Amplifier Shaper Discriminator Board(ASD)

ASD は TGC の側面に取り付けられ、4 チャンネル分の処理が出来る ASD ASIC が 4 個搭載され、1 つのボードで 16 チャンネルを処理する。ASD は TGC から受けたアナログ信号を増幅、整形し、一定の閾値電圧を越えた信号だけを LVDS 信号で出力する。また ASD 以降のエレクトロニクスの診断やタイミング調整のために、トリガー信号を受けて、擬似的な TGC の出力信号 (テストパルス) を出力する機能も持っている。ASD は後述する PP が設置される PS Board とツイストケーブルで接続され、動作電源、閾値電圧、テストパルスのためのトリガーは全て PS Board から供給される。また、ASD には 16 チャンネル目のアナログ信号をモニタできるアナログ出力もついている。ASD Board の写真を図 3.15 に示す。

### 3.4.3 Patch Panel ASIC(PP)

ASD から送られてくる信号は、TOF やケーブル遅延などから生じるタイミングのずれを調整しバンチ識別を行う。具体的には ASD からの LVDS 信号を LVDS レシーバーにより CMOS レベルの信号に変換する。次に variable delay 回路で、各チャンネルにそれぞれ

ディレイをかけることによりタイミングの調整が行われる。このディレイは 0.84nsec 単位での調整が可能になっている。タイミング調整された信号は BCID 回路に入り、TTC から供給される LHC クロックと同期がとられバンチ識別が行われる。最後に TGC が重なっている境界領域のダブルカウントを防ぐため、OR ロジックを通して SLB に送られる。また PP は ASD に向けてテストパルスを発生させるためのトリガーを出力するテストパルス回路も搭載している。テストパルスの振幅、タイミングは可変であり、JTAG プロトコルによって制御できる。PP は 1 つで 32 チャンネル分の信号を処理できるようになっている。PP は図 3.14 に示すように Triplet の前面、及び Doublet の背面に設置されている。

#### 3.4.4 Slave Board ASIC(SLB)

SLB はトリガー部とリードアウト部の両方を兼ね備えている。

トリガー部は TGC からの入力信号に対してコインシデンス処理を行う。SLB の段階ではワイヤーとストリップ、Doublet と Triplet を別々に処理するため、それらに応じて設定を切り替えられるようになっている。Doublet ではワイヤー、ストリップ共に 3/4 コインシデンス処理が行われる。そして M3 を基準にして M2 上の位置のずれによって Low- $p_T$  判定を行う。Triplet では  $p_T$  は測定されず位置のみが求められる。ワイヤーでは 2/3 コインシデンス、ストリップでは 1/2 コインシデンス処理が行われる。トリガー部には PP からの信号に 1/2 クロック単位でディレイをかける機能や、各チャンネルをマスクする機能や、連続するチャンネルにトリガーがあった場合にその中の一つのチャンネルのみの信号だけを出力させるデクラスタリング機能、SLB 以降のハードウェアの診断やタイミング調整を行うためのテストパルスを出力する機能を備えている。

リードアウト部は LVL1 トリガーの判定を受けたデータの読み出しを行う。LVL1 Buffer に一時的に保存されたデータを CTP からの L1A 信号を受けたデータとその前後 1 バンチずつのデータがデランダムマイザにコピーされる。コピーされたデータは 3 バンチ分それぞれ別個にシリアルに変換され SSW に送られる。

PP と SLB は PS Board(PSB) と呼ばれる 1 つのボード上に搭載されている。PSB には他にも JRC、eLMB が搭載されている。PSB の写真を、図 3.16 に示す。また PSB は Service Patch Panel(SPP) と呼ばれる TTC からの信号を受信する機能を持つエレクトロニクスと一緒に、PS Pack と呼ばれるパッケージにまとめられて、図 3.14 に示すように Triplet の前面、及び Doublet の背面に設置される。PS Pack は 1 つの SPP と複数の PSB から構成され、1/12 セクター毎に 2 つ設置される。

#### 3.4.5 High-Pt Board(HPT)

HPT は Doublet と Triplet の情報を用いて High- $p_T$  コインシデンスを判定する。そのために HPT 上では SLB までは独立に処理されてきた Doublet と Triplet のデータが統合される。Triplet は M2、M3 よりも衝突点に近く、M2 と M3 の間隔よりも離れた位置に設置されているため、トロイダルマグネットによりあまり曲げられなかった高い  $p_T$  を持つミューオン由来の信号を選別することができる。HPT からはワイヤーとストリップに対して別個にコインシデンス処理が行われ、 $\Delta R$ 、 $\Delta\phi$  が出力される。HPT は図 3.17 のように

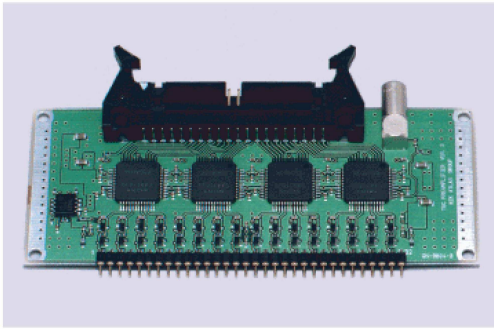


図 3.15: ASD Board の写真 [16]

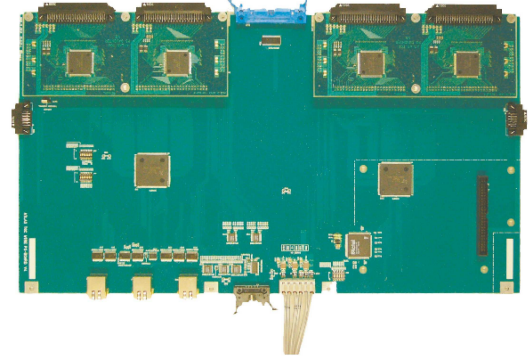


図 3.16: PS Board の写真 [16]

設計され、図 3.14 に示すように TGC の近傍に設置されている。出力データはシリアルライズされオプティカル信号に変換されて光ファイバーによって約 100m 離れた USA15 と呼ばれるカウンティングルームにある SL に送信される。HPT はエンドキャップ領域のワイヤー用、ストリップ用、フォワード領域用の計 3 種類があり、エンドキャップ用には 4 つの HPT ASIC がフォワード用には 3 つの HPT ASIC が搭載されている。図 3.17 の HPT はフォワード用である。

### 3.4.6 Sector Logic(SL)

SL は TGC のトリガーデータが最終的に集約されるモジュールであり、1 つで 2 トリガーセクター分の信号を処理する。それまでのエレクトロニクスとは違い図 3.14 に示すように検出器から約 100m 離れた計算機室 (USA15) のクレート上に設置されている。SL の信号処理は  $r$ - $\phi$  コインシデンス、Track Selector の 2 種類に分類される。

まず SL は HPT からの信号を受け取り、電気信号に変換した後パラレル変換をする。そしてこれまでは独立に行われていたワイヤーとストリップの情報を統合しコインシデンス処理を行い、ミューオンのトラックを再構築する。そのトラックを 6 段階の  $p_T$  の閾値 ( $p_T$  threshold) に分類する。この閾値はシミュレーションから作成された  $\Delta R$  と  $\Delta\phi$  の対応関係からユニークに判定できるコインシデンスウインドウ (コインシデンスマトリックス) を用いて判定される。コインシデンスウインドウは探索する物理により自由に変更することができるように、書き換え可能な FPGA の Look Up Table(LUT) によって実装されている。

Track Selector は Pre Selector と Final Selector の 2 種類に分類される。Pre Selector は 6 段階の  $p_T$  判定のそれぞれに用意され、それぞれ  $r$  の大きい順に 2 つのトラックを選択して、最大で計 12 トラックが Final Selector に送られる。Final Selector では、送られてきたトラックの中から  $p_T$  の大きいものを 2 つ選択して、位置情報と共に MuCTPI に送信する。SL が処理に使用した HPT のデータと SL での処理の結果は SL に搭載された SLB より USA15 に設置された SSW に送信される。SL の写真を 3.18 に添付する。

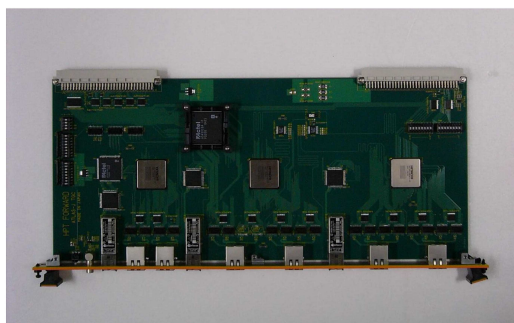


図 3.17: HPT Board 写真 [16]

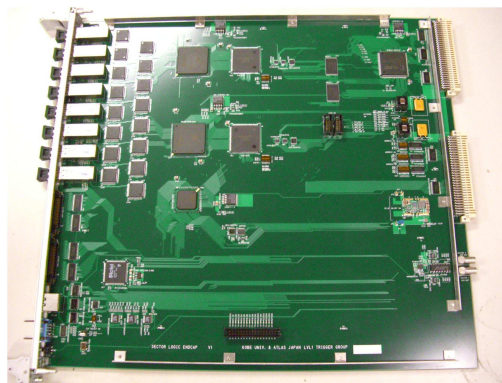


図 3.18: SL の写真 [16]

### 3.4.7 Star Switch(SSW)

これまでに説明したエレクトロニクスは主にトリガー系の処理を行うエレクトロニクスであったが、SSW と後述の ROD は読み出し系のエレクトロニクスである。SSW の役割は SLB のリードアウト部から送られてくるデータを圧縮し、ROD にデータを送る前にデータ量を減らして、効率よい読み出しができるようにすることである。図 3.14 に示すように SLB からデータを素早く受信できるように検出器の近傍に設置されている。データ圧縮は、データを cell と呼ばれる 8bit ごとの塊に分け、各 cell にアドレスを付け、値がゼロでない cell だけをアドレスと共に送る。TGC の全チャンネルのうちヒット信号を発生するのはごく一部なので、これによりデータ量を大幅に減らすことができる。1 つの SSW で最大 23 個の SLB のデータを受け取ることができる。具体的な動作は次の通りである。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。そのデータはレシーバーである SSWrx に送られ、そこでデータの圧縮が行われる。その後データはトランスミッターである SSWtx 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オプティカル信号に変換され USA15 内にある ROD に送信される。SSW の写真を図 3.19 に添付する。

### 3.4.8 Read Out Driver Board(ROD)

リードアウト系のデータが最終的に集約されるモジュールが ROD である。ROD は図 3.14 の通り、USA15 に設置される。複数の SSW から受け取った圧縮データを、オプティカル信号から電気信号に変換した後パラレルデータに戻し、FIFO メモリに一時格納する。このデータをトリガー情報を元に同じイベント毎にまとめ、決められたフォーマットに従ってヘッダー、トレーラーをつける。まとめられたデータは Simple Link(Simple Link Interface) というフロントエンドとリードアウト系エレクトロニクスを繋ぐために CERN で開発された光信号のリンクモジュールによって ROD に送信される。イベントの同定やヘッダー、トレーラ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 TTCrx が搭載さ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。ROD の写真を図 3.20 に示す。

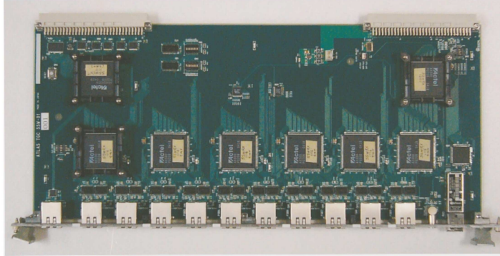


図 3.19: SSW の写真 [16]

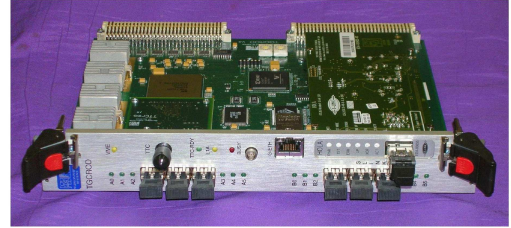


図 3.20: ROD の写真 [19]

### 3.5 TGC の現状

TGC の現状のトリガー効率 (Trigger Efficiency) の遷移を横軸に run number をとりプロットしたものを図 3.21 に示す。図 3.21 の run は RunNumber160387(2010 年 7 月 29 日) から RunNumber166850(2010 年 10 月 25 日) のデータを用いたものである。Trigger Efficiency の定義は、

$$\text{TriggerEfficiency} = \frac{\text{真の muon の中で RoI を通過した muon}}{\text{offline で reconstruction された muon(真の muon)}} \% \quad (3.1)$$

である。分子の RoI を通過したミューオンというのは、即ち TGC でミューオンと同定されたものを示す。図中にある setD、setF、setG というのは前節の SL に搭載されたコインシデンスウインドウのタイプの違いを表している。実験状況に合わせて、アップデートされている。図 3.21 は Low- $p_T$  判定が下されたミューオン全てに関して Trigger Efficiency を見積もったもので、十分に高い Efficiency を維持していることが伺える。ところどころ Efficiency が落ちているところは、統計量が少ない等が原因であるが、次の run ではその問題は解消され Efficiency は回復している。以上のように、Trigger Efficiency の観点に立つと、TGC システムは十分に高いパフォーマンスを提供できているといえる。

今回の私の研究は、これをハードウェアの観点から TGC システムの現状を評価することで、それについては第 4 章以降で述べる。

### 3.5. TGC の現状

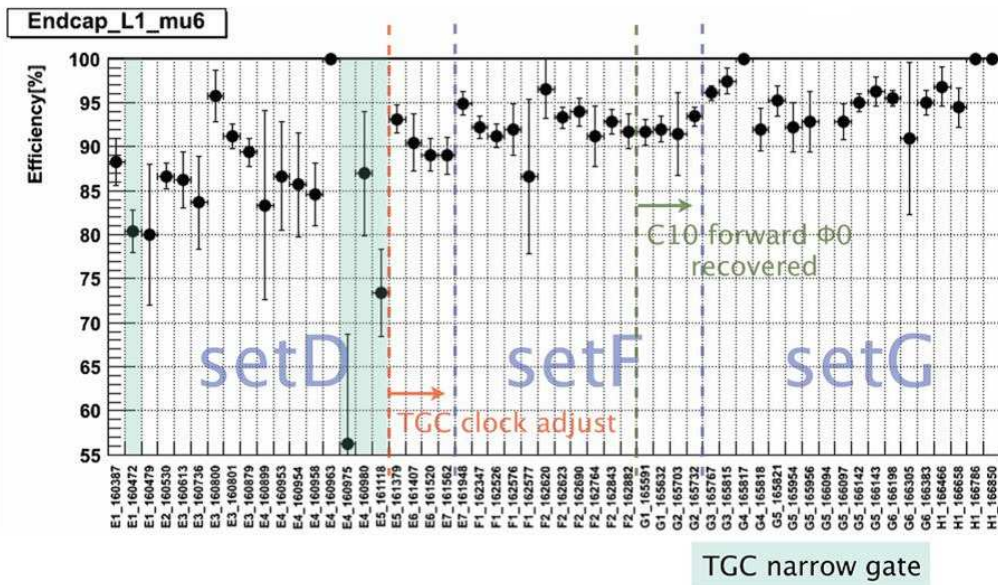


図 3.21: TGC のトリガー効率の遷移 [18]

## 第4章 オフラインデータ処理システム

本章では実験で得られたデータをオフラインで処理するシステム、とくにLVL1トリガーが成された後のデータ処理システムについて解説する。解説の準備として、まず ATLAS におけるデータの形式についてまとめたあと、ミューオンオフラインデータ処理システムについて述べ、最後に中でも私が改良を行った TrigMuonCoinHierarchy と呼ばれるオフライン解析用のソフトウェアパッケージについて述べる。

### 4.1 ATLAS 実験におけるデータの形式

以下に ATLAS 実験で使われるデータ形式について、そのシステムを述べ、上流から順番に説明する。

#### 4.1.1 グリッドシステム

実験で得られるデータは膨大で、毎秒約 200 イベント、そのデータサイズを約 1.5MB とすると、年間で約 4PB にもものぼる。これを 1 箇所では保管し、研究者が解析の度に同じ計算機にアクセスすることは危険でもある。そのため CERN(Tier0) サイトを頂点として世界各地に階層化したサイト (Tier1、Tier2、Tier3) を設置し、データの送信及び蓄積を行っている。グリッドとはネットワークを介して世界の計算機を結び、共通プラットフォームに統合し、仮想的な高性能計算機を形成するシステムのことである。ユーザーはグリッドシステムを用いることで、自らのジョブを処理する際に世界各地のどのサイトで処理するかを自動的に判断され、使用している CPU 量等を気にせずに解析を行うことができる。

#### 4.1.2 RAW Data

RAW Data はイベントフィルターから出力されるもので、Event Store の中では一番上流に位置する。これは各検出器のエレクトロニクスで処理された生のデータ情報である。保存される RAW Data の大きさは 1 イベントあたり約 1.5MB である。Raw Data は Byte Stream と呼ばれる 0 と 1 のバイト列で、特定の 0 と 1 の並びによってデータの始まりを表すヘッダーと、終わりを表すトレイラー、及びそれに挟まれたデータ部分から成る。このため Byte Stream を直接読むことは困難であり、人間が扱いやすいようにこれらは後々デコーディングされていく。イベントフィルターでの処理順にイベントが割り振られるために、ファイル内の各イベントはトリガーの種類や時系列順にソートされていない。ただし run の切れ目で一旦ファイルは閉じられるため、1 つのファイルの中に異なる run のイベントが混在することはない。



### 4.1.3 Event Summary Data(ESD)

ESDは後述するPRDというデータ形式の全ての情報と、リコンストラクションによって得られたトラック情報やカロリメータのエネルギークラス情報等が含まれている。サイズはRAW Dataと後述のAODの間であり、1イベントあたり約0.5MBである。ESDに含まれるデータからは解析者が自身のアルゴリズムを用いてトラックやジェットを再構成することが可能である。POOL ROOT fileという複雑な構成をしたフォーマットである。1つのRAWデータファイルに対して、1つのESDファイルが作られる。ESDはリコンストラクション後、世界中のグリッドにコピーされ、研究者が自由に使用できるようになっている。

### 4.1.4 Analysis Object Data(AOD)

AODはESDをさらに抽象化したもので、解析に最も適したものと言える。再構成されたイベント情報のみが含まれ、検出器レベルの情報は含まれない。そのサイズは1イベントあたり約0.1MBであり、ESDと同様、POOL ROOT fileの形式をとる。

## 4.2 オフラインデコーディングの流れ

データは前述の通りRAW Dataから始まり、順にESD、AODとデコーディングされ保存される。Raw DataとESDの間にはRaw Data Object(RDO)とPrep Raw Data(PRD)と呼ばれる中間状態が存在する。シミュレーションでは、検出器の応答などをシミュレートし、実データのRAW Dataに対応するヒット情報のRDOを作成する。実データとシミュレーションのデータはRDOレベルで統合され、それ以降は同じく扱われる。データのデコーディングの流れを図4.1に示す。

以降、RDOとPRDについて述べたあと、シミュレーションにおけるデータ生成の流れと、RDO以降のミューオンスペクトロメータにおけるデータデコーディングの流れについて述べる。

### 4.2.1 Raw Data Object(RDO)

RDOは、Byte Streamを値に変換し、各検出器におけるヒット情報やトリガー情報をオブジェクト化したフォーマットである。それぞれのオブジェクトの持つデータは検出器によって様々であるが、エレクトロニクスのモジュールやチップのIDやチャンネル番号など、Byte Stream上の数値がそのまま入っている。システムのコミッショニングには、デコーダーを数多く通過した下流のフォーマットよりもRDOレベルの情報が非常に有用である。ただし対象とする検出器のシステムに精通していなければ理解は難しい。

### 4.2.2 Prep Raw Data(PRD)

PRDはRDOの情報をさらに抽象化し、測定器の情報として解釈したものである。ヒットやトリガーの位置情報や、どのチェンバーからの情報かなど、実験者が理解しやすい形

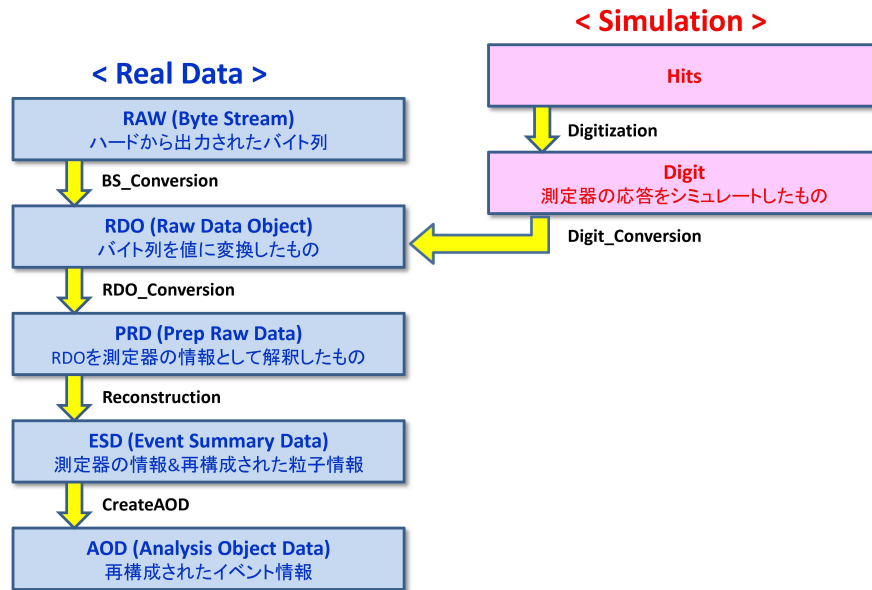


図 4.1: 実データとシミュレーションのデータフロー

にデコーディングされている。オブジェクトの型には各検出器グループ内で決められた制約がある。PRD の情報は ESD に全て含まれる。ただし ESD には PRD の情報以外の情報も含まれている。

#### 4.2.3 シミュレーションにおけるデータ生成の流れ

シミュレーションにおけるデータ生成の流れを図 4.2 に示す。

始めに Event Generator によって粒子のデータが生成される。Event Generator には pythia や AlpGen など幾種類かが用いられているが、それらのアウトプットは HepMC という共通のフォーマットに統一される。このデータを Geant4 ベースの G4ATLAS simulation 等を用いて、生成された粒子と ATLAS 検出器の相互作用のシミュレーションを行う。次に Digitization という作業が行われる。これは G4ATLAS simulation の結果を ATLAS の各検出器の応答の結果として変換、出力する作業である。Digitization されたデータは Digit と呼ばれ、これは実データにおける RDO に対応するものである。ただしこの Digit はパーシステントなオブジェクトではなく、その情報は RDO として得ることができる。RDO 以降は、順に PRD、ESD、AOD と処理される。これは実データにおける RDO 以降の処理と同じである。

#### 4.2.4 ミューオンスペクトロメータのデータデコードの流れ

図 4.3 にミューオンスペクトロメータの RAW Data からのデコードの流れを示す。図 4.3 の左側がシミュレーションデータをデコードする場合、右側が実データをデコードす

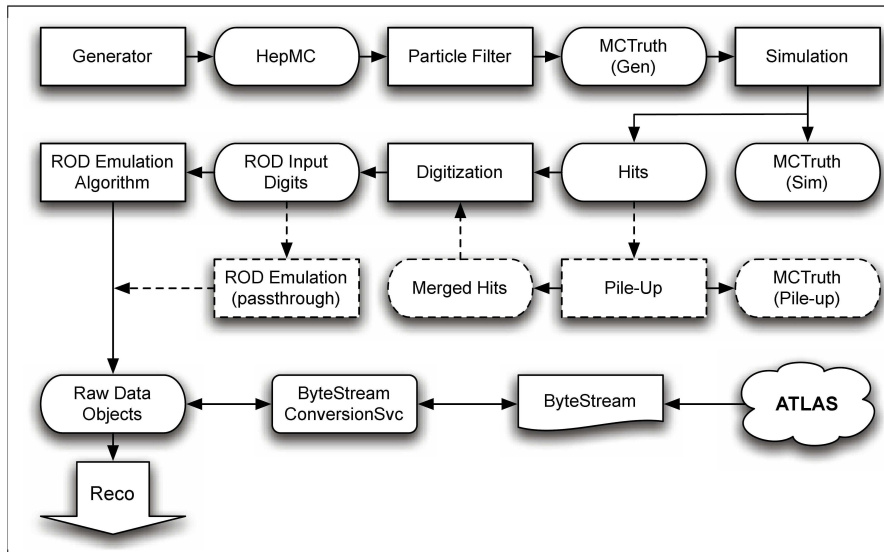


図 4.2: シミュレーションのデータフロー [11]

る場合の流れである。

ミュオンスペクトロメータのデータデコーディングは、MuonSpectrometer というパッケージ内の MuonCnv というコンバータによって行われる。Byte Stream として取得されたデータは MuonByteStream によって RDO に変換される。次に MuonRdoToPrepData によって RDO から PRD への変換が行われる。この際、各検出器とそれらの読み出しシステムの配線を反映した MuonSpectrometer 内の MuonCablings が用いられる。次に MuonReconstruction によって粒子のリコンストラクションが行われる。図 4.3 の右には AANTuple への出力があるが、AANTuple は Athena Aware Ntuple というフォーマットの略で、各段階での情報を AANTuple 形式で出力させ、プログラムのデバッグに役立てることができるものである。AANTuple は root に直接入力ファイルとして読み込ませることができるためデバッグには有効なのである。図 4.3 の流れで生成された PRD の情報は全て ESD に保存される。

### 4.3 TGC オフラインデコードソフトウェアの概要

前節ではミュオンスペクトロメータのデータデコードの流れについて述べたが、この節では、特に TGC システムにおけるデータデコーディングについて焦点を当て、ミュオン全体のパッケージである MuonSpectrometer 内に設けられたオフラインデコードソフトウェアの主なものについてまとめる。

MuonSpectrometer Software Package は、Coverter や Cabling 等の役割毎にサブパッケージに分類され、それが更に階層構造を持つ場合もある。プログラムはいずれかのパッケージに属し、その中の他のプログラムと共にバージョン管理される。TGC に関わる主なパッケージとして数点以下にまとめる。

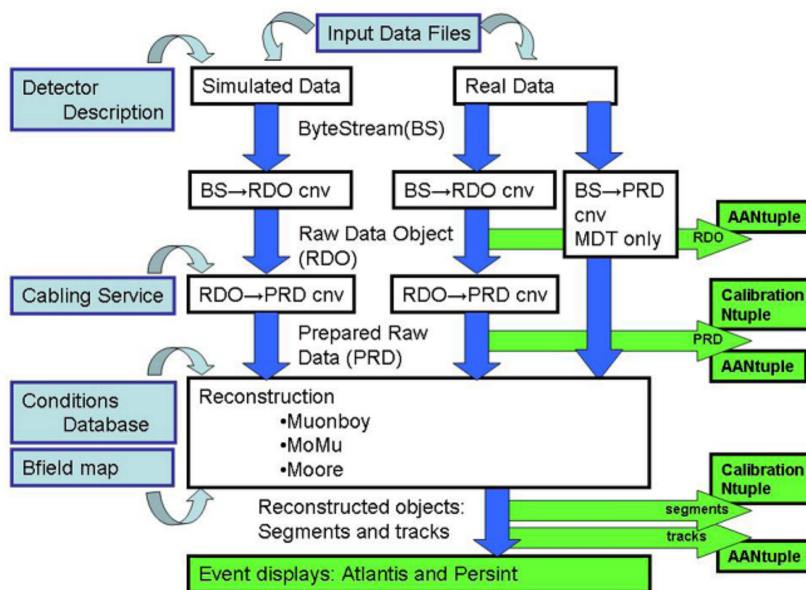


図 4.3: RAW Data から PRD までのデコーディングの流れ [12]

### 4.3.1 MuonCnv/MuonTGC\_CnvTools

RAW Data、RDO、PRD、Digit の変換を行うことができるクラスを含んでいるのが、MuonCnv/MuonTGC\_CnvTools である。MuonTGC\_CnvTools のクラス階層構造を図 C.1 に示す。MuonTGC\_CnvTools には、様々なデータ形式、変数が用いられている。図 4.4 に示すように、TGC の Identifier には 4 種類ある。

- OfflineID  
PRD 内で用いられる形式で、多くのツールの中で使用されている。チャンネルの同定に用いられる。
- ReadoutID  
リードアウト系のエレクトロニクスのチャンネルの同定に用いられる。RDO と後述する TrigT1TGC パッケージで用いられている。
- OnlineID  
メカニカルなレベルでのチャンネルの同定に使用される。
- TriggerID  
トリガー系のエレクトロニクスの同定に用いられる。trigger RDO や後述する TrigT1TGC パッケージの中で使われている。

図 4.4 に示すように、RDO には以下の 4 タイプがある。ReadoutID がベースとなって構成されている。

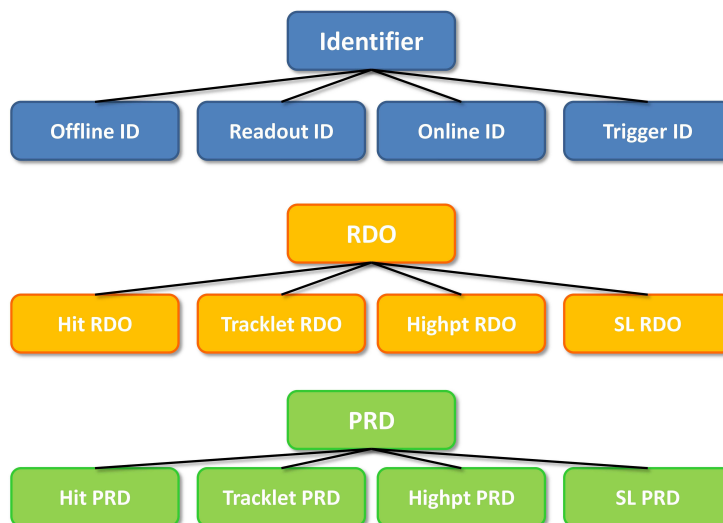


図 4.4: MuonCnv/MuonTGC\_CnvTools で使用されているデータ形式及び変数

- Hit RDO  
SLB から出力される Hit 情報が含まれた RDO である。
- Tracklet RDO  
Low- $p_T$  コインシデンスの情報と、Triplet のコインシデンス情報が含まれた RDO である。
- Highpt RDO  
High- $p_T$  コインシデンスの情報が含まれた RDO である。
- SL RDO  
 $r$ - $\phi$  コインシデンスの情報が含まれた RDO である。

図 4.4 に示すように PRD には以下の 4 タイプが存在する。OfflineID がベースとなって構成されている。

- Hit PRD  
Muon::TgcPrepData クラスが使用される。Hit 情報を含んだ PRD である。
- Tracklet PRD  
Muon::TgcCoinData クラスが使用される。Low- $p_T$  コインシデンスの情報が含まれた PRD である。
- Highpt PRD  
Muon::TgcCoinData クラスが使用される。High- $p_T$  コインシデンスの情報が含まれた PRD である。

- SL PRD

Muon::TgcCoinData クラスが使用される。r- $\phi$  コインシデンスの情報が含まれた PRD である。

以上の、Identifier、RDO、PRD のそれぞれの変数を構成する変数の詳細は、付録 B にまとめる。

#### 4.3.2 MuonCommissioning/MuonCommAlgs

このパッケージはコンバータによって作られた RDO、PRD のコンテナを呼び出して、イベント内の RDO、PRD オブジェクトに対してループさせ、その変数を AANtuple に格納するという役割を持つ。MuonCommAlgs のクラス階層構造を図 C.2 に示す。

TGC の部分は TgcPrepRawDataNtuple というコードが受け持つ。このパッケージによって作成された AANtuple はデバッグやキャリブレーションに用いられる。

#### 4.3.3 MuonConditions/MuonCablings/TGCCabling12

一般的に Cabling Service と呼ばれるのがこのパッケージである。ATLAS では読み出しのチャンネルが膨大なため、それらを処理するエレクトロニクスのチャンネル、モジュール間の配線等は膨大かつ複雑なものとなっている。また、エレクトロニクス系の ID 情報と読み出しチャンネルの ID 情報、そしてオフラインで用いる時の ID 情報のフォーマットは全て統一されていない。しかし後述の TrigMuonCoinHierarchy のようにオフラインの ID 情報を用いてエレクトロニクスの動作評価を行う場合やメンテナンス面からも、それらの間の対応関係を容易かつ迅速に行えることは実験において必須である。それを可能にするためには、配線やジオメトリのデータベースを 1 箇所に集中することが必要であり、その実装が Cabling Service である。

Cabling Service は、Cabling Service のベース部分となる TGCCablingInterface、12 回対称の配線を考慮した TGCCabling12、過去の 8 回対称の配線を考慮した TGCCabling から主に構成されている。特に TGCCabling12 は、現在の 1/12 セクターに基づく、12 回対称の配線を考慮した Cabling Service の本体部分である。実データでは基本的にこの TGCCabling12 が用いられる。TGCCabling12 のクラス階層構造を図 C.4 に示す。8 回対称の配線を考慮した TGCCabling は主にシミュレーションで使われてきた。ただし 2008 年後半から 12 回対称への移行準備が整ったため、現在ではほとんど用いられていない。

#### 4.3.4 MuonReconstruction/MuonRecExample

このパッケージはミュオンのリコンストラクションを行う。ユーザーはこれを用いることで、自らミュオンのリコンストラクションを行うことができる。主に以下の 3 つのジョブオプションが使われる。

- MuonRec\_jobOptions.py

ミュオンのリコンストラクションを行うためのメインのジョブオプションである。

- MuonRec\_myTopOptions.py  
ユーザーがミュオンスタンドアロンリコンストラクションを行うためのジョブオプション。モンテカルロイベントを入力とする。
- MuonDataRec\_myTopOptions.py  
ユーザーがミュオンスタンドアロンリコンストラクションを行うためのジョブオプション。実データを入力とする。

MuonRecExample のクラス階層構造を図 C.5 に示す。

#### 4.3.5 MuonReconstruction/MuonRecEvent/MuonRDO

このパッケージはミュオンの RDO の型を定義するものである。

MuonRDO のクラス階層構造を図 C.6 に示す。

図 C.6 中の TgcRawData というコードによって、TGC の RDO の型が定義される。図 4.4 に示したように、MuonRDO には hit、tracklet、highpt、sl の 4 種類が存在している。以下に各 RDO についてまとめる。ここでは次章の TrigT1TGC を用いた解析を視野に入れて、各 RDO オブジェクトに関して詳しくまとめる。オブジェクト内の変数の値域等は、付録 B に示す。

##### hitRDO

TGC の SLB リードアウト系の出力データ 200bit をもとに作られる。SLB のバイトストリームデータフォーマットを図 D.1 に示す。図 D.1 の通り、出力される 200bit のうち始めの 40bit はトリガーデータを表し、残りの 160bit の各 1bit ずつが検出器の 1 つのチャンネルに対応して、ヒットがあれば bit が立つ仕組みになっている。bit が 1 つ立つ毎に、それに対応したヒットの hitRDO が作られる。そのため 2 つのチャンネルに hit があった場合は 2 つの hitRDO が作られる。また、どの bit が立ったかは、そのまま hitRDO 内の channel という変数の中に int 型として代入される。

以下にオブジェクト内の各変数について説明する。

- bcTag  
データに割り振られたバンチ。SLB は L1A を受け取りデータを読み出す際に、対応するバンチの前後 1 バンチを含めて 3 バンチ分 (Previous、Current、Next) を読み出す。bcTag によって 3 バンチのうち、どのバンチ由来のものかを識別することができる。
- subDetectorId  
これは ATLAS 検出器内の全ての検出器に割り振られた番号である。TGC は Aside、Cside に分けて割り振られており、それぞれ 103、104 である。
- rodId  
どの ROD から読み出されたデータであるかを示す。ROD は 1/12 セクター毎に割り当てられているので、これは即ちそのデータがどの 1/12 セクター由来のものか識別することに使うことができる。

- sswId  
信号が通ってきた SSW の ID を示す。SSW は Big Wheel では 1/12 セクター毎に 8 枚用いられている。その 8 枚を区別する。
- slbId  
これは sbLoc を表している。SLB の ID は 0~15 までしかないので、そのまま用いると 1 つの SSW につながる SLB の ID に重複が生じてしまう。SSW 内での重複が生じないように割り振られているのが sbLoc である。
- l1Id  
レベル 1 トリガー信号の番号。TGC の PSB でカウントしたものである。
- bcId  
ビームバンチの番号。TGC の PSB でカウントしたものである。
- slbType  
どのタイプのチェンバーの SLB 由来の情報かを示す。Doublet Wire、Doublet Strip、Triplet Wire、Triplet Strip、Inner Wire、Inner Strip の区別をすることができる。
- adj  
SLB の adjacent 領域の bit がどうかを示す。隣り合うワイヤー領域をカバーする SLB 同士は、コインシデンスを取る都合上、端の数チャンネルを共有している。よってその領域のチャンネルにヒットがあると 2 つの SLB で bit が立ち、hitRDO が 2 つ作られてしまう。そこで hit オブジェクトを重複してカウントしないように片方の SLB では adjacent というタグを付ける。
- channel  
SLB の出力 200bit のうちヒット情報を司る 160bit を値域にもち、SLB のどのチャンネルからのヒットを示す。

上記の変数のうち、subDetectorId、rodId、sswId、slbId、adj、channel の 6 つを合わせて ReadoutID と呼んでいる。これらの情報を合わせると例えば、「Aside の sector2 の SSW4 につながる slbId0 の SLB の 68bit 目」というのは、「triggerSector4 の Endcap の Doublet Wire の layer7 の r が最大の位置にある channel」であるというように、ReadoutId が一意に決まると、エレクトロニクス上の約 32 万のチャンネルから 1 つに絞り込むことができるようになっている。Cabling Service 内の getReadoutIDfromOfflineID 関数、getOfflineIDfromReadoutID 関数を用いることによって OfflineID との相互変換が可能である。

### trackletRDO

TGC の SLB のリードアウト系の出力データをもとに作られる。図 D.1 の通り、SLB の出力 200bit のうち、始めの 40bit には SLB でのコインシデンスの結果が出力される。Triplet Wire は 1 つの SLB から最大 3 個、Triplet Strip は最大 8 個、その他のタイプのチェンバーからは最大 2 個のコインシデンス情報が含まれる。trackletRDO 自体は全ての



SLB から作られる。ただし後にも書くように trackletPRD には doublet の trackletRDO のみを変換される。コインシデンス情報では検出器のチャンネル単位よりも細かい trgch(トリガーチャンネル) という単位で位置情報を与える。これは doublet であれば2層のスタックリングにより 1/2 チャンネル、TripletWire であれば3層による 1/3 チャンネルの単位に相当する。

以下にオブジェクト内の各変数について説明する。

- bcTag  
hitRDO の bcTag の項目に同じ。
- subDetectorId  
hitRDO の subDetectorId の項目に同じ。
- rodId  
hitRDO の rodId の項目に同じ。
- sswId  
hitRDO の sswId の項目に同じ。
- slbId  
hitRDO の slbId の項目に同じ。
- l1Id  
hitRDO の l1Id の項目に同じ。
- bcId  
hitRDO の bcId の項目に同じ。
- slbType  
hitRDO の slbType の項目に同じ。
- delta  
TGC2 と TGC3 のヒット間の位置のずれを trgch 単位で表す。ワイヤーでは数値が正ならば、TGC3 のヒットに比べて、TGC2 のヒットが  $\eta$  の大きい方 ( $r$  の小さい方) にずれていることを表す。ストリップでは Aside の Forward チェンバーと Cside の Backward チェンバーでは、delta が正ならば TGC2 のヒットが  $\phi$  の大きい側にずれていることを表す。Cside の Forward チェンバーと Aside の Backward チェンバーでは、delta が正ならば TGC2 のヒットが  $\phi$  の小さい側にずれていることを表している。delta は doublet 用の Low- $p_T$  のコインシデンスウインドウの大きさに対応している。
- seg  
この変数は Triplet Strip のみで用いられるものである。Triplet Strip では1つの SLB で隣り合う2つの Triplet Strip チェンバーを見ているため、その2つを TRIGA(seg=0)、TRIGB(seg=1) と分けることで区別している。SLB からの出力 200bit のトリガーデータを司る始めの 40bit の部分が TRIGA、TRIGB と分けられていて、その構図は図 D.1 を参照されたい。

- sub  
slbsubMatrix のことである。後述の PRD オブジェクトの中で trackletId という変数が出てくるが、その実態がこれである。1つのイベントの中で、1つの slbsubMatrix からは1つのコインシデンス情報しか出てこない。もしも複数のヒットがあった場合はデクラスタリングが行われる。1つの slbsubMatrix は 32trgch から構成されている。
- rphi  
slbsubMatrix 内のどの trgch でコインシデンスがとれたかを表す変数。ただしこれは TGC3 上の値である。

上記の変数のうち、subDetectorId、rodId、sswId、slbId、sub、rphiをまとめて、LowPt-CoincidenceID と呼ぶ。ReadoutID の時と同様、Cabling Service を用いることで OfflineID との相互変換が可能である。用いる関数は Cabling Service 内の getOfflineIdfromLowPt-CoincidenceID 関数等を用いる。

### highptRDO

SLB のトリガー出力は HPT ボードを経由して SL に届く。HPT の出力は SL でのコインシデンス処理結果と共に SL の SLB から出力される。そのため highptRDO は SL の SLB の出力をもとに作られる。HPT では位置情報を RoI 単位で扱う。

以下にオブジェクト内の各変数について説明する。

- bcTag  
hitRDO の bcTag の項目に同じ。
- subDetectorId  
hitRDO の subDetectorId の項目に同じ。
- rodId  
hitRDO の rodId の項目に同じ。
- sswId  
hitRDO の sswId の項目に同じ。
- slbId  
hitRDO の slbId の項目に同じ。
- l1Id  
hitRDO の l1Id の項目に同じ。
- bcId  
hitRDO の bcId の項目に同じ。
- strip  
経由した HPT がワイヤーカストリップのものであったかを区別する。

- forward  
経路した HPT がエンドキャップかフォワードのものであったかを区別する。
- sector  
1/12 セクターの中の、どのサブトリガーセクター由来の情報かを表す。
- chip  
HPT ボードの何番目のチップを通過したかを表す。
- index  
1つの HPT チップからは最大 2つのトリガー候補が出るが、その 1 番目か 2 番目かを表す。
- hipt  
結果的に High- $p_T$  コインシデンスがとれたのか、Low- $p_T$  コインシデンスがとれたのかを表す。
- hitId  
6 個のマトリックスに分けられた HPT チップ内の、どのマトリックスを通過してきたかを表す。
- sub  
上記の 6 個に分かれたマトリックスは、更に上下に半分ずつ hptsubMatrix というものにわかれる。0 が上、1 が下に対応している。また hptsubMatrix は RoI1 つ分に対応する。
- delta  
High- $p_T$  コインシデンスがとれた時には、TGC1 と TGC2/3 の間の位置のずれが trgch 単位で代入される。このときの delta は High- $p_T$  コインシデンスウインドウの大きさに対応している。注意点として絶対値の大きいところでは delta は trgch の差分そのものを直接表すわけではない。ワイヤーの  $11 \leq |\text{delta}|$  は、trgch 差 12、14、16、18、20 に対応し、ストリップの  $5 \leq |\text{delta}|$  は、trgch 差 7、9、12 に対応する。結果的に High- $p_T$  コインシデンスがとれず Low- $p_T$  判定が下された時は、SLB からきた TGC2 と TGC3 の間の位置のずれがそのまま代入される。

上記の変数のうち、subDetectorId、rodId、strip、forward、sector、chip、hitId、sub をあわせて HighPtID と呼ぶ。これを用いることで TGC3 上のどの RoI から来た情報かを特定することができる。ただしワイヤーの場合は r 方向、ストリップなら  $\phi$  方向のみの RoI 判別しかできない。また Cabling Service を用いることで OfflineID との相互変換が可能である。変換をするには Cabling Service 内の getOfflineIDfromHighPtID 関数等を用いればよい。ただし前述の通り HPT では位置情報を RoI 単位で扱うため、HighptID 側から OfflineID を求める時は、概ね 8 チャンネル分に相当する RoI レベルまでしか特定することができないので、Cabling 関数では RoI の中心のチャンネルの OfflineID を取ってくるようになっている。

## sIRDO

SLにおいてHPTからのワイヤーとストリップの入力のコインシデンスをとった情報をもとに作られる。Look Up Table(LUT)を用いた6段階の $p_T$ の判別が行われ、1つのSLからはより大きな $p_T$ を持つ最大2つのSLトリガーの情報が出される。

- bcTag  
hitRDOのbcTagの項目に同じ。
- subDetectorId  
hitRDOのsubDetectorIdの項目に同じ。
- rodId  
hitRDOのrodIdの項目に同じ。
- l1Id  
hitRDOのl1Idの項目に同じ。
- bcId  
hitRDOのbcIdの項目に同じ。
- cand3plus  
3つ以上のトリガー候補があったときにbool型がtrueになるようになっている。
- forward  
highptRDOのforwardの項目に同じ。
- sector  
highptRDOのsectorの項目に同じ。
- index  
2つのSLトリガー候補のうちの1番目か2番目かを区別することができる。
- muplus  
SLにおけるdeltaRの値が正か負かを表す。deltaR=0の場合は正として扱われる。
- threshold  
6段階の $p_T$  thresholdのうち、どの閾値のトリガーが発行されたかを示す。
- roi  
TGC3上のどのRoIからのトリガー情報であるかを示す。

上記の変数のうち、subDetectorId、rodId、forward、sector、roiをまとめてSIIDと呼ぶ。SIIDからはワイヤー、ストリップ両方のOfflineIDを得ることが出来る。ただしCabling関数で得られるのはRoIの真ん中のチャンネルのOfflineIDであって、チャンネルレベルの位置情報を得られるわけではない。

### 4.3.6 MuonReconstruction/MuonRecEvent/MuonPrepRawData 及び Muon-TrigCoinData

この2つパッケージは PRD の型を定義する。PRD にも RDO と同様に4種類のタイプが存在しており、図 4.4 の通り hit、tracklet、highpt、sl となっている。MuonPrepRawData では、そのうちの hitPRD を定義する。そして MuonTrigCoinData では、tracklet、highpt、sl の型を定義する。MuonPrepRawData のクラス階層構造を図 C.7 に、MuonTrigCoinData のクラス階層構造を図 C.8 に示す

各 PRD は対応する RDO の情報をもとに作られる。PRD は必ず OfflineID を1つ以上持っていて、その OfflineID は RDO の ReadoutID 等から Cabling Service を通して取得する。また OfflineID とセットで対応する TgcReadoutElement へのポインタも持っている。TgcReadoutElement は TGC のチェンバー1枚に対応したオブジェクトであり、これと OfflineID を組み合わせることによって ATLAS のグローバルな座標系における位置情報を得ることが可能になっている。

また tracklet、highpt、sl の PRD は trackletId という数値をもつ。trackletId は、1つのイベント内でそこからは最大1つのトリガー情報しか出さないという単位にチェンバーの領域を分けて ID を振ったもので、各 PRD オブジェクト内の trackletId という変数を参照することで得られる。単位としては slbsubMatrix に対応している。すなわち同じイベント内の各 PRD の間で trackletId が同じであれば、同じ hit を起源とした trackletPRD、highptPRD、slPRD であるに対応づけをすることができる。後述する TrigMuonCoinHierarchy のパッケージも、各 PRD 間の対応づけに trackletId を用いている。

PRD の変数の詳細については付録 B でまとめる。以下では各種 PRD について簡単にまとめる。

#### hitPRD

hit に対応したチャンネルの OfflineID を保持する。OfflineID は PRD オブジェクト内の RDOid という変数を参照することで得られる。オブジェクトとして持つ位置情報は TgcReadoutElement 内でのローカル位置座標系 (チェンバーの中心を 0 とした 2次元座標系) だが、Cabling Service を通してグローバルな 3次元デカルト座標系に変換することができる。その他の基本的な情報は、Cabling Service の TgcIdHelper というクラスを用いることによって OfflineID から得ることができる。

#### trackletPRD

trackletPRD は trackletRDO の Doublet 部からのみ作られ、Triplet の RDO は用いられない。TGC2 の hit と TGC3 の hit に由来するコインシデンス情報で、それぞれの hit に対応した2つの OfflineID を持つ。それぞれの OfflineID は PRD オブジェクト内の Identifier 型の ChannelIdIn、ChannelIdOut の変数を参照することによって得ることができる。同様に位置情報なども2つずつ持っている。delta 値は trackletRDO のものがそのまま入る。

## highptPRD

TGC1 と TGC2/3 の hit に由来するコインシデンス情報であり、保持する OfflineID は TGC1 と TGC3 の hit に対応している。trackletPRD 同様、それぞれの OfflineID は PRD オブジェクト内の Identifier 型の ChannelIdIn、ChannelIdOut の変数を参照することによって得ることができる。ただし保持する位置情報、位置座標は RoI レベルであるので、OfflineID は RoI の中心のチャンネルのもので代表されることになっている。これが後述の TrigMuonCoinHierarchy パッケージにおいて、TGC1 上のヒット情報を取得するのが複雑になっている原因でもある。ただし width という変数はそのチャンネルの幅ではなく、その RoI の幅である。delta 値は highptRDO のものがそのまま入る。

## slPRD

ワイヤーとストリップのコインシデンス情報であり、保持する OfflineID はワイヤーとストリップに関して1つずつである。この時の OfflineID は対応する RoI の中心にある TGC3 のチャンネルで代表する。roi と pt の変数には slRDO の roi、threshold の値がそのまま入る。

### 4.3.7 MuonSimulation/MuonDigitization/TGC\_Digitization

TGC\_Digitization にはシミュレーションによって得られた TGC のヒット情報を TGC の digit 情報に変換する機能と、検出器の応答をシミュレートする機能を備えている。当然のことながらシミュレーション解析において使われるパッケージである。TGC\_Digitization のクラス階層構造を図 C.3 に示す。

## 4.4 TrigMuonCoinHierarchy

Trigger/TrigAnalysis/TrigMuonCoinHierarchy は、TGC のトリガーとヒットの階層関係を利用して、トリガー情報が存在した時に上の階層に対応するトリガー情報が存在するかや、トリガーが存在している時にそれに対応する (トリガーの元となった) ヒットが存在するかをチェックすることができるパッケージである。このパッケージを用いることで、ハードウェアの不具合の発見やデコードに用いられるソフトウェアの性能評価及びデバッグに役立てることができる。私は次章で述べる TrigT1TGC を使った LVL1 トリガーの動作検証において、その検証方法自体の性能を評価するにあたって、この TrigMuonCoinHierarchy パッケージを用いた。

本節では TrigMuonCoinHierarchy の概要と私が開発に携わった部分についてまとめた後、パッケージ自身の性能について考察する。

### 4.4.1 TrigMuonCoinHierarchy の概要

前節 4.3.6 で述べたように MuonPRD には4つのパターン、すなわち hitPRD、trackletPRD、highptPRD、slPRD がある。これらは TGC のハードウェアに置き換えることがで

きるといえる。hitPRD はイベント毎に TGC 上に存在したヒット情報そのもの、tracklet-PRD は Low- $p_T$  コインシデンスのトリガーが発行されたかどうか、highptPRD は High- $p_T$  コインシデンスのトリガーが発行されたかどうか、slPRD は  $r-\phi$  コインシデンスのトリガーが発行されたかどうか、である。そして、ここで特筆すべきは SL において  $r-\phi$  コインシデンスを満たすかチェックされるまでは、Wire のヒット情報及びトリガー情報、Strip のヒット情報及びトリガー情報は別個に扱われるということである。以上のことを SL のデータオブジェクトを頂点として階層構造で表せば図 4.5 のようになる。この階層構造は

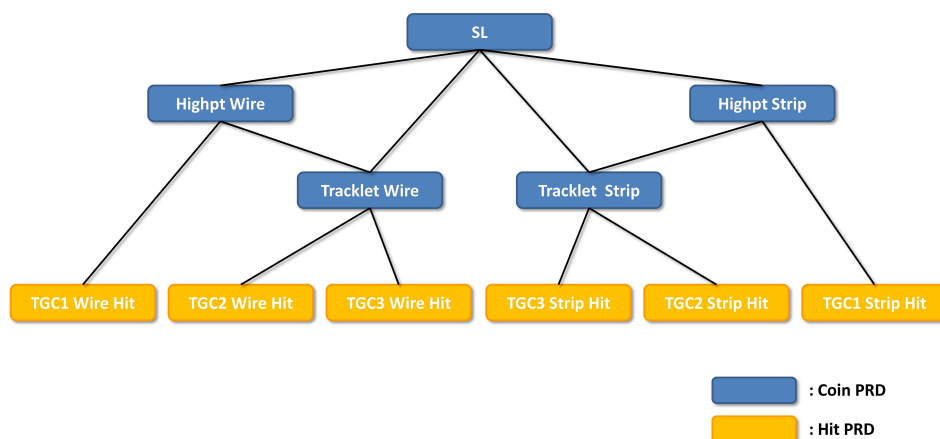


図 4.5: TGC のヒットとトリガーの階層構造図

次のように理解することができる。仮に頂点の SL のデータオブジェクトが存在すると、1 つ下位の HighptWire、HighptStrip、TrackletWire、TrackletStrip に対応するデータオブジェクトが存在しているはずである。また HighptWire のデータオブジェクトが存在する時は、その下位の TrackletWire と TGC1WireHit に対応するデータオブジェクトが存在しているはずである。このように上位のトリガーが存在したときに、下位のトリガーもしくはヒットが理論的には必ず存在する。この階層関係を利用したのが TrigMuonCoinHierarchy パッケージである。

図 4.5 のような階層構造が見つかったときに、4.3.6 で述べたヒットとトリガーの情報を保持する TgcPrepData と TgcCoinData のポインタを取得し、ヒット、コインシデンス、トラック等からアクセスすることができるようになっている。

またこのパッケージを用いると Trigger 用のハードウェアに蓄積されるヒット情報を取得し、ハードウェア間での情報の伝達がどの程度正しく行われているかを調べることができる。例えば上位の階層にオブジェクトが存在した時に下位の階層にオブジェクトが存在しなかったとする。本来、このような現象が起こるはずはないが、これを前述の考え方の通りハードウェアに置き換えたとなると、トリガー系のエレクトロニクスで、流れの中で途中から存在しないはずのトリガー情報が加わっていることを表す。これはハードウェアの誤作動、例えば常に信号を出し続けているハードウェアが存在する可能性、実験で用いられているデータのデコード用ソフトウェアに問題がある可能性などが示唆され、TrigMuonCoinHierarchy はそのチェックに役立てることができる。

まとめると TrigMuonCoinHierarchy パッケージを用いると以下のようなことが可能に

なる。

- オフライン上でのハードウェアの不具合の有無のチェック
- デコードに用いるソフトウェア内のバグの発見
- トラック情報から、元になったヒットが生成したトリガー情報の取得

TrigMuonCoinHierarchy の基本構成は ICEPP の織田勸氏によって開発されたものである。私はパッケージを構成するクラスの開発、改良を行った。

TrigMuonCoinHierarchy のクラス階層構造を図 4.6 に示し、以下にパッケージの構成についてまとめる。

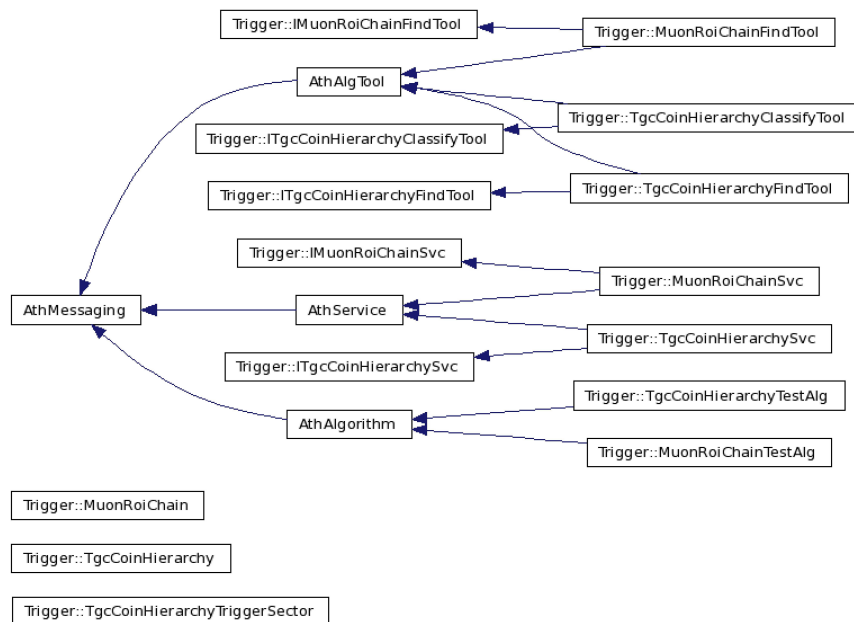


図 4.6: Trigger/TrigAnalysis/TrigMuonCoinHierarchy のクラス階層構造 [12]

### TgcCoinHierarchy クラス

このクラスは TgcCoinData(CoinPRD) と TgcPrepData(HitPRD) のポインタを前述の階層構造を保ったまま保持するためのメソッドを記述する。TgcCoinHierarchy 型の変数は TgcCoinHierarchySvc によって管理される。TgcCoinHierarchy の型は以下のものに基づいて構成されている。

- Identifier、TgcPrepData 等のヒット情報
- トラック情報から得られるヒット



- TgcCoinData からのコインシデンス情報
- トリガーセクター番号  
1/12セクターを4分割したもの。すなわち Endcap で48トリガーセクター、Forward で24トリガーセクターある。
- RoI 番号

#### TgcCoinHierarchySvc クラス

TgcCoinHierarchy のポインタを提供するためのメソッドを記述する。

#### TgcCoinHierarchyClassifyTool クラス

TgcCoinHierarchyTriggerSector を用いて、TgcCoinData と TgcPrepData をそれぞれのトリガーセクターにクラス分けする。これは、トリガー判定はそれぞれのトリガーセクター毎に行われるが TgcCoinData と TgcPrepData は1/12セクター毎に記録されているからである。

#### TgcCoinHierarchyTriggerSector クラス

1つのトリガーセクター内に存在する複数の TgcCoinData と TgcPrepData を保持する。TgcCoinHierarchyClassifyTool によって呼び出され、TgcCoinHierarchyFindTool を用いて TgcCoinHierarchy を構成するために呼び出される。

#### TgcCoinHierarchyFindTool クラス

トリガーもしくはヒットの階層関係において必要十分条件を満たすものを見つけ、TgcCoinHierarchy のインスタンスを作成するためのクラス。CoinPRD 同士の比較には PRD オブジェクト内の trackletId という変数の一致を確認することで行う。CoinPRD と HitPRD 間の比較には、TGC Cabling Service を用いる。パッケージの核となる部分である。

#### TgcCoinHierarchyTestAlg クラス

TgcCoinHierarchySvc を実装したサンプルプログラム。また、基本的な対応関係を確認するプログラムが含まれている。ここでいう対応関係は、

- SL → Tracklet Wire(Strip)  
SL で Low- $p_T$  判定が下された PRD が存在したとき、それと同じ trackletId を持つ PRD が Tracklet Wire(Strip) に存在するか否か
- SL → Highpt Wire(Strip)  
SL で High- $p_T$  判定が下された PRD が存在したとき、それと同じ trackletId を持つ PRD が Highpt Wire(Strip) に存在するか否か

- Highpt Wire(Strip) → Tracklet Wire(Strip)  
HPT でコインシデンスがとれたとき、Highpt Wire(Strip) の PRD 内の trackletId と同じ trackletId を持つ PRD が Tracklet Wire(Strip) に存在するか否か
- Highpt Wire(Strip) → TGC1 Wire(Strip) Hits  
HPT でコインシデンスがとれたとき、その Wire(Strip) の情報が TGC1 において元となるヒット (High- $p_T$  コインシデンス条件を満たしているヒット) が存在するか否か
- Tracklet Wire(Strip) → TGC2 and TGC3 Wire(Strip) Hits  
SLB でコインシデンスがとれたとき、その Wire(Strip) の情報が TGC2 と TGC3 において元となるヒット (Low- $p_T$  コインシデンス条件を満たしているヒット) が存在するか否か

の計 10 パターンについて調べることができる。

#### 4.4.2 TrigMuonCoinHierarchy の改良

##### mask2

私は TrigMuonCoinHierarchy のパッケージに mask2 の効果を組み込んだ。mask2 というのは、コインシデンスに先立って信号を常に high にしたり、low にするロジックのことである。図 4.7 は、Doublet における SLB のヒットからトリガーが発行されるまでのブロック図で、赤での点線で囲った部分を一般的に mask2 と呼んでいる。mask2 上で特定のチャ

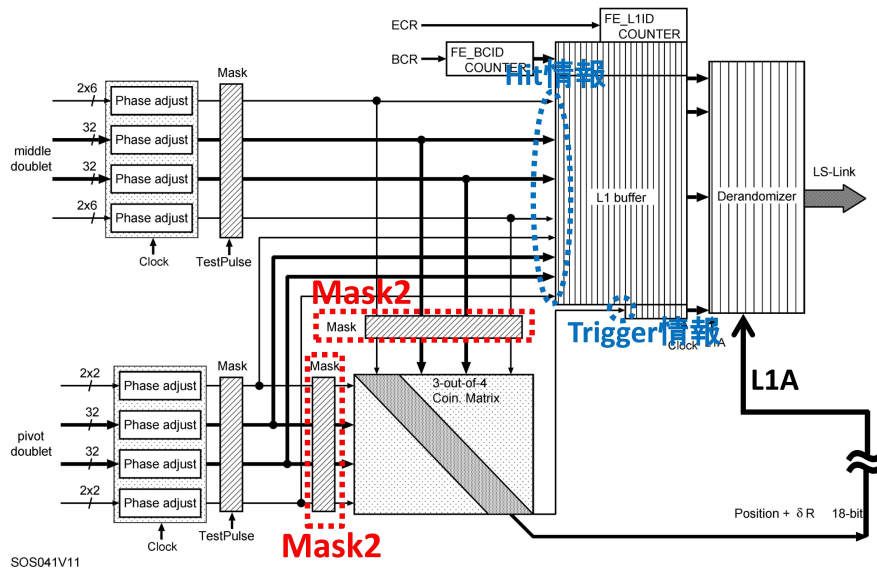


図 4.7: SLB(Doublet) のブロック図と mask2[13]

ンネルを常に high の状態 (mask in) にしておくということは、例えば Doublet の場合は

例え 4/4 コインシデンスを要求するコインシデンスロジックになったとしても、HV がわからない等の物理的要因によって生まれたデッドチェンバーや全く信号が出なくなってしまったデッドチャンネルによるアクセプタンスの損失を防ぐということにつながる。逆に特定のチャンネルを常に low の状態 (mask out) するという事は、ヒットがないのに高い頻度で信号を出し続けるノイジーチャンネル等のようにトリガーには使いたくないチャンネルを除外することができる。mask2 は以下の手順で更新される。

1. TGC シフターが calibration period にノイズランを取り出す。  
ノイズランというのは、ヒットと関係なくランダムなパターンで L1A 信号を出して、ランダムに引っ掛かるノイズの数を見るランのことである。
2. 10,000 回の L1A 信号 (ASD テストパルス) を出して、 $\frac{\text{そのチャンネルのヒット回数}}{\text{全トリガーの回数}}$  を計算する。
3. その数が  $10^{-4}$  よりも高いチャンネルをノイジーチャンネルとする。
4. ノイジーチャンネルとして同定されたものを mask out するように configuration parameter を更新する。  
ここでの解析と parameter の更新は on-call シフターの仕事である。
5. 同様に ASD テストパルスを使ってデッドチャンネルの有無も確認する。  
デッドチャンネルが存在した場合は mask in チャンネルの更新が行われる。

mask2 のリストはテキストファイルの形式で保存され、mask2 が更新された後、mask2 ファイルの更新を担っている研究者によって更新される。また更新された日付がファイル名の中に含まれており、且つ過去のファイルは全て保管される仕組みがとられているため、注目する run がとられた日付がわかると、その run 当時に適用されていた mask2 の configuration parameter を知ることができる。

保存されている mask2 ファイルは参照する人間にわかりやすいように独自のフォーマットがとられ、Identifier の ID 形式とは異なるフォーマットが用いられている。まずファイルの中身は大きく次の 2 つのパートから構成されている。

- BIT\_DEFINITION 1  
この行以下のチャンネルはダミーヒットを埋めている (mask in している)。
- BIT\_DEFINITION 0  
この行以下のチャンネルはノーヒットにしている (mask out している)。

次に各チャンネル情報は、例えば以下のような一行で表される。

A    01    3    E    S    6    1    18

各変数について左から順に表 4.1 にまとめる。

ここで注意しなければならないことは、mask2 の情報は図 4.7 からも明らかなように、コインシデンスをとる直前に与えられる。そのため mask in のチャンネルを使うことによって初めてコインシデンスがとれたイベントは、Readout ラインにはコインシデンスがとれたと情報が流れる。しかし、そのときのヒットパターンを参照すると mask in のチャンネルが含まれていないため、コインシデンスがとれる条件を満たしているように見えないう矛盾が生じてしまう。

変数名 (仮)	型	値域	意味
AorC	char	A/C : Aside/Cside	Aside か Cside を表す
sector	int	01~12	1/12 セクターの番号を表す
phi	int	Endcap : 0~3 Forward : 0/2	1/12 セクター内のサブセクターを表す (図 A.2 参照)
EorF	char	E/F : Endcap/Forward	Endcap か Forward かを示す
SorW	char	S/W : Strip/Wire	Strip か Wire かを示す
layer	int	1~7	7層の layer のうち内側から何層目かを示す
chamber	int	TGC1Endcap : 1~4 TGC1Forward : 0 TGC2,3Endcap : 0~4 TGC2,3Forward : 0	r の減少方向に何番目のチェンバーか
channel	int	Wire : 0~n Strip : 0~31	チャンネル番号 (OnlineID と同じ割り振られ方)

表 4.1: mask2 ファイルのフォーマット

### TrigMuonCoinHierarchy への mask2 の組み込み

TrigMuonCoinHierarchy-00-02-07 までは、この mask2 をパッケージ内に考慮に入れていなかった。そのため前述の通り、コインシデンス情報とヒットパターンの中に食い違いが生じ、

Tracklet Wire(Strip) → TGC2 and TGC3 Wire(Strip) Hits  
Highpt Wire(Strip) → TGC1 Wire(Strip) Hits

の計 4 パターンの対応関係を落とす大きな要因となっていた。私は mask2 の影響をパッケージに反映させるべく、TgcCoinHierarchyFindTool クラスの改良を行い、それぞれの場合において次のようなアルゴリズムを付け加えた。

### Tracklet Wire(Strip) → TGC2 and TGC3 Wire(Strip) Hits の場合

Tracklet PRD とヒット情報の対応関係を見るには、両者が持つ OfflineID を Cabling Service を用いて LowPtCoincidenceID に変換し比較する形がとられている。この作業は TgcCoinHierarchyFindTool クラス内の findTGC23HitFromTracklet 関数内で行われている。今回の改良では LowPtCoincidenceID への変換の前段階で、mask2 内の mask in の OfflineID を適用させることにした。具体的に加えたアルゴリズムは以下の通りである。

1. あらかじめ mask in の各チャンネルを mask2 ファイルのフォーマットから OnlineID に変換させておく。  
この変換には新たにパッケージ内に作成した DummyHitChannelToOnlineID というクラスを利用する。

2. Tracklet PRD に対応する TGC2,3 のヒットパターンが見つからなかったとき、その Tracklet PRD の元になった TGC2,3 の対応がつかないヒットの OfflineID を取得する。
3. 1 で OnlineID に変換させた mask2 チャンネル全てを OfflineID に変換させる。この変換には Cabling Service を用いる。
4. 2 で得た OfflineID が 3 で得た OfflineID の中に含まれていれば、mask in の影響でコインシデンスがとれたのだと考え、その対応関係はとれたものと認識する。

### Highpt Wire(Strip) → TGC1 Wire(Strip) Hits の場合

Highpt PRD が保持する位置情報は RoI レベルであるため、OfflineID は RoI の中心のチャンネルで”代表”される。よって Highpt PRD とヒット情報の対応関係を OfflineID レベルで得ることはできないため、以下の流れが TgcCoinHierarchyFindTool クラス内の findTGC1HitFromHiPt 関数内で組み込まれている。

1. Highpt の OfflineID をもとに由来となった TGC3 上の 2 つのビットポジション (bitpos) を取得する。  
OfflineID から Cabling Service を用いて SLBID を取得し、TGC3 の bitpos は SLBID から得ることが出来る。
2. TGC3 の 2 つの bitpos と HPT コインシデンスを取りうる TGC1 上の 2 つの bitpos を取得する。
3. 取得した TGC1 の 2 つの bitpos を SLB のチャンネル番号に変換する。
4. TGC1 Hits の OfflineID からその SLBID を取得し、そこから得られた bitpos をもとに TGC1 の SLB 上でのチャンネル番号を取得する。
5. 3 で求めた SLB のチャンネルの範囲の中に、4 で求めた SLB のチャンネルが位置していれば対応関係が成立したとする。

今回の改良では Highpt PRD と TGC1 Hits の間で対応がとれなかったとき、TGC1 上の mask in のチャンネル 1 つずつに対して 4 以降の操作を行うことにした。すなわち、各々の mask in のチャンネルがコインシデンスに使われたという立場に立ち、本当にそれが SLB の bit pos レベルで HPT コインシデンスを満たすことができるか否かというチェックを行うことにした。このチェックを満たす mask in のチャンネルが存在した場合、Highpt PRD と TGC1 Hits の間で対応がとれた判断することにした。具体的には以下のアルゴリズムを加えた。

1. あらかじめ mask in の各チャンネルを mask2 ファイルのフォーマットから OnlineID に変換させておく。  
この変換には新たにパッケージ内に作成した DummyHitChannelToOnlineID というクラスを利用する。

2. Highpt PRD と TGC1 Hits の間で対応がとれなかったとき、1 で OnlineID に変換した mask in のチャンネルを ReadoutID に変換する。  
この変換には Cabling Service を用いる。
3. さらに 2 で得た ReadoutID を SLBID に変換する。  
この変換には Cabling Service を用いる。
4. SLBID から TGC1 上の bitpos を取得し、それを SLB のチャンネル番号に変換する。
5. この SLB のチャンネル番号が Highpt PRD から取得した、取り得る TGC1 上のチャンネル番号の範囲内に一致していれば、対応関係はとれたものと認識する。

以上のように mask2 の影響が組み込まれた TrigMuonCoinHierarchy は、TrigMuonCoinHierarchy-00-03-05 以降のパッケージで用いられている。mask2 の影響を考慮するか否かは、

share/TgcCoinHierarchyTestAlg\_topOptions.py

内でフラグ付けをすることによって使用者によって変更ができるようにした。また 2010 年 11 月 2 日当時までに存在した mask2 のリストファイルは、TrigMuonCoinHierarchy-00-03-11 の share の下に全て置いている。使用したい mask2 ファイルは、

src/TgcCoinHierarchyFindTool.cxx

内で指定できるようになっている。

## 改良結果

mask2 の影響を組み込むことによって、パッケージの性能がどのように変化したかを以下にまとめる。

用いたデータは RunNumber167844 で 2010 年 10 月 29 日に収集されたものである。Athena のリリースは 16.0.2 を用い、パッケージは TrigMuonCoinHierarchy-00-03-11 を用いている。また解析の際には、当時の mask2 のリストである、

share/mask2channels20101029.txt

を用いている。このファイルの中には mask in されているチャンネルが合計で 252 チャンネル記述されてある。

mask in チャンネルを考慮に入れない場合の対応結果を表 4.2 に、mask in チャンネルを考慮に入れた場合の対応結果を表 4.3 に示す。表は左から順に比較した階層、その階層関係において上位階層にてオブジェクトが存在したとき下位階層に存在したオブジェクトの数、非対応率を表している。階層の上下関係は図 4.5 の通りである。非対応率は次式で定義する。

$$\text{非対応率} = 1 - \frac{\text{上位階層と対応がとれた下位階層のオブジェクト数}}{\text{上位階層に存在するオブジェクト数}} \quad (4.1)$$

両者を比較すると、トリガー情報をヒット情報と比較している 4 つの階層、

Tracklet Wire(Strip) → TGC2 and TGC3 Wire(Strip) Hits

Highpt Wire(Strip) → TGC1 Wire(Strip) Hits

において、mask2 を考慮したときの方が対応関係がよくなっていることが伺える。

## 4.4. TRIGMUONCOINHIERARCHY

比較した階層	下位階層 / 上位階層	非対応率
SL→Tracklet Wire	3,147,741 / 3,147,793	$1.7 \times 10^{-5}$
SL→Tracklet Strip	3,147,788 / 3,147,793	$1.6 \times 10^{-6}$
SL(PT=5 or 6)→HiPt Wire	914,351 / 914,351	0.00
SL(PT=5 or 6)→HiPt Strip	914,281 / 914,351	$7.7 \times 10^{-5}$
HiPt Wire→Tracklet Wire	1,911,697 / 1,911,708	$5.8 \times 10^{-6}$
HiPt Wire→TGC1 Wire Hits	1,901,656 / 1,911,708	$5.3 \times 10^{-3}$
HiPt Strip→Tracklet Strip	3,844,078 / 3,844,091	$3.4 \times 10^{-6}$
HiPt Strip→TGC1 Strip Hits	3,839,812 / 3,844,091	$1.1 \times 10^{-3}$
Tracklet Wire→TGC2 and 3 Wire Hits	3,285,935 / 3,292,454	$2.0 \times 10^{-3}$
Tracklet Strip→TGC2 and 3 Strip Hits	4,781,673 / 4,804,454	$4.7 \times 10^{-3}$

表 4.2: TrigMuonCoinHierarchy を用いた run167844 の対応結果 (mask2 を考慮しない場合)

比較した階層	下位階層 / 上位階層	非対応率
SL→Tracklet Wire	3,147,741 / 3,147,793	$1.7 \times 10^{-5}$
SL→Tracklet Strip	3,147,788 / 3,147,793	$1.6 \times 10^{-6}$
SL(PT=5 or 6)→HiPt Wire	914,351 / 914,351	0.00
SL(PT=5 or 6)→HiPt Strip	914,281 / 914,351	$7.7 \times 10^{-5}$
HiPt Wire→Tracklet Wire	1,911,697 / 1,911,708	$5.8 \times 10^{-6}$
HiPt Wire→TGC1 Wire Hits	1,904,049 / 1,911,708	$4.0 \times 10^{-3}$
HiPt Strip→Tracklet Strip	3,844,078 / 3,844,091	$3.4 \times 10^{-6}$
HiPt Strip→TGC1 Strip Hits	3,842,886 / 3,844,091	$0.3 \times 10^{-3}$
Tracklet Wire→TGC2 and 3 Wire Hits	3,285,943 / 3,292,454	$2.0 \times 10^{-3}$
Tracklet Strip→TGC2 and 3 Strip Hits	4,794,389 / 4,804,454	$2.1 \times 10^{-3}$

表 4.3: TrigMuonCoinHierarchy を用いた run167844 の対応結果 (mask2 を考慮した場合)

#### 4.4.3 TrigMuonCoinHierarchy に関するまとめ

表 4.3 に示す通り、各階層においてほぼ対応がとれていて大きな問題はないと言える。表 4.3 の場合に、それぞれどこのトリガーセクターで対応がとれていないかを図にしたものを図 4.8 に示す。横軸はトリガーセクターの番号を示していて、それぞれの領域で  $\phi = 0$  のトリガーセクターから順に Aside のエンドキャップが 1~48、フォワードが 49~72、Cside のエンドキャップが 73~120、フォワードが 121~144 とナンバリングされている。このナンバリングは TrigMuonCoinHierarchy の中でトリガーセクター ID と呼ばれている (図 A.4 参照)。図中の白いピンは対応関係がとれている量、赤いピンは対応関係がとれなかった量を示している。

Tracklet Wire(Strip) → TGC2 and TGC3 Wire(Strip) Hits と Highpt Wire(Strip) → TGC1 Wire(Strip) Hits において赤いピンが目立つ。Tracklet では特定のピンで対応関係がとれないという決まった構造が見られる。これはハードウェアの不具合が原因として考えられる。しかし Highpt の Strip のようにエンドキャップ領域で一様に対応がとれていないのは TrigMuonCoinHierarchy 自身のバグ、もしくは実験で使われているデコーディングソフトウェアのバグが考えられる。

また実験で得られたヒット情報を実機でリコンストラクションされたもの、そのヒット情報に対してシミュレーションでトリガー情報を発生させリコンストラクションしたもの、の2つに関して TrigMuonCoinHierarchy で各階層の対応関係を見たときには、シミュレーションで作られた PRD を入力としたときの方が対応関係が良い結果が出ている。(これについては次の章で詳細を述べる。) すなわち実機とシミュレーションのトリガーの生成の様子が異なることを示している。この原因にはシミュレーションで使われているソフトウェアのバグ、デコーディングで用いられているソフトウェアのバグも考えられるが、一番の原因として考えられるのは、実験中にハードウェアに不具合が発生しシミュレーションと異なる動きをしているということである。

図 4.8 に示すように、現状、Highpt Strip → TGC1 Strip Hits の階層を除いては、対応関係のとれないときの様子から、パッケージ自体にバグが残っている可能性は少ないと言える。そのため TGC エレクトロニクス の性能評価を行うに辺り TrigMuonCoinHierarchy を用いることは有用であると考えられる。

次章では、TrigT1TGC を用いた LVL1 トリガーの性能評価の妥当性について TrigMuonCoinHierarchy を用いて検証してみる。



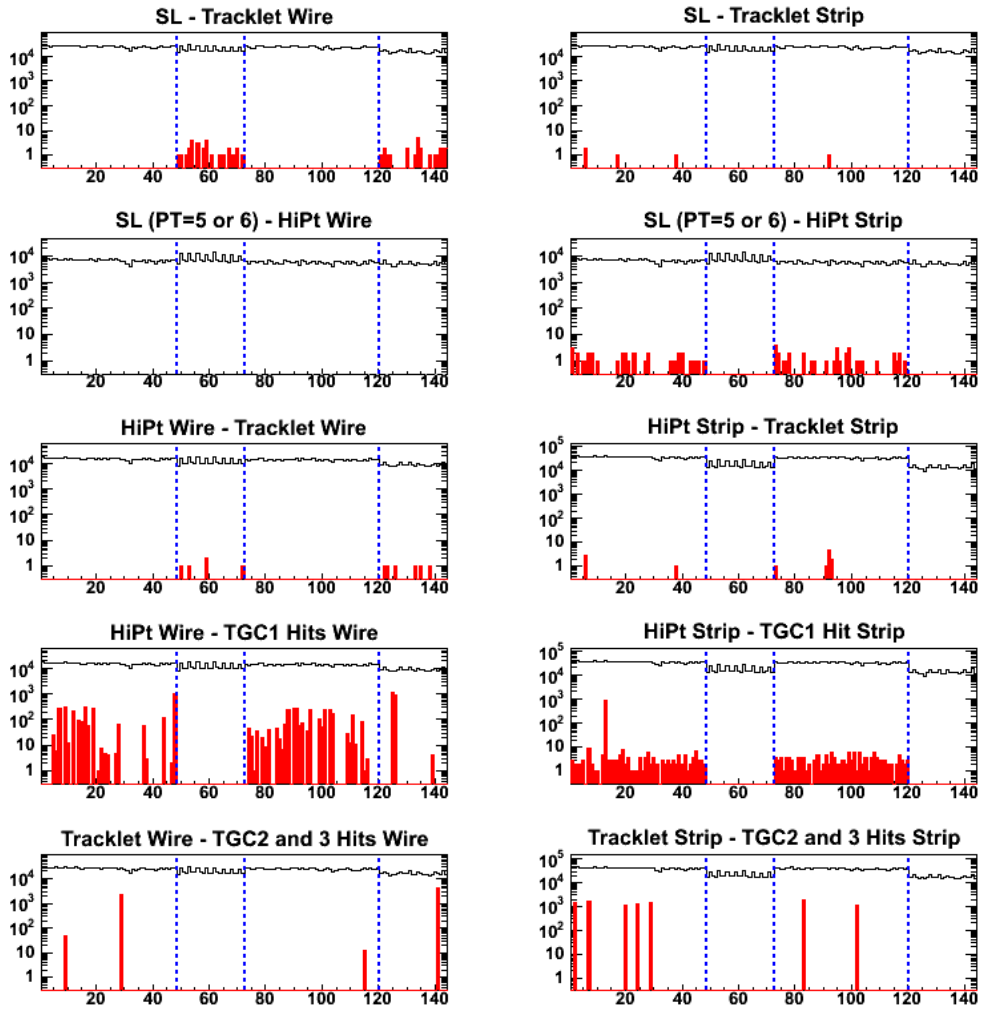


図 4.8: TrigMuonCoinHierarchy を用いた各階層のトリガーセクター毎の対応状況 (run167844、mask2 を考慮した場合)

## 第5章 実データを用いたTGCシステムの動作検証

第1章で述べた通り、TGCシステムの評価は実験において大変重要な作業である。またそれはツール化されていることが望ましい。今回、私はそのツールとして、これまでトリガーのエミュレーションに用いられてきたTrigT1TGCという標準ツールを選び、これを実験の実データに応用できないかと考えた。

TrigT1TGCはシミュレーションにより作られたヒット情報をもとに、TGCシステムのトリガー情報のエミュレーションを行うツールである。私はこのツールに実験で得られた実データを入力し、そこから得られるトリガー情報が、実験中に得られるトリガー情報とどの程度一致しているかを調べた。これによりTGCシステムで用いられているハードウェア、デコーディングソフトウェアが、想定通りに機能しているか否かを評価できると考えたためである。

本章では、本論文の主題である2010年のビームデータを用いたTrigT1TGCツールを用いたTGCシステムの評価についてまとめる。

### 5.1 TrigT1TGCを用いた動作検証の手法

第4章で紹介したTrigMuonCoinHierarchyは上位の階層にオブジェクトが存在したときに、下位の階層に対応するオブジェクトが存在するか否かという考え方をもって、TGCエレクトロニクスの動作の検証ができた。すなわちTGCシステムの動作検証ができた。ただしこれはハードウェアの観点に置き換えると、下流のハードに情報が存在したときに、その由来となった情報が上流のハードに存在するか否かという、トリガー系のハードウェアの動作を図5.1のように逆方向にチェックしていることになる。

今回確立されたTrigT1TGCを用いた動作検証の手法は、ヒットの情報が存在したときに、そのヒット情報から作られたトリガー情報が正しく存在しているか否かを調べる、いわばTrigMuonCoinHierarchyとは逆で、トリガー系のエレクトロニクスの流れを図5.1のように順方向に辿る検証である。TGCシステムを順方向からと逆方向からの両方から検証できるようになると、TGCシステムに不具合が見られたときのクロスチェックは勿論のこと、場合に応じた使い分けができ非常に便利になる。

本節では、始めにTrigT1TGCパッケージについて説明した後、TrigT1TGCを用いたTGCシステムの動作検証についてまとめる。

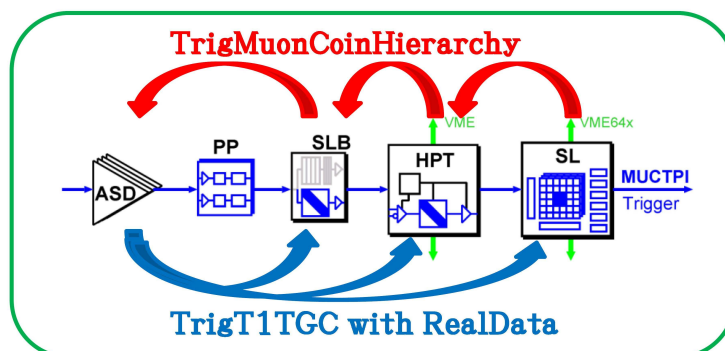


図 5.1: TrigMuonCoinHierarchy と TrigT1TGC を用いた TGC トリガーシステムの検証のイメージ図

### 5.1.1 Trigger/TrigT1/TrigT1TGC の概要

TrigT1TGC パッケージは TGC システムのトリガー系のハードウェアの動作をエミュレートするパッケージである。本来はシミュレーション解析の場で使われるツールで、インプットとして TGC Digit もしくは RDO でヒット情報を与えると、アウトプットとして RoI や  $p_T$  threshold などのトリガー情報を得ることができる。イメージ図を図 5.2 に示す。図 5.2 の中の HitRDO と CoinRDO は 4.3.5 でまとめたものである。ここで CoinRDO

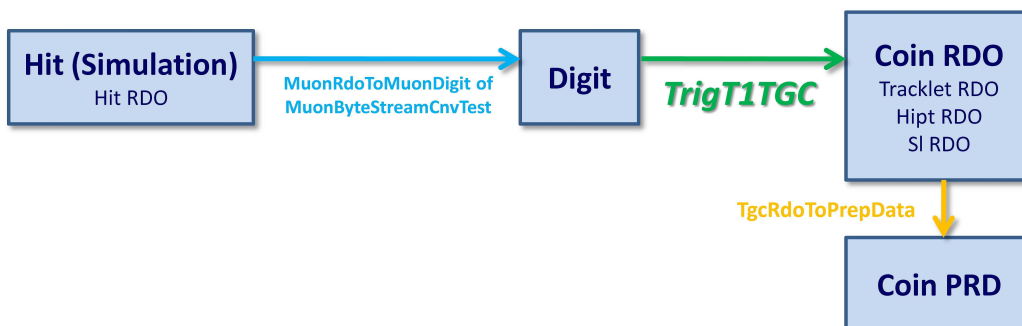


図 5.2: TrigT1TGC を MC で走らせるときのイメージ図

というのは Coincidence 情報を含んだ RDO という意味で、trackletRDO、highptRDO、slRDO の総称として扱う。CoinPRD は 4.3.6 で述べた TrigCoinData パッケージから作られる CoinRDO に対応した各種 PRD の総称である。

通常、TrigT1TGC パッケージは TgcDigitization から作られた Digit から Sector Logic のデータを作りだし、そこからコインシデンスウインドウの作成やハードウェアの設計の検証、テストパルスなどでの実機の評価、トラッキングのパフォーマンスの評価に用いられる。

次に TrigT1TGC のクラス階層構造について簡単に説明する。クラス階層の様子を図 5.3

に示す。

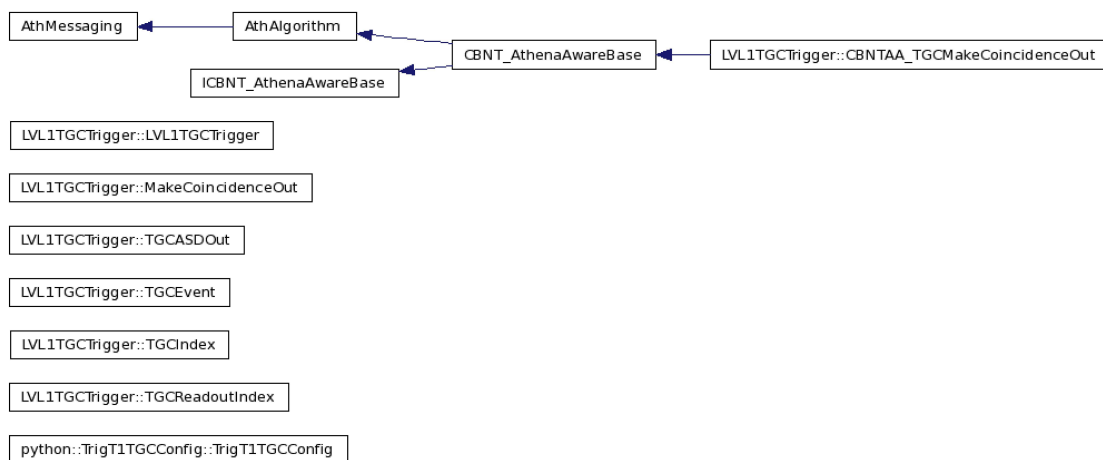


図 5.3: Trigger/TrigT1/TrigT1TGC のクラス階層構造 [12]

主な 3 つのクラスを以下に簡単にまとめる。

### LVL1TGCTrigger クラス

TrigT1TGC においてツールの核となるメインアルゴリズムが LVL1TGCTrigger クラスである。TGC Digit のコンテナを入力させると、SL からのトリガー判定までのエミュレートを行うことができる。クラス内の doMaskOperation 関数を用いることによつて、トリガー判定を行う際に mask in/mask out のチャンネルの情報を考慮に入れるか否かを指定することができる。当時、doMaskOperation 関数に関しては mask チャンネルが正しく TrigT1TGC に反映されないバグがあった。これに関しては TrigT1TGC-00-02-44 以降で改善された。また mask のチャンネルの影響を組み込む際には、4.4.2 で述べた mask2 のファイルと同じ形式のものを用いる。TrigT1TGC が mask in 及び mask out を読み込む際には Online ID が用いられるため、mask2 のテキストファイルを TrigT1TGC に加える際には Online ID に変換させて使用する。また TrigT1TGC の中で使う SL のコインシデンスウインドウのタイプはこのクラス内で指定することができる。TrigT1TGC-00-02-46 からは 2011 年のデータテイキング向けの新しいコインシデンスウインドウである setH がデフォルトとして指定されている。

### CBNTAA\_TGCMakeCoincidenceOut クラス

4.2.4 で述べた AANtuple(AANT) を作成するアルゴリズムである。AANT はヒット情報を含むもの、コインシデンス情報を含むものの 2 種類を作成することが可能で、AANT はデコーディング時のデバッグに有用である。

## TGCMakeCoincidenceOut クラス

Column Wise Ntuple(CWN)を作成するクラスで、CBNTAATGCMakeCoincidenceOut クラスとほぼ同等の枠組みである。

### 5.1.2 実データによる TrigT1TGC の動作

ここまで説明してきたように TrigT1TGC パッケージはモンテカルロ (MC) のヒット情報からトリガー情報をエミュレートするものとして用いられてきたが、以前から、実データで得たヒット情報を入力ファイルとして与え、そこからトリガー情報を得ることがはできないかという議論があった。そのイメージ図を図 5.4 に示す。図 5.4 の通り、実データ

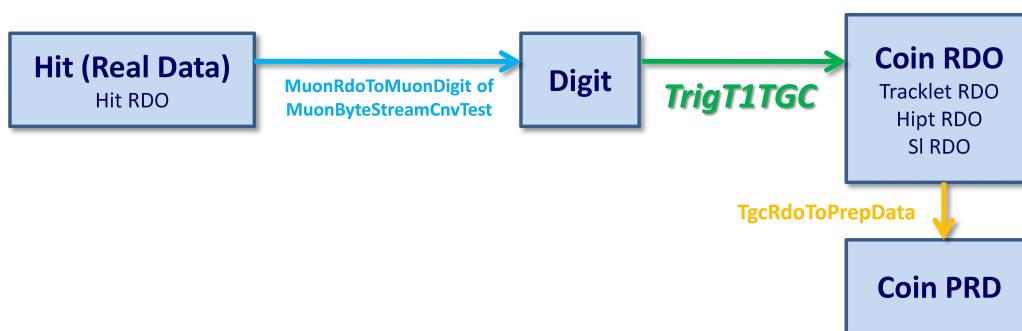


図 5.4: TrigT1TGC を実データで走らせるときのイメージ図

をデコードすることで得られた HitRDO を、あたかも MC から取得したヒット情報だと偽り TGC Digit に変換し、TrigT1TGC に入力するのである。出発は実データを用いた解析であるが、TrigT1TGC を通すことで途中からはシミュレーション解析となる。当然、トリガー情報をエミュレートする TrigT1TGC からは、ユーザーが想定しているトリガー情報、すなわちそれは実機から得られるトリガー情報と”同じもの”が生成されるはずである。しかし、もしも TrigT1TGC から生成されたトリガー情報が、実機のトリガー情報と”異なる”ものであった場合、それは実機の想定外の挙動、もしくは TrigT1TGC 自身のバグや実機で使われているデコーディングソフトウェアのバグなどの可能性が浮上することになる。この考え方を用いれば、この一連の流れは TGC LVL1 ミューオントリガーシステム (TGC システム) の現状を評価する手段となる。具体的には、図 5.5 の流れでの検証となる。

図 5.5 中の Standard Decoding というのは実験中に処理される実機のデコーディングを示している。CoinRDO with TrigT1TGC と CoinPRD with TrigT1TGC は TrigT1TGC によりヒット情報からトリガー情報をエミュレートして作成された Coincidence RDO と Coincidence PRD である。また CoinRDO with StDec と CoinPRD with StDec は Standard Decoding で処理された Coincidence RDO と Coincidence PRD であり、これは実験中に記録されるデータと同じである。4.3.5 と 4.3.6 で述べたように、RDO と PRD のレベルまでデータをデコーディングすれば、中の変数も参照しやすいため、CoinRDO with

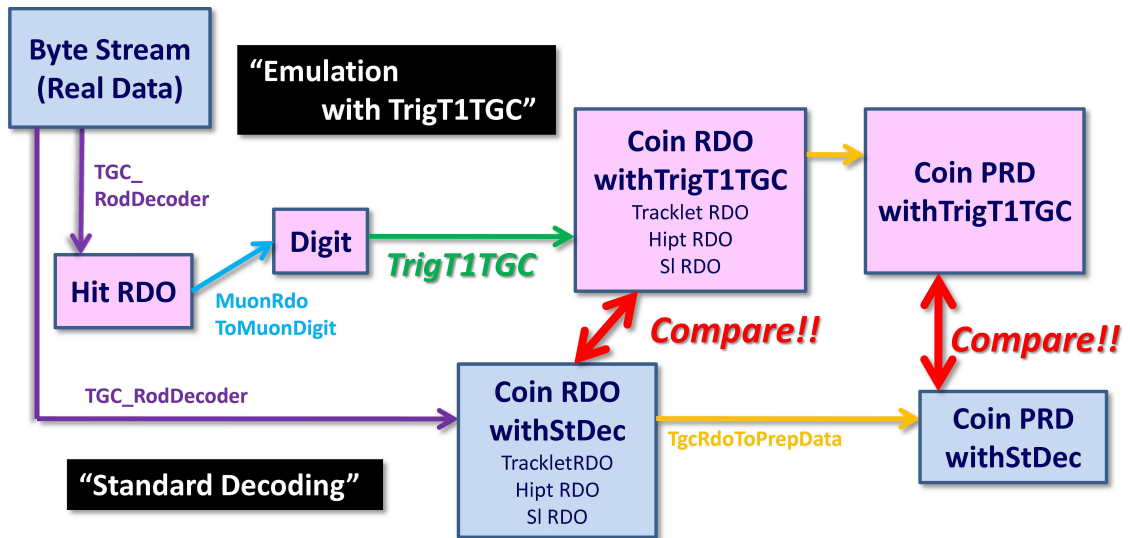


図 5.5: TrigT1TGC を用いた TGC システムの動作検証の流れ

TrigT1TGC と CoinRDO with StDec、CoinPRD with TrigT1TGC と CoinPRD with StDec を各々比較することで、実機とエミュレーションのトリガーシステムの一致度を確認することができる。十分に一致していれば、現状の TGC システムにハードウェアの故障などの想定外の事態は生じていないことがわかる。

そして、この手法の長所は、比較対象に RDO を用いることができる点である。Trig-MuonCoinHierarchy を用いた TGC システムの性能評価では、PRD(ESD) レベルでの比較であったため、対応関係がとれないときに、どのような箇所で対応がとれていないのか、どのハードウェアで不具合が生じているのか等を知るため、一度 OfflineID を取得して Cabling Service を用いてエレクトロニクスのレベルまでデータを変換しなければいけない。勿論、その変換の間にバグが入ってしまうことは十分に考えられる。しかし今回の手法では PRD よりも、より Byte Stream のデータの状態を保持しエレクトロニクスとの対応関係がとりやすい RDO での比較が可能のため、仮にエミュレーションと実機のトリガー情報が一致しない場合に、それがハードウェアの問題であれば、それがどのハードウェアであるのか等の認識が TrigMuonCoinHierarchy の場合に比べて安易に行える。ただし RDO は 4.3.5 でも述べた通り、TGC 検出器及びハードウェアに精通していなければ理解し難いという短所もある。

また TrigT1TGC に実データを入力できるようになれば、それは TGC システムの動作検証以外にも以下のような点で役立つことも期待される。

- TrigT1TGC パッケージのデバッグ  
 実機とエミュレーションのトリガー情報が一致しない時、その不一致が特定の箇所に集中していないなどの非構造的な場合はエミュレーションに使っている TrigT1TGC のバグの可能性が浮上する。

- コミット前の新しいコインシデンスウインドウの性能評価

実データを用いることで、新しいコインシデンスウインドウの検出効率等を、よりリアルにシミュレートすることができる。

以上のように、TrigT1TGC に実データのヒット情報をインプット情報として用いることは TGC システムの性能評価だけにとどまらず、その有用性に多くの可能性のある手法となる。

### 5.1.3 TrigT1TGC に実データのヒット情報を入力する方法

以下に実際にどのように実データを TrigT1TGC に入力するか、その手法を説明する。

そもそも TrigT1TGC パッケージは MC サンプルを処理するためのツールであり、実データを処理するためのツールではない。ここで生じる問題が、ATLAS では MC データと実データのデータフォーマットが異なるということである。すなわち、2つのデータは使うときに異なるデータベースを用いて解析を行わなければいけない。TrigT1TGC が開発された当初は、実データを入力ファイルとして使う考えはなかったため、現在でも実データを TrigT1TGC に入力しようとする、データフォーマットの問題で上手く処理されない。そのため TrigT1TGC に実データを読み込ませる際には、用いるデータベースを実データ用のもので処理させなければいけない。他にも TrigT1TGC を騙しながら走らせるために、TrigT1TGC が走るときに読み込まれる様々なパッケージの一部に変更を加えることにより初めて実データを用いた解析が可能となる。

もう1つの課題が入力とするヒット情報の RDO をいかにして入手するかである。4.1 で述べた通り、ユーザーが解析で利用できるデータ形式は、RAW、ESD、AOD の3種類である。図 4.1 に示したように、これら3種類のデータ形式は下流に行くほどその情報数が減少していく。そして RDO は RAW と ESD の中間の位置に存在するデータ形式である。そのためユーザーが特定の run の ESD データを入手したとしても、それを RDO に戻すことは理論上不可能なことである。HitRDO を入手するためには、それよりも上流のデータフォーマットを用いなければいけないため、RAW、すなわち Byte Stream を用いることになる。Byte Stream にはハードから出力されたヒット情報が全て詰まっているため、これを HitRDO に変換し入力ファイルとして使うことにした。ただし RAW からトリガー情報を出力させるためには、リコンストラクション処理を始めから行わなければいけないため、大変な労力と時間がかかる。

### 5.1.4 RDO with TrigT1TGC と RDO with StDec の入手方法

以下に、HitRDO に対してどのようなデコーディング作業を施して、CoinRDO with TrigT1TGC と CoinRDO with StDec を入手するかについての流れを説明する。

#### CoinRDO with TrigT1TGC の入手方法

1. RAW(Byte Stream) には TGC のコインシデンスも記録されているため、始めにこれを除く処理を施す。

今の場合 TGC のコインシデンスは後から TrigT1TGC によって付加されるべきであるためである。具体的には MuonSpectrometer/MuonCnv/MuonTGC\_CnvTools-00-07-03/src/TgcRODReadOut.cxx 内にその処理を施す。

2. 4.3.4 で述べた MuonDataRec\_myTopOptions.py を用いてリコンストラクションをすると HitRDO が生成される。  
1 で TGC のコインシデンス情報を除外しているため、CoinRDO は生成されない。

3. 2 で得られた HitRDO を入力として TrigT1TGC を走らせる。  
ここでは Reco\_trf.py というリコンストラクション用ツールを用いることで自動的に TrigT1TGC が走る仕組みになっていることを利用する。また TrigT1TGC を走らせる際には RAW がデータテイキングされた当時のコインシデンスウインドウを LVL1TGCTrigger クラスの中で指定してやる必要がある。また Reco\_trf.py の性質上、CoinRDO はパーシステントなオブジェクトとして保存せず、CoinESD のみが出力される。そのため、MuonSpectrometer/MuonReconstruction/MuonRecExample の

share/MuonRecOutputItemList\_jobOptions.py

内に変更を施して、CoinESD が CoinRDO のオブジェクトを保持したまま出力されるように設定する。

4. 3 の後、CoinRDO のオブジェクトを含んだ CoinESD が取得できる。
5. CoinRDO(HitRDO も中に含まれている) を解析のしやすいフォーマットに変換する。  
これは RDO を含む ESD が 4.1.3 で述べた通り POOL ROOT file という解析には向かない複雑なフォーマットをとるためである。この変換には自作のツールを用いて、HitRDO と CoinRDO 情報のみが含まれた Ntuple を取得することにした。

以上の方法で、HitRDO with TrigT1TGC と CoinRDO with TrigT1TGC の情報が含まれたデータを準備することができる。

### CoinRDO with StDec の入手方法

1. RAW(Byte Stream) に対して TrigT1TGC を用いるときに行った TGC のコインシデンス情報を除外する作業を行わない。  
これは RAW に含まれる TGC のコインシデンス情報、すなわち実機が行ったコインシデンス処理の情報を保持するためである。
2. 4.3.4 で述べた MuonDataRec\_myTopOptions.py を用いてリコンストラクションをすると CoinESD が取得できる。  
この時 TrigT1TGC の時と同じように、CoinESD の中に CoinRDO のオブジェクトが残るように処理をしておく。



3. CoinRDO(HitRDO も中に含まれている) を解析のしやすいフォーマットに変換する。

用いるツールは TrigT1TGC の場合の処理で用いたものと同じである。

以上の方法で、HitRDO with StDec と CoinRDO with StDec の情報が含まれたデータを準備することができる。

## 5.2 TGC システムの性能評価

図 5.5 で述べた通り、前節で得られた RDO with TrigT1TGC と RDO with StDec をそれぞれ比較すれば、実機とエミュレーションがどの程度一致しているかを検証することができる。4.3.5 の通り、RDO には hitRDO、trackletRDO、highptRDO、slRDO があり、それぞれの RDO は更に、

- hitRDO  
DoubletWire、DoubletStrip、TripletWire、TripletStrip の情報を含む。
- trackletRDO  
DoubletWire、DoubletStrip、TripletWire、TripletStrip の情報を含む。
- highptRDO  
Wire、Strip の情報を含む。
- slRDO  
r- $\phi$  コインシデンスがとられるため slRDO は一種類のみ。

の計 11 種類に区別することができる。

今回、私は、この 11 種類の RDO に対して、withTrigT1TGC の RDO、withStDec の RDO を相互に比較することで、TGC システムの実機とエミュレーションの一致度を割り出し、TGC システムの動作検証を行った。また、両者が一致しないときに、その原因を追究することで、冬季シャットダウンを終え 2011 年 2 月から実験を再開する ATLAS 検出器の TGC エレクトロニクスの交換、使用されている TrigT1TGC そのもののデバッグの必要性の有無について考察した。

以下では 11 種類の RDO について、ヒット情報、トリガー情報の一致性を調査し、実機とエミュレーションの両方で完全に一致する RDO の数、TrigT1TGC のみに含まれる RDO の数、実機のみに含まれる RDO の数の 3 項目の調査結果を起点として TGC システムの性能評価を行った結果をまとめる。順に 11 種類の RDO の対応結果について述べ、考察する。

### 5.2.1 データサンプル

RAW(Byte Stream) 形式のデータの中で入手可能な一番最新のデータを用いた。用いたのは RunNumber159179 のデータである。これは 2010/07/16,17:40:43~07/17,04:50:37 にデータテイキングされたデータである。計 1,486,581 イベント含まれる。ここで”入手可

能な一番最新の”とあるのは、RAW 形式のデータはグリッド上にコピーされるものではなく、一般の研究者が自由に取得できないためである。

この RAW から前述の方法でもって、実機由来のコインシデンス情報、TrigT1TGC 由来 (エミュレーションで得られる) のコインシデンス情報を取り出し、そこで得られた Ntuple を用いて解析を行った。

また各 RDO 共に、どのバンチ由来のものかを示す bcTag は 2 のみ、すなわち Current のみを用いた。

### 5.2.2 解析環境

用いた Athena のリリースは 16.0.2 である。

TrigT1TGC のバージョンは TrigT1TGC-00-02-44 を使用した。そしてこの TrigT1TGC には、run159179 当時に適用されていた mask2 ファイル (mask2\_channels\_20100705.txt) の mask in 及び mask out の影響を組み込んだ。また当時使われていたコインシデンスウインドウは setD で、今回の解析でもそのコインシデンスウインドウを用いている。そしてシミュレーションで用いる ATLAS のジオメトリバージョンは ATLAS-GEO-16-00-00 である。

以下、各 RDO の解析結果をまとめる。

各 RDO 内で出てくる変数名は、4.3.5 及び、B.2 を参照されたい。また、TrigT1TGC のみ、または実機 (StDec) のみに含まれる RDO が存在するとき、その数を用いて非対応率を次のように定義する。

$$\text{非対応率} = \frac{\text{TrigT1TGC(StDec) のみに含まれる RDO の数}}{\text{共通に含まれる RDO の数} + \text{TrigT1TGC(StDec) のみに含まれる RDO の数}} \quad (5.1)$$

### 5.2.3 hitRDO(DoubletWire)

RDO 内の変数である subDetectorId、rodId、sswId、slbId、channel が全て一致しているかどうかを確認した。DoubletWire の hitRDO の対応関係は表 5.1 の通りである。

	個数	非対応率
共通に含まれる RDO	5,468,876	-
TrigT1TGC のみに含まれる RDO	660	$1.2 \times 10^{-2}$
StDec のみに含まれる RDO	0	0.00

表 5.1: hitRDO(DoubletWire) の対応関係

ここで TrigT1TGC のみにだけ見られる RDO があるが、これは TrigT1TGC の入力となったヒット情報と StDec の入力となったヒット情報のフォーマットの違いにより生じたものである。TrigT1TGC の入力となったのは RawdataFormat と呼ばれるフォーマットで、SSW からリードアウトされた情報がそのまま用いられている。一方、StDec の入力と

なったヒット情報のフォーマットは ReadoutFormat と呼ばれるフォーマットで ROD 内の Data Converter を通過することで、不必要な情報が除かれたものである。本来、この2つは一致すべきであるが、run159179 では、eventNumber が 2045473、11774779、13207549 の3つのイベントにおいて RawdataFormat と ReadoutFormat に違いがある。2.4.1 で述べたように ROD のデータは ROB(Read Out Buffer) を複数持つ ROS(Read Out System) へと送られるが、ハードウェアの性質により ROB が送信される間にタイムアウトが起き、全ての ROB の情報が ROS に送信されなかったことが原因である。これはハードウェアの性質であって、不具合ではないため TGC システムには問題はない。そこで、上記の3イベントを除いて、再び対応関係を見ると表 5.2 となり、両者の hitRDO が完全に一致していることがわかる。

	個数	非対応率
共通に含まれる RDO	5,466,662	-
TrigT1TGC のみに含まれる RDO	0	0.00
StDec のみに含まれる RDO	0	0.00

表 5.2: hitRDO(DoubletWire) の対応関係 (データフォーマット間で情報が一致しないイベントを除外した場合)

全く同じ現象が DoubletStrip においても見られた。このことは即ち、TrigT1TGC と StDec がそれぞれ RawdataFormat と ReadoutFormat という異なるフォーマットをとることを逆に利用して、ここで対応がとれない RDO が存在すると、ReadoutFormat を作成するときの特性によるヒット情報の差異を検証することができるといえる。

以下で述べる残りの 10 パターンの対応関係の検証においては、上記の3イベントは除外して解析を行うことにした。

#### 5.2.4 hitRDO(DoubletStrip)

RDO 内の変数である subDetectorId、rodId、sswId、slbId、channel が全て一致しているかどうかを確認した。DoubletStrip の hitRDO の対応関係は表 5.3 の通りである。

	個数	非対応率
共通に含まれる RDO	8,004,993	-
TrigT1TGC のみに含まれる RDO	0	0.00
StDec のみに含まれる RDO	0	0.00

表 5.3: hitRDO(DoubletStrip) の対応関係

完全に一致している。

### 5.2.5 hitRDO(TripletWire)

RDO 内の変数である subDetectorId、rodId、sswId、slbId、channel が全て一致しているかどうかを確認した。TripletWire の hitRDO の対応関係は表 5.4 の通りである。

	個数	非対応率
共通に含まれる RDO	3,031,770	-
TrigT1TGC のみに含まれる RDO	0	0.00
StDec のみに含まれる RDO	0	0.00

表 5.4: hitRDO(TripletWire) の対応関係

完全に一致している。

### 5.2.6 hitRDO(TripletStrip)

RDO 内の変数である subDetectorId、rodId、sswId、slbId、channel が全て一致しているかどうかを確認した。TripletStrip の hitRDO の対応関係は表 5.5 の通りである。

	個数	非対応率
共通に含まれる RDO	2,373,985	-
TrigT1TGC のみに含まれる RDO	0	0.00
StDec のみに含まれる RDO	0	0.00

表 5.5: hitRDO(TripletStrip) の対応関係

完全に一致している。

### 5.2.7 trackletRDO(DoubletWire)

RDO 内の変数である subDetectorId、rodId、sswId、slbId、delta、sub、rphi が全て一致しているかどうかを確認した。DoubletWire の trackletRDO の対応関係は表 5.6 の通りである。

	個数	非対応率
共通に含まれる RDO	650,993	-
TrigT1TGC のみに含まれる RDO	31	$4.8 \times 10^{-5}$
StDec のみに含まれる RDO	68	$1.0 \times 10^{-4}$

表 5.6: trackletRDO(DoubletWire) の対応関係

## 5.2. TGC システムの性能評価

対応がとれない RDO も存在するが、数量は僅かである。実機の方が対応がとれない RDO の数が多いのは、当時実機に何らかの不具合があった等が原因でトリガー信号を余分に出してしまっていたと考えられる。また対応がとれない時の原因となっている変数は多くの場合が delta 値であった (図 5.6)。ただし、その delta 値は若干の構造が見られるた

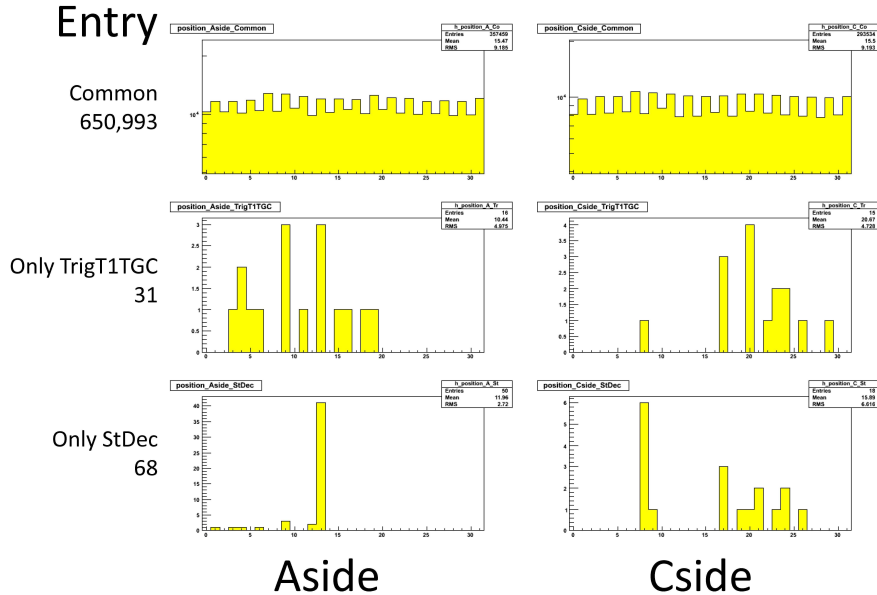


図 5.6: TrackletDoubletWire における delta 値の対応関係

め、ハードウェアの不具合が原因と考えられる。ただ表 5.6 の通り対応は十分にとれているため、無視できるレベルであると思われる。

### 5.2.8 trackletRDO(DoubletStrip)

RDO 内の変数である subDetectorId、rodId、sswId、slbId、sub、rphi が全て一致しているかどうかを確認した。DoubletStrip の trackletRDO の対応関係は表 5.7 の通りである。

	個数	非対応率
共通に含まれる RDO	739,693	-
TrigT1TGC のみに含まれる RDO	22	$3.0 \times 10^{-5}$
StDec のみに含まれる RDO	19	$2.6 \times 10^{-5}$

表 5.7: trackletRDO(DoubletStrip) の対応関係

DoubletWire の時と同様、対応がとれていない RDO が存在するがその数は僅かである。ただし DoubletWire の時と違って実機から余分なトリガーが出ていないことが伺える。対応がとれていない RDO は DoubletWire の時と同様、delta 値であった (図 5.7)。

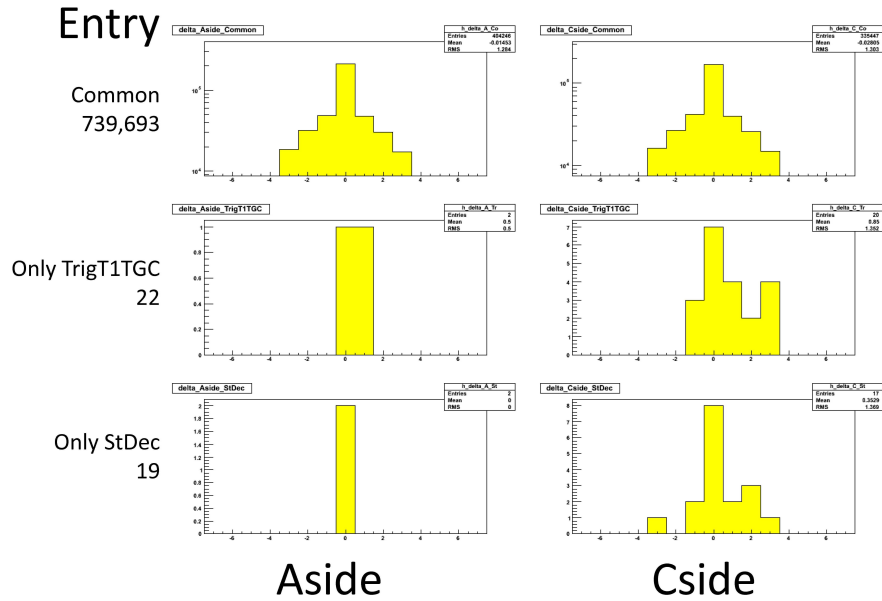


図 5.7: TrackletDoubletStrip における delta 値の対応関係

### 5.2.9 trackletRDO(TripletWire)

RDO 内の変数である subDetectorId、rodId、sswId、slbId、delta、sub、rphi が全て一致しているかどうかを確認した。TripletWire の trackletRDO の対応関係は表 5.8 の通りである。

	個数
共通に含まれる RDO	2,213,426
TrigT1TGC のみに含まれる RDO	852
StDec のみに含まれる RDO	730

表 5.8: trackletRDO(TripletWire) の対応関係

一見、十分に対応がとれているように見えるが、共通に含まれる RDO が非常に多いことが分かる。共通に含まれているということは常にヒット情報において bit が high になっているということであり、ハードウェアの不具合の可能性が非常に高い。実際に rphi 値 (slbsubMatrix 内の位置を trgch 単位で表したものを) をヒストグラム (図 5.8) にすると、特定のチャンネルに集中していることが伺える。このような常にトリガーが出ているようなチャンネルがあればノイジーチャンネルとして、その状態をエキスパート確認、対処してもらう必要がある。今回の場合は、このチャンネルは TGC 上では、

Aside, layer:2, Endcap, sector:01, phi:1, SLBchannel:196

で layer2 の一番外側のチャンネルである。このチャンネルは全イベントでトリガーを出し

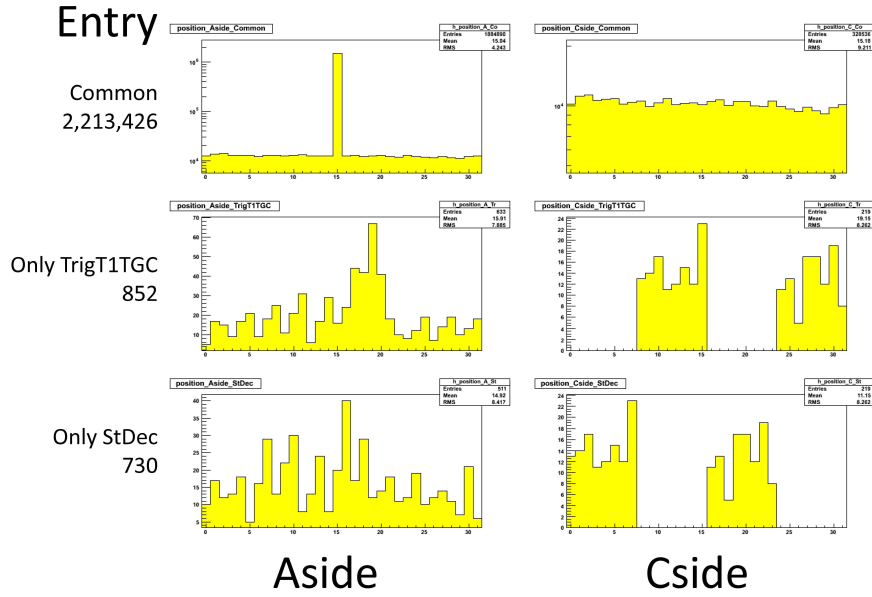


図 5.8: TrackletTripletWire における rphi 値の対応関係

ていた。ただし今回の原因はハードウェアの故障ではなく、ノイジーチャンネルを埋めていたことが原因であったことが判明した。この辺りのチャンネルは3層連続でデッドチャンネルになっているため、 $p_T = 0$  で3station コインシデンスが出せるように、tracklet が出力され続けるように設定されていたためであった。結果的にハードウェアの故障ではなかったが、実際の TGC のトリガーの現状を TrigT1TGC を用いて再現できることが確認できた。

ノイジーチャンネルの部分を除いた場合、対応関係は、表 5.9 の通りである。

	個数	非対応率
共通に含まれる RDO	726,845	-
TrigT1TGC のみに含まれる RDO	852	$1.2 \times 10^{-3}$
StDec のみに含まれる RDO	730	$1.0 \times 10^{-3}$

表 5.9: trackletRDO(TripletWire) の対応関係 (ノイジーチャンネルを除外した場合)

非対応率がこれまでの RDO に比べて高いことが伺える。調査の結果、対応がとれないときのトリガーの情報は、

- Aside, Endcap, sector:10, phi(triggerSector):2, chamber:T7, TRIG1
- Aside, Endcap, sector:10, phi(triggerSector):2, chamber:T7, TRIG2
- Aside, Endcap, sector:10, phi(triggerSector):2, chamber:T6, TRIG0

に集中し、これは実機の場合、対応がとれない 730 の RDO のうち 75% の 511 の RDO に相当することが判明した。

原因はケーブルリングスワップであった。

Aside, Endcap, sector:10, phi(triggerSector):2, chamber:T7

は Triplet が本来内側から layer1→layer2→layer3 と並ばなければいけないところを、ケーブルリングスワップが原因で、実質 layer2→layer0→layer1 の順番で並んでいる。9月以降、ケーブルリングスワップを考慮に入れた Cabling Service がアップデートされた。しかし TrigT1TGC ではこのケーブルリングスワップが考慮に入れられていない。TrigT1TGC のインプットファイルである TGC Digit を作成するときには ReadoutID で書かれた hitRDO を一度、OfflineID で処理される TGC Digit に変換される。この変換の際には Cabling Service が用いられるため、TGC の現状を正しく反映した TGC Digit が得られる。しかし、TrigT1TGC は ReadoutID がベースとなる CoinRDO を出力するときに、Cabling Service を使わずに TrigT1TGC 内でその変換を行うためにケーブルリングスワップが考慮されずに変換されてしまっている。そのため Cabling Service を用いてデコーディングが成された実機の CoinRDO とは対応関係がとれなくなっているのである。また T6 は一部 T7 と重なっている領域があるため、その部分が寄与しているのだと考えられる。

これは TrigT1TGC のバグであり、上記の3つからのトリガー情報と前述のノイズチャンネルを除外して対応関係を見ると、表 5.10 のようになる。

	個数	非対応率
共通に含まれる RDO	726,845	-
TrigT1TGC のみに含まれる RDO	219	$3.0 \times 10^{-4}$
StDec のみに含まれる RDO	219	$3.0 \times 10^{-4}$

表 5.10: trackletRDO(TripletWire) の対応関係 (ノイズチャンネルと疑わしい SLB を除外した場合)

表 5.10 の通り、ノイズチャンネルが入っている SLB、前述の疑わしい SLB 由来の tracklet を除くと、完全ではないものの十分に対応がとれていて、TrackletTripletWire に関しては十分な性能を維持していると考えられる。

また、このとき対応がとれないのは全て Cside で、以下のところに集中している。

Cside, Endcap, sector:06, phi(triggerSector):2, chamber:T3, TRIG1

これを TGC 上での位置に直すと図 5.9 の桃色のハイカラーにした部分である。図 5.9 は紙面を時計回りに 90 度回転させて参照されたい。右の 3 層は左から layer1、layer2、layer3 と並んでいて、左の 3 層は T6 チェンバー、右の 3 層は T3 チェンバーを表している。下から上に r の増加方向のチャンネルの並びとなっている。

この時、対応がとれない原因となっているのはトリガーの rphi 値 (slbsubMatrix 内の位置 (trgch 単位)) の差であることが判明した。この様子は図 5.8 の右の Cside の図の通りである。TrigT1TGC から得られるトリガー情報と実機のトリガー情報の rphi 値がずれていることが伺える。run159179 当時の mask2 を参照しても、C06 に関するチャンネルは mask in/mask out されていないので、mask2 が原因ではないと思われる。また、冬季シャットダウンの間に ASD テストパルスを用いて、該当する SLB の調査を行ったところ、



## 5.2. TGC システムの性能評価

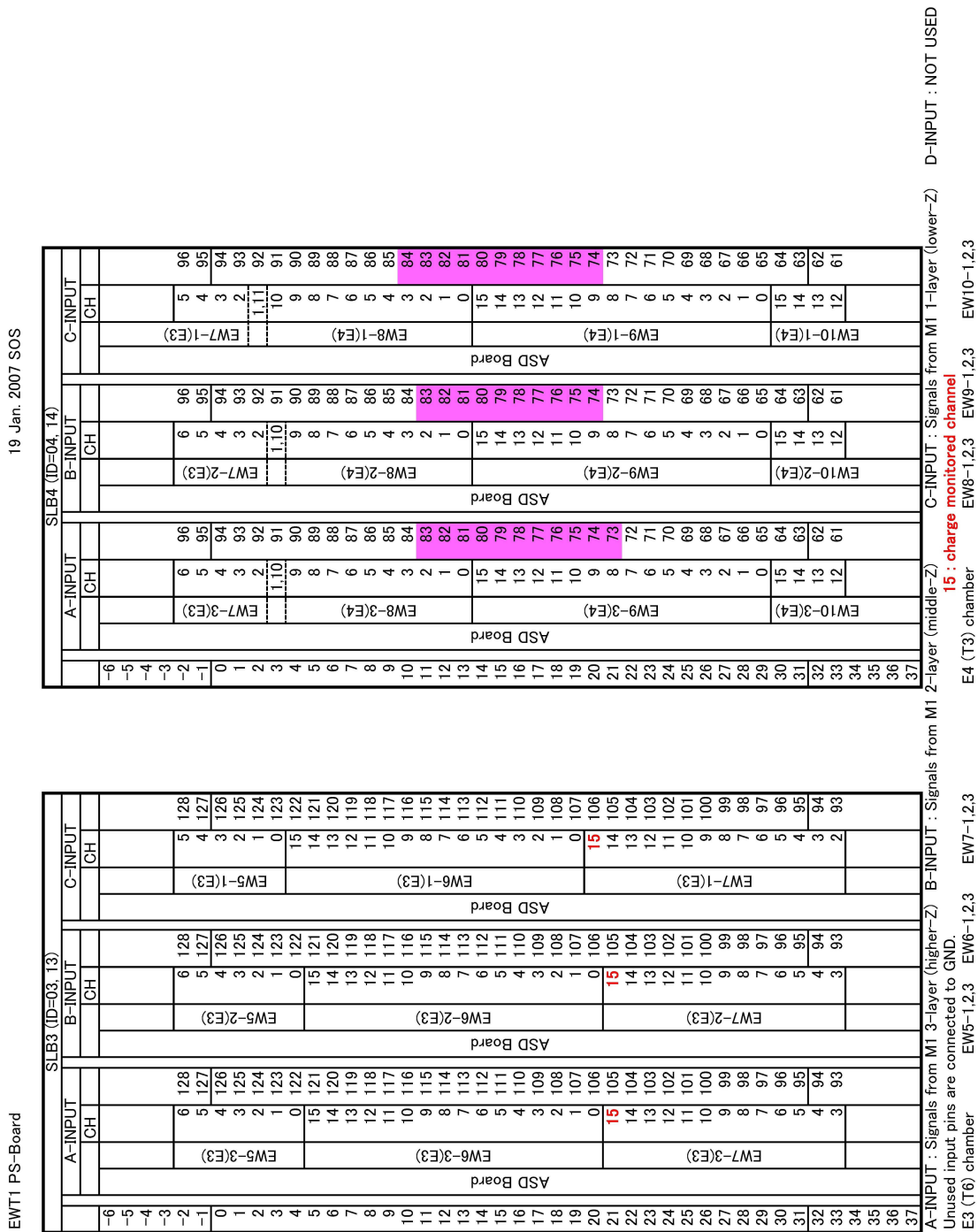


図 5.9: TrackletTripletWire で対応がとれない Cside sector=6 の SLB の TGC 上での位置

他のセクターの対称となる SLB と同じトリガー情報が出力されることが確認された。そのため現状の該当箇所の SLB は正常に動作しているといえる。ただし、run159179 当時は、ASD テストパルスと同じヒットパターンの際に、シミュレーションの phi 値と実機からの phi 値の両方が、正常に実機が動作した場合と異なる値を出力していることも判明した。シミュレーションに関してはケーブリングスワップの可能性も示唆されるが、実機に関しては現在も調査中である。

### 5.2.10 trackletRDO(TripletStrip)

RDO 内の変数である subDetectorId、rodId、sswId、slbId、delta、sub、rphi、seg が全て一致しているかどうかを確認した。TripletStrip の trackletRDO の対応関係は表 5.11 の通りである。

	個数
共通に含まれる RDO	2,788
TrigT1TGC のみに含まれる RDO	969,833
StDec のみに含まれる RDO	969,517

表 5.11: trackletRDO(TripletStrip) の対応関係

対応がとれないときの RDO には特徴がある。図 5.10 に示すように、segment の値が反転していることが伺える。segment は隣り合う 2 つの TripletStrip のチェンバーのどちらか

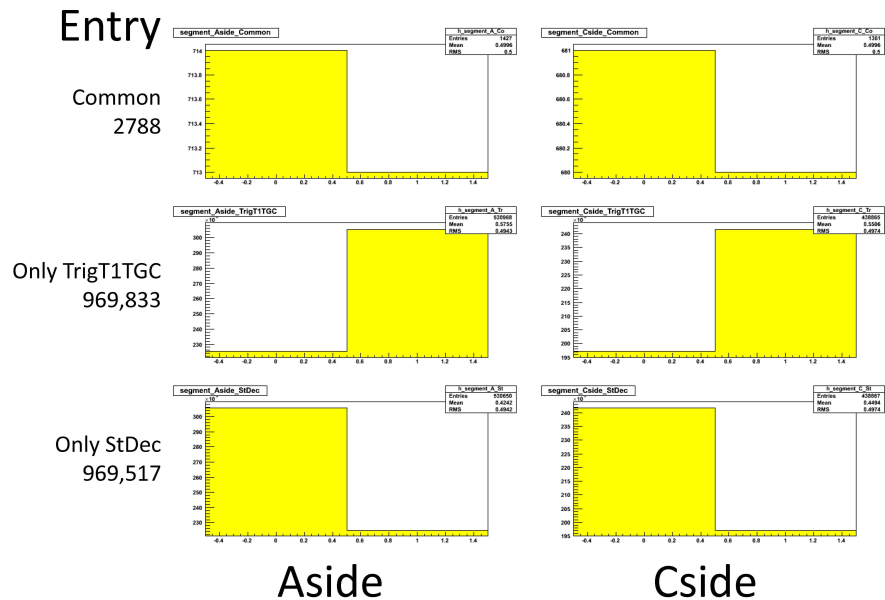


図 5.10: TrackletTripletStrip における seg 値の対応関係

ら読み出されているかを司る変数である。これはハードウェアのバグではなく TrigT1TGC

のバグで、segment 内の TRIGA、TRIGB を反転させるコードが用いられていたことが原因だった。TrackletTripletStrip の seg は PRD(ESD) を作成するときに用いられていないため、これまで判明しなかった。逆に言えば、使われていないため highptStrip や SL でその影響が見られることがないため、これまでの SL 上のコインシデンスウインドウを作成する際等に影響は及ぼしていない。対応がとれているのは、偶然にも 1 つのイベント中に隣り合う 2 つの TripletStrip のチェンバーの両方にヒットがあり、比較する際に隣のチェンバーからの異なる RDO 情報でもって、本来とれるはずのない対応関係がとれてしまったことが原因であった。このバグは TrigT1TGC-00-02-45 でデバッグされコミットされた。このバグを考慮した上で対応関係をみると、表 5.12 のように大幅に回復した。

	個数	非対応率
共通に含まれる RDO	972,289	-
TrigT1TGC のみに含まれる RDO	332	$3.4 \times 10^{-4}$
StDec のみに含まれる RDO	16	$1.6 \times 10^{-5}$

表 5.12: trackletRDO(TripletStrip) の対応関係 (seg の反転を考慮した場合)

### 5.2.11 highptRDO(Wire)

RDO 内の変数である subDetectorId、rodId、sswId、strip、forward、sector、chip、hipt、hitId、sub、delta が全て一致しているかどうかを確認した。Wire の highptRDO の対応関係は表 5.13 の通りである。

	個数	非対応率
共通に含まれる RDO	648,956	-
TrigT1TGC のみに含まれる RDO	1,325	$2.0 \times 10^{-3}$
StDec のみに含まれる RDO	2,071	$3.2 \times 10^{-3}$

表 5.13: highptRDO(Wire) の対応関係

実機とシミュレーションの間であまり対応がとれていない。対応関係がとれない原因の 1 つとして考えられるのは TrackletTripletWire での実機とシミュレーションの不一致が、HPT でも引き継がれているということである。この場合は原因が TrigT1TGC にケーブリングスワップの影響を組み込んでいないことが原因のため、TGC システムに問題はない。そこでケーブリングスワップにより、実機とシミュレーションで対応がとれなかった TrackletTripletWire を含むイベントを除外して、HiptWire の対応関係を調べると、表 5.14 のようになる。

表 5.14 の結果では、Aside と Cside を区別しても、Aside で対応がとれないトリガー情報が依然として存在した。TrackletTripletWire の Aside はケーブリングスワップの影響で対応がとれないイベントとノイジーチャンネルを除外すると、100% 対応関係がとれるた

	個数	非対応率
共通に含まれる RDO	648,648	-
TrigT1TGC のみに含まれる RDO	1,218	$1.9 \times 10^{-3}$
StDec のみに含まれる RDO	1,964	$3.0 \times 10^{-3}$

表 5.14: highptRDO(Wire) の対応関係 (TrackletTripletWire でケーブルリングスワップにより対応がとれなかったイベントを除外した場合)

め、このことから HighptWire で対応がとれないのは TrackletTripletWire におけるシミュレーションの間違いが影響を及ぼしているわけではないと言える。

結果、対応関係がとれていない他の原因として浮上するのが HiptWire における delta 値の間違いである。delta 値が実機とシミュレーション間で一致しないことが原因でトリガー情報の対応がとれなかったのは計 826 個あり、表 5.14 における対応関係がとれない RDO の個数の多数を占める。delta 値の不一致の原因を考える。

1 つ目に考えられるのが、HPT における delta の算出方法の間違いである。これが原因であれば、delta 値が原因で実機とシミュレーションのトリガー情報が一致しない RDO が激増するため、可能性としては考えにくい。

次に原因として考えられたのが Aside の sub 値の間違いである。この変数は hptsubMatrix(0/1) の値を示している。sub 値はトリガーが成される物理的な位置を司る変数のうちの 1 つであり、これが間違えると delta も間違ふことになるからである。実際、HighptWireRDO における対応関係を見ると、図 5.11 のようになり、実機とシミュレーションの間で大きくずれていることが分かる。

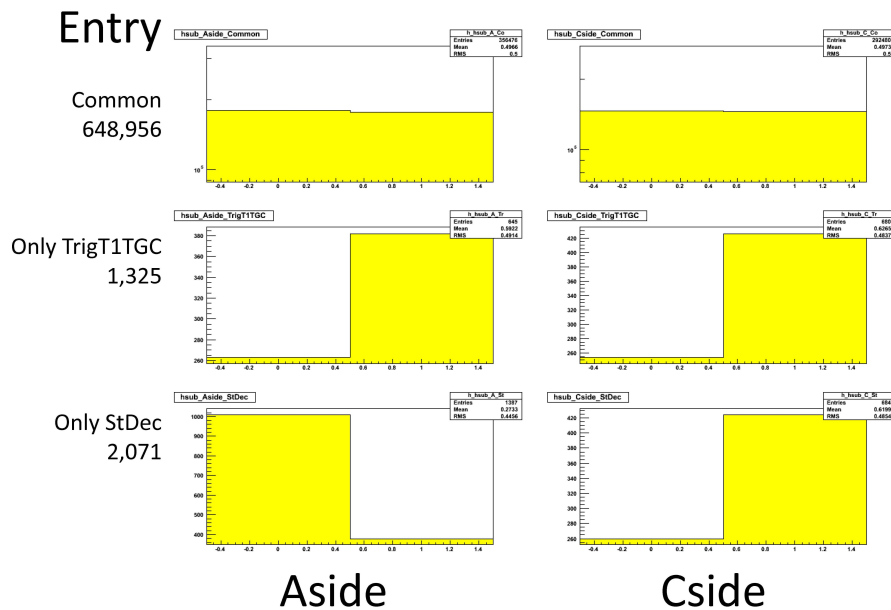


図 5.11: HighptWire における sub 値の対応関係

TrackletTripletStrip での seg 値の取り間違いは TrigT1TGC が原因であったが、今回の sub 値の場合は全イベントでその影響が見られないため、ソフトウェアのバグとは考えられない。原因として考えられるのは、PS Pack からの情報が HPT に送られる LVDS 信号の中で hptsubMatrix の値が反転している可能性が考えられる。これが原因の場合は、それが物理的な原因ではないため、ランダムに起こると予想される。ただしこの場合、何かしらの対処が施せるわけではない。しかし、その影響は、表 5.14 の通り実機では約 0.3 %と、TGC システムに大きな影響を及ぼしていないため無視をし、ケーブル等の交換は見送ることになった。

ただし、実機から出力される delta 値の解析により疑わしい実機の存在が示唆された。図 5.12 は Aside で実機とシミュレーションの対応がとれなかった RDO の delta 値のヒストグラムで、その値が-8 のところで高いビンが立っていることが伺える。

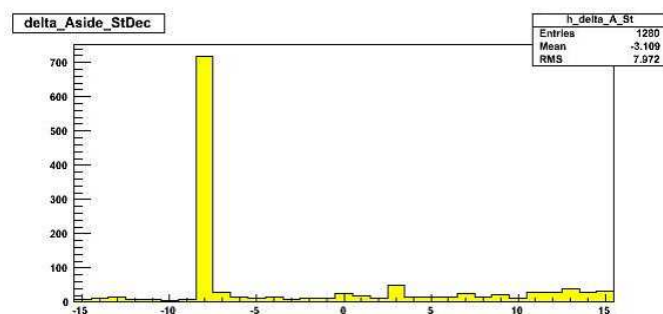


図 5.12: HighptWire における実機から出力される delta 値 (Aside でシミュレーションと一致しなかったトリガー情報)

この RDO は約 700 個あり表 5.14 の対応がとれない実機とシミュレーションのトリガー情報の個数の差にほぼ一致する。そのためシミュレーションでは想定されない実機の不具合が示唆される。この delta=-8 を頻繁に出すハードウェアは、

A12, Endcap, phi(triggerSector):3, chip:1or3 内, hitId:1

に相当する。この原因は当時の SL の読み出しチップに問題があったためであり、対処済みである。これは実機の highptRDO は SL の読み出しを元に作られるため、hitRDO を元に作られた TrigT1TGC での highptRDO にずれが生じたのだと考えられる。

### 5.2.12 highptRDO(Stirp)

RDO 内の変数である subDetectorId, rodId, sswId, strip, forward, sector, chip, hipt, hitId, sub, delta が全て一致しているかどうかを確認した。Strip の highptRDO の対応関係は表 5.15 の通りである。

100 %対応がとれているわけではないが、無視できる程度であり、TGC システムには異常はないと見受けられる。

	個数	非対応率
共通に含まれる RDO	1,089,906	-
TrigT1TGC のみに含まれる RDO	591	$5.4 \times 10^{-4}$
StDec のみに含まれる RDO	655	$6.0 \times 10^{-4}$

表 5.15: highptRDO(Strip) の対応関係

### 5.2.13 slRDO

RDO 内の変数である subDetectorId、rodId、cand3plus、forward、sector、index、muplus、threshold、roi が全て一致しているかどうかを確認している。slRDO の対応関係は表 5.16 の通りである。

	個数	非対応率
共通に含まれる RDO	584,391	-
TrigT1TGC のみに含まれる RDO	41,364	$6.6 \times 10^{-2}$
StDec のみに含まれる RDO	41,458	$6.6 \times 10^{-2}$

表 5.16: slRDO の対応関係

対応関係が非常に悪い。この原因は run159179 ときに SL の実機にバグがあったからである。対応関係を落としているのは変数 muplus の値 (SL における deltaR の値が正か負か) の取り違いによるものであり、本来 deltaR=0 のときは muplus は正 (true) が出力されなければいけないにも関わらず、負 (false) が出力されるコードが SL に搭載されていた。これは run159179 以降にデバッグが行われている。この実機のバグを TrigT1TGC で再現するように、すなわち deltaR=0 のときに muplus が負 (false) を出力するように変更し、対応関係を見ると、表 5.17 のようになる。

	個数	非対応率
共通に含まれる RDO	624,334	-
TrigT1TGC のみに含まれる RDO	1,421	$2.3 \times 10^{-3}$
StDec のみに含まれる RDO	1,515	$2.4 \times 10^{-3}$

表 5.17: slRDO の対応関係 (SL における muplus 値のバグを除外した場合)

対応関係は回復し、十分なパフォーマンスを維持していることが伺える。若干対応がとれていない箇所があるが、その数から察するに、これが HighptWire における delta 値の出力ミスにより、その delta 値を用いて SL のコインシデンスウインドウが参照されたことが大きな原因であろうと考えられる。よって SL におけるハードウェアは正常に動作していると結論づける。

### 5.2.14 TGC システム全体としての性能

本節の最後に、TGC システム全体としての性能評価を行う。以下のように、実際に動いている実機の稼働率を見積もるため、TGC システムの稼働率を定義する。

$$\text{稼働率} = 100 \% - \frac{\text{StDec のみに含まれる全 RDO 数}}{\text{共通に含まれる全 RDO 数} + \text{StDec のみに含まれる全 RDO 数}} \% \quad (5.2)$$

ここで、全 RDO 数というのは、トリガー情報を含む、trackletRDO、highptRDO、slRDO のことであり、ここで用いる RDO は、全て TrigT1TGC のデバッグ後の解析結果、ハードの不具合を対処した後の解析結果を用いる。すなわち、表 5.6、表 5.7、表 5.10、表 5.12、表 5.14、表 5.15、表 5.17 の共通の RDO 数の合計数と StDec のみに含まれる RDO 数の合計を用いることにする。ただし、表 5.14 に関しては対処済みの A12 の誤作動による約 700 イベントの対応のとれないイベントを考慮するために、StDec のみに含まれる RDO の数から 700 引いた数で見積もる。この情報はシミュレーションには存在せず、当然共通に含まれることもないので共通に含まれる RDO の数は変化しない。以上の情報を用いて、TGC システムの稼働率を計算すると、

$$\text{稼働率} \approx 99.9 \%$$

となり、高いパフォーマンスを維持していると言える。また、これは表 2.4 における LVL1 Muon TGC Trigger の稼働率 100 % にほぼ一致する値となっている。

### 5.2.15 TrigMuonCoinHierarchy を用いた TrigT1TGC による実データ解析の性能評価

本節では TrigT1TGC を用いた TGC システムの動作検証自身の性能を評価する。手法としては、TrigT1TGC の解析結果が第 4 章でまとめた TrigMuonCoinHierarchy のパッケージによって再現できるか否かで判断することにした。

前節では highptWireRDO の解析において、実機において誤った sub 値が原因となって、想定外の delta 値が出力されている可能性について述べた。これが TrigMuonCoinHierarchy で再現できるか調べた。hitRDO から作られた highptWireRDO について解析することの逆方向からの解析というのは、TrigMuonCoinHierarchy において、HPT でコインシデンスがとれたときに、その Wire の情報が TGC1 に存在しているか否かを見ることと同義であり、この 2 つを比較することによって、その目的を果たせると考えたためである。以下での TrigMuonCoinHierarchy の解析には mask2 は run159179 当時の mask2 を適用し、TrigT1TGC と同じ条件での解析を行っている。

まず、TrigMuonCoinHierarchy に run159179 の RAW を TrigT1TGC に入力することで得られた ESD の中で、HighptWire において誤った delta 値を出力したイベントのみを入力すると、

$$\text{HiPt Wire} \rightarrow \text{TGC1 Wire Hits} : \quad 868 / 887 \quad \text{非対応率} : 2.0 \times 10^{-2}$$

となり、表 4.3 における同階層の対応関係を見ると、完全ではないがほぼ同程度の非対応率を示している。しかし、TrigMuonCoinHierarchy に run15179 の RAW を通常のデコー

デコーディングで得られた実機由来の ESD の中で、HighptWire において誤った delta 値を出力したイベントのみを入力すると、

HiPt Wire→TGC1 Wire Hits : 483 / 887 非対応率 :  $4.6 \times 10^{-1}$

のように対応関係を著しく落としている。ここで対応がとれなかった 404 のトリガー情報は、TrigMuonCoinHierarchy に run159179 の RAW を通常のデコーディングで得られた実機由来の ESD の全イベントを入力したときの、

HiPt Wire→TGC1 Wire Hits : 187,629 / 188,615 非対応率 :  $5.2 \times 10^{-3}$

において対応がとれない 986 のトリガー情報の約 41 % を占め、TrigT1TGC の実データ解析で発覚した実機からの誤った delta 値の出力が、TrigMuonCoinHierarchy でも対応関係を落としている大きな原因となっていることが伺える。

また同様に TrigMuonCoinHierarchy における HiPt Strip→TGC1 Strip Hits の対応関係は、TrigT1TGC の実データ解析における highptRDO の解析とコンシステントな結果が得られるはずである。run15179 の RAW を通常のデコーディングで得られた実機由来の ESD の全イベントを TrigMuonCoinHierarchy に入力したところ、

HiPt Strip→TGC1 Strip Hits : 743,582 / 743,824 非対応率 :  $3.3 \times 10^{-4}$

の結果が得られる。これは表 5.15 における実機における非対応率  $6.0 \times 10^{-4}$  とコンシステントな結果が得られている。

次に TrigMuonCoinHierarchy における Tracklet Wire(Strip)→TGC2 and 3 Wire(Strip) Hits の対応関係は、TrigT1TGC の実データ解析における TrackletDoubletWire(Strip)RDO の解析とコンシステントな結果が得られるはずである。run159179 の RAW を通常のデコーディングで得られた実機由来の ESD の全イベントを TrigMuonCoinHierarchy に入力したところ、

Tracklet Wire→TGC2 and 3 Wire Hits : 649,194 / 651,113 非対応率 :  $2.9 \times 10^{-3}$   
Tracklet Strip→TGC2 and 3 Strip Hits : 988,040 / 991,616 非対応率 :  $3.6 \times 10^{-3}$

の結果が得られる。これは表 5.6 の実機における非対応率  $1.0 \times 10^{-4}$ 、表 5.7 の実機における非対応率  $2.6 \times 10^{-5}$  よりも対応関係が悪い。これは第 4 章のまとめ部分でも述べた通り、構造的に見られる実機のバグを TrigMuonCoinHierarchy に反映できていないことが原因として考えられる。

以上のように、TrigT1TGC に実データを入力し、それを TGC システムの性能評価に用いることは、十分に妥当な手段であると結論づけられる。

### 5.2.16 TrigT1TGC を用いた TGC システムの動作検証のまとめ

以上のように、TrigT1TGC を実データで用いることによって、ハードウェア上のバグを確認することができた。このバグは run159179 以降にデバッグの対処がとられている。バグをツールの中で確認できるということは、このツール、アルゴリズムが TGC システムを評価するに十分な性能を持っていることを表している。またその性能は前節で述べた



ように TrigMuonCoinHierarchy を用いて評価しても十分なパフォーマンスを提供できるツールであることが確認された。

よって今回の解析により現状の TGC システムにはハードウェア的な問題は残されておらず、この状態のまま TGC は 2011 年 2 月以降の実験再開に臨んで支障がないと結論づけられる。

また TrigT1TGC パッケージに潜んでいたバグを取り除くことに繋げることができた。これにより、今後 TrigT1TGC を扱う時、その性能が上がると思われる。

ただし以上の TGC システムの評価の体系が ATLAS 実験における標準ツールとなるかどうかはまだ未定である。その理由としては、前述の通り実データを TrigT1TGC に読み込ませる際には、実データのフォーマットで書かれた RAW を、シミュレーション用のフォーマットに変換しなければならない。もしもその変換の際に致命的なバグがあれば、このツールは破綻する。また RAW から順番にリコンストラクションし直して解析を行うために時間と労力がかかるので、仮に標準ツールになっても、オンラインで、且つ毎回の run でシステム評価を行うツールとしては不適當である。

またオフラインで解析を行うにしても、RAW はグリッド上に記録されないので研究者が自由に取得できない。RAW の取得にはデータ管理を行っているエキスパートにその旨と、どこのグリッド上にコピーして欲しいかを申請する必要がある。今回の研究で最新の RAW をリコンストラクションして解析することが出来なかった大きな原因でもある。

しかし、これらの事項を解決することができれば、TrigT1TGC を用いた実データ解析は TGC システムの性能評価において十分な力を発揮でき、且つ得られるトリガー情報をハードウェアの出力のより近い段階でシステムを評価することができるため、システム評価を代表するツールとなるだろう。またシステム評価以外にも、TrigT1TGC 自身のデバッグ用ツール、アップデートされたコインシデンスウインドウの評価等にも用いることができると考えられ、需要は高いと考えられる。

## 第6章 まとめ

第2章、第3章で述べたように、LHC加速器は2009年11月以降、順調にビームの衝突に成功しており、ビーム衝突点に設置されたATLAS検出器を用いたATLAS実験も順調にデータテイキングを行うことに成功している。そして実験において重要な役割を担うミュオントリガーシステムの1つを担当するミュオン検出器TGCは、Trigger Efficiencyの観点から見ても十分高いパフォーマンスを実験に提供している。

また第4章で述べたように、現在実験で標準ツールとして用いられているTrigMuonCoinHierarchyというオフラインソフトウェアのパッケージを用いて、2010年の最新の実験データを用いてLVL1 TGC ミュオントリガーシステム (TGC システム) の性能評価を行ったところ、現状TGCシステムはハードウェアの観点から見ても高いパフォーマンスを維持していることがわかった。

そして第5章で述べたように、TrigT1TGCというこれまでTGCシステムの再現で用いられてきたシミュレーションツールを、実験データを使って実際のトリガー情報と同じトリガー情報がシミュレートされるかどうかを検証する枠組みを確立することができた。この枠組みを用いれば、TGCシステムをハードウェアの観点から評価することができる。

そして2010年の実験データを用いてこのツールを動かしたところ、開発された枠組みはTGCの現状を正しく反映する結果を出し、またTrigT1TGC内に残っていたバグを発見することができた。そしてTrigMuonCoinHierarchyを用いて、その性能の高さを確認づけることができた。そして2010年の実験データの性能評価の結果より、TGCシステムはハードウェアの観点から見て、約99.9%の稼働率で動作していると見積もられた。以上から2011年2月以降、冬季シャットダウン明けから再開されるデータテイキングにTGCシステムは現状のまま臨むことができると結論づける。

# 付録A TGCのナンバリングの補足

## A.1 RoIナンバリング

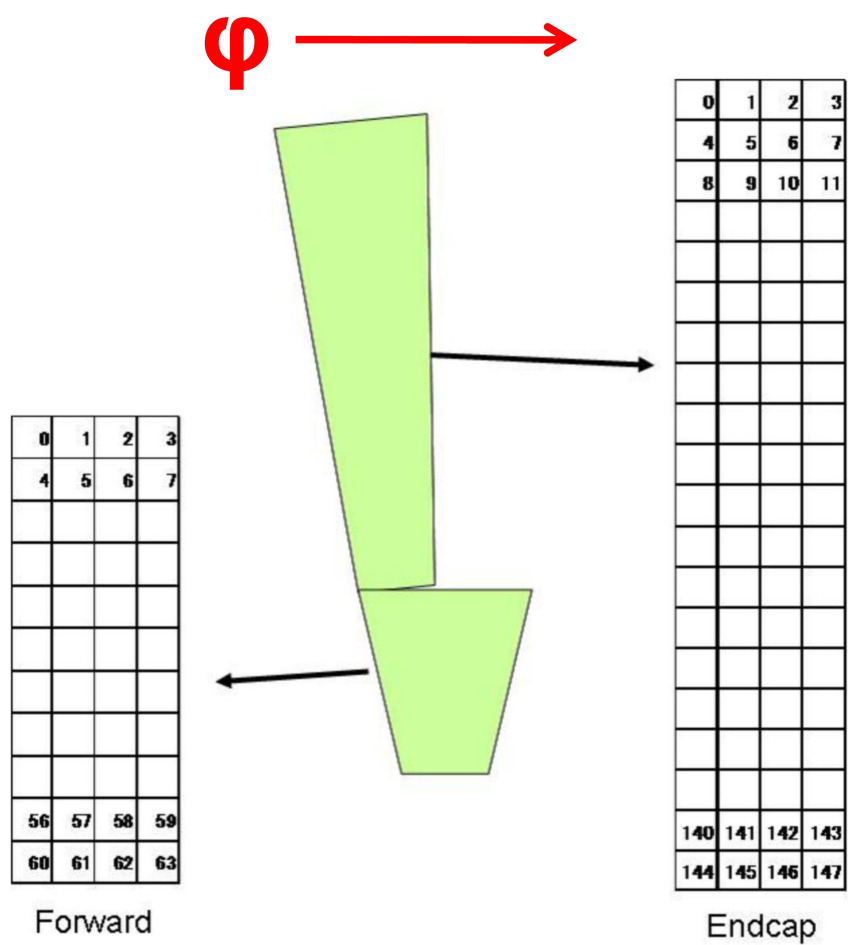


図 A.1: RoI のナンバリング

## A.2 1/12セクターナンバリング

図 A.2 のにおいて、座標軸の  $\phi = 0$  を含む 1/12 セクターを 1/12 セクター番号の 1 番とし、1~12 まで割り振られる。

セクターの中でもチェンバー毎に phi 番号が割り振られている。phi 番号については図 A.2 の通りである。

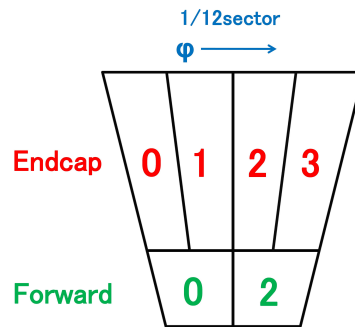


図 A.2: 1/12 セクターにおける phi 番号

## A.3 1/8セクターナンバリング

図 A.3 のにおいて、座標軸の  $\phi = 0$  を含む 1/8 セクターを 1/8 セクター番号の 0 番とし、0~7 まで割り振られる。

セクターの中でもチェンバー毎に octant 番号が割り振られている。octant 番号については図 A.3 の通りである。

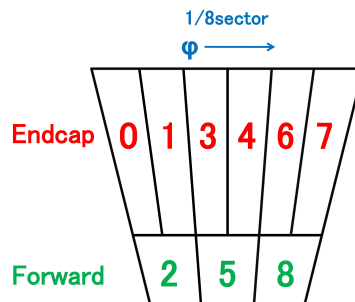


図 A.3: 1/8 セクターにおける octant 番号

## A.4 トリガーセクター ID

図 A.4 は、Aside のナンバリングを表している。 $\phi = 0$  のトリガーセクターから順に Aside のエンドキャップが 1~48、フォワードが 49~72 でナンバリングされる。

Cside は  $\phi = 0$  のトリガーセクターから順にエンドキャップが 73~120、フォワードが 121~144 とナンバリングされている。

この ID は主に TrigMuonCoinHierarchy パッケージの中で使用されている。

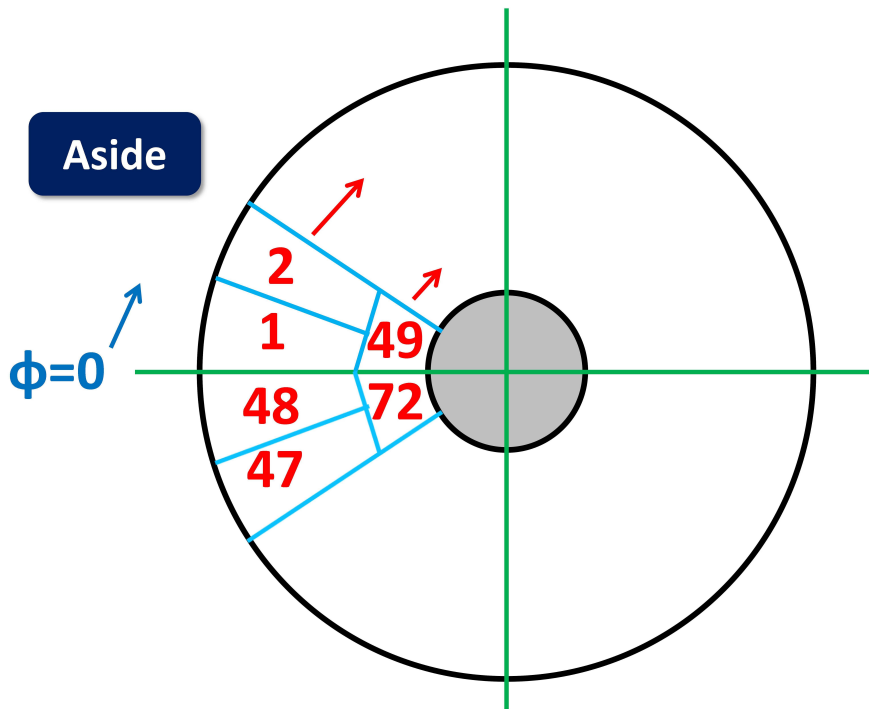


図 A.4: トリガーセクター ID のナンバリング

## 付 録 B 各種変数のまとめ

### B.1 Identifier

#### B.1.1 OfflineID

変数名	型	値域	意味
stationName	int	41 : TGC1 Forward 42 : TGC1 Endcap 43 : TGC2 Forward 44 : TGC2 Endcap 45 : TGC3 Forward 46 : TGC3 Endcap 47 : TGC FI 48 : TGC EI	TGC の位置を示す
stationEta	int	M1 : 1~4 M2,M3 : 1~5 Forward : 1	R の増加方向にチェンバーに割り振られた番号 Aside はプラス、Cside はマイナス
stationPhi	int	Endcap : 1~8 Forward : 1~24	Phi の増加方向に向かって、 チェンバーに割り振られた番号
channel	int	Wire : 1~n Strip : 1~32	Wire は R の増加方向に、Strip は Phi の増加方向に、 各チャンネルに割り振られた番号

表 B.1: OfflineID の変数一覧

## B.1.2 ReadoutID

変数名	型	値域	意味
subDetectorID	int	A/C : 0x67/0x68	Aside か Cside かを示す
rodID	int	1~12	Phi の増加方向に 1/12 のセクターのどこから読み出されたかを示す
sswId	int	0,1 : EndcapTriplet 2 : ForwardTriplet 3~6 : EndcapDoublet 7 : ForwardDoublet 8 : InnerStation 9 : SLB on SL	SSW のアドレスを示す
sbLoc	int	0~n	SSW 内の SLB の区別をすることができる番号
channelId	int	0~39 : coincidece info 40~199 : hitmap	SLB の出力 200bit を表す 図 D.1 参照のこと

表 B.2: ReadoutID の変数一覧

### B.1.3 OnlineID

変数名	型	値域	意味
subsystemNumber	int	A/C : +1/-1	Aside か Cside を示す
octantNumber	int	0~7	Phi の増加方向に Wheel を 8 等分した時の番号 (A.3 参照)
moduleNumber	int	0~12	Wheel を 8 等分したときに、Endcap、Forward を Phi の増加方向に区別した番号
layerNumber	int	0~6 : M1~M3 7,8 : EI/FI	z の増加方向に各 layer に割り振られた番号
rNumber	int	M1 : 1~4 M2,M3 : 0~4 Forward : 0	Eta の増加方向に各チェンバーに割り振られた番号
wireOrStrip	bool	T/F : Strip/Wire	Strip と Wire の区別
channelNumber	int	0~n Wire : increase with Eta Strip : AsideForward CsideBackward :increase with Phi Strip : AsideBackward CsideForward :decrease with Phi	各チャンネルに割り振られた通し番号

表 B.3: OnlineID の変数一覧



## B.2 MuonRDO

### B.2.1 hitRDO

変数名	型	値域	意味
bcTag	int	0 : Unknown 1 : Previous 2 : Current 3 : Next	どのバンチ由来のものを識別する
subDetectorId	int	A/C : 103/104	Aside か Cside を区別する
rodId	int	1~12	1/12 毎の ROD の区別
sswId	int	0,1 : EndcapTriplet 2 : ForwardTriplet 3~6 : EndcapDoublet 7 : ForwardDoublet 8 : Inner 9 : SLBonSL	1/12 毎の SSW の区別
slbId	int	0~25	slbLoc を表す
l1Id	int	-	LVL1 トリガーの ID
bcId	int	-	バンチの ID
slbType	enum	0 : DoubletWire 1 : DoubletStrip 2 : TripletWire 3 : TripletStrip 4 : InnerWire 5 : InnerStrip 6 : Unknown	どのタイプのチェンバーの SLB かを示す
adj	bool	T/F	SLB の adjacent 領域の bit か否か
channel	int	40~199	SLB の出力 200bit のうち どのチャンネルかを示す

表 B.4: hitRDO の変数一覧

### B.2.2 trackletRDO

変数名	型	値域	意味
bcTag	int	0 : Unknown 1 : Previous 2 : Current 3 : Next	どのバンチ由来のものを識別する
subDetectorId	int	A/C : 103/104	Aside か Cside を区別する
rodId	int	1~12	1/12 毎の ROD の区別
sswId	int	0/1 : EndcapTriplet 2 : ForwardTriplet 3~6 : EndcapDoublet 7 : ForwardDoublet 8 : Inner 9 : SLBonSL	1/12 毎の SSW の区別
slbId	int	0~25	sbLoc を表す
l1Id	int	-	LVL1 トリガーの ID
bcId	int	-	バンチの ID
slbType	enum	0 : DoubletWire 1 : DoubletStrip 2 : TripletWire 3 : TripletStrip 4 : InnerWire 5 : InnerStrip 6 : Unknown	どのタイプのチェンバーの SLB かを示す
delta	int	Wire : -7~7 Strip : -3~3	TGC2 と TGC3 のヒット位置の差 (trgch 単位)
seg	int	0/1 : TRIGA/B	TripletStrip の TRIGA/TRIGB の区別
sub	int	0~3	slbsubMatrix を表す (TriggerData の TRIG0~3 に対応)
rphi	int	0~31	TGC3 の slbsubMatrix 内の位置 (trgch 単位)

表 B.5: trackletRDO の変数一覧

## B.2.3 highptRDO

変数名	型	値域	意味
bcTag	int	0 : Unknown 1 : Previous 2 : Current 3 : Next	どのバンチ由来のものかを識別する
subDetectorId	int	A/C : 103/104	Aside か Cside を区別する
rodId	int	1~12	1/12 毎の ROD の区別
sswId	int	0/1 : EndcapTriplet 2 : ForwardTriplet 3~6 : EndcapDoublet 7 : ForwardDoublet 8 : Inner 9 : SLBonSL	1/12 毎の SSW の区別
slbId	int	0~25	sbLoc を表す
l1Id	int	-	LVL1 トリガーの ID
bcId	int	-	バンチの ID
strip	bool	T/F : Strip/Wire	Strip/Wire の区別
forward	bool	T/F : Forward/Endcap	Forward/Endcap の区別
sector	int	Forward : 0/1 Endcap : 0~3	1/12sector 内の subTriggerSector の値
chip	int	0~3	HPT ボード上のチップの区別
index	int	0/1	未使用
hipt	bool	T/F : High/Low	HPT/LPT coincidence の区別
hitId	int	1~6	HPT チップ内のマトリックスの区別
sub	int	0/1	hptsubMatrix(マトリックスの区別)
delta	int	Wire : -15~15 Strip : -7~7	TGC3 と 1 のヒット位置の差 (trgch 単位)

表 B.6: highptRDO の変数一覧

B.2.4 sLRDO

変数名	型	値域	意味
bcTag	int	0 : Unknown 1 : Previous 2 : Current 3 : Next	どのバンチ由来のものを識別する
subDetectorId	int	A/C : 103/104	Aside か Cside を区別する
rodId	int	1~12	1/12 毎の ROD の区別
l1Id	int	-	LVL1 トリガーの ID
bcId	int	-	バンチの ID
cand3plus	bool	T/F	trigger 候補が 3 つ以上あれば True
forward	bool	T/F : Forward/Endcap	Forward/Endcap の区別
sector	int	Forward : 0/1 Endcap : 0~3	1/12sector 内の subTriggerSector の値
index	int	0/1	2 つの trigger 候補の区別
muplus	bool	T/F : $dR \geq 0/dR < 0$	deltaR の符号の区別
threshold	int	1~6	$p_T$ threshold の値
overlap	bool	-	未使用
roi	int	Forward : 0~63 Endcap : 0~147	RoI 番号 (図 A.1 参照)

表 B.7: sLRDO の変数一覧

## B.3 MuonPRD

### B.3.1 hitPRD

変数名	型	意味
RDOId	Identifier	channel の OfflineID
idDE	IdentifierHash	所属する Collection の Hash
locpos	Trk:: LocalPosition	TgcReadoutElement 内での位置 [mm]
rdoList	vector<Identifier>	同一イベントの hitRDO の OfflineID リスト
locErrMat	Trk:: ErrorMatrix	hit 位置のエラー
detEI	MuonGM:: TgcReadoutElement	属する TgcReadoutElement へのポインタ

表 B.8: hitPRD の変数一覧

### B.3.2 trackletPRD

変数名	型	意味
ChannelIdIn	Identifier	TGC2 の hit の OfflineID
ChannelIdOut	Identifier	TGC3 の hit の OfflineID
collectionIdHash	IdentifierHash	所属する Collection の Hash
detElIn	MuonGM:: TgcReadoutElement	TGC2 の TgcReadoutElement へのポインタ
detElOut	MuonGM:: TgcReadoutElement	TGC3 の TgcReadoutElement へのポインタ
type	TgcCoinData:: CoinDataType	CoinData の Type、tracklet は 0
isAside	bool	T/F : Aside/Cside
phi	int	triggerSector 番号 (Foward : 1~24、Endcap : 1~48)
isForward	bool	T/F : Forward/Endcap
isStrip	bool	T/F : Strip/Wire
trackletId	int	trackletId を示す
posIn	Trk:: LocalPosition	TGC2 の TgcReadoutElement 内での位置 [mm]
posOut	Trk:: LocalPosition	TGC3 の TgcReadoutElement 内での位置 [mm]
widthIn	double	TGC2 の channel の幅 [mm]
widthOut	double	TGC3 の channel の幅 [mm]
delta	int	sagitta(trgch 単位)、trackletRDO の delta 値
sub	int	trackletRDO の subMatrix の値

表 B.9: trackletPRD の変数一覧

## B.3.3 highptPRD

変数名	型	意味
ChannelIdIn	Identifier	TGC1 の hit の OfflineID
ChannelIdOut	Identifier	TGC3 の hit の OfflineID
collectionIdHash	IdentifierHash	所属する Collection の Hash
detElIn	MuonGM:: TgcReadoutElement	TGC1 の TgcReadoutElement へのポインタ
detElOut	MuonGM:: TgcReadoutElement	TGC3 の TgcReadoutElement へのポインタ
type	TgcCoinData:: CoinDataType	CoinData の Type、highpt は 1
isAside	bool	T/F : Aside/Cside
phi	int	triggerSector 番号 (Foward : 1~24、Endcap : 1~48)
isForward	bool	T/F : Forward/Endcap
isStrip	bool	T/F : Strip/Wire
trackletId	int	trackletId を示す
posIn	Trk:: LocalPosition	TGC1 の TgcReadoutElement 内での位置 [mm]
posOut	Trk:: LocalPosition	TGC3 の TgcReadoutElement 内での位置 [mm]
widthIn	double	TGC1 の hit が属する RoI の幅 [mm]
widthOut	double	TGC3 の hit が属する RoI の幅 [mm]
delta	int	sagitta(trgch 単位)、highptRDO の delta 値
sub	int	highptRDO の hptsubMatrix の値

表 B.10: highptPRD の変数一覧

### B.3.4 slPRD

変数名	型	意味
ChannelIdOut	Identifier	TGC3 の hit(Wire) の OfflineID
collectionIdHash	IdentifierHash	所属する Collection の Hash
detElOut	MuonGM:: TgcReadoutElement	TGC3 の TgcReadoutElement へのポインタ
type	TgcCoinData:: CoinDataType	CoinData の Type、sl は 2
isAside	bool	T/F : Aside/Cside
phi	int	triggerSector 番号 (Foward : 1~24、Endcap : 1~48)
isForward	bool	T/F : Forward/Endcap
trackletId	int	Wire の trackletId を示す
trackletIdStrip	int	Strip の trackletId を示す
posOut	Trk:: LocalPosition	TGC3 の TgcReadoutElement 内での位置 [mm]
errMat	Trk:: ErrorMatrix	TGC3 の RoI の幅の行列 [mm]
roi	int	RoI 番号、slRDO の roi 値 (図 A.1 参照)
pt	int	$p_T$ threshold の値、slRDO の threshold 値
isPositiveDeltaR	bool	T/F : +/-、slRDO の isMuplus 値

表 B.11: slPRD の変数一覧



## 付録C 各種パッケージのクラス階層構造図

### C.1 MuonCnv/MuonTGC\_CnvTools

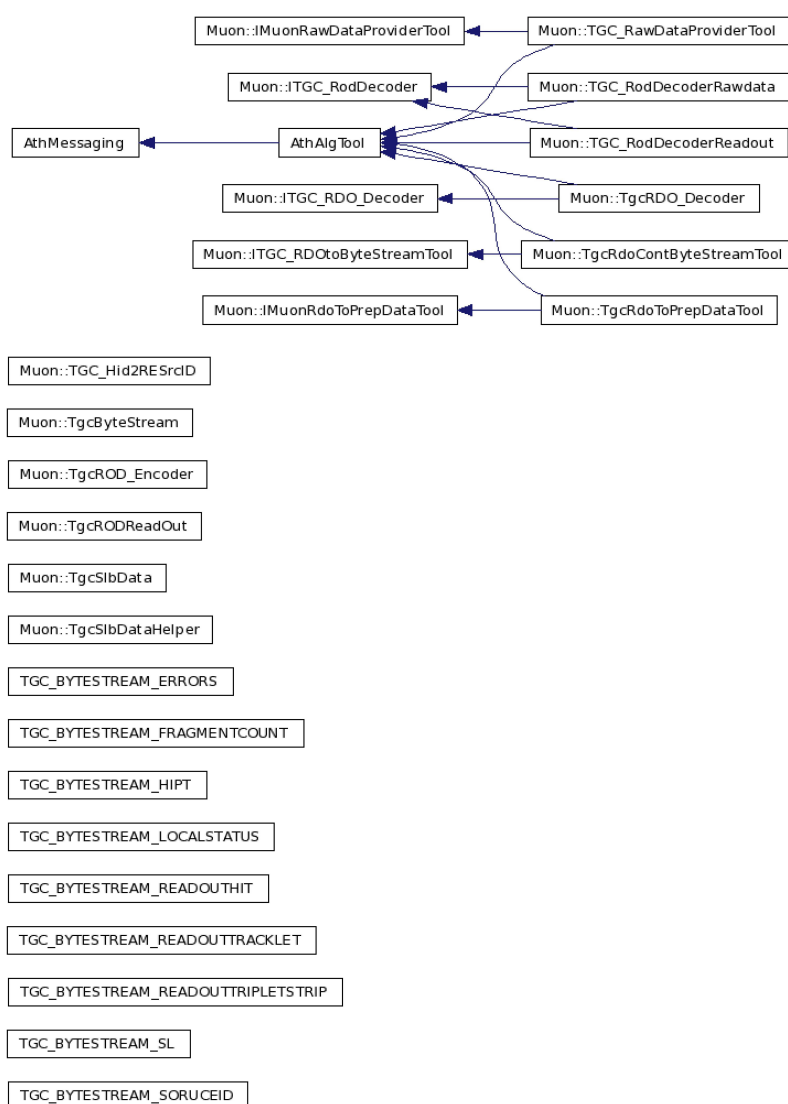


図 C.1: MuonCnv/MuonTGC\_CnvTools のクラス階層構造 [12]

## C.2 MuonCommissioning/MuonCommAlgs

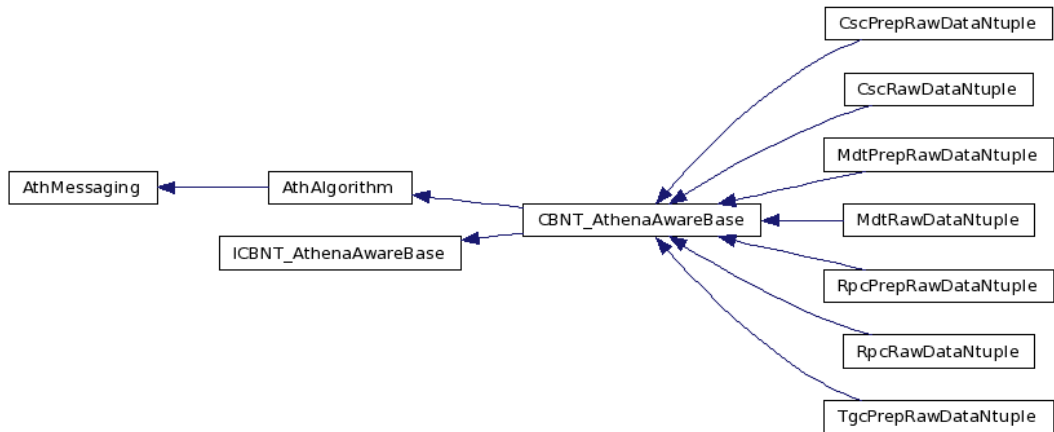


図 C.2: MuonCommissioning/MuonCommAlgs のクラス階層構造 [12]

## C.3 MuonSimulation/MuonDigitization/TGC\_Digitization

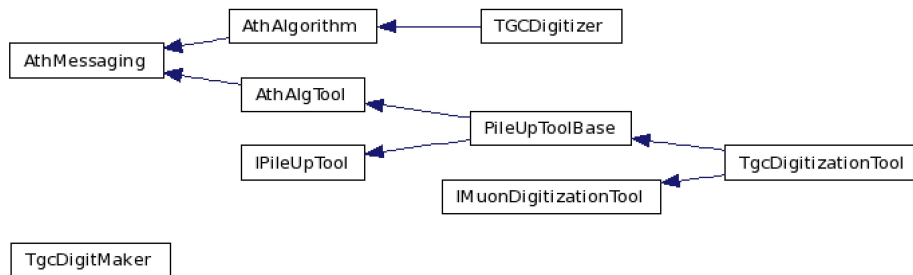


図 C.3: MuonSimulation/MuonDigitization/TGC\_Digitization のクラス階層構造 [12]

## C.4 MuonConditions/MuonCablings/TGCCabling12

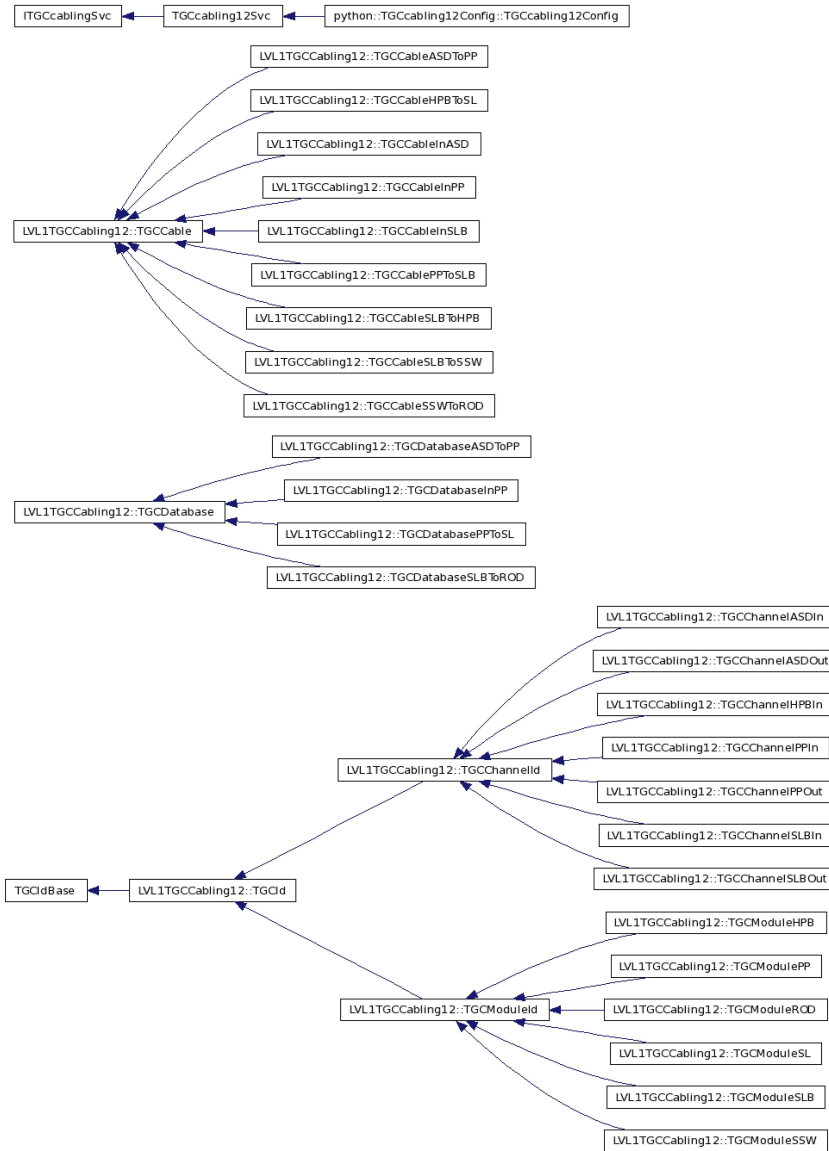


図 C.4: MuonConditions/MuonCablings/TGCCabling12 のクラス階層構造 [12]

## C.5 MuonReconstruction/MuonRecExample

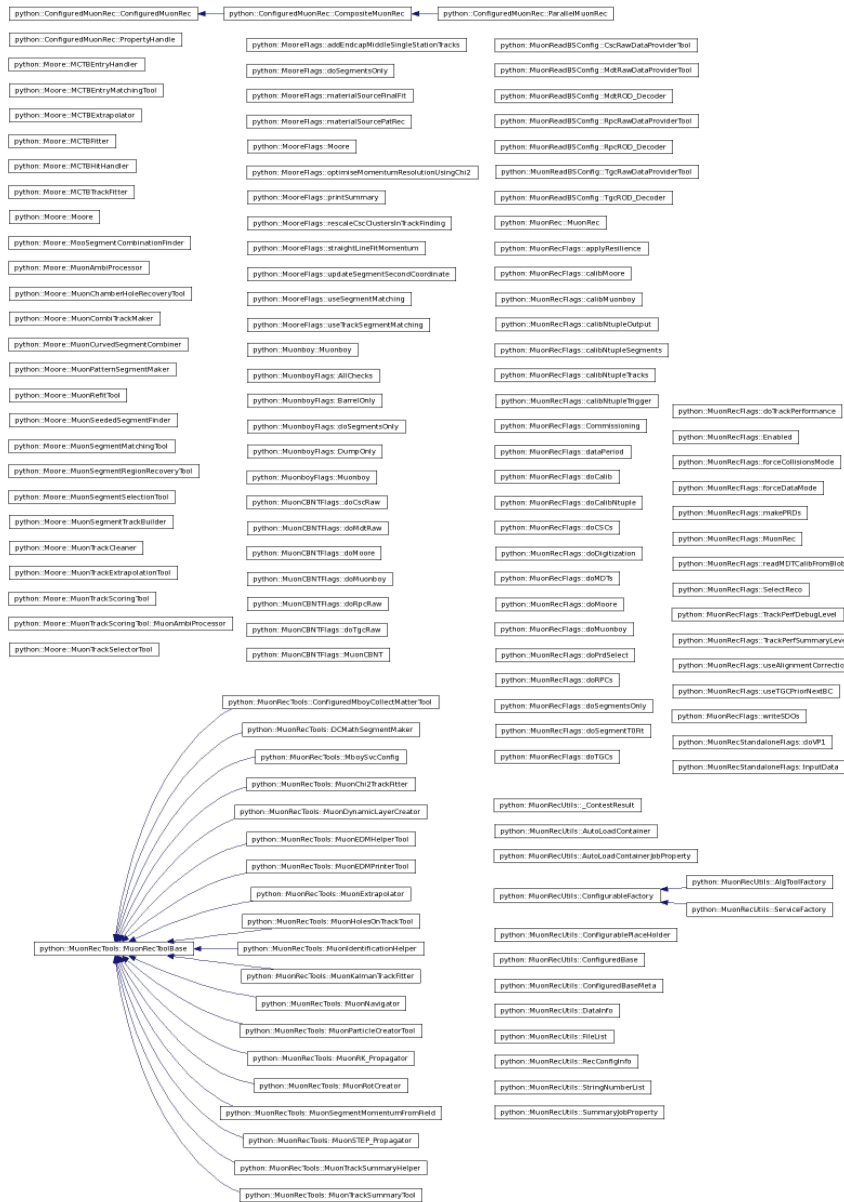


図 C.5: MuonReconstruction/MuonRecExample のクラス階層構造 [12]

## C.6 MuonReconstruction/MuonRecEvent/MuonRDO

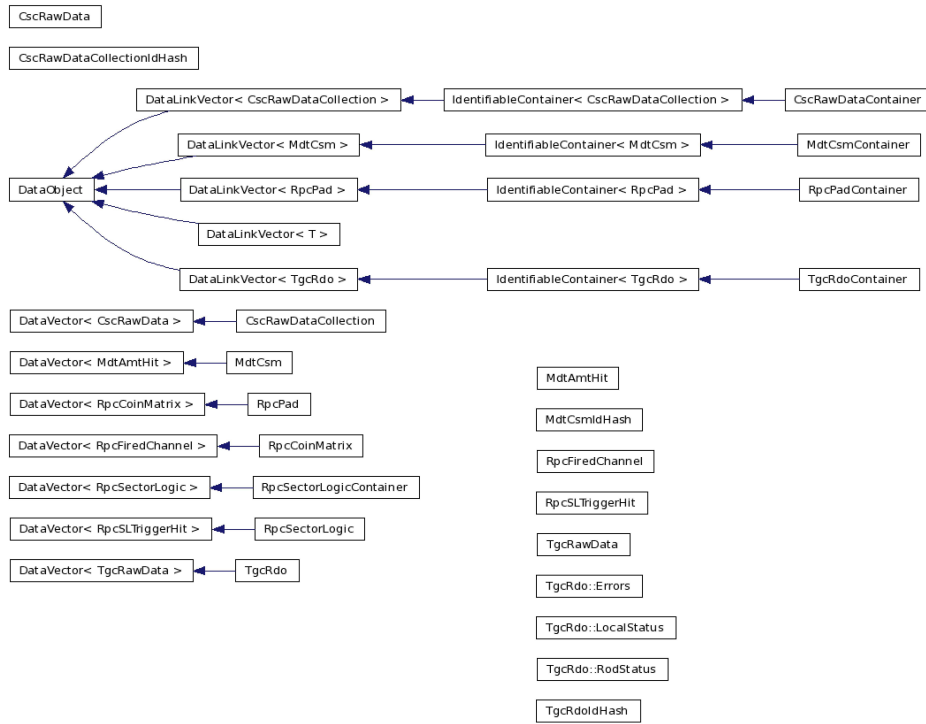


図 C.6: MuonReconstruction/MuonRecEvent/MuonRDO のクラス階層構造 [12]

## C.7 MuonReconstruction/MuonRecEvent/MuonPrepRawData

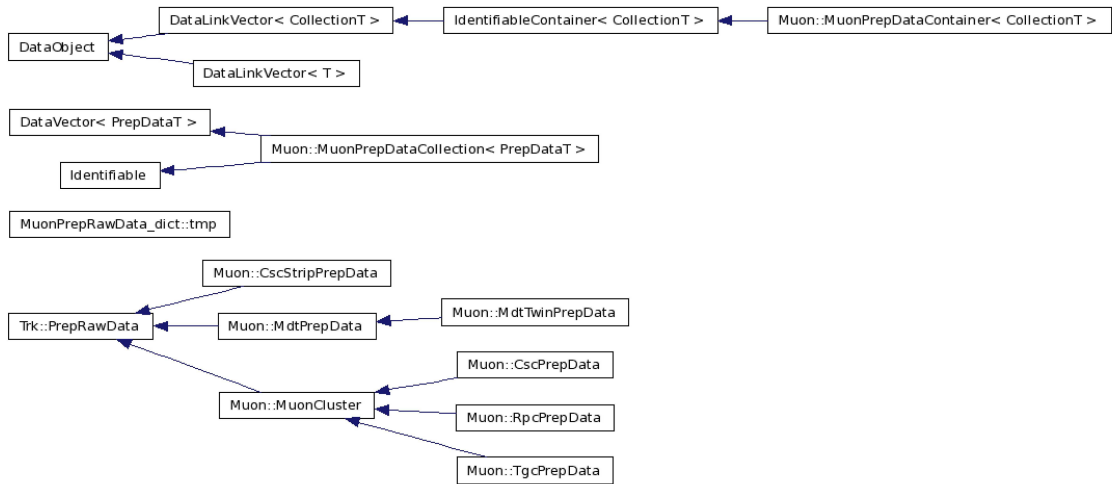


図 C.7: MuonReconstruction/MuonRecEvent/MuonPrepRawData のクラス階層構造 [12]

## C.8 MuonReconstruction/MuonRecEvent/MuonTrigCoinData

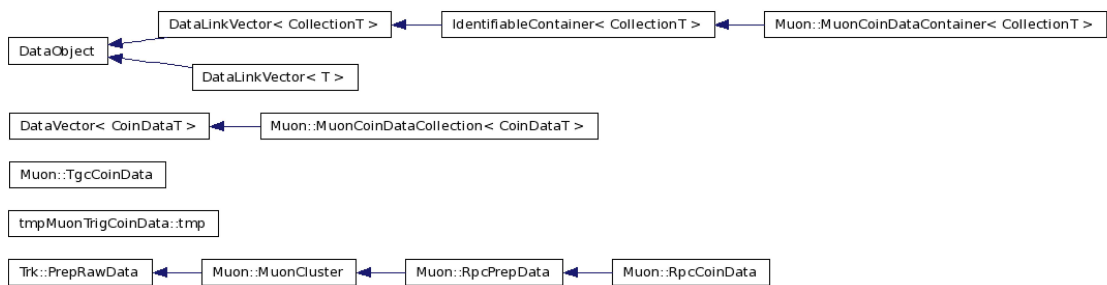
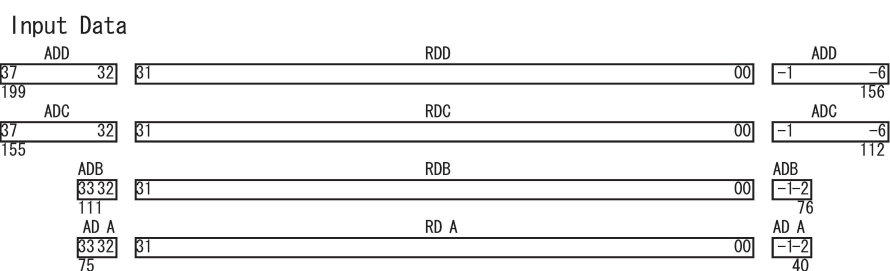


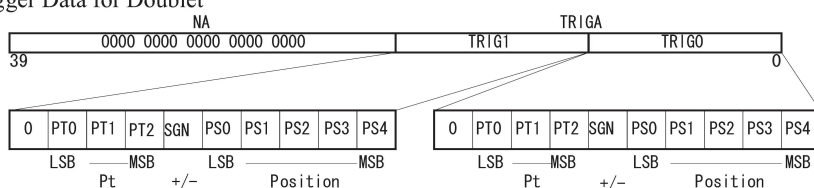
図 C.8: MuonReconstruction/MuonRecEvent/MuonTrigCoinData のクラス階層構造 [12]

# 付 録 D 各種データフォーマット

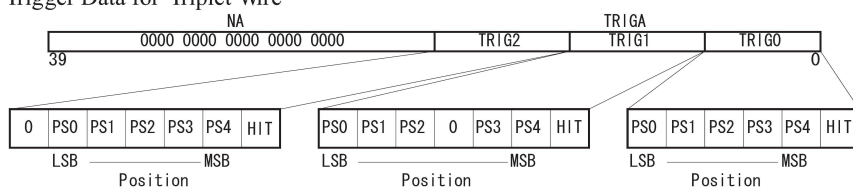
## D.1 SLB のデータフォーマット



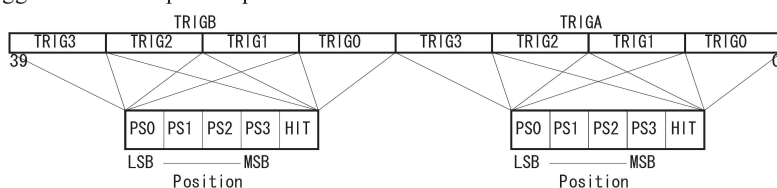
Trigger Data for Doublet



Trigger Data for Triplet Wire



Trigger Data for Triplet Strip



◀ To SSW

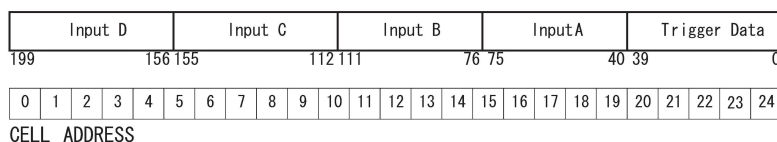


図 D.1: SLB のデータフォーマット

## 参考文献

- [1] ATLAS Collaboration. "ATLAS Photos".  
<http://www.atlas.ch/photos/index.html>.
- [2] ATLAS Collaboration. "ATLAS EXPERIMENT - Public Results".  
<https://twiki.cern.ch/twiki/bin/view/AtlasPublic/RunStatsPublicResults2010>.
- [3] CMS Collaboration. "The CMS experiment at the CERN LHC". 2008 JINST 3 S08004.
- [4] ALICE Collaboration. "The ALICE experiment at the CERN LHC". 2008 JINST 3 S08002.
- [5] The LHCb Collaboration. "The LHCb Detector at the LHC". 2008 JINST 3 S08005.
- [6] 岡田勝吾. "ATLAS 実験における質量が縮退した超対称性粒子の発見可能性に関する研究". 神戸大学修士論文, 2008.
- [7] Michael Spira. "QCD Effects in Higgs Physics". arXiv:hep-ph/9705337v2 12 Nov 1997.
- [8] ATLAS Collaboration. "ATLAS DETECTOR AND PHYSICS PERFORMANCE Technical Design Report Volume II". CERN/LHCC 99-15, 1999.
- [9] ATLAS Collaboration. "The ATLAS Experiment at the CERN Large Hadron Collider". 2008 JINST 3 S08003.
- [10] ATLAS Muon Collaboration. "ATLAS Muon Spectrometer Technical Design Report". CERN/LHCC 97.22, 1997.
- [11] ATLAS Computing Group. "ATLAS Computing Technical Design Report". CERN-LHCC-2005-022, 2005.
- [12] ATLAS Computing Group. "ATLAS TWiki".  
<https://twiki.cern.ch/twiki/bin/viewauth/Atlas/AtlasComputing>.
- [13] ATLAS Level-1 Trigger Group. "ATLAS Level-1 Trigger Group Technical Design Report". ATLAS TDR-12, 2000.
- [14] ATLAS 日本グループ. "ATLAS-Japan Home Page".  
<http://atlas.kek.jp/research/index.html>.



- 
- [15] ATLAS Collaboration. "ATLAS Collaboration Site".  
<http://atlas.web.cern.ch/Atlas/Collaboration/>.
- [16] 早川俊. "LHC シングルビームを用いた ATLAS 実験前後方ミュオントリガーシステムの性能評価". 神戸大学修士論文, 2009.
- [17] ATLAS Level-1 muon trigger group. "Naming and numbering scheme for the Endcap muon trigger system Version 1.19". ATL-M UON-2001-002, 2005.
- [18] 西山知徳. "L1 Trigger Efficiency summary". ATLAS 西日本グループ 2010 年 11 月会合, 2010.
- [19] 越前谷陽佑. "ATLAS 前後方ミュオントリガーシステムのコミッショニングとアップグレードに向けた研究開発". 東京大学修士論文, 2010.

## 謝辞

本研究を行うにあたり、御指導を頂きました指導教官の藏重久弥教授<sup>a</sup>に心よりの感謝を申し上げます。藏重様には研究面だけでなく私生活の面でも相談に乗って頂きました。自分自身を見つめ直す機会を作って頂くこともありました。素晴らしい指導教官に恵まれた学生生活を送ることができました。本当にありがとうございます。

本研究に着手する契機を作って頂き、且つ日頃から助言を多く頂きました織田勸研究員<sup>b</sup>には大変感謝しております。研究に行き詰まりなかなか進展しないときや、CERN 出張の折には隣で多くのアドバイスを頂きました。私が日本で研究している折にもメールで頻繁に助言を頂きました。本当にありがとうございます。

神戸大学 ATLAS グループの川越清以氏<sup>a</sup>、山崎祐司氏<sup>a</sup>、越智敦彦氏<sup>a</sup>、松下崇氏<sup>a</sup>、石川明正氏<sup>a</sup>の皆様には、研究室でのグループミーティングを始めとする様々な機会において多くの研究の助言、及び学生生活のサポートして頂きました。

また石野雅也氏<sup>c</sup>、佐々木修氏<sup>c</sup>、戸本誠氏<sup>d</sup>を始めとする ATLAS TGC 日本グループの皆様には、TGC ミーティングを始めとする場で研究に関する助言を多く頂きました。

神戸大学粒子物理学研究室の岡田勝吾氏<sup>a</sup>、早川俊氏<sup>a</sup>、Matthew King 氏<sup>a</sup>、西山知徳氏<sup>a</sup>、秋山邦裕氏<sup>a</sup>、谷和俊氏<sup>a</sup>、岸本巴氏<sup>a</sup>、鈴木雄太氏<sup>a</sup>にはソフトウェアの開発、解析の手法を始めとする研究におけるサポート、助言は勿論のこと、私生活でもアシストしていただき楽しい研究生生活を送ることができました。

奥村恭幸氏<sup>d</sup>、長谷川慧氏<sup>d</sup>、鈴木友氏<sup>c</sup>、結束晃平氏<sup>b</sup>、東裕也氏<sup>b</sup>、道前武氏<sup>e</sup>、越前谷陽佑氏<sup>b</sup>、二ノ宮陽一氏<sup>b</sup>、神谷隆之氏<sup>b</sup>を始めとする ATLAS TGC グループ及び ATLAS 日本グループの皆様には、研究のサポートや助言、そして CERN 出張の折には研究の面だけでなく生活面でもアシストしていただきました。

また宮崎一樹氏<sup>a</sup>、小西拓也氏<sup>a</sup>、大塚康平氏<sup>a</sup>、駒井英俊氏<sup>a</sup>を始めとする神戸大学粒子物理学研究室の皆様のお陰で、日頃の研究室生活が大変楽しく、有意義なものになりました。

粒子物理学研究室秘書の横山有美氏<sup>a</sup>、井上奈保氏<sup>a</sup>には、出張の折の手続きを始めとする様々な面でお世話になりました。

皆様には深く感謝しております。ありがとうございます。

物理学専攻の物理教室秘書の竹田修子氏<sup>a</sup>には、6年間にわたり、公私のあらゆる面でサポートして頂きました。本当に感謝しております。

そして、私が大学入学後、私生活、学業、研究で支えてくれた素晴らしい親友の、飯田

---

直幸氏、岡村航氏、出倉卓氏、徳永香氏、藤兼典史氏には本当に感謝しています。つまづいたときに背中を押してもらったこともありました。皆様のお陰で大変充実した学生生活となりました。最高の友人にめぐりあえたことを誇りに思っています。

最後に私のこれまでの生活、学業を始めとする全ての面において支え続けて頂いた父、母、祖母、姉に深く感謝しております。これからは少しずつ、私がこれまで受けてきた以上の恩を返し、支えることができる存在になりたいと思っています。この場で感謝の意を述べたいです。本当にありがとうございました。

所属:

神戸大学 理学研究科 (自然科学研究科)<sup>a</sup>

東京大学 素粒子物理国際研究センター (ICEPP)<sup>b</sup>

高エネルギー加速器研究機構 (KEK)<sup>c</sup>

名古屋大学 理学研究科<sup>d</sup>

東京大学 理学系研究科<sup>e</sup>