

修士学位論文

ATLAS 前後方ミュー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

東京大学大学院

理学系研究科 物理学専攻

素粒子国際研究センター 坂本研究室

越前谷 陽佑

2010 年 2 月 4 日

概 要

2009 年 11 月にスイスにある欧州原子核研究機構 (CERN) で建設が進んでいた大型ハドロン衝突型加速器 (LHC) は再稼働を始め、12 月には世界最高エネルギー (重心系で 2.36 TeV) を記録した。大型汎用検出器の一つである ATLAS 検出器はビーム衝突に向けコミッショニングを進めてきた。中でも私は ATLAS 前後方ミューオントリガーシステムを担当する TGC グループに参加し、トリガータイミングの最終調整に用いるモジュールの開発、システム構築、動作検証に関わってきた。またそれと同時に我々はアップグレードに向けたデータ読み出し系の研究開発も進めている。

本論文では ATLAS 前後方ミューオントリガーシステムの詳細、開発と構築について、動作検証の結果、アップグレードに向けた研究開発の現状について論じる。

目次

第 1 章	序論	13
1.1	LHC	14
1.2	4 つの実験	15
1.2.1	ATLAS 実験	16
1.2.2	CMS 実験	17
1.2.3	ALICE 実験	18
1.2.4	LHCb 実験	19
1.3	ATLAS で期待されている物理	19
1.3.1	ハドロンコライダーでの素過程	19
1.3.2	Higgs 粒子の探索	21
1.3.3	Higgs 粒子の崩壊過程	22
1.3.4	ATLAS 実験における Higgs 粒子の発見能力	23
1.4	超対称性粒子の探索	23
1.5	ATLAS 実験におけるミューオントリガーの重要性	25
第 2 章	ATLAS 検出器について	27
2.1	ATLAS 検出器の構成	27
2.2	Tracking	28
2.2.1	Pixel	28
2.2.2	SCT(Semi-Conductor Tracker)	29
2.2.3	TRT(Transition Radiation Tracker)	29
2.3	Calorimetry	31
2.3.1	LAr electromagnetic calorimeter	31
2.3.2	Hadronic calorimeters	32
2.4	Magnets	34
2.5	Muon Spectrometer	35
2.5.1	MDT(Monitored Drift Tube chambers)	35
2.5.2	CSC(Cathode Strip Chambers)	36
2.5.3	RPC(Resistive Plate Chambers)	39
2.5.4	TGC(Thin Gap Chambers)	39
2.6	Trigger and DAQ	41
2.6.1	Level-1 Trigger	41

2.6.2	Timing Trigger and Control system	42
2.6.3	Level-2 Trigger	43
2.6.4	EF(Event Filter)	44
2.7	データ解析	44
第 3 章	ATLAS 前後方ミューオントリガーシステム	46
3.1	Thin Gap Chamber(TGC) の構造と動作原理	46
3.1.1	基本構造	46
3.1.2	動作原理	47
3.1.3	チェンバーの構造	48
3.2	TGC の配置	49
3.2.1	チャンネルのオーバーラップ	50
3.3	TGC における Pt 測定の原理	52
3.4	TGC のエレクトロニクス	54
3.4.1	トリガー系	55
3.4.2	リードアウト系	55
3.4.3	コントロール系	56
3.5	エレクトロニクスの詳細	57
3.5.1	ASD(Amplifier Shaper Discriminator)	57
3.5.2	PS Board	57
3.5.3	SPP(Service Patch Panel)	61
3.5.4	PS Pack	62
3.5.5	HPT(High- p_T Board)	62
3.5.6	SL(Sector Logic)	63
3.5.7	SSW	63
3.5.8	ROD(Read Out Driver)	64
3.5.9	CCI(Control Configure Interface)	65
3.5.10	HSC(HPT SSW Controller)	65
第 4 章	TGC システムのコミッショニング	68
4.1	タイミング調整の重要性	68
4.2	TGC システムのタイミング調整	68
4.2.1	タイミング情報がどのように分配されるか	68
4.2.2	TGC システム内でのタイミング調整	68
4.3	2008 年時点でのタイミング調整の結果	73
4.4	Clock Phase Scan に向けて	75
4.4.1	Clock Phase Scan とは	75
4.4.2	VME Variable Logic Delay Module の製作	76
4.4.3	VME Variable Logic Delay の精密測定	81
4.4.4	Delay Module のインストール	88

4.4.5	インストール後のテスト	88
4.4.6	Delay Module 用 Online Software の開発	91
4.4.7	Clock Phase Scan の準備状況	96
4.5	TGC オフラインソフトウェアの開発	97
4.5.1	オフラインソフトウェア	97
4.5.2	TGC オフラインデコードソフトウェアの概要	98
4.5.3	TrigMuonCoinHierarchy の開発	98
4.6	まとめ	110
第 5 章	LHC アップグレードに向けた TGC 読み出し回路の研究開発	111
5.1	Super LHC 計画	111
5.1.1	物理的なモチベーション	111
5.1.2	加速器のシナリオ	111
5.1.3	ATLAS MuonSpectrometer のアップグレード	115
5.2	TGC 読み出し系のアップグレード	116
5.3	新 ROD のデザイン	119
5.4	CPU コアの評価	120
5.4.1	StandardIO の出力の確認	122
5.4.2	IO の操作	124
5.4.3	性能の評価	126
5.4.4	外部メモリの操作	131
5.4.5	MicroBlaze のリソースの使用量	134
5.5	TGC 読み出し回路の研究開発のまとめ	136
第 6 章	まとめと今後について	137
付 録 A	Delay Module の回路図	138
付 録 B	PS Board と SSW の接続	145
付 録 C	SLB データフォーマット	149
付 録 D	bitpos	150
付 録 E	放射線環境下に耐えうる VME コントローラと PCI インタフェースの開発研究	152
E.1	序論	152
E.2	ATLAS 前後方ミューオントリガーのコントロールシステムの構成要素	154
E.2.1	HSC	154
E.2.2	CCI	155
E.2.3	G-Link	156
E.3	開発計画	156

E.3.1	RTC	156
E.3.2	PCI-CCI	156
E.4	開発状況	158

目 次

1.1	Tevatron によって棄却された Higgs の質量領域 [7]	13
1.2	LHC の俯瞰図 [11]	14
1.3	加速器の構成 [16]	15
1.4	LHC トンネル中の超伝導双極電磁石 [11]	15
1.5	超伝導双極電磁石の断面図 [11]	15
1.6	LHC と 4 つの衝突点 [11]	16
1.7	ATLAS Cavern [11]	17
1.8	ATLAS 検出器 [11]	17
1.9	CMS 検出器 [8]	17
1.10	CMS Cavern [3]	18
1.11	CMS 検出器の断面図 [12]	18
1.12	ALICE Cavern [3]	18
1.13	ALICE 検出器 [8]	18
1.14	LHCb Cavern [3]	19
1.15	LHCb 検出器 [3]	19
1.16	ハドロンコライダーでの素過程 [27]	19
1.17	$Q^2 = 20 \text{ GeV}$ でのパートンの確率分布関数	20
1.18	Higgs 粒子の主な生成過程	21
1.19	標準模型における Higgs 質量と 生成断面積の関係 [20]	23
1.20	標準模型における Higgs 質量と 崩壊分岐比の関係 [20]	23
1.21	積分ルミノシティ 10 fb^{-1} における ATLAS の Higgs 発見能力 [20]	24
1.22	積分ルミノシティ 30 fb^{-1} における ATLAS の Higgs 発見能力 [20]	24
1.23	3 つの力の強さのスケール依存性 [28]	24
1.24	超対称性粒子の発見能力 [28]	25
1.25	LHC での主要物理過程の生成断面積 [27]	26
2.1	ATLAS 検出器の全体像 [11]	27
2.2	Inner Detector [9]	28

2.3	Overview of Pixcel Detector [9]	28
2.4	Pixcel Module [9]	28
2.5	The Cross Section of SCT [9]	29
2.6	The Picture of Barrel SCT [9]	29
2.7	Barrel Module [9]	29
2.8	Endcap Module [9]	29
2.9	The Picture of Endcap TRT [3]	30
2.10	The Picture of Barrel TRT [3]	30
2.11	The Straw Structure of TRT [9]	30
2.12	ATLAS calorimeter system [9]	31
2.13	The Sketch of a barrel module [9]	32
2.14	The picture of a partly stacked barrel LAr module [9]	32
2.15	The picture of a side view of end-cap LAr module [9]	32
2.16	Schematic view of Tile calorimeter [9]	33
2.17	Schematic view of a HEC module [9]	33
2.18	ATLAS のマグネットの中のコイルのイメージ [9]	34
2.19	η の関数でみた積分磁場強度 [9]	34
2.20	Central solenoid magnet [9]	34
2.21	Barrel toroid [9]	34
2.22	Endcap Toroid [9]	34
2.23	ATLAS muon system [9]	35
2.24	Cross-section of the barrel muon system [9]	36
2.25	Cross-section of the muon system in y-z plane [9]	36
2.26	MDT Chamber の構造 [9]	36
2.27	Cross-section of a MDT tube [9]	36
2.28	Longitudinal cut through a MDT tube [9]	36
2.29	Layout of a CSC [9]	37
2.30	Structure of the CSC [9]	38
2.31	Structure of the CSC cell [9]	38
2.32	Charge distribution on the CSC cathode [9]	38
2.33	Readout strips of CSC [9]	38
2.34	Cross-section thorough the upper part of the barrel with the RPCs	39
2.35	Structure of the RPC	40
2.36	Trigger and DAQ diagram	41
2.37	L1 Trigger block diagram	42

2.38	TTC システムによるタイミングシグナルの分配の概要 [9]	43
2.39	クラウドモデル [26]	44
2.40	ICEPP にある Tier2 Tokyo cite [26]	45
2.41	Tokyo Tier2 のテープストレージ [26]	45
3.1	TGC の構造 (断面図)[9]	46
3.2	TGC の主要な運転パラメータ [9]	46
3.3	タウンゼントなだれのイメージ [15]	47
3.4	TGC の平面図 [30]	48
3.5	TGC のワイヤーサポート [30]	48
3.6	TGC のトリプレットチェンバー (左) とダブルットチェンバー (右) [9]	48
3.7	TGC のワイヤーチャンネルのスタグリングのイメージ図	49
3.8	R-Z 平面でみた TGC のレイアウト [9]	50
3.9	TGC3 のチェンバー配置 [30]	50
3.10	EIFI のチェンバー配置 [30]	50
3.11	トリガーセクターとサブセクター [13]	51
3.12	Wire のオーバーラップ [31]	51
3.13	TGC3 での Strip のオーバーラップ [13]	52
3.14	TGC2、TGC3 におけるストリップシグナルの OR とコインシデンス [13]	52
3.15	Pt 測定の原理	53
3.16	TGC エレクトロニクスシステムの全体像 [30]	54
3.17	トリガーラインのモジュール構成 [30]	55
3.18	リードアウトラインのモジュール構成 [30]	55
3.19	コントロールラインのモジュール構成 [30]	56
3.20	ASD の写真 [30]	57
3.21	ASD ASIC のダイアグラム [6]	57
3.22	PS Board [30]	58
3.23	PS Board のダイアグラム [30]	58
3.24	PP ASIC のブロックダイアグラム [30]	59
3.25	SLB ASIC のブロックダイアグラム [30]	60
3.26	デクラスタリングの概念図 [13]	60
3.27	JRC 内部の構造 [30]	61
3.28	DCS board の写真 [30]	61
3.29	SPP の写真 [30]	62
3.30	PS Pack の写真 [30]	62
3.31	HPT の写真 [30]	62
3.32	SL の写真 [30]	62
3.33	SL のブロックダイアグラム [13]	64
3.34	SSW の写真 [30]	65

3.35	ROD の写真 [30]	65
3.36	SSW のブロックダイアグラム [17]	66
3.37	SSW のゼロサプレスのスキーム [17]	66
3.38	CCI の写真 [30]	67
3.39	HSC の写真 [30]	67
4.1	TOF [29]	69
4.2	シグナルケーブル [29]	69
4.3	ASD テストパルスの生成と信号受信の概略図 [10]	70
4.4	2.8 m と 48.1 m のケーブルで観測された信号の減衰 [10]	70
4.5	ASD のタイミング分布 [10]	71
4.6	TGC システムに実装されている遅延回路 [29]	72
4.7	HSC クレートと USA15 の位置関係 [31]	72
4.8	HSC クレートと USA15 を結ぶファイバーの長さ [31]	72
4.9	ACR でのシフト時の様子	73
4.10	2008 年 First Beam でのトリガータイミング [1]	74
4.11	シングルビームでの TOF [29]	74
4.12	Clock Phase Scan のシュミレーション [10]	75
4.13	Clock の配布を表した模式図	76
4.14	Orcad Capture	77
4.15	VME Slave モジュールのブロックダイアグラム [22]	78
4.16	ISE による Behavioral Simulation	79
4.17	VME Variable Logic Delay	80
4.18	VME Variable Logic Delay のフロントパネル	80
4.19	Delay Module の動作原理	80
4.20	Delay Module の測定風景 1	81
4.21	Delay Module の測定風景 2	81
4.22	Delay Module の各 Step 測定の際のセットアップ図	82
4.23	測定した 4 枚のボードの 1 Channel 目の結果	84
4.24	4.23 図の拡大 1	84
4.25	4.23 図の拡大 2	84
4.26	4.23 図の拡大 3	84
4.27	温度特性測定の様子	85
4.28	NIM Fan-out モジュール	85
4.29	温度特性測定の設定アップ図	85
4.30	測定開始直後 (26.0 °C) の波形	86
4.31	測定終了時 (56.0 °C) の波形	86
4.32	Delay Module の回路の一部	86
4.33	TTC クレートのコネクション図	89

4.34	ECL-NIM LEVEL CONVERTER	90
4.35	インストール前の A-side TTC クレート	90
4.36	インストール後の A-side TTC クレート	90
4.37	0ns Delay	91
4.38	4ns Delay	91
4.39	8ns Delay	91
4.40	12ns Delay	91
4.41	Offline Software のコンポーネント	92
4.42	ステートマシンのダイアグラム	93
4.43	TGCDelayModule の Segments&Resources	94
4.44	TGC Onlins Software の Barrier 機能	95
4.45	Delay Module ソフトウェアの TDAQ パネル	95
4.46	IS Monitor	96
4.47	ATLAS 検出器のトリガーと DAQ のブロックダイアグラム [9]	97
4.48	TGC のトリガーの階層構造	99
4.49	TrigMuonCoinHierarchy の Class の階層図 [21]	101
4.50	TgcCoinHierarchyFindTool のアルゴリズム	102
4.51	SLB が出力する 200bit のデータ	103
4.52	SLB が出力する 200bit のデータ中の Hit 情報	103
4.53	Triplet Wire SLB の bitpos	104
4.54	Triplet Wire の SLB Channel	104
4.55	Wire の場合の TGC3 から TGC1 へのプロジェクション	105
4.56	Triplet Strip SLB の bitpos	105
4.57	Triplet Strip の SLB Channel	105
4.58	TrigMuonCoinHierarchy-00-00-10-07 の結果	106
4.59	TrigMuonCoinHierarchy-00-00-19 の結果	106
4.60	Trigger Sector 別にみた対応関係の結果 (TrigMuonCoinHierarchy-00-00-10-07)	107
4.61	Trigger Sector 別にみた対応関係の結果 (TrigMuonCoinHierarchy-00-00-19)	107
4.62	run137898 の対応関係の結果	108
4.63	run137898 のトリガーセクター別に見た対応関係の結果	108
4.64	モンテカルロで見た対応関係の結果	109
4.65	モンテカルロデータをトリガーセクター別に見たときの対応関係の結果	109
5.1	IR のアップグレードプラン [25]	113
5.2	Peak Luminosity [25]	114
5.3	Integrate Luminosity [25]	114
5.4	IP における (a)head-on 衝突と (b)crab crossing を持った衝突 [16]	115
5.5	様々なバックグラウンド環境下でみた MDT の分解能	116
5.6	(a)	116

5.7 (b)	116
5.8 $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ で期待されている inclusive Level-1 rate	117
5.9 現在の TGC リードアウトシステム	117
5.10 Triplet PSB と SSW のコネクション図	118
5.11 新しい TGC 読み出し系の案	119
5.12 新 ROD のデザイン	120
5.13 FPGA 内部のブロックダイアグラム	120
5.14 Spartan 3E Starter Kit	121
5.15 Base System Builder(BSB)	122
5.16 Board Selection	122
5.17 Peripheral Configuration	122
5.18 Address Generation	122
5.19 MicroBlaze のブロックダイアグラム	123
5.20 開発風景	123
5.21 TeraTerm	123
5.22 tutorial.c の出力	124
5.23 led_ctrl.c の出力	126
5.24 (a)	126
5.25 (b)	126
5.26 簡単な四則演算の計測結果	127
5.27 forloops.c の出力結果	128
5.28 一般的な CPU(Pentium D) での実行結果	129
5.29 for loop の回数を変えて実行した結果	130
5.30 PT5 のブロックダイアグラム	133
5.31 MemoryTest.c の実行結果	133
5.32 DPM アクセスの実行結果	133
5.33 DDR SDRAM へのアクセス時間の計測	134
5.34 Synthesis Summary	134
5.35 Device Utilization Summary	135
A.1 Delay Module の回路図 (1)	138
A.2 Delay Module の回路図 (2)	139
A.3 Delay Module の回路図 (3)	140
A.4 Delay Module の回路図 (4)	141
A.5 Delay Module の回路図 (5)	142
A.6 Delay Module の回路図 (6)	143
A.7 Delay Module の回路図 (7)	144
B.1 Doublet PSB と SSW の接続 (1) [14]	145
B.2 Doublet PSB と SSW の接続 (2) [14]	146

B.3	Triplet PSB と SSW の接続 (1) [14]	147
B.4	Triplet PSB と SSW の接続 (2) [14]	148
C.1	SLB のデータフォーマット [30]	149
D.1	Triplet Wire の bitpos	150
D.2	Triplet Strip の bitpos	151
E.1	CCI と HSC	152
E.2	CCI と HSC を用いた ATLAS 前後方ミューオントリガーのコントロールシステム	153
E.3	HSC のブロックダイアグラム	154
E.4	CCI のブロックダイアグラム	155
E.5	PCI 評価ボード	157
E.6	WinDriver のスクリーンショット	157
E.7	PCI-CCI の写真	158
E.8	RTC の写真	158

表 目 次

1.1	LHC parameters	16
2.1	Main MDT chamber parameters [9]	37
2.2	Main CSC chamber parameters [9]	39
2.3	RPC parameters [9]	40
4.1	各モジュールに実装されている遅延回路のステップと用途	71
4.2	2008 年 First Beam 時のトリガー条件	73
4.3	Delay Step の測定結果	83
4.4	MC100EP16DG と MC100LVEL58DG の Propagation Delay	87
4.5	Run124887 の時点でのトリガー条件	106
5.1	Main parameters of the proposed schema for the IRs	114
5.2	MicroBlaze と Pentium D での実行結果	129
5.3	for loop の回数と処理にかかったクロック数の関係	130
5.4	Spartan-6 FPGA Feature Summary (抜粋) [24]	135

第1章 序論

素粒子の研究は 1960-1970 年代に飛躍的に躍進し、強い相互作用、電磁相互作用、弱い相互作用を正確に記述する標準模型が急速に発展した。主にそれは加速器実験によって理論が実証される形で牽引されてきた。1989 年に始動した CERN(Conseil European pour la Recherche Nucleaire、欧州原子核研究機構)における LEP(Large Electron-Positron Collider)では Z^0 粒子の精密測定や素粒子の世代数の決定がなされ、アメリカ合衆国のフェルミ研究所で 1983 年に始動した Tevatron 加速器では Top クォークの発見がなされた。しかし、素粒子の標準模型において唯一見つかっていないのが、素粒子の質量起源を説明する Higgs 粒子である。LEP と Tevatron によってもその存在は確認されておらず、LEP によって 114.4 GeV より軽い領域が、Tevatron によって 160 GeV と 170 GeV の間のある領域が exclude されたのみである (1.1 図参照)。

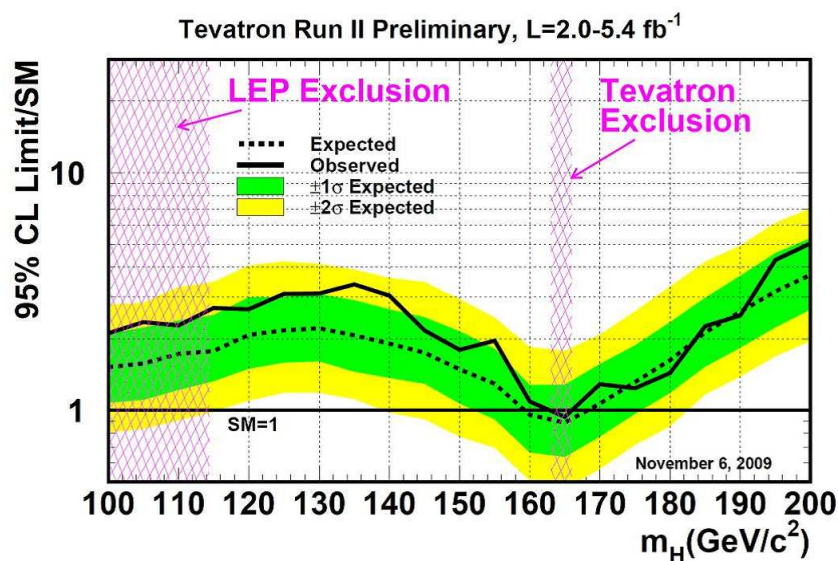


図 1.1: Tevatron によって棄却された Higgs の質量領域 [7]

1980 年代には Higgs 粒子の発見を目指して TeV エネルギーの超大型加速器の計画が立ち上がった。SSC(Superconducting Super Collider)計画と LHC(Large Hadron Collider)計画である。しかし、SSC 計画は建設半ばで米国の財政難により中止が余儀なくされた。一方 LHC 計画は世界的なプロジェクトとして、各国の協力のもとに完成し、2009 年に安定稼働に入った。

私は LHC 計画の中の ATLAS 実験に参加し、TGC 日本グループの一員として研究に携わった。本

論文では TGC のコミッショニングと LHC のアップグレードに向けた TGC 読み出し回路の研究開発について述べる。まずこの章では LHC の概要と 4 つの実験、ATLAS で期待されている物理についてまとめる。

1.1 LHC

LHC はジュネーブ郊外のスイスとフランスの国境に位置する CERN に建設された大型ハドロン衝突型加速器である。LHC は 2000 年まで稼働していた LEP のトンネルを利用して、周長 26.7km ほどの大きさがある。LHC の建設にあたり、LEP トンネル内のマグネットは超伝導磁石に置き換えられた。1.2 図は上空から LHC を俯瞰した写真である。写真中央よりやや左上あたりにレマン湖が見え、またそのさらに奥にはアルプス山脈が見える。写真の中で赤色の円で示されているのが、LHC のリングであり、スイスとフランスの国境を跨いでいる。



図 1.2: LHC の俯瞰図 [11]

LHC 内では、デザイン上の値で、 10^{11} 個の陽子が 40 MHz の頻度で衝突し、それは重心系のエネルギーにして 14 TeV に至る。また、デザインルミノシティは $10^{34} \text{ cm}^{-2}\text{s}^{-1}$ である。重イオン (なかでも鉛イオン) の衝突も計画されていて、5.5 TeV のエネルギーを持つ原子核がルミノシティ $10^{27} \text{ cm}^{-2}\text{s}^{-1}$ で衝突する。

1.3 図は LHC に入射されるまでの加速器の構成を表している。Linac から入射された陽子は PS にて 26 GeV まで加速された後に、SPS によって 450 GeV まで加速され、LHC リングに入射する。LHC リングには超伝導双極磁石が敷き詰められている。ニオブチタン線を超流動ヘリウムの温度である 1.9 K まで冷却して 8.3 T の磁場を作り出している。1.4 図が超伝導双極磁石が位置する LHC のトンネルの様子で、1.5 図がその断面図である。一つの鉄ヨークの中に 2 セットの超伝導コイルとビームパイプが入っており、逆方向に回るビームも同時に扱うことができる。表 1.1 に LHC の主要なパラメータを載せる。

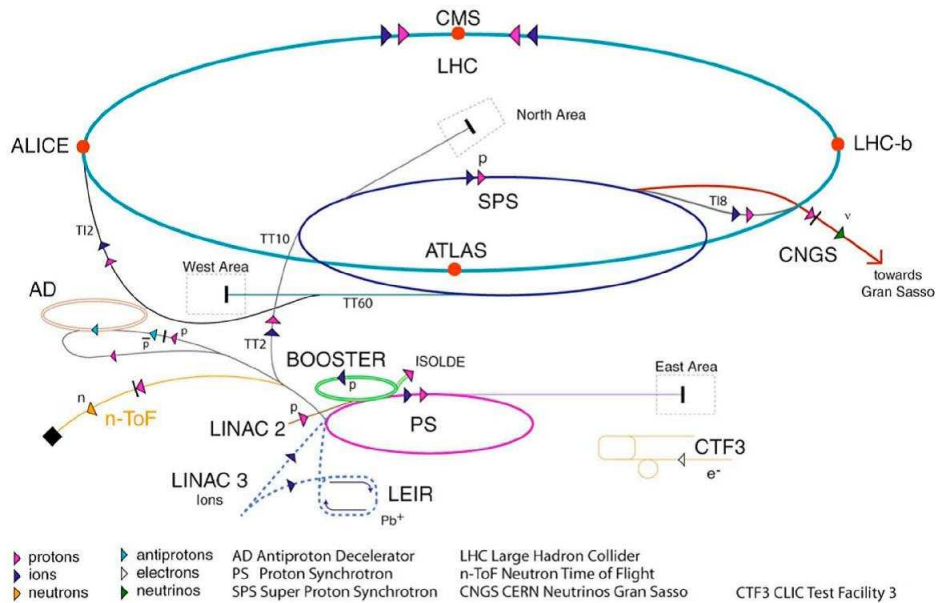


図 1.3: 加速器の構成 [16]

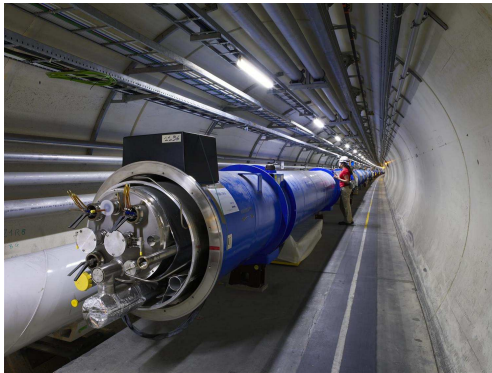


図 1.4: LHC トンネル中の超伝導双極電磁石 [11]

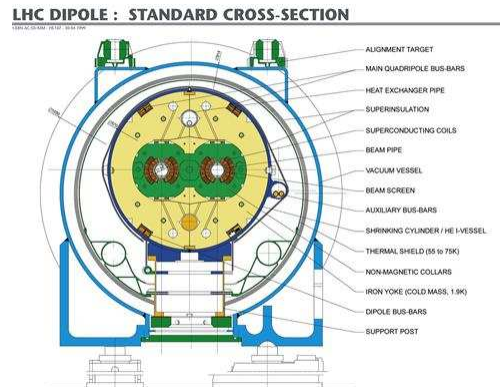


図 1.5: 超伝導双極電磁石の断面図 [11]

1.2 4つの実験

1.6 図は LHC リングと地下の実験ホールを描いたイメージ図である。図にあるように LHC では、4 つの実験ホール内に ATLAS、CMS、ALICE、LHCb の 4 台の検出器が建設されている。その他にも CMS の衝突点の近傍に建設された検出器で行う TOTEM(TOTAL Elastic and diffractive cross section Measurement) 実験と ATLAS の検出器の前方約 140m に設置される小型の検出器を用いて行われる LHCf(Large Hadron Collider forward) 実験がある。以下では主要な 4 つの実験について簡単に解説する。

表 1.1: LHC parameters

Parameter	Design value
Luminosity	$10^{34} \text{ cm}^{-2} \text{ s}^{-1}$
Number of bunches	2808
Bunch spacing	24.95 ns
Number of protons per bunch	1.15×10^{11}
Beam Current	0.58 A
Transverse emittance	$3.75 \mu\text{m}$
Longitudinal emittance	2.5 eVs
Crossing Angle	$285 \mu\text{rad}$

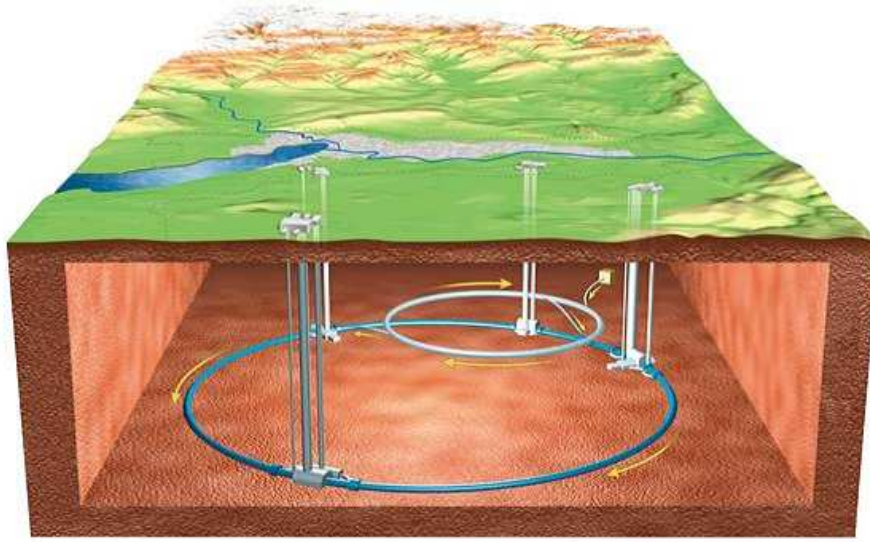


図 1.6: LHC と 4 つの衝突点 [11]

1.2.1 ATLAS 実験

ATLAS 実験には日本の研究者が集中して参加している。ATLAS は A Toroidal LHC ApparatuS の略である。名前にある通り、巨大な超伝導トロイダルマグネットがこの検出器を特徴付けている。検出器は円筒形の形で直径が 44m、直径 25m、重量は 7,000 トンに達する。ATLAS 検出器は Higgs の探索を主目的とする汎用検出器である。次章で ATLAS 検出器については詳しく説明する。1.7 図が ATLAS 検出器のある実験ホールの写真、1.8 図が ATLAS 検出器の全体像である。

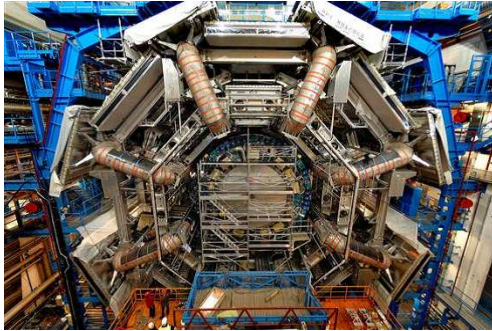


図 1.7: ATLAS Cavern [11]

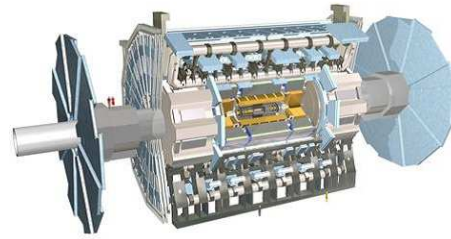


図 1.8: ATLAS 検出器 [11]

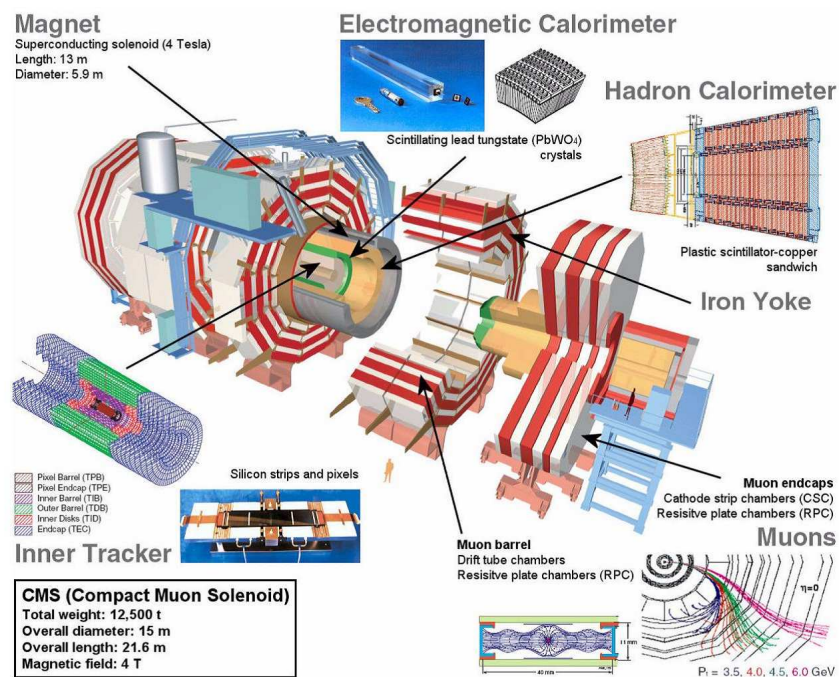


図 1.9: CMS 検出器 [8]

1.2.2 CMS 実験

CMS(Compact Muon Solenoid) も ATLAS 実験と同様 Higgs 粒子の探索を主目的とする汎用検出器である。世界最大のソレノイド型超伝導マグネットが作る 4 T の磁場を用いて荷電粒子の運動量を測る一方、外側の鉄ヨークのリターン磁場を使って再度ミューオンの運動量が測定される。検出器は、長さ 21m、直径 15m の円筒形で、重量は 12,500 トンと ATLAS よりも重い。電磁カロリメーターには 75848 本の PbWO_4 の結晶が使われている。この結晶は 10 MRad までの耐放射線があり、かつ高

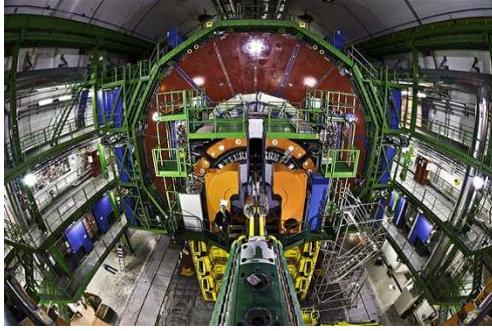


図 1.10: CMS Cavern [3]

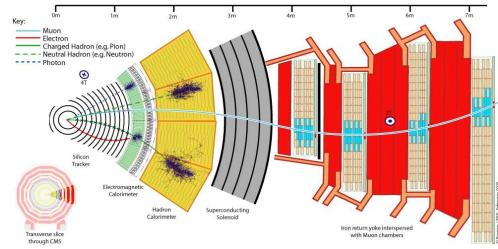


図 1.11: CMS 検出器の断面図 [12]

いエネルギー精度 $\Delta E/E = 3\%/\sqrt{E} [\text{GeV}]$ をもつため、Higgs 粒子の崩壊の一つである $H \rightarrow \gamma\gamma$ の探索に有利である。1.9 図は CMS 検出器の全体像で、1.11 図は検出器の断面図を描いたものである。1.10 図は CMS 検出器の置かれている実験ホールの様子である。

1.2.3 ALICE 実験

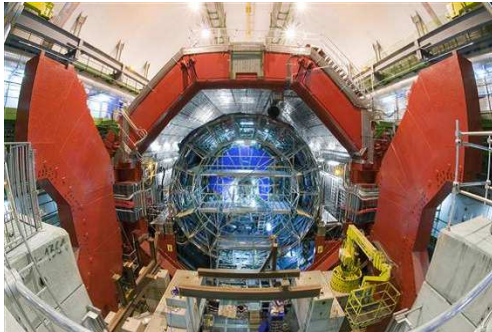


図 1.12: ALICE Cavern [3]



図 1.13: ALICE 検出器 [8]

ALICE(A Large Ion Collider Experiment) 実験は、クォーク・グルーオンプラズマ (QGP) などの超高エネルギー重イオン衝突現象の研究を行う。LHC 加速器では鉛イオン ($^{208}\text{Pb}^{82+}$) を核子あたり 2.76 TeV まで加速することができ、全重心エネルギーは 1.15 PeV に達する。これは米国のブルックヘブン国立研究所の RHIC コライダーの 28 倍のエネルギーに相当する。検出器の大きさは長さ 26m、直径が 16m、重量 10,000 トンである。1.13 図が ALICE 検出器の全体像で、1.12 図が ALICE 検出器が置かれる実験ホールの写真である。



図 1.14: LHCb Cavern [3]

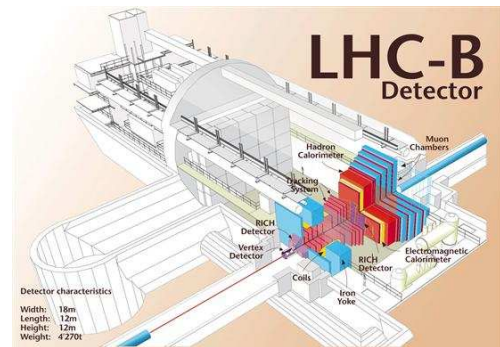


図 1.15: LHCb 検出器 [3]

1.2.4 LHCb 実験

LHCb(Large Hadron Collider beauty) 実験は b クォークを含む B ハドロン粒子の CP 非保存などの稀崩壊現象を高統計で観測し、間接的に標準モデルを超える新しい物理の影響を検証する実験である。検出器は長さ 21m、高さ 10m、幅 13m、重量 5,600 トンである。従来の衝突点より、11.25 m 離れた位置に衝突点が位置している。また、検出器は前方方向に特化し、可動式のトラックカーを有している。1.15 図が LHCb 検出器の全体像で、1.14 図が LHCb が置かれている実験ホールの様子である。

1.3 ATLAS で期待されている物理

先に述べたように LHC は Higgs 粒子の探索を主な目的としている。ATLAS 実験においても Higgs 粒子の探索を行う。またそれに加え、超対称性 (Supersymmetry=SUSY) 粒子の探索やミニブラックホールの生成など標準理論を超えた物理の研究も期待されている。

1.3.1 ハドロンコライダーでの素過程

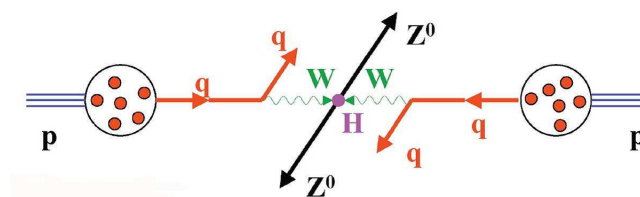


図 1.16: ハドロンコライダーでの素過程 [27]

陽子を構成するパートンが反応に寄与する。他のパートンはほとんどがビームパイプ方向に逃げる。この図は 1.3.2 図の Higgs 粒子生成過程を模式的に描いたものである。

LHCは陽子と陽子を重心エネルギー 14 TeV で衝突させるハドロンコライダーである。ハドロンコライダーでは、1.16 図のように”パートン”と呼ばれる陽子を構成しているクォークやグルーオン、反クォークが反応を行う。それぞれのパートンが担っている運動量の割合を x_1, x_2 とすると、実際の衝突エネルギーは $\sqrt{x_1 x_2} \times 14 \text{ TeV}$ であり、事象ごとに異なる。さらに、この系は $(x_1 - x_2) \times 7 \text{ TeV}$ でビーム軸方向に運動しているため、ビーム軸方向の保存則は使えない。 $x_1 \sim x_2$ のときは生成された粒子が中心に集まった事象となるが、 $x_1 \gg x_2$ または $x_2 \gg x_1$ のようなアンバランスな事象では、前後方にブーストしている。ハドロンコライダーの生成断面積は、素過程の断面積に各パートンの確立分布関数 (PDF) をかけて積分したものとなる。1.17 図が陽子中の各パートンの確率分布関数のグラフである。

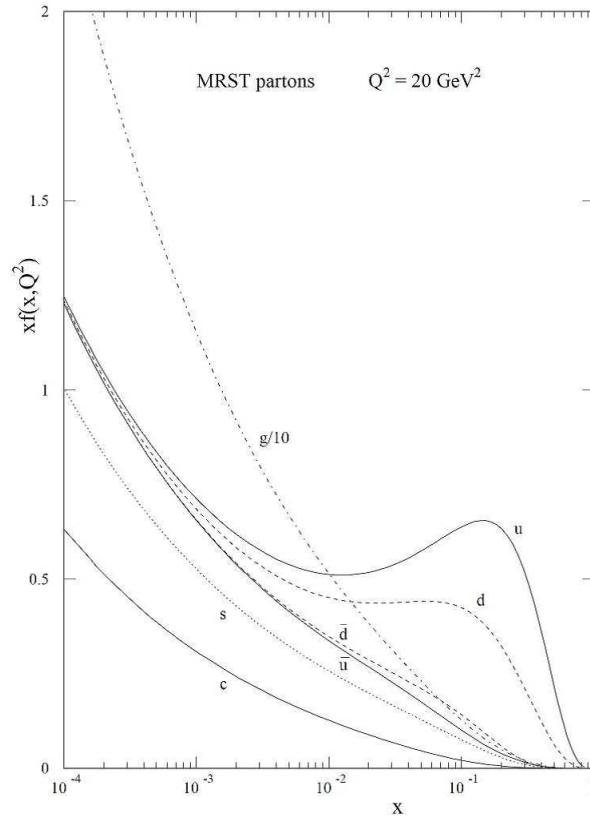


図 1.17: $Q^2 = 20 \text{ GeV}^2$ でのパートンの確率分布関数

例えば、質量 1 TeV 程度の粒子を作ろうとすると、 $\sqrt{x_1 x_2} > O(0.1)$ のような大きな運動量を担うパートンの寄与が主となるのに比べ、100 GeV 程度の粒子では $\sqrt{x_1 x_2} = O(0.01)$ 程度になるようなパートンからの寄与が主となる。1.17 図を見ると、 $x > O(0.1)$ では電荷を担う価クォーク (u クォークや d クォーク) が主要な寄与になる。 $x < O(0.1)$ では、グルーオンの存在確率が $1/x$ で急激に大きくなり、 $O(0.1)$ でもまだ寄与は大きい。LHC で質量の重い粒子を作ろうと思うと、大きな $\sqrt{x_1 x_2}$ が必要になり、必然的に x_1, x_2 はバランスする。そのため重い粒子を含む事象は中央 (バレル部) で

観測され、比較的軽い粒子では前後方まで広がる。

1.3.2 Higgs 粒子の探索

LHC での Higgs 粒子の主な生成過程は 4 つある。1.18 図がその 4 つのファインマンダイアグラムである。

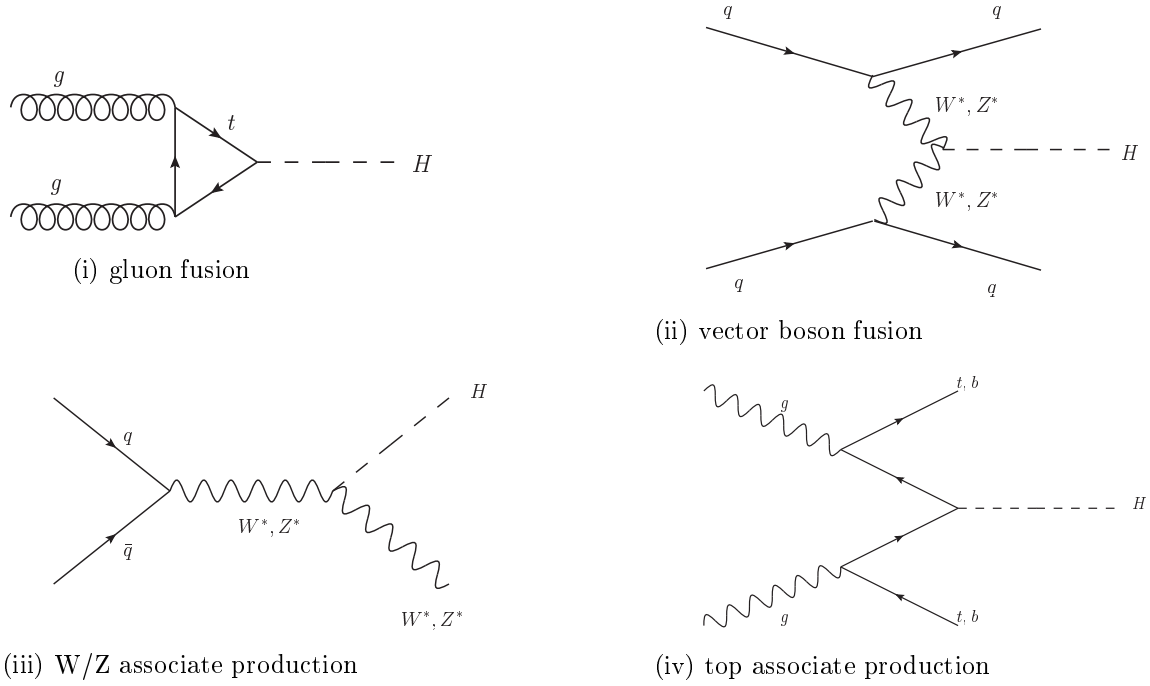


図 1.18: Higgs 粒子の主な生成過程

断面積が大きい順に以下に解説する。

(i) $gg \rightarrow H$ (gluon fusion)

グルーオン同士が反応し、 t クォークのループを介して Higgs を生成する過程である。LHC ではグルーオンの寄与が大きいため、ループを介した過程にもかかわらず、断面積が $O(10)$ pb と大きい。

(ii) $qq \rightarrow qqH$ (VBF: vector boson fusion)

クォークから放出された $W^\pm$ や Z^0 ボソンから Higgs が生成される。LHC では大きな運動量を担うクォーク同士の反応も起こりやすいので、この過程の断面積も $O(1)$ pb と大きい。

また、この過程では $W^\pm/Z^0$ 粒子を放出して反跳したクォークに由来する 2 本のジェットが観測されるという特徴がある。

(iii) $q\bar{q} \rightarrow (W/Z) H$ (W/Z associate production)

Tevatron や LEP などの粒子・反粒子コライダーでは有望な生成過程であったが、粒子・粒子コライダーである LHC ではあまり魅力的ではない。しかし、一緒に生成される $W^\pm/Z^0$ 粒子がレプトンに崩壊した場合には、レプトンをトリガーとして用いることでイベントの選別が可能になる。

(iv) $gg \rightarrow t\bar{t}H$ (top associate production)

対生成された t クォークから Higgs が放出される過程であり、断面積は $O(10)$ fb と小さいが、2 つの t クォークを含む特徴的な事象が観測される。 t クォークの湯川結合の直接測定が可能である。同様に b クォークを随伴する過程もあるが、バックグラウンドとの識別が難しい。

1.19 図に主要な生成過程の生成断面積と Higgs の質量の関係図を載せる。

1.3.3 Higgs 粒子の崩壊過程

1.20 図に示すように、Higgs 粒子の質量によって分岐比と崩壊過程が変わってくる。Higgs 粒子は質量の重い粒子に強く結合するので、第三世代のフェルミ粒子や質量のあるゲージ粒子への結合が大きい。

Higgs の質量が 140 GeV より軽い場合は、 b クォーク対 ($b\bar{b}$)、タウレプトン対 ($\tau^+\tau^-$) が主要な崩壊モードである。また、分岐比は小さいが二光子 ($\gamma\gamma$) への崩壊も重要である。質量が 140 GeV より重くなると、主要な崩壊モードは (W^+W^-) や (Z^0Z^0) となる。

$H \rightarrow b\bar{b}$ への崩壊は低質量 Higgs の場合の主力チャンネルである。しかし、この崩壊によって現れる b クォーク由来のジェット対は QCD バックグラウンドの膨大な 2 ジェットイベントと区別がつかなく、(iii) や (iv) のような生成過程に対してのみ研究がなされていた。生成過程に伴う W 、 Z 、 t が崩壊してできる高 p_T (横方向運動量、transverse momentum) のレプトンをトリガーとして利用可能だが、生成断面積の小ささが崩壊分岐比の大きさと打ち消しあい、イベントレートはそれほど高くはないことも相まって、現在ではこの崩壊モードに対する研究は ATLAS においては断念されている。

$H \rightarrow \tau\tau$ は (ii) の VBF 生成過程との組み合わせで、有望視されているチャンネルである。 τ の崩壊でニュートリノを放出するため、 E_T^{miss} (消失横運動量) の精密な測定が必要とされる。このため電磁カロリメータ、ハドロンカロリメータともに高い性能が要求される。この崩壊過程のときの主要なバックグラウンドは $Z \rightarrow \tau\tau$ のドレル・ヤン過程であるが、この質量ピークは Z の質量 (91 GeV) の位置に集中するので、分離可能である。

$H \rightarrow \gamma\gamma$ は、CMS の電磁カロリメータの分解能が 0.9 GeV、ATLAS の液体アルゴン電磁カロリメータで 1.3 GeV と高分解能なので、連続的に分布するバックグラウンドから信号のシャープなピークが分離可能である。二つの γ 線のみを要求する inclusive 解析では比較的バックグラウンドが大きい、1 本または 2 本の高い p_T を持つジェットを要求する exclusive 解析をすることで発見能力が向上する。Exclusive 解析により、生成過程の中の (i) gluon fusion の割合が減り、(ii) vector boson fusion の割合が増える。さらに高い p_T を持ったジェットを要求することで、バックグラウンドを効率よく抑えることが可能である。以上から 10 fb^{-1} の統計で発見能力が 5σ となる。 $H \rightarrow \gamma\gamma$ と $H \rightarrow \tau\tau$ が Higgs 発見の最初のチャンネルとして期待されている。

Higgs の質量が 140 GeV より重い場合にはベクターボソン対への崩壊が主となる。 Z^0 に崩壊する場合は $H \rightarrow Z^0 Z^0 \rightarrow l^+ l^- l^+ l^-$ と終状態に四つのレプトンが観測される。四つのレプトンの不変質量を計算すると、Higgs の質量の位置に綺麗なピークが観測される。Higgs の質量が 200 GeV より重い場合には Higgs の崩壊幅が直接観測できる。

$H \rightarrow W^+ W^-$ に崩壊し、 $W^\pm$ がレプトンに崩壊した場合は、二つの ν が勝手な方向に出ているため、質量の再構成ができない。そこで二つのレプトンの横運動量と二つのニュートリノに由来する E_T^{miss} から横方向質量を計算する。

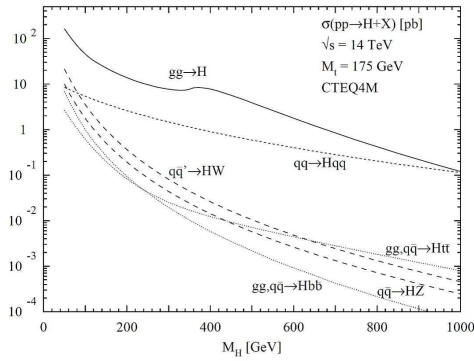


図 1.19: 標準模型における Higgs 質量と生成断面積の関係 [20]

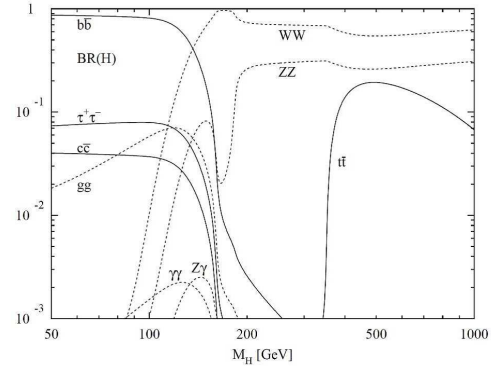


図 1.20: 標準模型における Higgs 質量と崩壊分岐比の関係 [20]

1.3.4 ATLAS 実験における Higgs 粒子の発見能力

1.21、1.22 図はそれぞれ 10 fb^{-1} 、 30 fb^{-1} の積分ルミノシティ時の ATLAS における Higgs の発見能力を示している。 10 fb^{-1} は LHC が実験初期の $10^{33} \text{ cm}^{-2} \text{ s}^{-1}$ のルミノシティで一年稼働して蓄積することができるデータ量に相当し、 30 fb^{-1} はその 3 年分に相当する。図の縦軸はシグナルのイベント数を S 、バックグラウンドのイベント数を B として、 $S/\sqrt{B}$ をとったものである。点線で示されたラインはこの値が 5 のラインを表し、点線を越えたものは発見したと断言することができる。1.22 図より 30 fb^{-1} で発見可能なのは、Higgs の質量が低い領域では $H \rightarrow \tau\tau$ で、 $H \rightarrow ZZ^{(*)}$ 、 $H \rightarrow WW^{(*)}$ などとなっている。

1.4 超対称性粒子の探索

自然界の基本粒子は統計性の違いにより、ボーズ粒子とフェルミ粒子に大別されている。超対称性は、ボーズ粒子とフェルミ粒子とを交換する最も基本的な対称性である。超対称性が存在すると、スピンが $1/2$ 違う超対称性パートナーが観測されるはずであるが、未だ観測されていない。LEP でのゲージ結合定数の精密測定の結果から 1 TeV 程度の質量を持った超対称性粒子が存在すると、力

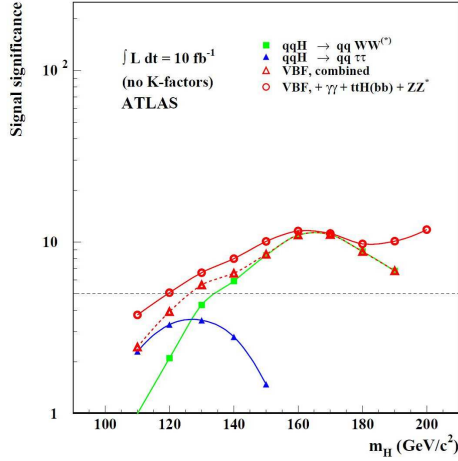


図 1.21: 積分ルミノシティ 10 fb^{-1} における ATLAS の Higgs 発見能力 [20]

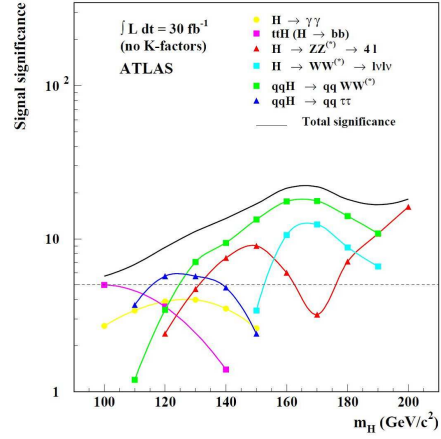


図 1.22: 積分ルミノシティ 30 fb^{-1} における ATLAS の Higgs 発見能力 [20]

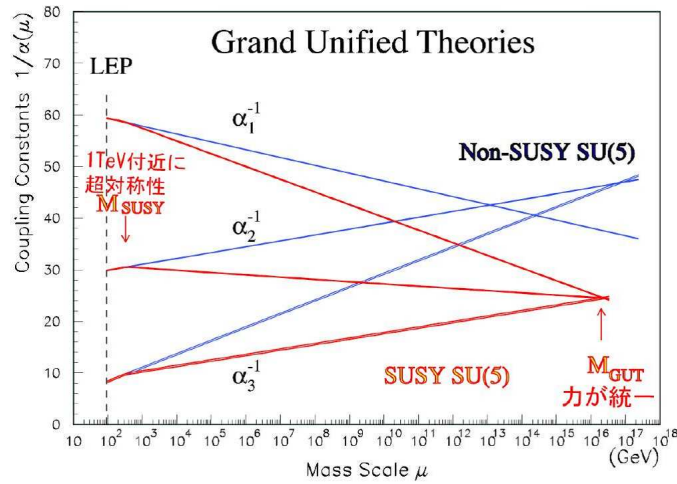


図 1.23: 3つの力の強さのスケール依存性 [28]

の大統一があることが示唆されている (1.23 図参照)。また、R パリティが保存しているとき、一番軽い超対称性粒子が安定で、暗黒物質のよい候補となる。

LHC では、強い相互作用を通して、スカラークォーク (クォークの超対称性パートナー) とグルイーノ (グルーオンのパートナー) がまず生成される。R パリティが保存すると、超対称性粒子は対で生成され、カスケード的に次々と崩壊していく。最終的には一番軽いスーパー粒子 (LSP: Lightest Super-Particle) 二つが検出器と反応せずに通り抜ける。超対称性事象の特徴として、

1. 終状態で逃げる二つの LSP による大きな E_T^{miss}

2. カスケード崩壊の上流で放出される高い p_T を持った複数のジェット
3. 下流で時折放出されるレプトンやハドロンに崩壊した τ を含む

などが挙げられる。超対称性粒子の探索には何よりも E_T^{miss} の理解が重要だが、レプトンを要求することで大部分のバックグラウンドを抑えることができる。

1.24 図に超対称性粒子の発見能力を示す図を載せる。縦軸、横軸は超対称性のモデルの一つであるミニマム重力伝搬 SUSY モデル (mSUGRA) のパラメータである。対応するグルイーノとスカラークォークの質量を等高線で加えてある。これらの質量が 2 TeV 以下の場合、 10 fb^{-1} で 5σ 発見が可能である。

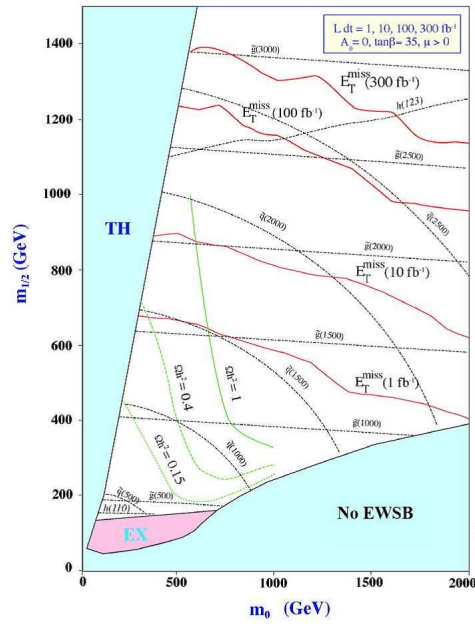


図 1.24: 超対称性粒子の発見能力 [28]

1.5 ATLAS 実験におけるミューオントリガーの重要性

1.25 図は LHC での主要な物理過程の生成断面積の図である。LHC では、陽子・陽子の非弾性散乱の断面積は 80 mb あり、デザインルミノシティで 1 秒間に 10^9 個の非弾性散乱イベントを生じる。また、 t クォークの生成断面積も 840 pb と大きく、 W や Z などを生成する過程も $10 \sim 100 \text{ nb}$ 程度ある。一方、Higgs 粒子や超対称性粒子の生成断面積は 10 pb から 10 fb である。つまり、多くのバックグラウンドの中から効率よく興味のあるイベントを選別する必要がある。Higgs のある種の崩壊過程や超対称性粒子の崩壊過程ではレプトン、とりわけミューオンが重要な役割を果たす。高 p_T ミューオンを要求することで、効果的にバックグラウンドを落とすことが可能である。

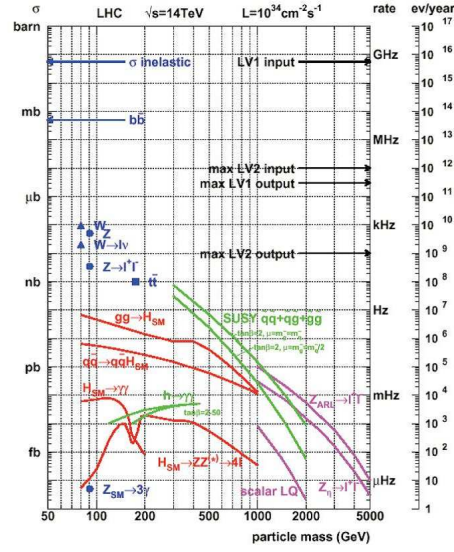


図 1.25: LHC での主要物理過程の生成断面積 [27]
 右側のレートの値はデザインミノシティ
 $(10^{34} \text{ cm}^{-2} \text{ s}^{-1})$ 時のものである。

ATLAS 検出器では Muon Spectrometer がミューオントリガーを担当する。前後方部を担当するのが TGC(Thin Gap Chamber) である。次章では ATLAS の各サブシステムについて詳細に解説する。

第2章 ATLAS 検出器について

2.1 ATLAS 検出器の構成

ATLAS 検出器は LHC での陽子・陽子衝突と核子・核子衝突のために建造された汎用検出器である。ATLAS 検出器の構築にあたっては標準理論 Higgs の探索を基準として、各サブシステムの性能が決められてきた。ATLAS 検出器は内部から、衝突によって生成された粒子の軌跡を再構成し、磁場中での曲がり具合から粒子の p_T や電荷を測定するとともに衝突点の測定も行う (Tracking を行う) 内部飛跡検出器 (Inner Detector, ID)、電磁シャワー、ハドロンシャワーからエネルギーやジェットの角度の測定を行うカロリメータ (Calorimetry)、ミューオントリガーとミューオンの p_T 測定を行うミューオンスペクトロメーター (Muon Spectrometer) がある。また、2 種類のマグネットがあり、ID での運動量測定に用いる超伝導ソレノイドマグネットと Muon Spectrometer での p_T 測定を行うための超伝導トロイドマグネットがある。

以下で各サブシステムについて解説する。2.1 図が ATLAS 検出器の全体像である。

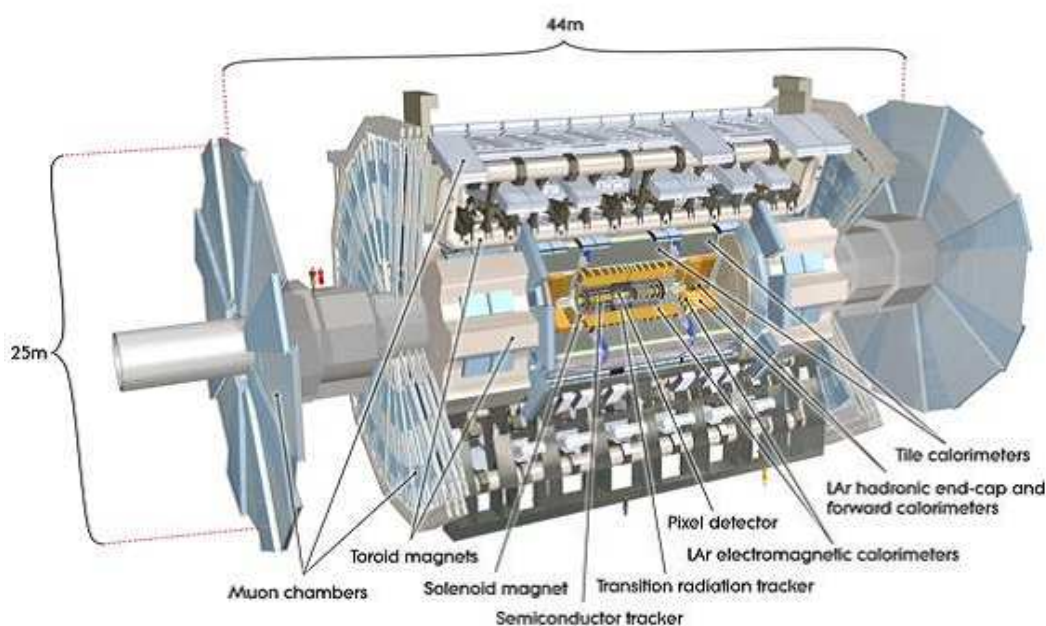


図 2.1: ATLAS 検出器の全体像 [11]

2.2 Tracking

Tracking を担当するのが内部飛跡検出器 (ID) である。2.2 図が ID の全体像である。ID は Pixel 型半導体検出器 (Pixel)、シリコンストリップ検出器 (Semi-Conductor Tracker, SCT)、遷移輻射飛跡検出器 (Transition Radiation Tracker, TRT) から構成される。 $\eta < 2.5$ の領域をカバーし、ソレノイド磁石によって作り出される 2 T の磁場を用いて粒子の運動量測定が行われる。

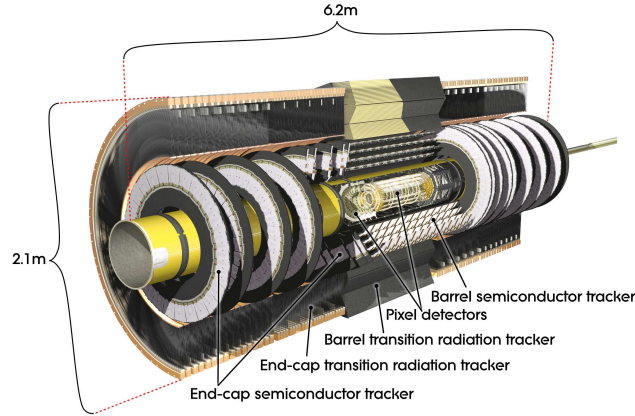


図 2.2: Inner Detector [9]

2.2.1 Pixel

Pixel 検出器は $R \leq 10$ cm の衝突点から最も近い領域に設定される検出器である。2.3 図が Pixel 検出器の全体像を描いたものである。全ての Pixel センサは同一で $50\text{ }\mu\text{m} \times 400\text{ }\mu\text{m}$ を 1 ピクセルとし、47268 ピクセル集めたものを一つのモジュールと呼ぶ (2.4 図)。1 ピクセル毎に読み出しが可能で、高イベントレートに対応するために 2 次元読み出しが可能である。バレル部が 3 層、前後方部が 3 層ずつとなっていて、読み出しチャンネル数は 8040 万チャンネルに及ぶ。 $R - \phi$ 方向で $10\text{ }\mu\text{m}$ 、 z 方向で $115\text{ }\mu\text{m}$ という高い位置分解能があり、衝突点の測定に主に使用される。

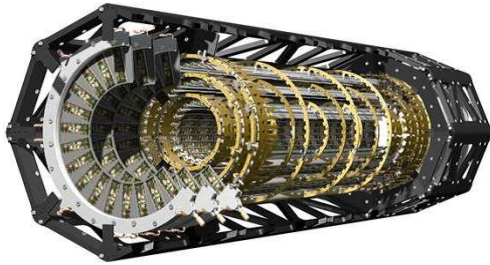


図 2.3: Overview of Pixel Detector [9]

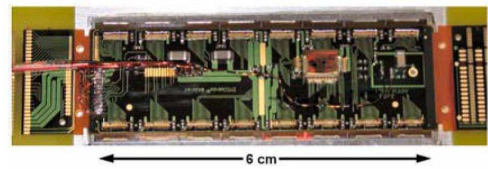


図 2.4: Pixel Module [9]

2.2.2 SCT(Semi-Conductor Tracker)

SCT は Pixel 検出器と同様に多数のシリコン検出器の集合体からなる。2.5 図は SCT 検出器の前後方部側からみた写真で、2.6 図は SCT 検出器のバレル部の写真である。1 つの SCT モジュールは 6 ～ 7cm 角の長方形シリコンウェハー 2 枚からなり、それぞれのウェハーには一定間隔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いる。2.7、2.8 図はバレル用、前後方部用それぞれのモジュールの写真であるが、外形や読み出しピッチの間隔が若干異なっている。バレル部ではモジュールが組み合わって 4 層の円筒状となっており、前後方部ではモジュールを円盤状に並べたものが片サイド 9 層ある。SCT の位置分解能は $R - \phi$ 方向で $17\mu\text{m}$ 、 z 方向で $580\mu\text{m}$ である。

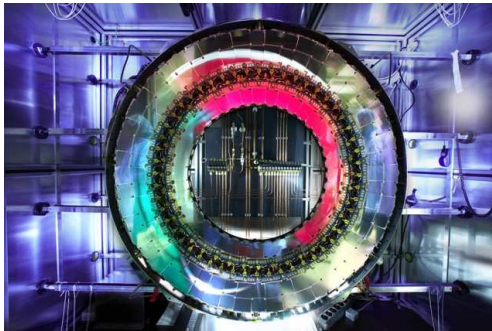


図 2.5: The Cross Section of SCT [9]

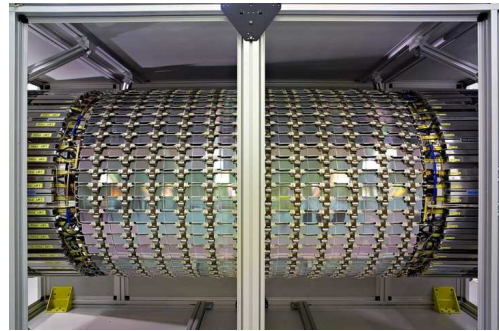


図 2.6: The Picture of Barrel SCT [9]

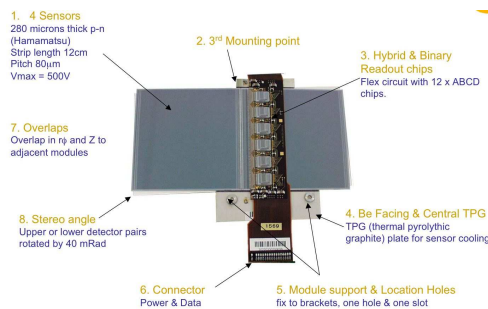


図 2.7: Barrel Module [9]

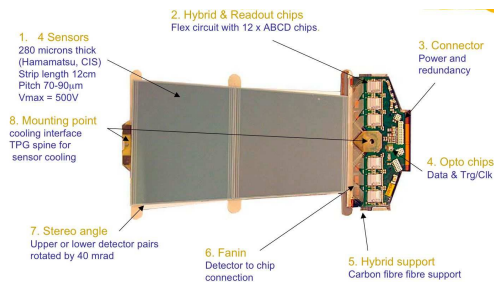


図 2.8: Endcap Module [9]

2.2.3 TRT(Transition Radiation Tracker)

TRT は遷移輻射の原理を用いた検出器である。遷移輻射とは荷電粒子が誘電率の異なる 2 つの物質を通るときに境界面において電磁的な放射をする現象である。放射されるエネルギーは荷電粒子の γ に比例するので、荷電粒子のエネルギーが高いほど顕著である。放射される光子 (X 線) は大体数 keV 程度である。2.9 図は TRT の前後方部の写真で、2.10 図は TRT のバレル部の写真である。

TRT はストロー状のチューブを重ね、その隙間はポリプロピレンファイバーで埋めた構成になっている。ポリプロピレンファイバーによって遷移輻射が起きる。2.11 図はストローチューブの層を

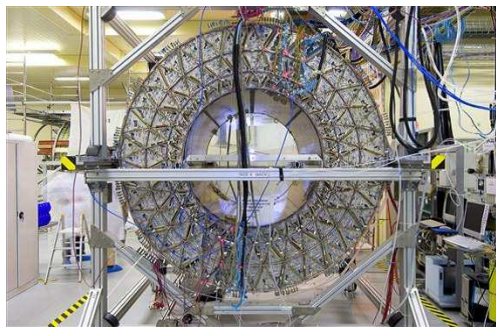


図 2.9: The Picture of Endcap TRT [3]

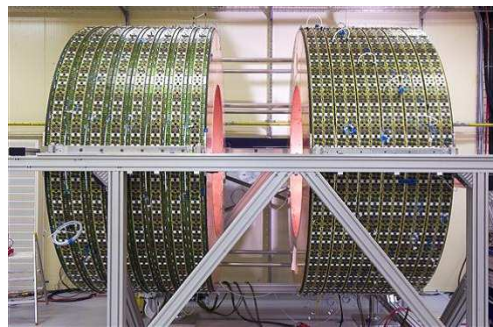


図 2.10: The Picture of Barrel TRT [3]

作り、組み立てを行っているときの写真である。

SCT より外の領域のトラッカーに TRT を用いることで、シリコン検出器を使う場合に比べ、コストの削減ができる。TRT は $R - \phi$ 方向のみの情報を提供し、位置分解能は一本のストローにつき $130\ \mu\text{m}$ である。SCT に比べ、モジュール単体での位置分解能は低い、TRT はより広い領域をカバーし、1トラックにつき、平均 36 個のヒットが得られることから運動量測定においては威力を発揮する。

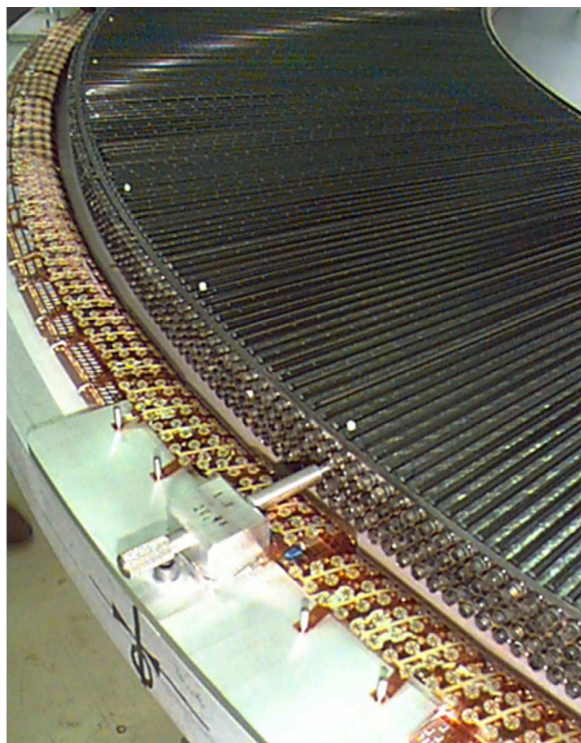


図 2.11: The Straw Structure of TRT [9]

2.3 Calorimetry

ATLAS 検出器のカロリメータは LAr(液体アルゴンを用いた) 電磁カロリメータとハドロンカロリメータに大別される。電磁カロリメータでは主に電子や γ の精密測定が行われる。またカロリメータの残りの部分はジェットのリコストラクションと E_T^{miss} 測定が物理的な要請に適うように作られている。カロリメータ全体で $|\eta| < 4.9$ の広範な領域をカバーしている。2.12 図が ATLAS のカロリメータシステムの全体像である。

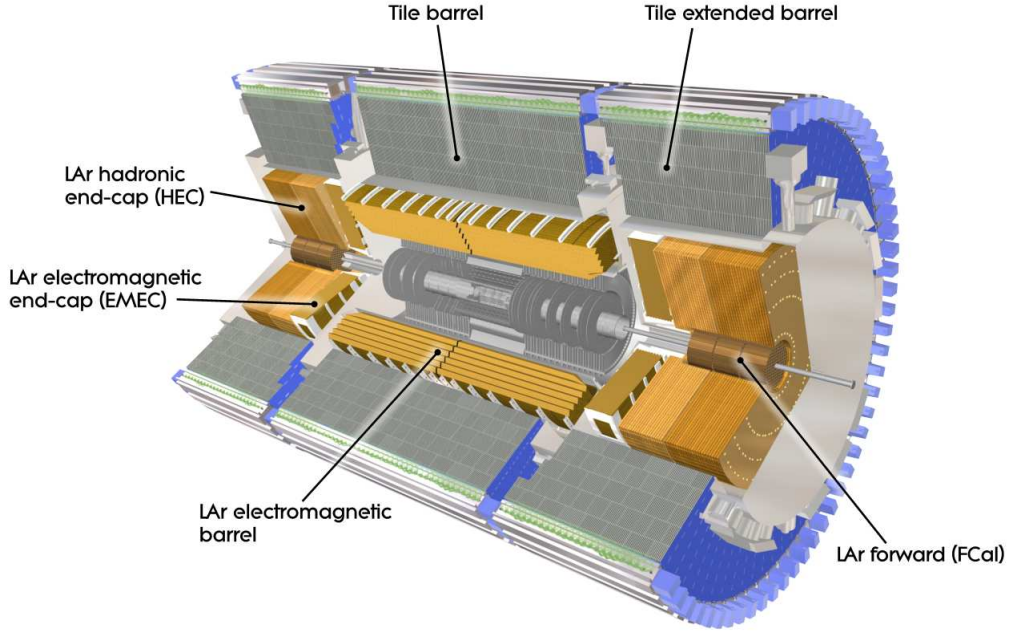


図 2.12: ATLAS calorimeter system [9]

2.3.1 LAr electromagnetic calorimeter

ATLAS の電磁カロリメータは LAr を使用したものである。鉛の吸収体 (absorber) と LAr を組み合わせたサンプリングカロリメータである。アコーディオン構造しているのが特徴で、 ϕ 方向の不感領域を無くしている。 $|\eta| < 1.5$ 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。読み出しチャンネル数はバレル部で約 10 万チャンネル、前後方部で 7 万チャンネルである。

電磁カロリメータは 3 層構造になっている (2.13 図参照)。図中の Layer1 では、 η 方向の超精密測定を行い、 $\pi_0 \rightarrow \gamma\gamma$ による γ の同定を行う。Layer2 では、主にシャワーの測定を行う。Layer3 はハドロンシャワーと電磁シャワーの区別を行う。 $|\eta| < 1.8$ の領域ではプリサンプラーが設置されている。プリサンプラーには LAr の層が使われており、ここでエネルギーの一部をサンプルすることで、カロリメータの手前で起きた粒子のエネルギー損失の補正を行う。

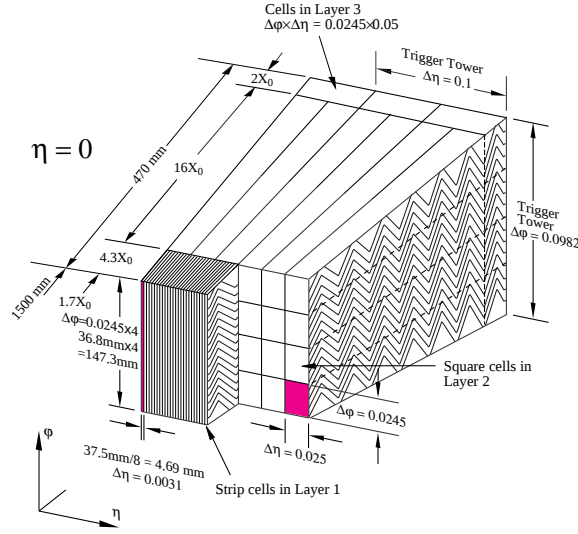


図 2.13: The Sketch of a barrel module [9]

2.14 図は部分的に組み立てられた LAr 電磁カロリメータのバレル部の写真で、2.15 図はその前後方部の写真である。

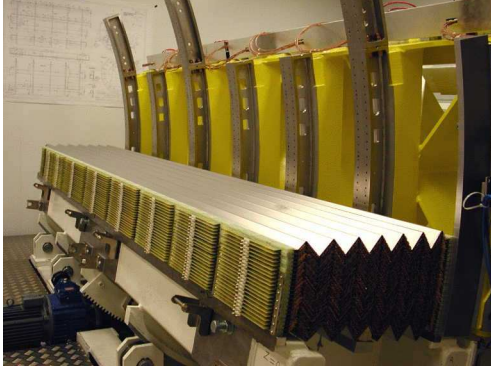


図 2.14: The picture of a partly stacked barrel LAr module [9]



図 2.15: The picture of a side view of end-cap LAr module [9]

2.3.2 Hadronic calorimeters

ハドロンカロリメータは $|\eta| < 1.7$ のバレル部と $1.5 < |\eta| < 3.2$ のエンドキャップ部で構造が異なる。バレル部は Tile カロリメータが使われ、エンドキャップ部には LAr Hadronic End-cap Calorimeter (HEC) が使われている。

Tile calorimeter

Tile カロリメータは鉄板を吸収体とし、タイル状のシンチレータを交互に重ね合わせたサンプリングカロリメータである。検出器の厚さは $\eta = 0$ で 9.7λ である (λ は Interaction length)。シンチレータタイルの両端に波長変換ファイバーが繋がれていて、二つの光電子増倍管から読み出しを行う。2.16 図は Tile カロリメータの構造を表している。

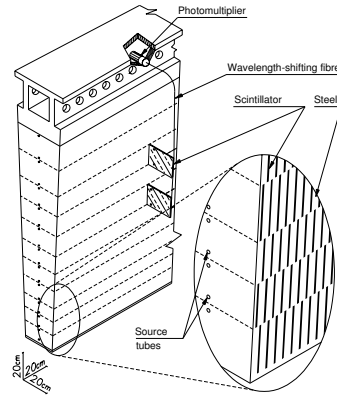


図 2.16: Schematic view of Tile calorimeter [9]

LAr hadronic end-cap calorimeter

HEC は電磁カロリメータと同じく LAr を用いたサンプリングカロリメータである。ただし、absorber には銅が使われている。HEC は 32 個の同一な形状のくさび型モジュールを組み合わせて円盤状になっている。2.17 図が HEC モジュールの概略図である。

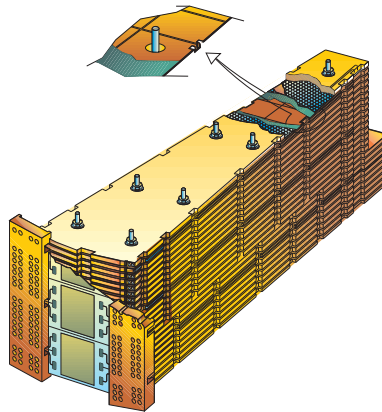


図 2.17: Schematic view of a HEC module [9]

2.4 Magnets

ATLAS 検出器には 3 種類のマグネットがインストールされている。ID での運動量測定のために用いる超伝導ソレノイド磁石とミューオンスペクトロメーターでミューオンの運動量測定に使用するバレル部超伝導トロイド磁石 (Barrel Toroid) と前後方部超伝導トロイド磁石 (End-cap Toroid) である。2.18 図はマグネットの中のコイルのイメージ図であるが、ソレノイド磁石では磁場がビーム軸方向にかかるため、 ϕ 方向に曲がる。バレル部と前後方部のトロイド磁石では ϕ 方向に磁場が発生する。バレル部トロイド磁石は ϕ に対して 8 回対称で作られており、一つ一つのコイルが独立した真空容器に入っている。前後方部もコイルは 8 回対称で作られているが、一つの真空容器の中に収まっている。2.20 図はソレノイド磁石の写真、2.21 図はバレルトロイドの写真、2.22 図はエンドキャップトロイドの写真である。

ソレノイドで発生させることができる磁束密度は 2 T 程度、バレルトロイドでは平均で 0.5 T、エンドキャップトロイドでは 1.0 T となっている。2.19 図は横軸を η に縦軸を積分磁場強度 $\int \mathbf{B} \cdot d\mathbf{l}$ をとったものである。1.4 < $|\eta|$ < 1.6 の領域はバレルトロイドとエンドキャップトロイドの遷移領域で打ち消しあって磁場が弱くなっている。

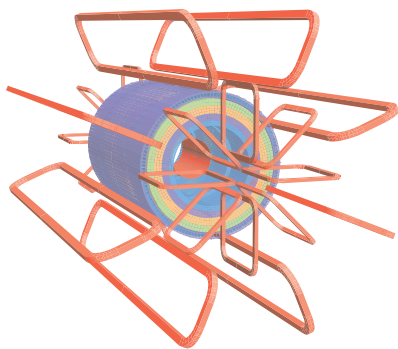


図 2.18: ATLAS のマグネットの中のコイルのイメージ [9]

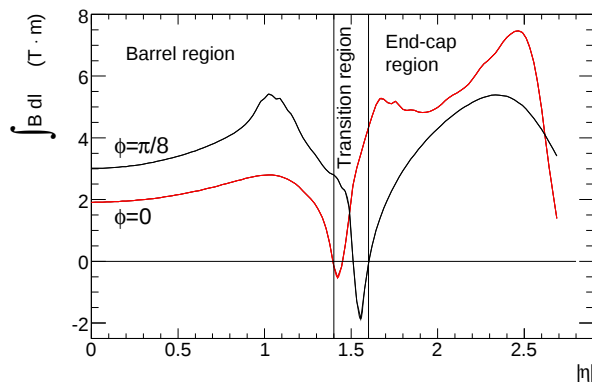


図 2.19: η の関数でみた積分磁場強度 [9]



図 2.20: Central solenoid magnet [9]

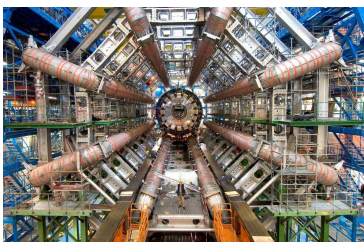


図 2.21: Barrel toroid [9]

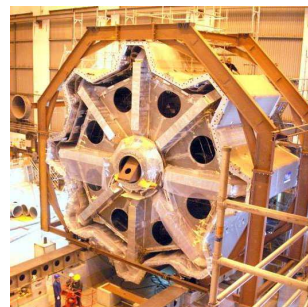


図 2.22: Endcap Toroid [9]

2.5 Muon Spectrometer

2.23 図が ATLAS 検出器のミュオンシステム (またはミュオンスペクトロメーターと呼ばれる) の全体像である。ATLAS 検出器の最も外側に配置されている。ミュオンは $2.2\mu\text{s}$ と寿命が長く、物質の透過力が強いので (カロリメータでエネルギー損失を十分にしないので) ミュオンスペクトロメーターにまで到達する。ミュオンスペクトロメーターはトリガーを担当する Thin Gap Chamber(TGC)、Resistive Plate Chamber(RPC) と、位置検出を行う Monitor Drift Tube(MDT)、CSC(Cathode Strip Chamber) からなる。

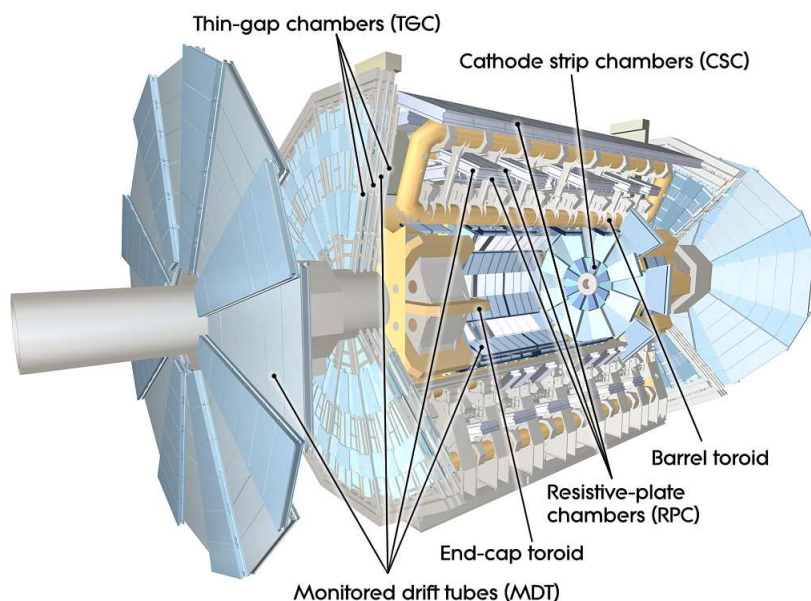


図 2.23: ATLAS muon system [9]

2.24 図はミュオンスペクトロメーターを $x-y$ 平面で見た図で、2.25 図は $y-z$ 平面で見た図である。

2.5.1 MDT(Monitored Drift Tube chambers)

MDT はドリフトチューブの一種で、約 30 mm 径のカソードチューブの中に $50\mu\text{m}$ 径のアノードワイヤーを張った構造をしている。2.26 図は MDT チェンバーの図で、2.27 図は MDT チューブの断面図で、2.28 図は MDT チューブを横方向から見た図である。一つの MDT チューブの位置分解能は $80\mu\text{m}$ で、最大のドリフト時間は 700 ns である。MDT の主要なパラメータを表 2.1 に載せる。

2.26 図にあるように 3 ないし 4 のチューブを層状に並べて、仕切りを挟んで反対側にも同じように層状に並べられたものが一つのチェンバーとなる。位置情報の精密測定のためには重力や温度による歪みが大きな障害となる。そこでチェンバー内にはチューブのずれを監視する alignment system が設置されている。その他にもチェンバーの位置も監視する alignment system も設置されている。

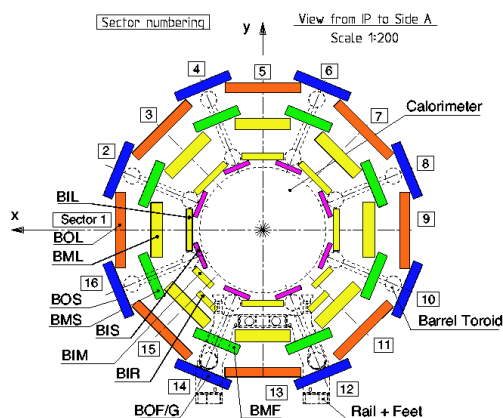


図 2.24: Cross-section of the barrel muon system [9]

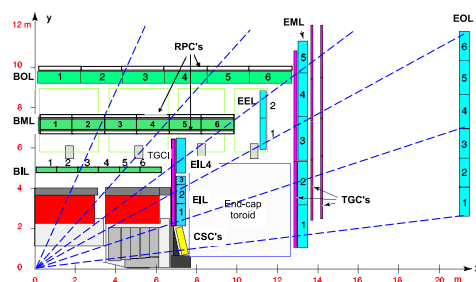


図 2.25: Cross-section of the muon system in y-z plane [9]

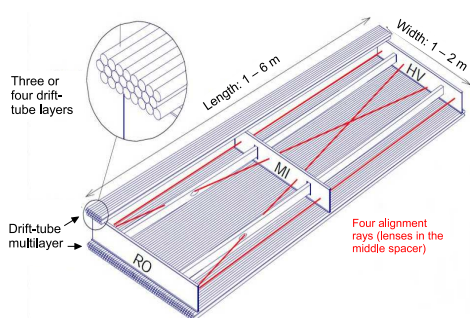


図 2.26: MDT Chamber の構造 [9]

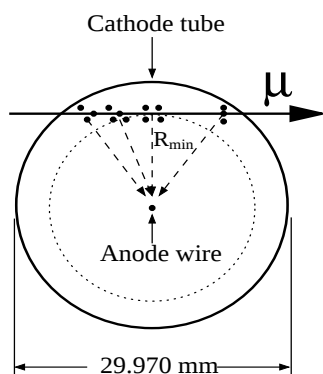


図 2.27: Cross-section of a MDT tube [9]

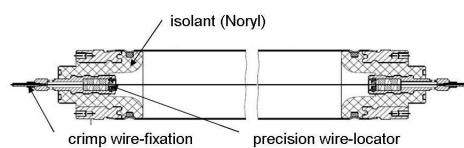


図 2.28: Longitudinal cut through a MDT tube [9]

2.5.2 CSC(Cathode Strip Chambers)

MDT はバレル、エンドキャップ領域全てにインストールされているが、粒子の飛来する頻度が 150 Hz/cm^2 を越える、エンドキャップの一番内側の領域では MDT ではなく、CSC が用いられている。

Parameter	Design value
Tube material	Al
Outer tube diameter	29.970 mm
Tube wall thickness	0.4 mm
Wire material	gold-plated W/Re (97/3)
Wire diameter	50 μ m
Gas mixture	Ar/CO ₂ /H ₂ O (93/7/ $\leq$ 1000 ppm)
Gas pressure	3 bar(absolute)
Gas gain	2×10^4
Wire potential	3080 V
Maximum drift time	~ 700 ns
Average resolution per tube	80 μ m

表 2.1: Main MDT chamber parameters [9]

CSCは 1000 Hz/cm^2 のレートにまで耐えられる。CSCは構造的にはMWPC(Multi-Wire Propotional Chamber)の一種で、ドリフト時間が40 nsと短く、2トラックに対して高い分解能を持ち、ガス量が少なく、水素を含まないガスを用いるので、中性子線への感度が低い。2.29 図がエンドキャップにおけるCSCのレイアウトである。8個の大きなチェンバーと8個の小さなチェンバーから構成されている。

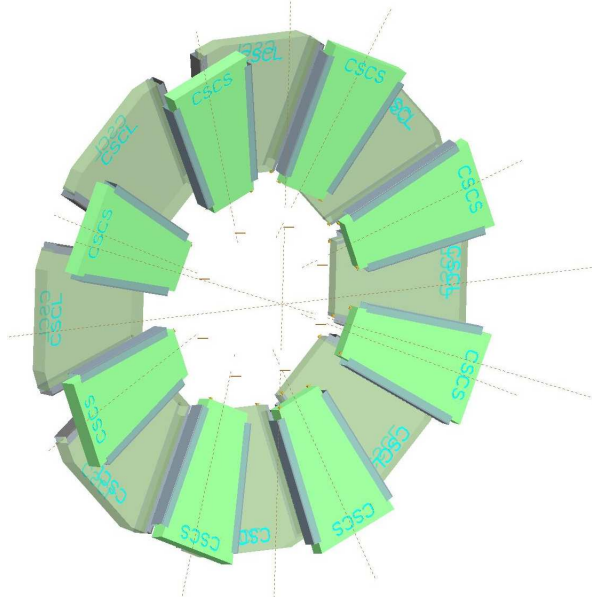


図 2.29: Layout of a CSC [9]

2.30 図はチェンバーの構造を描いていて、ワイヤーとストリップは 2.31 図のようになっている。CSC はワイヤー間のピッチとワイヤーとカソードストリップ間の距離が 2.54 mm と等しい。読み出しを行うストリップの間隔は 5.08 mm となっている。

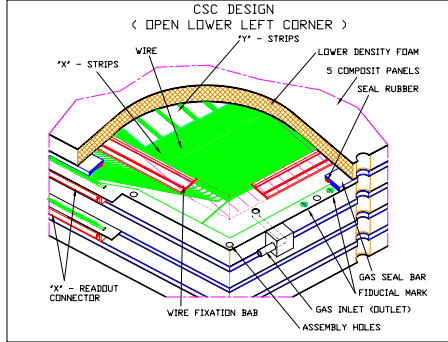


図 2.30: Structure of the CSC [9]

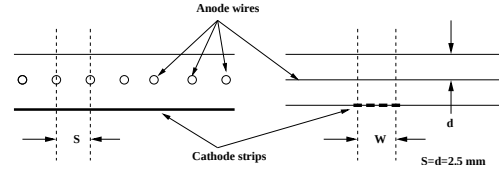


図 2.31: Structure of the CSC cell [9]

CSC はワイヤーでなだれ増幅によって誘導された電荷をカソードストリップから読みだし、複数のストリップからの信号に対して重心法を用いることで $60\mu\text{m}$ の分解能での位置検出を可能にしている。2.32 図は 3 ～ 5 の読み出しストリップに誘導された電荷の分布である。2.33 図はカソードストリップで読み出し行うイメージ図であるが、読み出しストリップの間に挟まったストリップからは読み出しを行っていない。読み出しに使わないストリップは隣の読み出しストリップとコンデンサを介して結合されていて、正規分布のような電荷分布を生じさせるのに役立っている。表 2.2 に CSC の主要なパラメータを載せる。

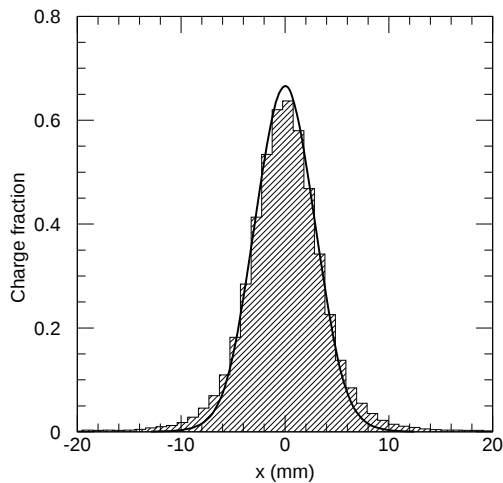


図 2.32: Charge distribution on the CSC cathode [9]

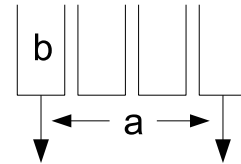
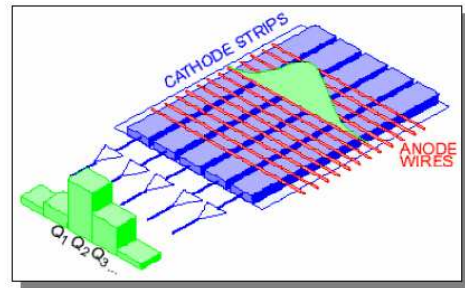


図 2.33: Readout strips of CSC [9]

表 2.2: Main CSC chamber parameters [9]

Parameter	Design value
Operating voltage	1900 V
Anode wire diameter	$30\ \mu\text{m}$
Gas gain	6×10^4
Gas mixture	Ar/CO ₂ (80/20)
Total ionisation (normal track)	90 ion pairs
Position resolution	$60\ \mu\text{m}$

2.5.3 RPC(Resistive Plate Chambers)

RPC はバレル部のトリガーを担当する他に、MDT では測定されない ϕ 方向の位置を測定する。2.34 図はバレル上部の RPC の断面図である。

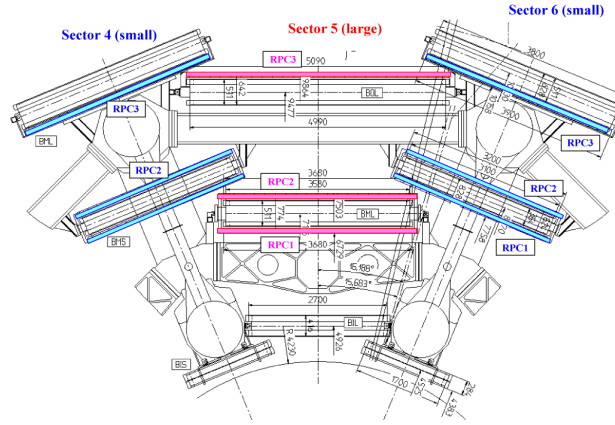


図 2.34: Cross-section thorough the upper part of the barrel with the RPCs

RPC は平行電極板を使用したガスチェンバーである。RPC にはワイヤーは存在しない。2.35 図は RPC の構造を表しているが、フェノールメラミンの薄板で作られた電気抵抗のある板が 2 mm の間隔をおいて平行に置かれている。電極間には 4.9 kV/mm の高電圧がかけられていて荷電粒子が通るとトラックに沿ってなだれ増幅がおき、アノードに達する。シグナルはフェノールメラミンの薄板の外側にある金属ストリップから読みだされる。 ϕ 方向の位置分解能は $5 \sim 10\text{ mm}$ 程で、検出器の Time jitter は 1.5 ns より小さく、反応時間が短い。表 2.3 に RPC の主要なパラメータを載せる。

2.5.4 TGC(Thin Gap Chambers)

TGC の動作原理とエレクトロニクスについては次章で詳しく説明をするので、ここでは省略する。

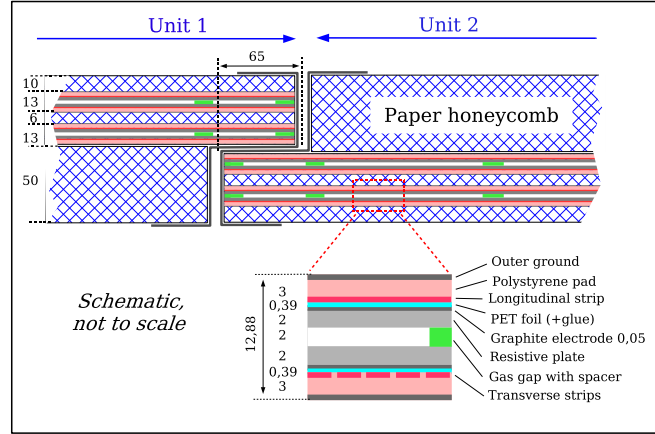


图 2.35: Structure of the RPC

表 2.3: RPC parameters [9]

Parameter	Design value
Electric field in gap	4.9 kV/mm
Gas gap	2 mm
Gas mixture	$C_2H_2F_4$ /Iso – C_4H_{10} /SF ₆ (94.7/5/0.3)
Readout pitch of η and ϕ -strips	23 – 55 mm
Detection efficiency per layer	$\geq 98.5\%$
Efficiency including spacers and frames	$\geq 97\%$
Intrinsic time jitter	≥ 1.5 ns
Jitter including strip propagation time	≥ 10 ns
Local rate capability	~ 1 kHz/cm ²
Streamer probability	$\geq 1\%$

2.6 Trigger and DAQ

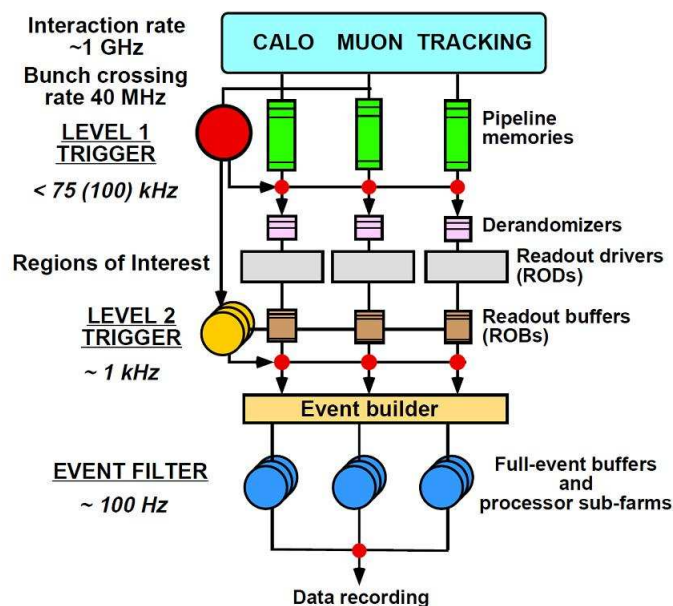


図 2.36: Trigger and DAQ diagram

2.36 図は ATLAS のトリガーと DAQ のダイアグラムである。ATLAS では 1 GHz で起こる反応の中から興味のある事象のみをデータとして保存するために、多段階的なトリガーシステムを用いている。トリガーは 3 種類あり、Level-1、Level-2、Event Filter(EF) と呼ばれる。Level-1 トリガーはハードウェアによるトリガーで、ここでレートは 75 kHz にまで落とされる。Level-2、EF はまとめて HLT(High Level Trigger) と呼ばれるが、これらはソフトウェアのトリガーで Level-2 の時点で 3.5 kHz にまで落とされている。最終的には EF のアウトプットの時点で 200 Hz となり、データは CERN の計算機センターのストレージに保存される。

2.6.1 Level-1 Trigger

2.37 図は Level-1 トリガーのブロックダイアグラムである。Level-1 トリガーはカロリメータトリガーと TGC と RPC からなるミューオントリガーから構成されている。カロリメータで e 、 γ 、ジェット等のシグナルが検出されるとその情報を CTP(Central Trigger Processor) に送る。一方、RPC、TGC でミューオンが検出されたときは p_T 閾値と興味のある領域の情報 (ROI、Region Of Interest) をまず MUCTPI(Muon trigger CTP Interface) に送る。MUCTPI は RPC と TGC の情報を統合し、CTP に送る。CTP が最終的な Level-1 トリガーの判定を行う。CTP の判定を通過した場合は、CTP は L1A(Level-1 Accept) と呼ばれるトリガー信号を TTC(Timing Trigger and Control system) を用いて ATLAS の各検出器のフロントエンドに配布する。

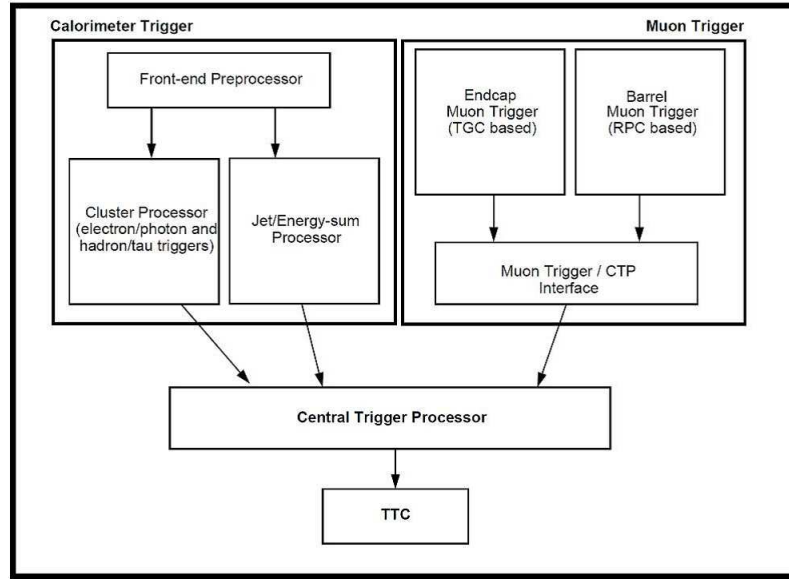


図 2.37: L1 Trigger block diagram

各検出器のフロントエンドには L1A の判定がなされる間データを保持する L1 バッファ (Level-1 バッファ) が実装されている。L1A 判定がなされるのにかかる時間 (L1 Latency) は $2.5\ \mu\text{s}$ と決まっています、TGC 検出器の場合は $3.2\ \mu\text{s}$ の間データを保持できる L1 バッファを搭載している。L1A を受け取ったフロントエンドエレクトロニクスは後段にデータを送り、ROD (Read Out Driver) で検出器内でのデータのイベントビルディングを行い、ROS (Read Out System)、正確には ROS に搭載されている ROB (Read Out Buffer) にデータを送る。

2.6.2 Timing Trigger and Control system

図 2.38 に ATLAS 実験におけるタイミングに関するシグナルの分配の様子を示した概要図を載せる。ATLAS 実験においてタイミングに関するシグナルの配布を担うのが TTC (Timing Trigger and Control system) である。TTC システムは RD12¹ によって開発された optical fan-out の技術に基づいたシステムである。各検出器は L1A を TTC システムから受け取り、その L1A に対応したヒット情報を読み出す。各測定器の側には Bunch Counter、Event Counter が存在し、連続して入射する Bunch のうちそれが何番目の Bunch かを識別する BCID と何番目の L1A かを識別する L1ID をヒット情報に付加して読み出す。TTC システムはまたこれらの ID が実験全体を通して整合性を保つために、BCR (Bunch Counter Reset) と ECR (Event Counter Reset) という信号も各測定器に分配する。BCR は orbit 信号² と同期して送られ、Bunch Counter をリセットする。ECR はある決まった

¹LHC 共通のプロジェクトで EP (Experimental Physics)、SL (SPS and LHC) 部門と 4 つの LHC の実験から出資されたプロジェクトである。初期には RD12 によって TTC システムの研究開発が行われた。[2]

²一つの bunch が LHC リング内を一周するのにかかる時間 ($88.934\ \mu\text{s}$) と同じ周期で送られる信号

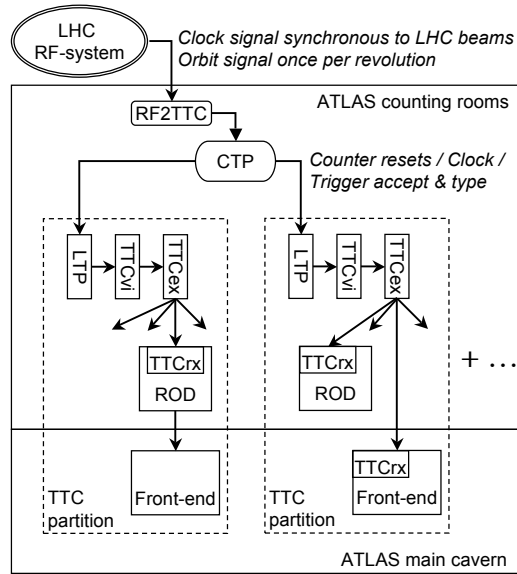


図 2.38: TTC システムによるタイミングシグナルの分配の概要 [9]

周期で TTC システムから発行され、Event Counter をリセットする。

LHC ビームに同期したクロックと orbit 信号は RF2TTC モジュールを介したあとに CTP に送られる。RF2TTC では、シグナルを整形したり、シグナルにディレイをかけることができる。CTP は検出器固有の信号、例えば L1A や ECR を付加し、各検出器の LTP(Local Trigger Processor) に送る。LTP から先では TTCvi(TTC VME Interface) によってシグナルはシリアルライズされ、TTCEX(TTC Encoder Transmitter) から Optical fiber を通して検出器のフロントエンドにシグナルが分配される。フロントエンドでは TTCCrx(TTC receiver chip) が受信したシグナルをデコードし、エレクトロニクスでできるように電気信号に変換している。TTC システムの実装の仕方と使用の方法は各検出器によって異なる。

2.6.3 Level-2 Trigger

Level-2 トリガーでは Level-1 トリガーから送られてきた ROI の情報と MDT、CSC、ID からのトラック情報とカロリメータの高精度情報を合わせてコンピュータで計算しトリガー判定を行う。Level-1 からの ROI 情報は ROI Builder を通った後に、L2SV(Level-2 Supervisor) に渡される。L2SV は 500 台の PC からなる L2PU(Level-2 Processing Unit) にそれぞれのイベントをアサインする。L2PU はネットワークを介して ROS から必要なイベント情報を受け取り Level-2 判定を行う。Level-2 判定がなされると DFM(Data Flow Manager) はそれぞれのイベントをイベントビルディングノードの SFI(Sub-Farm Input) に割り当てを行う。イベントビルディングがされたデータは SFI のバッファに保持され、EF による処理を待つ。Level-2 トリガーの処理は 40 ms かかる。

2.6.4 EF(Event Filter)

EF ではさらに選別が行われる。SFI からデータを受け取り、約 1800 台の EF ノードによって並列処理が行われる。EF での処理には 4s かかる。EF を通過したイベントは SFO(Sub-Farm Output) に送られ、CERN 計算機センターにあるテープストレージに記録される。SFO にデータが送られる順番は、処理が終わったものから送られるので、必ずしも時系列と一致していない。

2.7 データ解析

EF を通過したデータは 1 イベントあたり 1.6 MB の大きさがあり、これが 200 Hz で読みだされる。これが 1 年間蓄積されると 3.2 PB にも及ぶ。さらにデータ解析に必要とされるシュミレーションデータも加えるとデータ量は高エネルギー実験がいままで経験したことのない規模となる。LHC 実験ではこの難問に打ち勝つために「グリッド (Grid)」が採用されている。グリッドは多数の計算機クラスターを広域ネットワークで結び、あたかも単一の計算機システムとして運用する技術である。LHC 実験では各国が計算機資源を提供しており、それぞれが階層構造 (Tier) の各部に位置づけられている。CERN の計算機センターを頂点の Tier0 とし、Tier0 の下には Tier1 を担当する計算機群があり、Tier1 の下に Tier2 の計算機群があるという構造になっている。これを表したのが 2.39 図である。

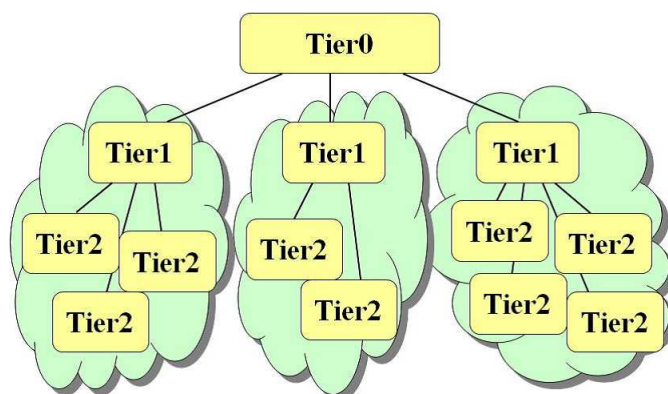


図 2.39: クラウドモデル [26]

実験データは再構成 (Reconstruction) されるとまず ESD(Event Summery Data) と呼ばれるデータになる。ESD には検出器の情報、例えば飛跡検出器のトラックの情報やカロリメータのクラスター情報が詰まっている。ESD は 1 イベントあたり 0.5 MB 以下のサイズに抑えることを目標としている。

ESD を変換して AOD(Analysis Object Data) が生成される。AOD は物理の情報、各粒子の情報を持っている。AOD は 1 イベントあたり 100 kB 以下であるが、物理の解析対象ごとにデータセットが作られている。

Tier0 には生データが保管され、生データをもとに ESD を生成する。生データと ESD は Tier1 にも送られる。Tier1 は世界に 10 箇所程度分散して配置されていて、Tier0 から送られてきた生デー

タと ESD を保管する。また ESD を変換し AOD 生成を行う。生成された AOD は他の Tier1 にも送られ、いずれの Tier1 にもフルセットの AOD が保管される。AOD や ESD の一部は Tier2 からの要求に応じて Tier2 にも送られる。Tier1 は恒久的な記録装置を持ち、Tier2 で生成されたシミュレーションデータも保管する。Tier2 は Tier1 からユーザー解析に必要な AOD、ESD を受け取り、解析を行う。また、シミュレーションデータの生成も行う。

東京大学素粒子国際研究センター (ICEPP) にある地域解析センターは Tier2 を担当している。2.40 図は ICEPP にある解析センターの写真で、計算機サーバ、ディスクサーバ、テープサーバから構成されている。2.41 図はテープストレージの写真である。



図 2.40: ICEPP にある Tier2 Tokyo cite [26]



図 2.41: Tokyo Tier2 のテープストレージ [26]

第3章 ATLAS 前後方ミュオントリガーシステム

ATLAS の前後方部をカバーするのが TGC チェンバーで、TGC チェンバーとそのエレクトロニクスによって担われるのが ATLAS 前後方ミュオントリガーシステムである。ATLAS のトリガーシステム内では Level-1 トリガーを担当する。また ϕ 方向の位置の測定も行い、ミュオントラックの再構成の補助的役割も果たす。ATLAS 前後方ミュオントリガーシステムは単に TGC システム、TGC とも呼ばれる。

以下では、TGC チェンバーの構造と動作原理に続いて ATLAS 前後方ミュオントリガーシステムを構成する各エレクトロニクスについて説明する。

3.1 Thin Gap Chamber(TGC) の構造と動作原理

3.1.1 基本構造

Thin Gap Chamber(TGC) は MWPC(Multi-Wire Proportional Chamber) の一種であり、ガス検出器に属する。LEP の OPAL 実験において 12 年間の安定動作の実績があり、そのときに使用されたものを基に改良されたものが ATLAS 実験で使われている。アノードにはタングステンワイヤーが用いられ、カソードにはストリップ状の銅板が使用されている。ワイヤーが $R(\eta)$ 方向、ストリップが ϕ 方向の読み出しに使われる。3.1 図が TGC の断面図で、3.2 図が主要な TGC のパラメータである。

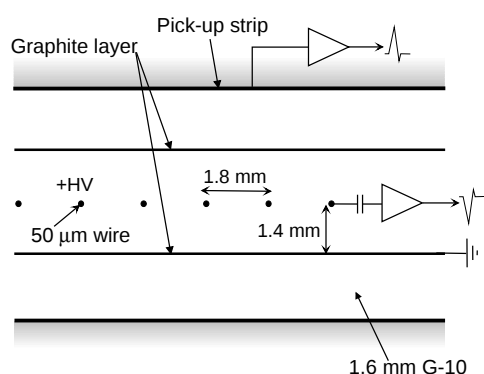


図 3.1: TGC の構造 (断面図)[9]

Parameter	Design value
Gas gap	$2.8 \pm 0.10 \text{ mm}$
Wire pitch	$1.8 \pm 0.05 \text{ mm}$
Wire diameter	$50 \mu\text{m}$
Wire potential	$2900 \pm 100 \text{ V}$
Operating plateau	200 V
Gas mixture	$\text{CO}_2/\text{n-pentane} \quad (55/45)$
Gas amplification	3×10^5

図 3.2: TGC の主要な運転パラメータ [9]

アノードワイヤーには金メッキが施された $50\text{ }\mu\text{m}$ 径のタングステンワイヤーが使用され、 $1.80 \pm \text{mm}$ の間隔で張られている。必要とされる運動量分解能に合うように、ワイヤーは η の関数として $4 \sim 33$ 本をひとまとめにして読み出しを行っている。

ガラスエポキシ板 (FR4、プリント基板の素材) の片面に表面抵抗 $1 \pm 0.5\text{M}\Omega/\text{cm}^2$ のカーボンを塗布し、もう片面に $30\text{ }\mu\text{m}$ の厚さの銅板がワイヤーと直角をなすように敷き詰められている。ストリップは ϕ 方向に 32 分割されている。3.1 図を見てもわかるように、ワイヤーストリップ間の方がワイヤー間隔より狭くなっている (このために "Thin" と呼ばれる)。これによって陽イオンがカソードに達するまでにかかる時間が短くなり、不感時間が減少する。

TGC は CO_2 と $n\text{-C}_5\text{H}_{12}$ が 55:45 で混合されたガスを使用している。 CO_2 が電離ガスの役割を $n\text{-pentane}$ が放電を防ぐクエンチガスの役割を担う。

3.1.2 動作原理

MWPC ではガスギャップ中を荷電粒子が通過すると、電離作用によって電子と陽イオンが生じる。このときにできた電子は一次電子と呼ぶ。アノードワイヤーとカソードストリップ間には高電圧がかけられているので、一次電子はアノードワイヤーへ、陽イオンはカソードストリップへと移動を始める。電場の強さがある閾値以上あると、一次電子は他のガス分子を電離するだけのエネルギーを得ることができる。十分なエネルギーを得た一次電子はまた電離作用によって電子、陽イオンペアを作り、さらにその電子が加速されまた電子、陽イオンペア作るといったように雪崩的に電子、陽イオンペアを生成する。これはタウンゼントなだれ (Townsend avalanche) と呼ばれる。電子雲はアノードワイヤーに向かってドリフトし、ワイヤーを取り囲む。雪崩の効果によって増幅された信号を読み出すことができる。陽イオンは質量が大きいためしばらくワイヤー表面にとどまるが、その後ワイヤー半径方向に離れていく。ガスギャップと読み出し用のストリップの間には FR4 があるが、ガスギャップ側の面に高抵抗のカーボンが塗布されているので、読み出し用ストリップにも電荷が誘起され信号として読み出せる。ワイヤーに生じるシグナルの立ち上がり時間は非常に速いが、立下り時間は増幅器の前に挿入される微分回路の時定数 RC によって決まる。TGC ではこの電子雪崩による増幅率は 3×10^5 である。3.3 図はこのタウンゼントなだれのイメージ図である。

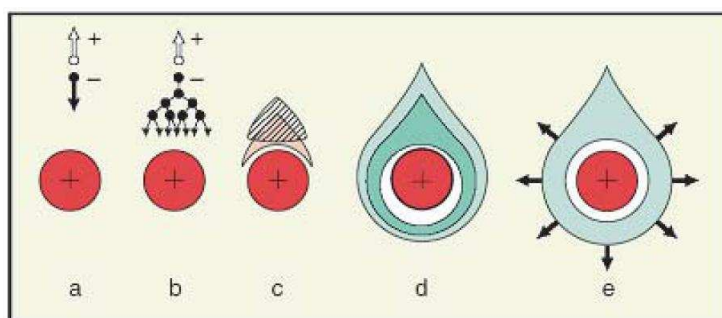


図 3.3: タウンゼントなだれのイメージ [15]

3.1.3 チェンバーの構造

TGC のチェンバーの形は 3.4 図のように扇形をしている。ワイヤーは $r(\eta)$ 方向の読み出しに使われ、ストリップは ϕ 方向の読み出しに使われる。ワイヤーは η に応じて求められる granularity を満たすように 4~33 本を一つにまとめて読み出すようになっている (wire-ganging と呼ばれる)。ストリップはチェンバータイプによらず 32 本あり、それぞれが一つのチャンネルとして読み出される。各ストリップは End-cap 領域では 4 mrad、Forward 領域では 8 mrad に対応する幅を持つ。また、3.5 図にあるようにチェンバーにはワイヤーサポートも備えられている。ワイヤーサポートにはボタン型のもつと棒状のものがある。ワイヤーサポートはワイヤーのたるみを防ぐだけでなく、ガスの流路を形成する役割と TGC 自体の歪みを防ぐ役割もある。

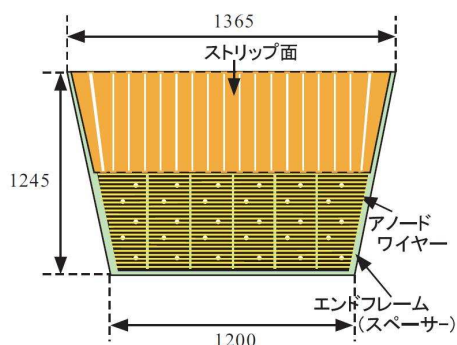


図 3.4: TGC の平面図 [30]

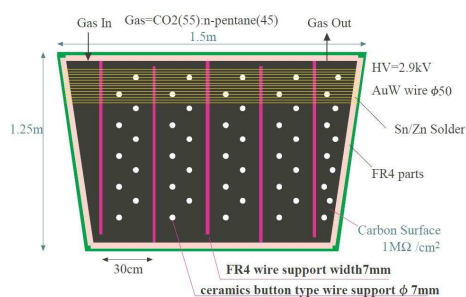


図 3.5: TGC のワイヤーサポート [30]

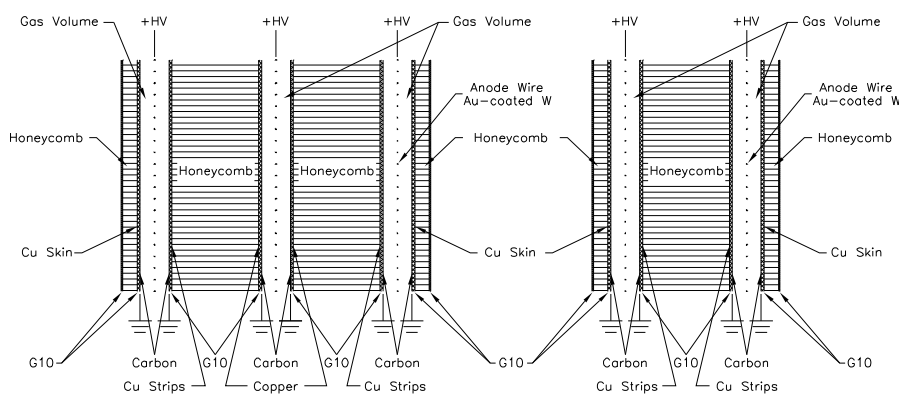


図 3.6: TGC のトリプレットチェンバー (左) とダブルットチェンバー (右) [9]

TGC チェンバーにはガスギャップが 2 層のダブルットチェンバーと、3 層のトリプレットチェンバーがある (3.6 図参照)。ダブルットチェンバーはワイヤー 2 層とストリップ 2 層それぞれから読み出しが可能である。トリプレットチェンバーは真ん中の層に読み出し用のストリップはないので、ワ

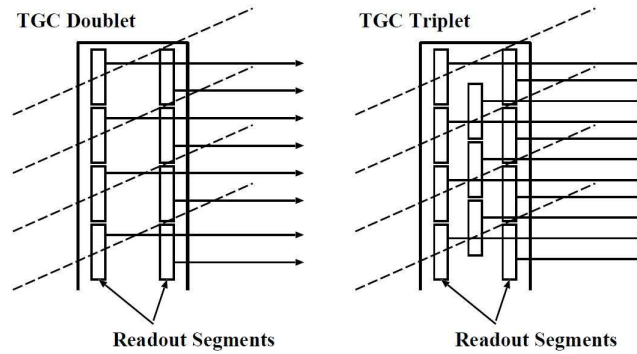


図 3.7: TGC のワイヤーチャンネルのスタッガリングのイメージ図

ワイヤー 3 層、ストリップ 2 層の読み出しとなる。ガスギャップ間にはハニカムが使われ、強度を保つ役割を果たしている。

また、ダブルットチェンバーとトリプレットチェンバーのワイヤーチャンネルは、ダブルットの場合で $1/2$ 、トリプレットの場合で $1/3$ ずらして配置されている (3.7 図参照)。これは不感領域を減らすことができるのに加え、実行的な位置分解能を 2 倍、3 倍にすることができる。

3.2 TGC の配置

ATLAS 実験で用いられている TGC チェンバーの総数は約 3600 枚ある。台形型のチェンバーを敷き詰め円形にして設置される。3.8 図は R-Z 平面でみた TGC のレイアウトである。エンドキャップトロイドの内側に位置する EI、FI をインナーと呼び、M1、M2、M3 と M1 と M2 の間にある MDT を含めたものはビッグウィールと呼ばれる。M1、M2、M3 という呼び方の他に TGC1、TGC2、TGC3 という呼び方もあるが、本論文では後者の呼び方を採用する TGC1 はトリプレットチェンバーからなり、TGC2、TGC3 はダブルットチェンバーからなる。合わせてワイヤー 7 層、ストリップ 6 層となる。TGC3 は Pt 測定の起点となることからピボットプレーン (Pivot plane) とも呼ばれる。

ビッグウィールを ϕ 方向に 12 分割したものをセクターと呼んでいる。セクターはさらに ϕ 方向に、エンドキャップでは 4 分割され、フォワードでは 2 分割される。それらはエンドキャップでは $\phi_0 \sim \phi_3$ と番号付けがされ、フォワードでは ϕ_0 または ϕ_2 の番号がつけられる。この " ϕ " 一つがトリガーセクターと呼ばれ、トリガーデータを処理するときの一つの単位となる (3.11 図)。つまり、TGC のエンドキャップでは " ϕ " は 48 個あり、フォワードでは 24 個あることになる。" ϕ " はさらにサブセクターと呼ばれるものに分割される。エンドキャップではサブセクターは 148 個あり、フォワードでは 64 個ある。このサブセクターの大きさが TGC での ROI (Region Of Interest) に相当する。

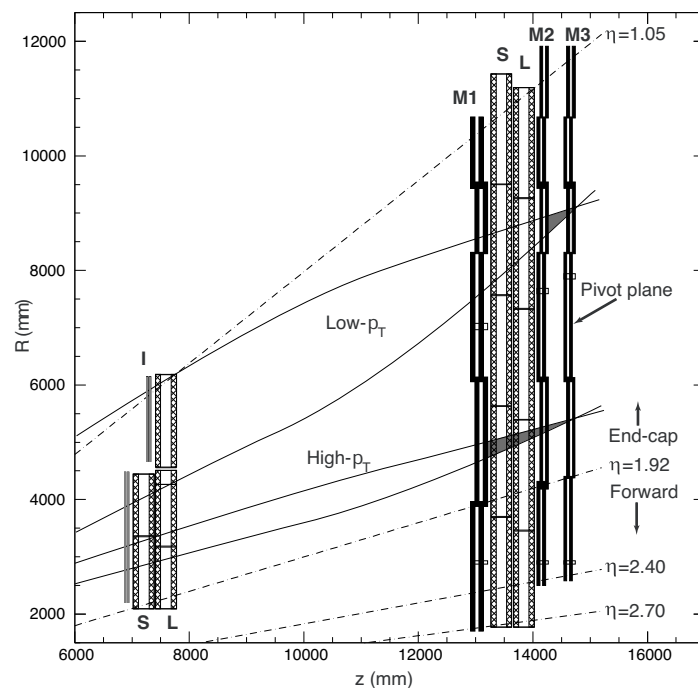


図 3.8: R-Z 平面でみた TGC のレイアウト [9]

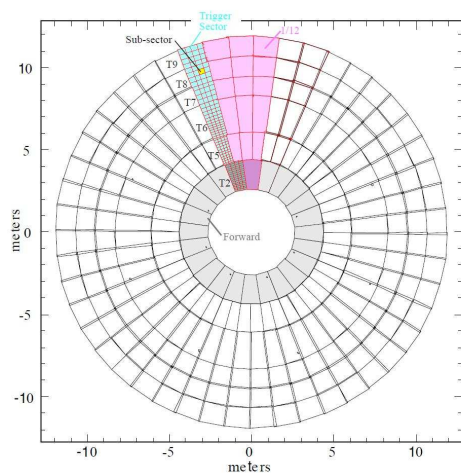


図 3.9: TGC3 のチェンバー配置 [30]

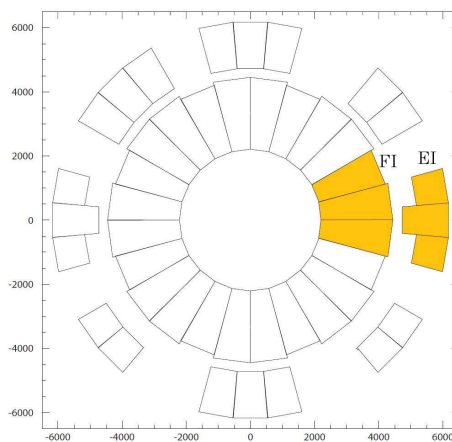


図 3.10: EIFI のチェンバー配置 [30]

3.2.1 チャンネルのオーバーラップ

隣り合うチェンバーの境界のチャンネルをオーバーラップさせることでチェンバー境界で不感領域ができるのを防いでいる。R 方向のワイヤーでは、隣り合うチェンバーのオーバーラップ部のチャンネルは OR をとっている。オーバーラップするチャンネル数は場所によって異なる (3.12 図)。

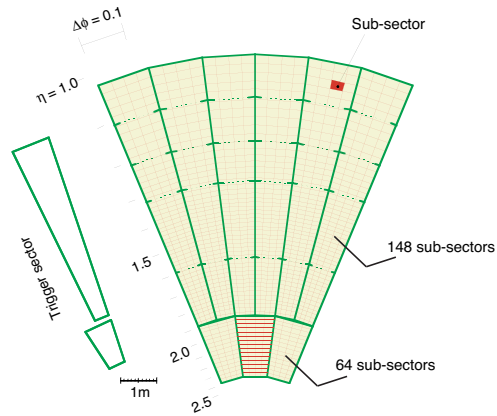


図 3.11: トリガーセクターとサブセクター [13]

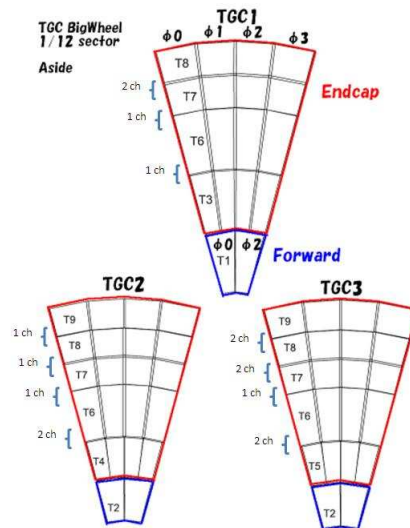


図 3.12: Wire のオーバーラップ [31]

チェンバー境界の脇に数字で示したものが OR がとられているチャンネルの数である。

一方、 ϕ 方向では、 ϕ 方向に隣り合うチェンバーは異なるトリガーセクターに属するので、ワイヤー、ストリップ共に OR はとっていない。しかし TGC3 ではヒットのダブルカウントを防ぐために、3.13 図のようにオーバーラップしているチャンネルをマスクしている。

オーバーラップという観点からは少々異なるが、TGC2 と TGC3 のストリップでコインシデンスをとるところで、TGC2 のチェンバーは一つ下のチェンバーと OR をとったあとにコインシデンスがとられる。3.14 図はそれを示したものである。

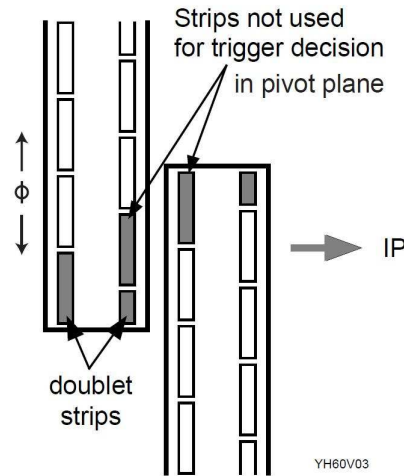


図 3.13: TGC3 での Strip のオーバーラップ [13]

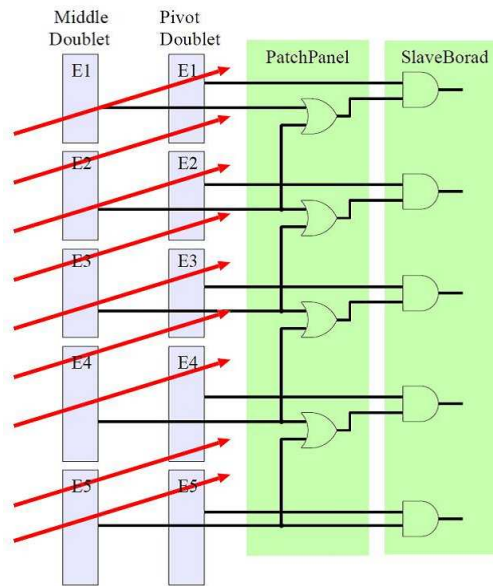


図 3.14: TGC2、TGC3 におけるストリップ信号の OR とコインシデンス [13]

3.3 TGC における P_t 測定の原理

Level-1 Trigger を担当する TGC システムは 6 段階の p_T のラベル付けを行い、トリガーを発行する。よってハードウェアによって p_T の測定を行わなければならない。3.15 図は TGC における p_T 測定の原理を表した図である。ここでは簡単にその原理について解説する。

まず TGC3 上のヒット位置と衝突点を結んだ無限大運動量トラック (Infinite p_T line) を考える。

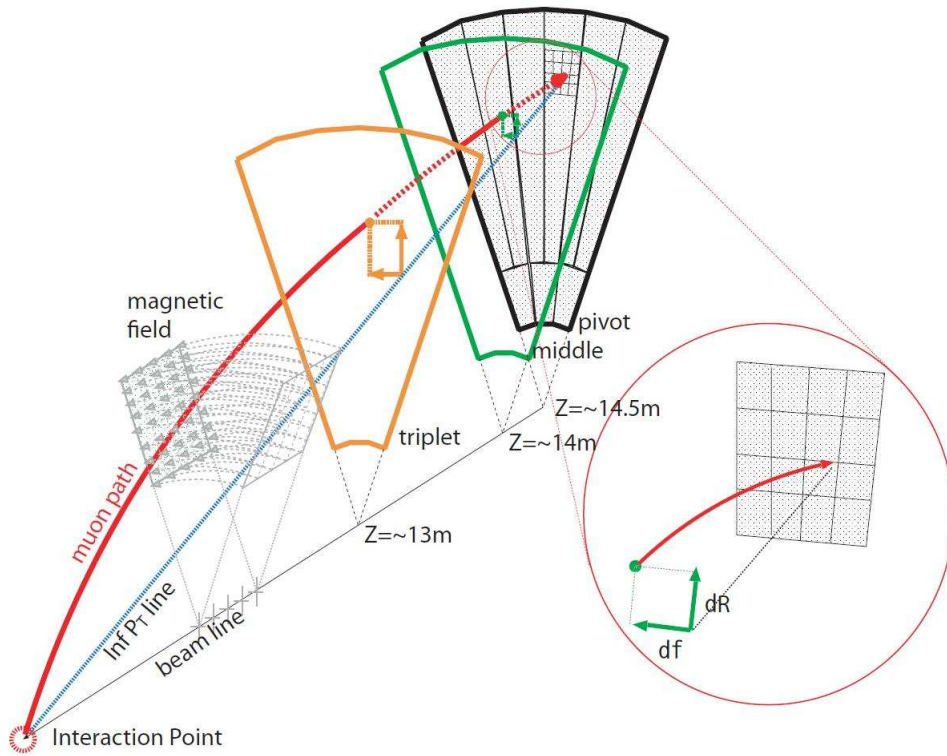


図 3.15: Pt 測定 の原理

TGC2 上でのヒットの位置から R 方向、 ϕ 方向それぞれの δR 、 $\delta\phi$ を測定する。エンドキャップトロイドの磁場は一様でなく、粒子は ϕ 方向にも曲がるので、 ϕ 方向も測定する必要がある。次に、TGC1 にもヒットがあった場合、TGC1 上での δR 、 $\delta\phi$ を測定する。実際には TGC2 上での δR 、 $\delta\phi$ の測定を行うのは SLB と呼ばれるエレクトロニクスで、TGC1 上での δR 、 $\delta\phi$ の測定を行うのは HPT と呼ばれるエレクトロニクスである。 δR 、 $\delta\phi$ の値は SL と呼ばれるエレクトロニクスに送られ、そこで R 方向、 ϕ 方向のデータが合わせられ、コインシデンスウィンドウと照らし合わされて p_T の閾値が決められる。コインシデンスウィンドウは LUT (Look Up Table) と呼ばれるテーブル状のデータとして SL に保存されている。参照する LUT を変えることで、6 段階の p_T 閾値の変更が容易にできる。

TGC システムでは 6 GeV 以上が Low- p_T トリガーで 20 GeV 以上が High- p_T トリガーとなっている。

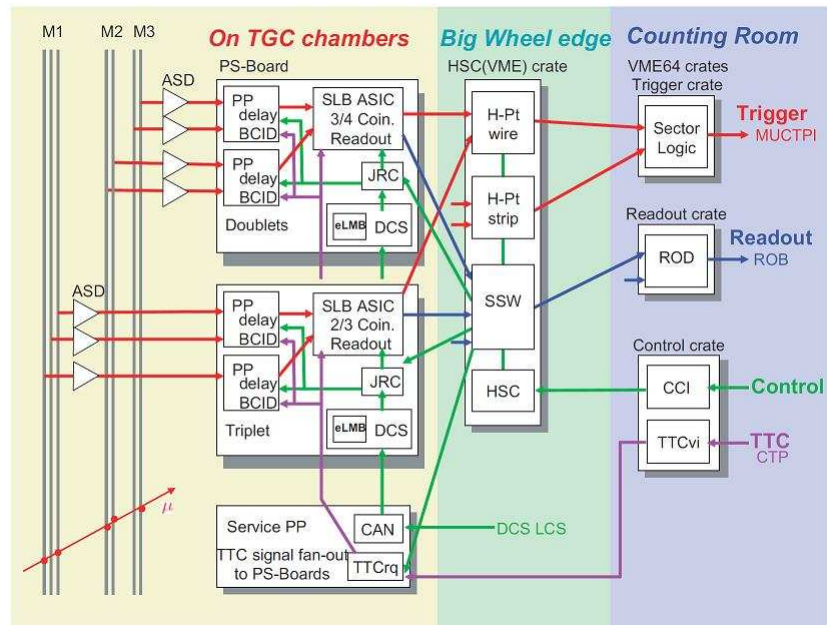


図 3.16: TGC エレクトロニクスシステムの全体像 [30]

3.4 TGCのエレクトロニクス

3.16 図は TGC エレクトロニクスシステムの全体図である。TGC のエレクトロニクスは設置される場所から大きく以下の 3 つに分けられる。

1. On-detector エレクトロニクス
2. ビッグウィールのエッジに置かれるミニラックにインストールされるエレクトロニクス
3. カウンティングルーム (USA15) にインストールされるエレクトロニクス

1、2. にインストールされるエレクトロニクスは放射線耐性が求められる。よって、1、2. にインストールされるエレクトロニクスにはカスタムメイドの ASIC や Anti-fuse FPGA などが用いられる。また、ROD(Read Out Driver) を除くすべてのエレクトロニクスが 40.08 MHz のクロックで動作する。

TGC のエレクトロニクスは設置される場所の他にも、機能によって以下の 3 つにも分けられる。

- トリガー系
- リードアウト系
- コントロール系

以下ではこの 3 つの機能とそれを構成するエレクトロニクスについて解説をする。

3.4.1 トリガー系

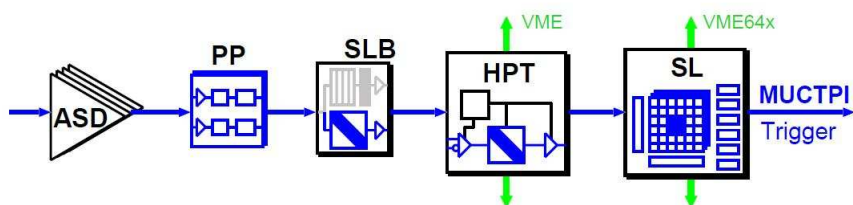


図 3.17: トリガーラインのモジュール構成 [30]

3.17 図はトリガーラインの流れとそれを構成するエレクトロニクスを表している。トリガーラインでは、TGC 全体が揃ってトリガーを発行できなければいけないのでタイミング調整が重要となる。

ASD (Amplifier Shaper Discriminator) によって増幅、整形、デジタル化された信号は LVDS (Low Voltage Differential Signal) ケーブルを通して PS Board (Patch panel and Slave board ASIC board) の中の SPP (Service Patch Panel board) に送られる。PP (Patch Panel) ASIC では、各チャンネルに粒子が飛来するまでにかかる時間 (= TOF) やケーブル遅延などから生じるタイミングのずれを吸収し、バンチ識別 (BCID、Bunch Crossing Identification) を行う。SLB (Slave Board) ASIC では PP ASIC からの出力をもとにコインシデンス処理が行われる。TGC2-3 のワイヤー、ストリップでは 3 out of 4 のコインシデンスがとられ、TGC1 ではワイヤーだと 2 out of 3、ストリップだと 1 out of 2 のコインシデンスがとられる。HPT (High- p_T Board) では TGC1、TGC2-3 のコインシデンスの結果をもとにさらにコインシデンス処理を行い、HPT トリガー判定を行う。SL (Sector Logic) ではこれまでワイヤーとストリップ別々に処理してきた情報を合わせて最終的なトリガー判定を行う。

3.4.2 リードアウト系

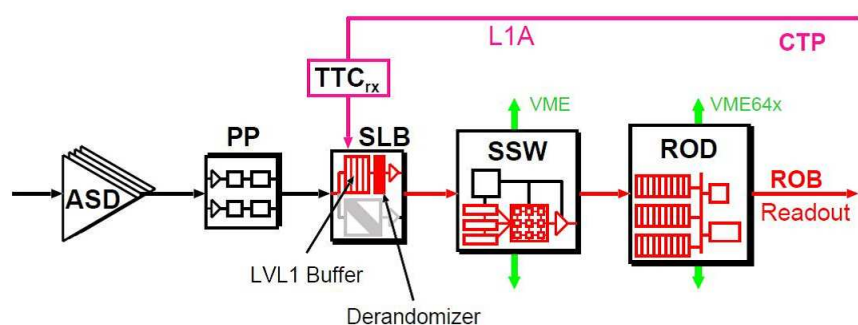


図 3.18: リードアウトラインのモジュール構成 [30]

リードアウト系は TGC のデータを読み出すためのシステムである。リードアウトラインの流れとそれを構成するエレクトロニクスを表したものが 3.18 図である。

PP ASIC から送られてきたデータは、SLB 中のレベル 1 バッファに蓄えられ、128 clock(= 3.2 μ s) の間トリガーの処理が行われ、L1A 信号が来るのを待つ。SLB は L1A 信号を受けたデータとその前後 1BC のデータ、さらにトリガー情報もあればそれらを全て SSW(Star Switch) に送る。SSW は複数の SLB からのデータをまとめ、ゼロサプレスによって圧縮を行った上で、G-Link を通して ROD にデータを送る。ROD はさらに複数の SSW からのデータを一つにまとめ、S-Link を用いて ROS にデータを送る。HPT、SL のトリガーデータは SL 上の SLB から読み出され SSW に送られ、同様に読み出しを行っている。

3.4.3 コントロール系

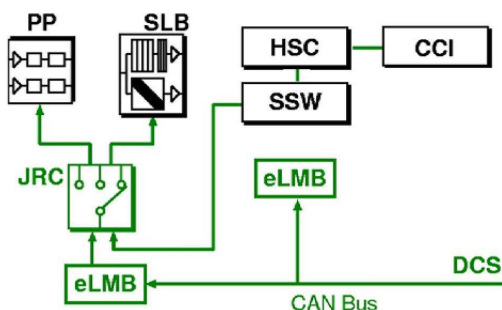


図 3.19: コントロールラインのモジュール構成 [30]

実験が開始すると、実験ホール内は放射線環境になってしまうため、遠隔でエレクトロニクスと検出器をコントロールする必要がある。それを行うのがコントロール系である。3.19 図はコントロールラインの信号の流れと、それを構成するエレクトロニクスを表している。

ミニラック中のモジュールのコントロールはカウンティングルームにある CCI(Control Configure Interface) から HSC(HPT SSW Controller) を通じて行う。PS Board 上の SLB、PP のコントロールはさらに SSW を介して行う。PS Board 上の JRC(JTAG Routing Controller) が PP、SLB の経路選択を行う。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 DCS(Detector Control System) が導入されている。TGC では ADC(Analog to Digital Converter)、DAC(Digital to Analog Converter)、eLMB(embedded Local Monitor Box)などを搭載した DCS カードが PS Board に配置されている。DCS の操作は専用の CAN バスで行われる。TGC の DCS は温度状態の管理や供給電源の監視の他に、JRC のコントロールが可能なので、CCI-HSC、SSW-JRC 間に不具合が起きた時に CAN バス経由で PP、SLB のコントロールをすることが可能である。

3.5 エレクトロニクスの詳細

3.5.1 ASD(Amplifier Shaper Discriminator)

ASD(Amplifier Shaper Discriminator) は TGC チェンバーの読み出し口に直に取り付けられている。一つのモジュールには ASD ASIC が 4 つ搭載されていて、ASIC 一つが 4 チャンネル分の信号処理ができるので、一つのモジュールで 16 チャンネル分の処理を行うことが可能である。ASD ASIC はチェンバーからのアナログ信号を増幅 (Amplify)、整形し (Shape)、ある閾値電圧を越えた信号をデジタル化し (Discriminate)、LVDS 信号として出力する。また ASD ASIC はエレクトロニクスの動作チェックやタイミング調整のために、PP からのテストパルスを受け取ることも可能である。ASD と PP の接続にはツイストケーブルが使用される。ASD には 16 チャンネル目のアナログ信号をモニターするための ADC もあり、DCS を介してその情報にアクセスすることも可能である。3.20 図が ASD の写真、3.21 図が ASD ASIC のダイアグラムである。

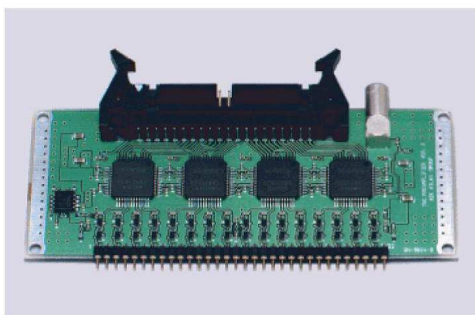


図 3.20: ASD の写真 [30]

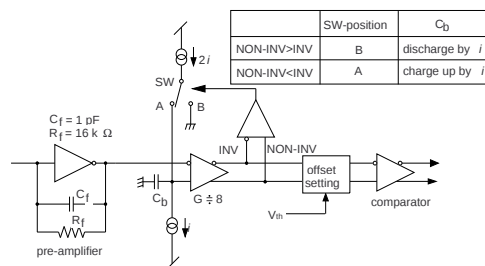


図 3.21: ASD ASIC のダイアグラム [6]

3.5.2 PS Board

PS Board には 3.23 に示されているように PP、SLB、JRC、DCS が搭載されている。PS Board は ASD に電源電圧と閾値電圧を供給する。また、SPP から供給される LHC クロックや各種リセット信号、トリガー信号を Board 上のエレクトロニクスに分配している。3.22 図が PS Board の写真で、3.23 図は PS Board のダイアグラムである。

PP(Patch Panel)

PP(Patch Panel) ASIC ではタイミングの調整とバンチ識別 (BCID) を行う。3.24 図は PP ASIC のブロックダイアグラムである。ASD から送られてくる信号はチャンネルによって TOF が異なり、ケーブルの長さも一様でないため、タイミングが揃っていない。そこで、PP ASIC には可変ディレイが搭載されている。サブナノ秒 (0.83 ns) 単位で信号を遅延させることができる PLL を利用したディレイが実装されている。さらに BCID 回路の前にはクロック (25 ns) 単位での遅延も可能な回路が実装されている。また、ASD に向けてテストパルスを送る機能も搭載している。テストパルスの



図 3.22: PS Board [30]

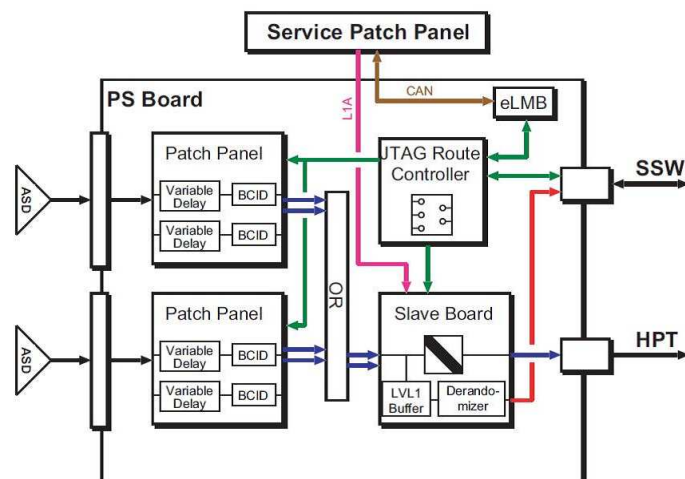


図 3.23: PS Board のダイアグラム [30]

振幅は可変で、さらにテストパルスを送るタイミングもクロック単位、サブナノ単位で設定できる。テストパルスの制御は JTAG プロトコルによって行う。

SLB(Slave Board)

3.25 図は SLB ASIC のブロックダイアグラムである。SLB には大きく分けてトリガーとリードアウトの機能がある。

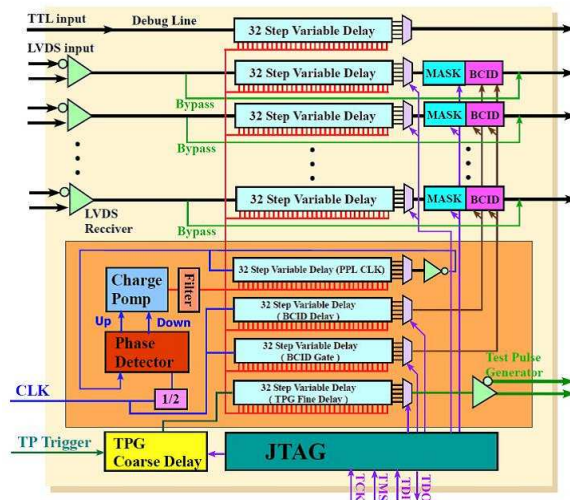


図 3.24: PP ASIC のブロックダイアグラム [30]

SLB ASIC のトリガー部

トリガー部は主にコインシデンスマトリックス回路からなる。コインシデンスマトリックスの仕様は Doublet Wire、Doublet Strip、Triplet Wire、Triplet Strip、EIFI に応じて異なる。Doublet の SLB では 3 out of 4 のコインシデンスをとり、さらに δR 、 $\delta \phi$ の値を決める。 δR は 15bit のデータ幅があり、 -7 から 7 のまでの値を保持することができる。 $\delta \phi$ は 7bit のデータ幅があり、 -3 から 3 までの値を保持することができる。ただし、このときの単位は読み出しチャンネル一つ分ではなく、スタグガリングを考慮した実際のチャンネルの $1/2$ の granularity の大きさになっている。Triplet Wire では 2 out of 3 コインシデンスがとられ、Triplet Strip では 1 out of 2 コインシデンスがとられる。連続したチャンネルにヒットがあった場合は、その中の 1 チャンネルを選択してトリガー判断に用いようになっており、その機能をデクラスタリングと呼ぶ。デクラスタリングの概念を 3.26 図に載せる。

SLB ASIC のリードアウト部

リードアウト部はデータの読み出しを行うための機能からなる。レベル 1 バッファとデランダムイザにより構成される。チェンバーからのデータは L1A が到着するまで、トリガーデータと共に $212 \text{ bit} \times 128$ 段のシフトレジスタに一旦蓄えられる。 212 bit の内訳はヒットデータ 160bit、トリガーデータ 40bit、BCID や L1ID 等のタグが 12bit となっている。L1A を受け取ると、該当バンチとその前後 1 バンチ分のデータがデランダムイザにコピーされる。デランダムイザは 3 バンチ分のデータを別々にシリアル変換し、SSW に送る。

これ以外にも SLB ASIC 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、チャ

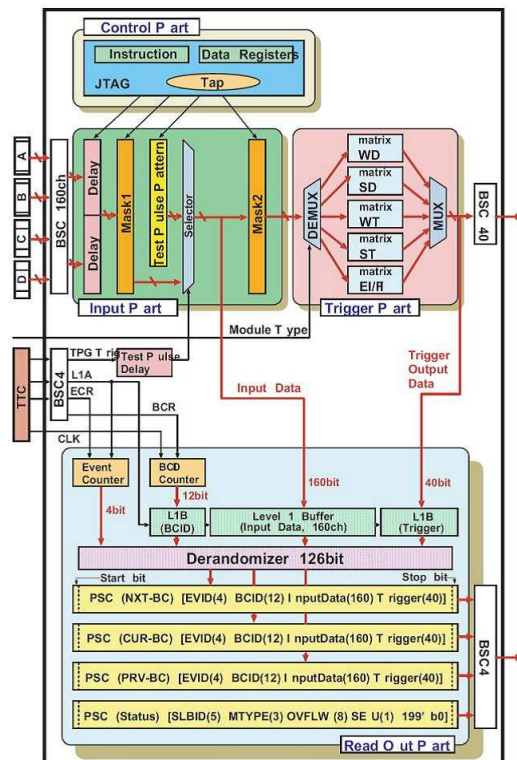


図 3.25: SLB ASIC のブロックダイアグラム [30]

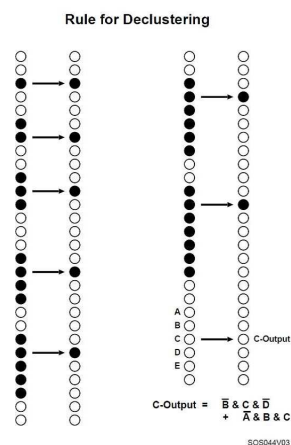


図 3.26: デクラスタリングの概念図 [13]

ンネル単位でマスクする機能、SLB 以降のエレクトロニクス動作確認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などが搭載されている。各種設定は JTAG 信号で行う。

JRC(JTAG Routing Controller)

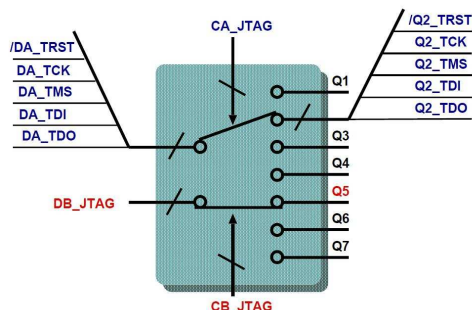


図 3.27: JRC 内部の構造 [30]



図 3.28: DCS board の写真 [30]

JRC(JTAG¹ Routing Controller) は JTAG ポートのスイッチングルータである (3.27 図)。入力が SSW と eLMB からの 2 系統あり、出力が PP と SLB 用に 7 系統ある。7 系統の内、4 つが PP ASIC へのもので、3 つが SLB ASIC へのものである。JTAG のラインとしては TCK、TMS、TDI、TDO、TRST を使用する。

JRC は放射線耐性を持たせるため Anti-fuse FPGA を使用している。

DCS(Detector Control System board)

DCS(Detector Control System board) は eLMB と呼ばれるサブモジュールを使用し、ADC や DAC が搭載され、センサーからのアナログ信号をモニターしたり、制御用のアナログ信号を作り出せる。TGC では DCS を温度モニタ、(エレクトロニクス用の)LV 電圧・電流モニタ、アノードワイヤーに印加する HV の電圧・電流モニタとコントロール、ASD の閾値電圧の設定を行う。eLMB は CAN バス²を通してデバイス上のデータの読み書きが行える。また、JRC を利用して PP と SLB のコントロールも可能である。3.28 図に DCS の写真を載せる。

3.5.3 SPP(Service Patch Panel)

SPP(Service Patch Panel) は TTC(2.6.2 参照) の信号を受信する機能を持つ TTCrx³が搭載されている。TTC システムからの LHC Clock、L1A、BCR、ECR、Test Pulse Trigger 等の信号を受け取り、PS Board に分配する。3.29 図が SPP の写真である。

¹JTAG(Joint Test Action Group) は、集積回路や基板の検査、デバッグなどに使えるバウンダリスキャンテストの標準方式である。JTAG に対応した端子及びレジスタを構成し、外部からテストコードを入力することでテストを行うことが可能である。

²CAN(Controller Area Network) はマルチマスターのシリアルバスである。アドレスの概念がなく、送信側は識別子を付けてメッセージを配信し、受信側はその識別子を見てメッセージを受信するかを決める。

³TTCrx は TTC が配布するシグナルを受信するレシーバーモジュールである。

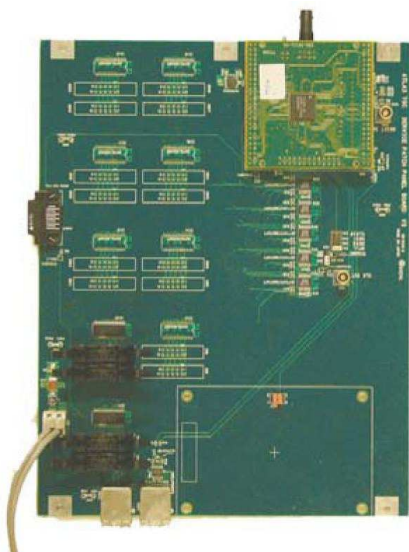


図 3.29: SPP の写真 [30]



図 3.30: PS Pack の写真 [30]

3.5.4 PS Pack

PS Board と SPP は一つの PS Pack にまとめられ、トリプレットの前面、ダブレットの背面に設置される。PS Pack は 1/12 毎に 2 つ設置される。一つの PS Pack 内には一つの SPP と、トリプレットの場合は 10 枚、ダブレットの場合は 17 枚の PS Board が格納される。3.30 図の写真である。

3.5.5 HPT(High- p_T Board)



図 3.31: HPT の写真 [30]

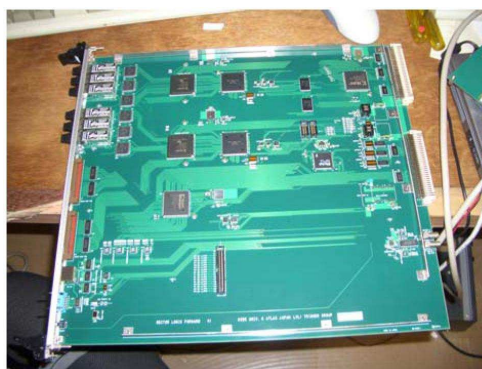


図 3.32: SL の写真 [30]

HPT(High- p_T Board) は 9U サイズの VME スレーブモジュールで、ウィールのエッジのミニラックにインストールされる。3.31 図が HPT の写真である。End-cap Wire、End-cap Strip、Forward の 3 種類が製造された。HPT はダブレットとトリプレットの情報を用いて High- p_T トリガー判定を行う。HPT では、SLB ASIC まで独立に処理されてきたダブレットとトリプレットのデータを統合して HPT コインシデンス情報を生成する。トリプレットは 2 つのダブレットよりも衝突点に近く、しかもダブレット同士の間隔よりも離れた位置に設置されているため、トリプレットを用いることによりトロイダルマグネットによってあまり曲げられることがなかった大きな p_T を持つミューオン信号を選別できる。HPT ではワイヤとストリップは独立に処理が行われ、 δR 、 $\delta\phi$ を出力する。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、カウンティングルーム (USA15) にある SL に送信される。HPT で処理されたデータは SL にある SLB ASIC からリードアウトされる。

3.5.6 SL(Sector Logic)

SL(Sector Logic) は 9U サイズの VME スレーブモジュールでカウンティングルーム (USA15) にインストールされる。3.32 図が SL の写真である。SL は TGC エレクトロニクスシステムのトリガー最終段に当たるモジュールである。一枚で 2 トリガーセクター分の信号を処理する。

3.33 図は SL 内部のブロックダイアグラムである。SL は主に R- ϕ コインシデンス、プレトラックセクター、ファイナルトラックセクターから構成される。まず SL は、HPT から送られてきたシリアルライズされているオプティカル信号を受け取り、電気信号に変換した後パラレル変換をする。そして、HPT まで独立に処理されていた R 方向 (ワイヤ) と ϕ 方向 (ストリップ) の HPT 信号から両者のコインシデンス (R- ϕ コインシデンス) を取ることで、ミューオンのトラックを構築する。それらのトラックを SSC(Sub-sector Cluster⁴) ごとに 6 段階の p_T 閾値によって分類する。6 GeV 以上でトリガーを出すということは決められているものの、閾値の設定は LUT(Look-up Table) の書き換えによって変更できる。Track Pre-Selector は 6 段階の p_T それぞれにつき 2 つのトラックを選ぶことができる。Track Pre-Selector からくる 12 個のトラックのうち p_T の大きいもの 2 つに p_T のラベルを付け、ROI 情報と共に MuCTPI に送る。SL で処理されたデータは SL 上の SLB ASIC(SLB on SL) から読みだされ、SSW に送られる。このとき同時に HPT から送られてきたデータも読みだされる。

3.5.7 SSW

SSW(Star Switch) は 9U の VME スレーブモジュールでウィールのエッジのミニラックにインストールされる。3.34 図が SSW の写真である。

SSW の主な役割は、複数の SLB から送られてくるリードアウトデータを圧縮し、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。SSW までデータは全く圧縮されていない。データを圧縮するために、SSW ではゼロサプレスをを行っている (3.37 図がゼロサプレスのスキーム図である)。データをセルと呼ばれる 8bit に分け、各セルにアドレスを

⁴R 方向に 2 つ、 ϕ 方向に 4 つのサブセクターの集まり。

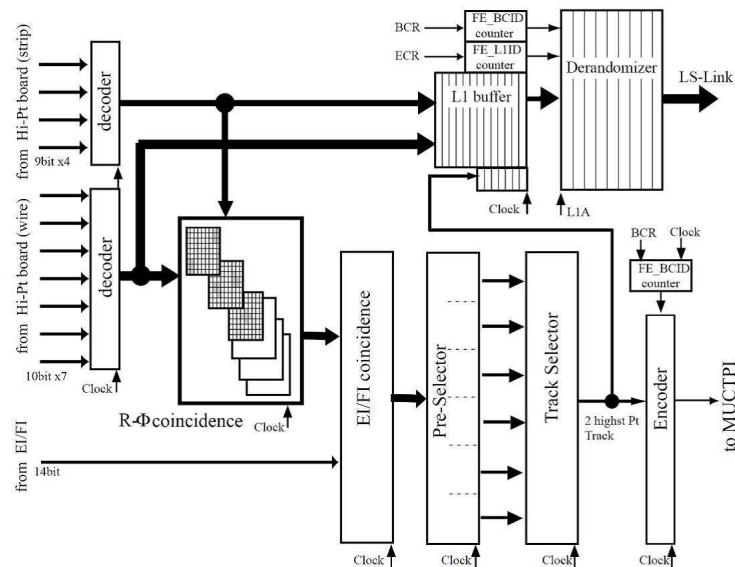


図 3.33: SL のブロックダイアグラム [13]

付加し、値が全てゼロでないセル情報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、これによりデータを減らすことが出来る。

3.36 図は SSW のブロックダイアグラムである。VME コントローラが 1 個、Rx FPGA(レシーバ) が 4 個、Tx FPGA(トランシーバ) が 1 個搭載されている。Rx FPGA 内にはさらに 4 個の Rx ロジックがある。1 つの SSW で最大 23 個の SLB のデータを受けとることが可能だが、現在はそのような接続の仕方にはなっていない (SSW と PS Board の接続表を付録 B に載せる)。Rx FPGA にてデータの圧縮が行われた後に、Tx FPGA(トランスミッター) に送られ、フォーマットされる。フォーマットされたデータはオプティカル信号に変換され、G-Link を用いてカウンティングルームにある ROD に送られる。また SSW は、PSB 上の PP、SLB に JTAG プロトコルによってレジスタ設定と、I2C⁵ による SPP 上の TTCrx の設定も行う。

3.5.8 ROD(Read Out Driver)

ROD(Read Out Driver) は VME の 6U モジュールでカウンティングルームにインストールされている。3.35 図が ROD の写真である。一つのモジュールが 1/12 セクターつ分の SSW(= 10 個) からの入力を受ける。よって ROD は片サイドで 12 枚ある。

ROD は SSW からのデータを一旦 FIFO に格納し、トリガー情報を元に同じイベントごとにまとめ、ヘッダーとトレーラーをつけて一つのイベントデータにまとめる。そしてそのデータを S-Link

⁵ フィリップス社によって開発されたシリアルバス。抵抗でプルアップされた双方向のオープンコレクタ信号線によって通信を行う。

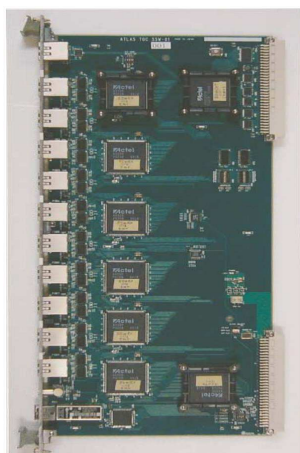


図 3.34: SSW の写真 [30]



図 3.35: ROD の写真 [30]

を用いて ROS に送る。ROD はイベントビルディングを行うときに TTC からのトリガー情報が必要となるので、TTCrx を搭載したメザニンカードを使って TTC からの信号を受信している。

3.5.9 CCI(Control Configure Interface)

CCI(Control Configuration Interface) は 6U サイズの VME スレーブモジュールで、カウンティングルームにインストールされる。3.38 図が CCI の写真である。CCI は G-Link を用いて実験ホールにある HSC とつながっている。CCI 中に搭載されたレジスタの読み書きを行うことで、HSC がインストールされているクレートのコントロールをすることができる。また、CCI は VME の優先割り込みにも対応している。

3.5.10 HSC(HPT SSW Controller)

HSC(HPT SSW Controller) は 9U サイズの VME マスタモジュールで、HPT、SSW と同じクレートにインストールされる。3.39 が HSC の写真である。実験ホールに置かれるため、耐放射線性が求められる。CCI からの命令を G-Link で受け取り、VME のマスタとしてスレーブに指示を出す。HSC には VME バスの他にも JTAG バスも搭載されている。このため、CCI との情報のエンコード、デコードのために PPE(Primary Protocol Encorder) という JTAG 用のものと、SPE(Secondary Protocol Encorder) という VME 用のものが実装されている。

私はこの CCI-HSC を改良したコントロールシステムの開発にも携わった。その概要について付録 E に載せる。

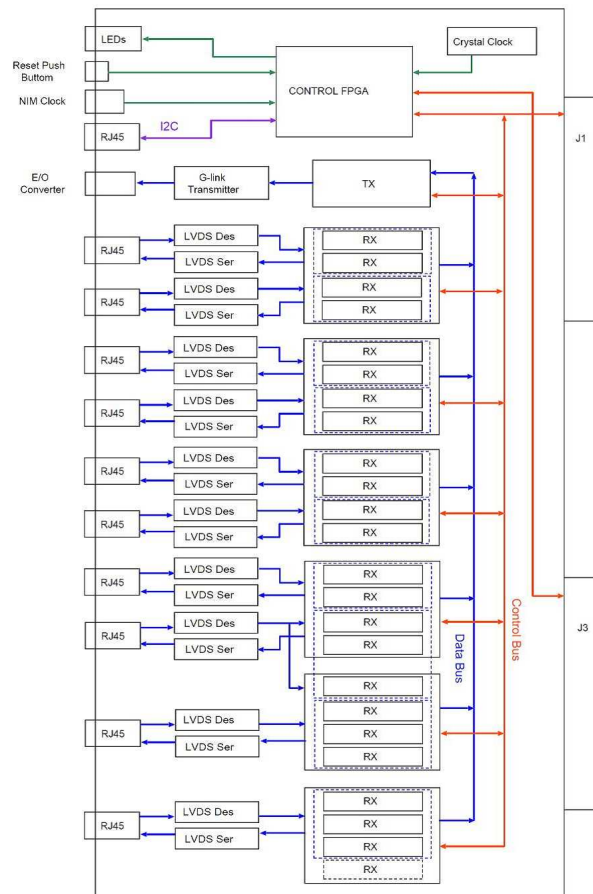


図 3.36: SSW のブロックダイアグラム [17]

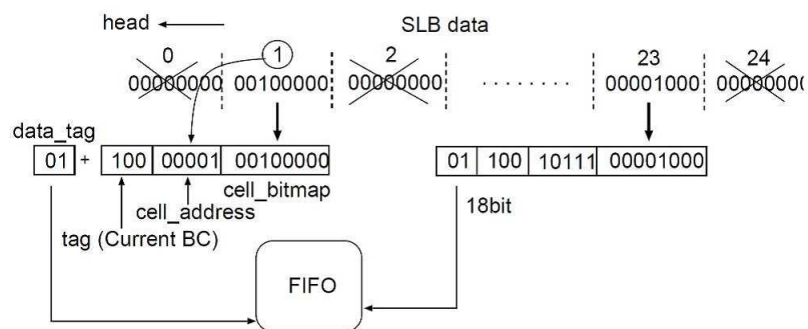


図 3.37: SSW のゼロサプレスのスキーム [17]

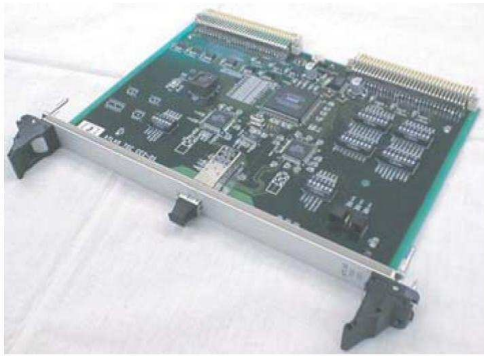


図 3.38: CCI の写真 [30]

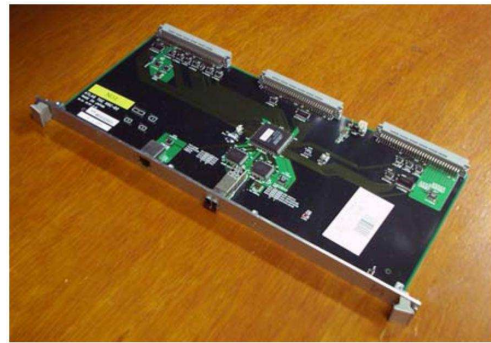


図 3.39: HSC の写真 [30]

第4章 TGCシステムのコミッショニング

4.1 タイミング調整の重要性

ATLAS 検出器において Level-1 Trigger を担当する TGC システムは、40 MHz で起きる衝突の中から興味のある bunch を一意的に特定できなければならない。さらに TGC システム全体としてタイミングが調整され、粒子が飛来する位置によって不定性があるといけない。エンドキャップの広大な領域を占め、各エレクトロニクスも位置も様々である TGC システムにとって、タイミング調整は重要な課題であるとともに、チャレンジングな課題である。私はタイミングと調整の中でも最後の調整となる TGC システム内で使用しているクロックとバンチ衝突の間の位相調整 (以下では Clock Phase Scan と呼ぶ) を行う VME のモジュールを開発した。

以下では TGC システムにはどのような機構が備わっていて、どのようにシステム内でタイミングをそろえているのかについて触れたあと、私が Clock Phase Scan のために開発した VME のモジュールの詳細とどのように Clock Phase Scan を行うのかについて言及する。

4.2 TGCシステムのタイミング調整

4.2.1 タイミング情報がどのように分配されるか

ATLAS 実験において、クロック信号や L1A などのタイミングに関するシグナルは 2.6.2 に示したように TTC システムを用いて各検出器に分配される。TGC ではカウンティングルーム (USA15) に LTPi、LTP、TTCvi、TTCex がインストールされたクレートがあり、TTC クレートと呼ばれている。TTC クレート中の TTCex モジュールからオプティカルファンアウトを通してフロントエンドに配布されている。TGC では SPP に搭載されている TTCrx モジュールがシグナルを受け取り、フロントエンドのエレクトロニクスに使用できるように電気信号に変換している。

4.2.2 TGC システム内でのタイミング調整

TGC システム内でのタイミング調整において一番重要だと言えるのは、PP ASIC でのタイミング調整だ。TGC システムのエレクトロニクスは On-Detector と Off-Detector の大きく二つに区別されるが、上流に位置するのは On-Detector のエレクトロニクスで、On-Detector のエレクトロニクスの中での個々の接続は、ASD→PP→SLB となっている。PP ASIC では衝突に対応して 25 ns ごとに開く BCID ゲートにその bunch 衝突で生じたミューオンの信号を収める処理 (BCID) を行っている。PP の後にシグナルは SLB のコインシデンスマトリックスに入るので PP の時点でのタイミ

ング調整が重要となる。しかし、ミューオンが衝突点から TGC チェンバーに届くまでの TOF は、TGC がカバーする領域が大きいために光速で粒子が走ったとしても 19 ns ほどの差があり (図 4.1 参照)、さらに ASD から PP に送るケーブルもチェンバーとエレクトロニクスの位置関係により 1.8 m ~ 12.5 m ほどの長さの違いがある (4.2 図参照)。このように PP ASIC のバンチ識別回路にシグナルが到達するタイミングは各々のチャンネルで一様でない。

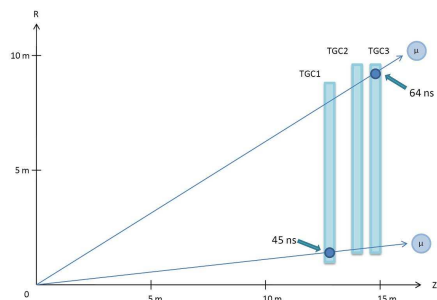


図 4.1: TOF [29]

図のように TGC1 の R のもっとも小さいチャンネルで TOF は約 45 ns で、一方 TGC3 の R の一番大きいチャンネルで TOF は約 64 ns である。その差は約 19 ns である。

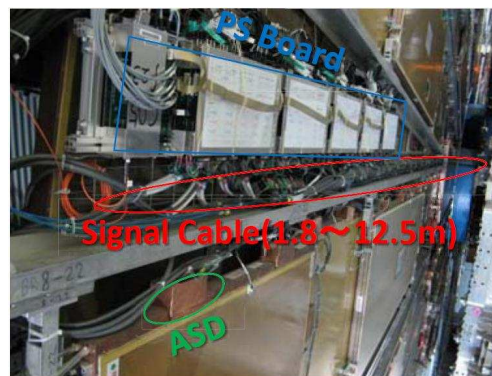


図 4.2: シグナルケーブル [29]

各 AS からの信号は様々な長さのシグナルケーブルを通して PP に届く

この問題を解決するために PP ASIC には ASD からの信号をサブナノ秒 (25/30 ns ~ 0.83 ns) 単位で遅らせる Signal Delay とクロック単位 (25 ns) での調整が可能な BCID Delay が実装されている。これらの Delay を駆使し、PP ASIC 上で全てのチャンネルからくるシグナルのタイミングを調整しなければならない。我々は PP ASIC に搭載されているテストパルスを生成する機構を用いて全ての ASD-PP 間のタイミングの調整を行った。TGC システムには後段の回路にもテストパルスを生成する機構が備わっているために、このテストパルスはそれらと区別するために ASD テストパルスと呼ばれる。ASD-PP 間での ASD テストパルスの生成と信号受信の概略図を 4.3 図に示す。

ASD テストパルスは Test Pulse Trigger が届いた後に一定の時間間隔を置いて (Delay をかけて) シグナルを ASD に送ることができる。この Delay も 2 種類あり、Coarse Delay と Fine Delay と呼ばれる。Coarse Delay は BCID Delay のようにクロック単位での設定が可能で、Fine Delay は 25/30 ns での設定が可能である。この二つの Delay の設定値を

$$\text{the cable length} \times \text{the propagation speed of signal} - \text{the TOF} \quad (4.1)$$

とすることで、特定のチャンネルでの IP から来る無限運動量を持ったミューオンによって生成されたシグナルをシミュレートすることができる。TOF は ASD がカバーするジオメトリの情報から計算することができるので、この方法を用いることで ASD-PP 間でのシグナルの伝播遅延を求めることができる。

ここで注意しなければいけないのは、シグナルが長いケーブルを伝わっていくときには減衰が起きることである。4.4 図に示すように長いケーブルほどこの効果が顕著に見られる。この減衰

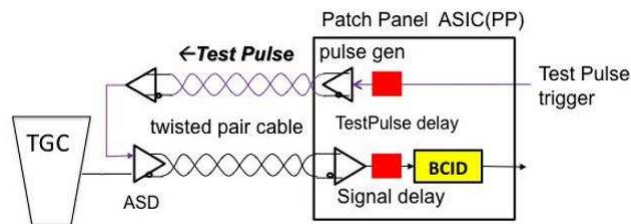


図 4.3: ASD テストパルスの生成と信号受信の概略図 [10]

PP ASIC で生成されたシグナルは ASD に送られ、ASD はそのシグナルを送り返す。注意すべきはテストパルスはケーブルを往復しているということである。

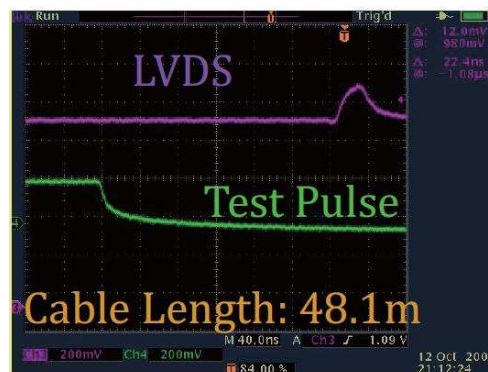
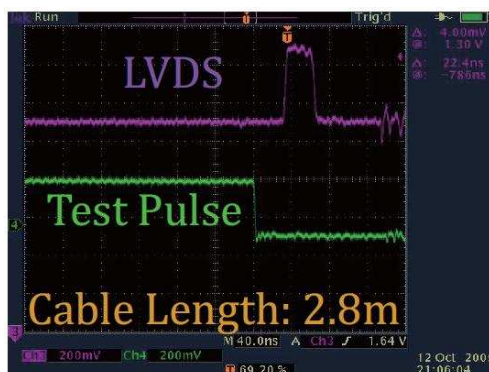


図 4.4: 2.8 m と 48.1 m のケーブルで観測された信号の減衰 [10]

”LVDS”は PP ASIC の直前で測定された ASD からの出力信号を表す。また、”Test Pulse”は PP ASIC で生成された信号を ASD に入力される前に測定したものである。

の効果については体系的な研究がすでになされており、現在では 1 m につき 0.2 ns という Delay をテストパルスの方に余分にかけることこの効果も考慮に入れている。

上記のシグナルの減衰の効果も考慮に入れて取得したタイミングの分布を図 4.5 に示す。ASD-PP 間をつなぐケーブルは 10008 個あるが、一つのケーブルで 2 つの ASD を担当し、ASD ごとにタイミングの設定が可能なので、調整できるパラメータの数としては 20016 個になる。しかし、実際には Threshold がうまくかからない ASD や Noisy な ASD は除外するため、図 4.5 のエントリーは 20016 より少なくなっている。まず青色で示されたヒストグラムはケーブルを作った会社から送られてきた情報を信用して測定した分布である。測定をしてみると、予想に反してタイミングの分布が広いことがわかった。そこで 834 あるタイプごとに測定してみると、タイプによって予想していた値からのずれがあることがわかった。このケーブルタイプごとの予想値からのずれも考慮したのが赤色で示されているヒストグラムである。青色の分布より分布の幅は小さくなったが、まだいくつかの ASD では ± 4 ns の領域より外にエントリーがあるため、正しくタイミングを調整できていないもの

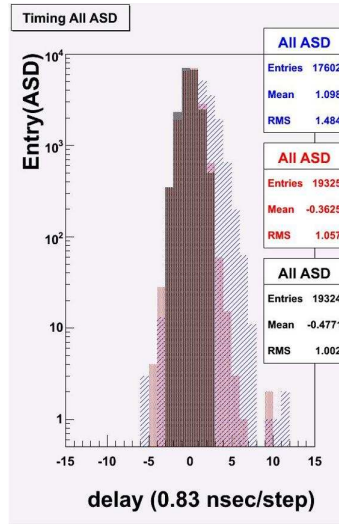


図 4.5: ASD のタイミング分布 [10]

3 つの異なるケーブル長の補正の結果を示す。1bin は 0.83 ns に相当する。

があることがわかった。そのため、我々は個々のケーブルの長さを実際に測ることにした。そして得られたのが、黒色で示された分布である。最終的に ± 2.5 ns 以内に全ての ASD のタイミングを収めることができた。

PP ASIC にて bunch 識別が行われた後も、トリガー判定を行う各モジュール間を接続するケーブルは様々な長さをとるため、正しくコインシデンス処理を行うためにはクロック単位でのタイミング調整が必要である。そのために後段のエレクトロニクスにも各種の Delay が実装されている。4.6 図は Delay が実装されているモジュールを示している。

表 4.1 は PP 以後に実装されている Delay についての説明である。

Delay	Step	Description
SLB Delay	1/2 clock (12.5 ns)	可変範囲は 0 ~ 3.5 clock。異なるステーション間でのタイミングを調整する。
HPT Delay	1/2 clock (12.5 ns)	可変範囲は 0 ~ 3.5 clock。PSB と HPT 間を結ぶケーブル長の差を吸収する。
SL Delay	1/2 clock (12.5 ns)	可変範囲は 0 ~ 15.5 clock(ただしこの値は FPGA の書き換えで変更可能である)。複数の HPT から届くケーブル長の差を吸収する。
MuCTPI Delay	1 clock (25 ns)	可変範囲は 0 ~ 10 clock。異なるセクターから届く信号のタイミングを吸収する。

表 4.1: 各モジュールに実装されている遅延回路のステップと用途

表 4.1 の中で SL Delay だけは使われていない。4.7 図は HSC が置かれている実験ホールのミニ

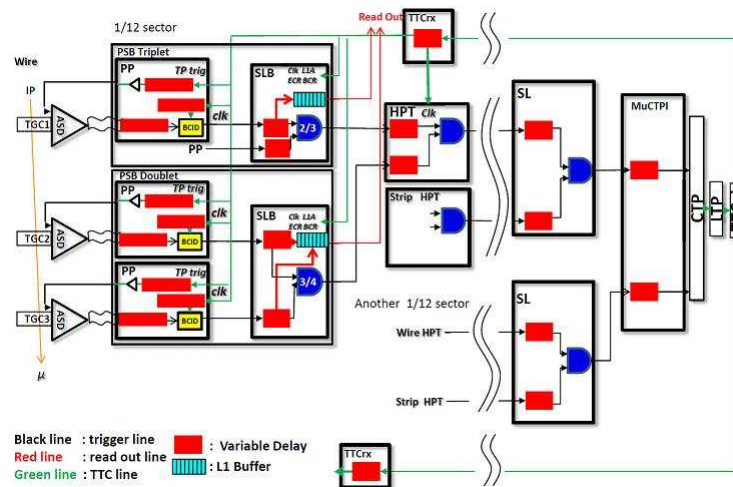


図 4.6: TGC システムに実装されている遅延回路 [29]
PP 以後も各種モジュールに遅延回路が実装されている。

ラックの位置と SL がインストールされている USA15 の位置関係を示しているが、セクター間では HSC-SL をつなぐ光ファイバーの長さに違いがあるものの、セクター内であればケーブルの長さは同じなので、同一セクターからの入力しか持たない SL ではタイミング調整は行っていない。SL の時点で各セクター内のタイミングはそろっているが、セクター間では差ができてしまう。その差を吸収するのが MuCTPI Delay である。

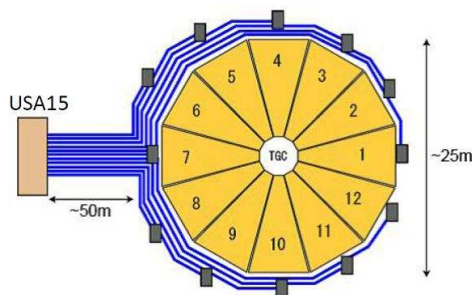


図 4.7: HSC クレートと USA15 の位置関係 [31]

Big Wheel のそばに長方形で示されたものが HSC がインストールされているミニラックを表している。セクターの位置によってファイバーの長さが違ってくる。

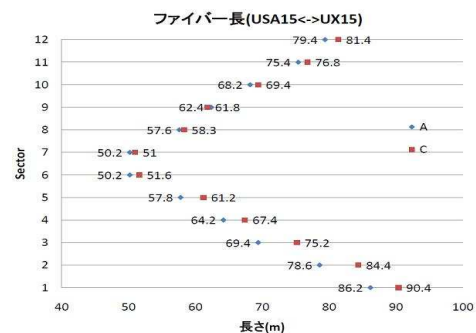


図 4.8: HSC クレートと USA15 を結ぶファイバーの長さ [31]

縦軸はセクター、横軸は光ファイバーの長さを表している。

以上のように各種の Delay の値を最適化することで TGC 全体のタイミングを揃った状態でのトリガー発行を可能にしている。

4.3 2008 年時点でのタイミング調整の結果

2008 年 9 月 10 日に LHC は最初の Beam の周回に成功した。私はそのとき Atlas Control Room(ACR)にて TGC shifter として作業していた (4.9 図参照)。



図 4.9: ACR でのシフト時の様子

First Beam の周回に歓喜したが、それから 9 日後の 2008 年 9 月 19 日にヘリウムリークのために LHC は稼動を停止してしまう [4]。

Trigger Type	Trigger Condition
TGC_HALO	TGC2 と TGC3 を使用。ビームラインと並行に走るビームハローにアクセプタンスを持たせるため、ワイヤーは TGC3 の 2 レイヤー使用し、ストリップは 2 station の 3 out of 4 コインシデンスを利用。
TGC_MU0	TGC_HALO の条件に加え、TGC1 の 2 out of 3 コインシデンスを要求したもの (ロードはフルにオープンされている)。
TGC_MU6	TGC_HALO の条件に加え、より狭いロードの領域で TGC1 の 2 out of 3 コインシデンスを要求したもの。

表 4.2: 2008 年 First Beam 時のトリガー条件

一週間弱ほどの運転で LHC は稼動を停止してしまったが、我々は貴重な First Beam でのデータを得ることができた。またこのデータから我々が行っていたタイミング調整はうまくいっていることを示す結果をが得られた。4.10 図はシングルービームのときに TGC のトリガーがどの Bunch Crossing

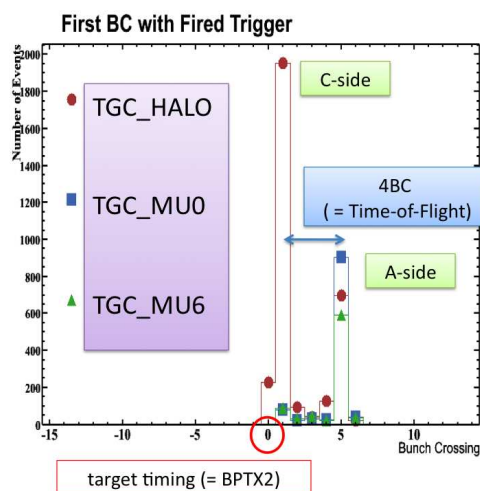


図 4.10: 2008 年 First Beam でのトリガータイミング [1]

シングルビーム時のトリガータイミングを表している。横軸は Bunch-Crossing である。

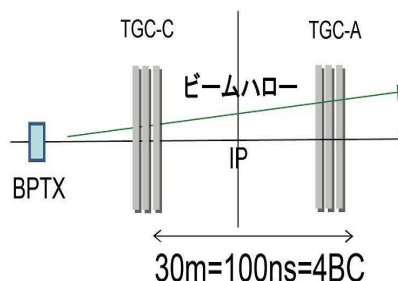


図 4.11: シングルビームでの TOF [29]

4.10 図で見られる A-side と C-side のトリガータイミングの差は、ちょうど μ 粒子が C-side から A-side まで飛ぶのにかかる時間 (TOF) と一致する。

で発行されたかを示している。4.10 図にある TGC_HALO、TGC_MU0、TGC_MU6、はトリガータイプを表している。それぞれのトリガータイプごとのトリガー条件を表 4.2 に載せる。

このデータが取られたときにはビームは C-side の側から A-side の側に向かって周回していた。Beam は ATLAS 検出器の約 100 m ほど手前のコリメーターに衝突し、たくさんの μ 粒子が飛来したと考えられている。4.10 図に見られる C-side と A-side の 4 Bunch-Crossing の差はちょうど μ 粒子が C-side から A-side までの距離 30 m を飛ぶのにかかる時間と一致している (4.11 図参照)。Delay パラメータの設定は IP から来る粒子に対して最適化されているため、C-side の TGC_HALO トリガーは 2bin にまたがっているが、このプロットから我々が行ってきた TGC システム内のタイミング調整はうまくいっていることが示された。

以上のようにして、TGC システム内のタイミング調整のよい結果が得られたが、最後の調整すべきパラメータが残されていた。TGC システム内で使用しているクロックと実際の LHC のバンチ衝突との間にはある位相差 δ だけの違いがある。この位相差の調整を行って初めて、TGC システムはバンチ衝突に同期してトリガーを発行することが可能になる。2008 年の時点ではこの位相差を調べ、また調整する機構は備わっていなかった。2009 年の LHC 再開までにできた時間を用いて、我々はこの位相差を測定し、調整を行うためのシステムを作ることにした。

4.4 Clock Phase Scanに向けて

4.4.1 Clock Phase Scan とは

TGC の信号は理想的には (チェンバー信号のジッターが 25 ns 以内ならば)、25 ns の時間間隔以内に分布するはずである。そのため、TGC システムは 3 つの連続するバンチ (Previous、Current、Next) を読み出しているが、LHC のバンチ衝突と TGC システムのクロックの位相差 $\delta = 0$ であれば、Current にのみヒットが存在するということになる。しかし、実際にはこの位相差のために本来は Current の位置にあるはずのヒットが隣のバンチにもこぼれ、Previous、Next にもヒットが存在する。

この事実より、TGC システム内を流れるクロックを何らかの方法で徐々に遅らせる操作を行うと、位相差が変化し、Current にヒットが存在する割合も変化するはずである。この操作を 25 ns 以内で実行することで (Scan することで)、LHC のバンチ衝突と TGC システム内のクロックの位相が一致する、つまり Current のヒットの割合が 100 % になるところを探すことができる。これをシュミレートしたものが 4.12 図である。

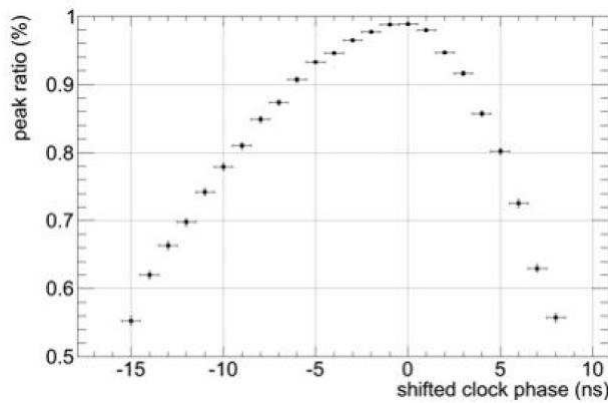


図 4.12: Clock Phase Scan のシュミレーション [10]

縦軸は TGC システムがリードアウトしている 3 つのバンチに対する Current バンチの割合を示している。

4.12 図の場合は 0 ns のところでもっとも位相があっていることになる。このように TGC システムのクロックに Delay をかけ、TGC システムからリードアウトされる Hit 情報の中の Current バンチが占める割合に注目することで、TGC システム内を流れるクロックと LHC のバンチ衝突の間にある位相差 δ を知ることができる。

この Clock Phase Scan を行うためには TGC システム内を流れるクロック全てに Delay をかけなければならない。クロックが各フロントエンドの電子クスに配布される前に Delay をかけるというのがもっとも効率がよい方法である。4.2.1 で示したように、カウンティングルーム (USA15) にある TTCex モジュールから各フロントエンドの電子クスにクロックが配布されている。再度クロックの分配がどのように行われているのかを 4.13 図に示す。

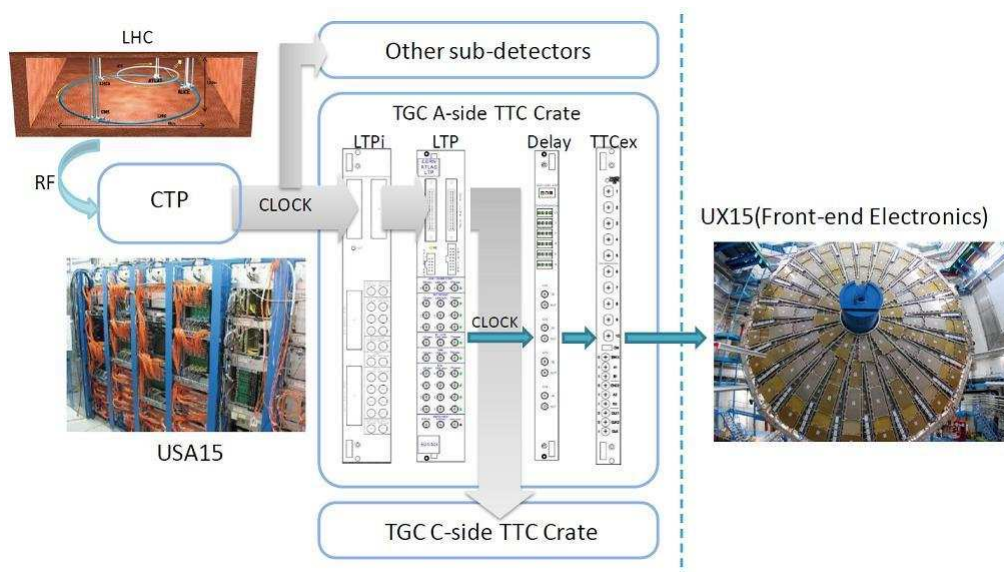


図 4.13: Clock の配布を表した模式図

TTCex の手前に Delay をインストールすることで TGC システム全体のクロックに Delay をかけることができる。

4.13 図からもわかるように TTCex の手前に Delay をかける VME のモジュールをインストールするのが最も効率が良い。そのような役割を果たすモジュールは 2008 年の時点でインストールされていなかった。そこで、この Delay として VME Variable Logic Delay Module を製作することにした。

4.4.2 VME Variable Logic Delay Module の製作

モジュールを製作するにあたり、仕様として要求されるものがあったのでそれを以下に箇条書きの形で挙げる。

1. TTC クレートにインストールされることから 6U サイズの VME モジュールでなければならない。
2. 遅延は同軸ケーブルを伝わる際の信号の伝播遅延を利用する。
3. Clock Phase Scan としては 1 ns の精度で Scan を行いたいので、Delay Module のステップとしてはサブナノ秒ほどが欲しい。
4. クロック以外のタイミングシグナルにも同様に Delay をかけなければいけないので、適切なカップリングで信号を受け適切なチャンネル数の入力が必要。
5. オンライン DAQ システムから Delay 値を設定できるようにするために、VME バス経由で値が設定できなければならない。

4. の項目について補足をする、TGC システムのフロントエンドエレクトロニクスでは 40 MHz のクロックの他にも、TTCvi モジュールによって供給される A Channel、B Channel と呼ばれるシグナルを使用している。A Channel を用いて L1A シグナルが伝達されていて、B Channel の方では Bunch に関連したシグナル、例えば BCR などが伝達されている。これらのシグナルにもクロックと同様に Delay をかけなければエレクトロニクスが正常に動作しないので、これらのシグナルにも同様に Delay をかけられるようにしたい。クロックは NECL レベルの¹AC Couple²のシグナルであるが、A-Ch、B-Ch のシグナルは NECL の DC Couple のシグナルである。そのために Delay Module は 4 Channel 入出力で、1 Channel 目と 2 Channel 目が NECL の AC Couple、3 Channel 目と 4 Channel 目は NECL の DC Couple という仕様にした。また 5. からの要請のために、バスコントローラとして 1 つの CPLD を搭載し、VME バス経由でのリード、ライトが可能ようにした。

私はこの Delay Module の製作において回路図の作製、CPLD のロジックの作製、値を設定するソフトウェアの作製を行った。まず回路図の作製についてであるが、回路図を書くのに Cadence 社の Orcad Capture を使用した。4.14 図は Orcad Capture のスクリーンショットである。Orcad Capture

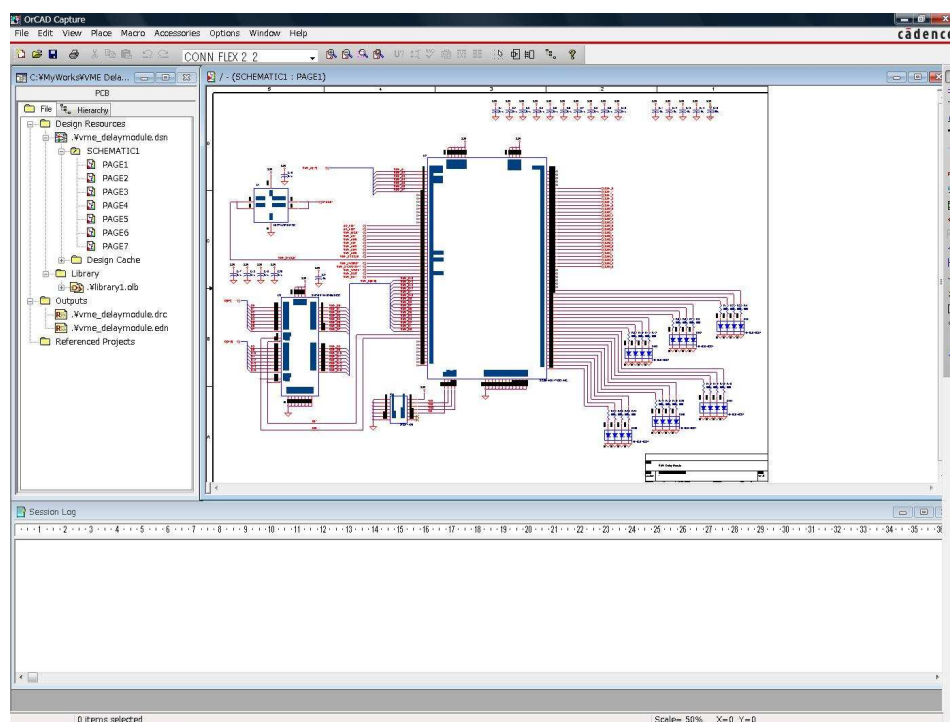


図 4.14: Orcad Capture

¹Negative ECL(Emitter-Coupled Logic)。Logic 1(High) が -1.75 V 、Logic 2(Low) が -0.90 V である [23]。負の電源電圧特別に供給しなければならない。この他に正の電源電圧で動作する PECL(Positive ECL) と呼ばれるものもある。

²キャパシタを直列に接続して直流成分を除去する回路構成のこと。後述の DC Couple はキャパシタが直列に挿入されていない回路構成のことである。

は電気回路設計のための CAD ツール³である。部品をライブラリとして管理できるため、次の開発にかかる時間を短縮できる。Orcad Capture は回路図からネットリスト⁴を作成してくれるので、基板の配線設計が容易になる。

回路の設計には 4 月から取り掛かり、完成までに約 1ヶ月ほど要した。完成版の Delay Module の回路図を付録 A に載せる。基板設計と製作は (有) ジー・エヌ・ディーに発注をした。試作機が完成したのは 6 月中旬ほどで、その後 CPLD のロジックの製作に取り掛かった。

CPLD のロジックは VME Bus の A24D16 のデータ転送に対応し、VME Bus 経由でレジスタの読み書きが行えるものにした。4.15 図は一般的な VME のスレーブモジュールとバスの関係を示して

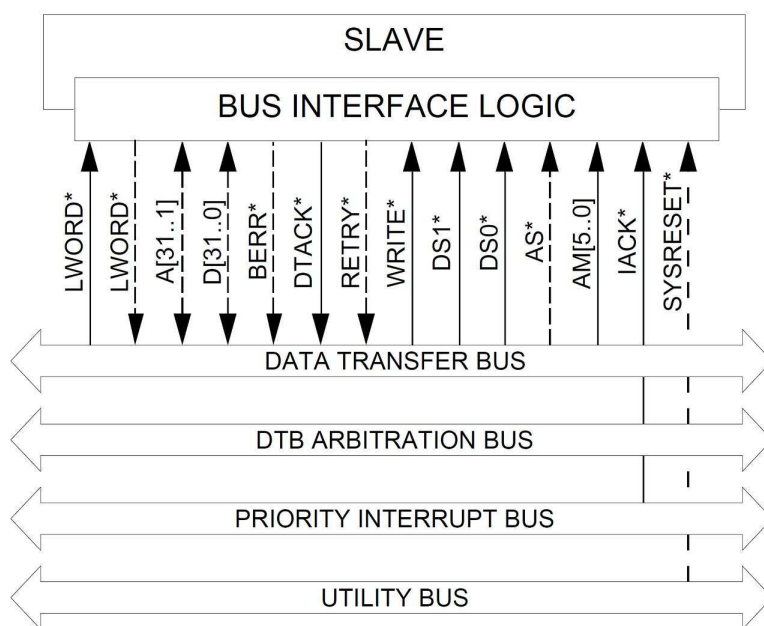


図 4.15: VME Slave モジュールのブロックダイアグラム [22]

いる。スレーブモジュールは図中に示された信号線をモニターし、場合によっては自らドライブしなければならない。今回の Delay Module では BERR と RETRY 線は実装しなかったが、他の信号線は使用した。また、ブロック転送とインタラプトには対応しないバスインターフェースロジックを作ることにした。

ロジックの設計には Xilinx 社の ISE(Integrated Software Environment) を使用した。HDL(Hardware Description Language) としては Verilog を用いた。ISE では、言語から実際の回路への合成 (Synthesis)、配置配線 (Place and Route)、ターゲットへのダウンロードを一括して行うことができる。また付属のシュミレータによる 2 種類のシュミレーション、Behavioral Simulation と Timing Simulation を行うことができる。Behavioral Simulation は設計した回路が意図した動作をするかを確認するた

³Computer Aided Design. 今まで人の手によって行われていた設計作業をコンピュータによって支援し、効率を高めるという発想、またはそれを実現するもの。機械設計では 3DCAD などが使用されている。

⁴電子回路における端子間の接続情報のデータ。

めだけのシュミレーションであり、Timing Simulation は配置配線まで行った後の実際の回路で起こるだろう遅延も考慮したシュミレーションが可能である。今回はそれほど複雑なロジックでんもなかったの、コードを書いた後に Behavioral Simulation で機能を確認し、その後すぐに実機にダウンロードして動作を確認するという方法でロジックの開発を進めた。

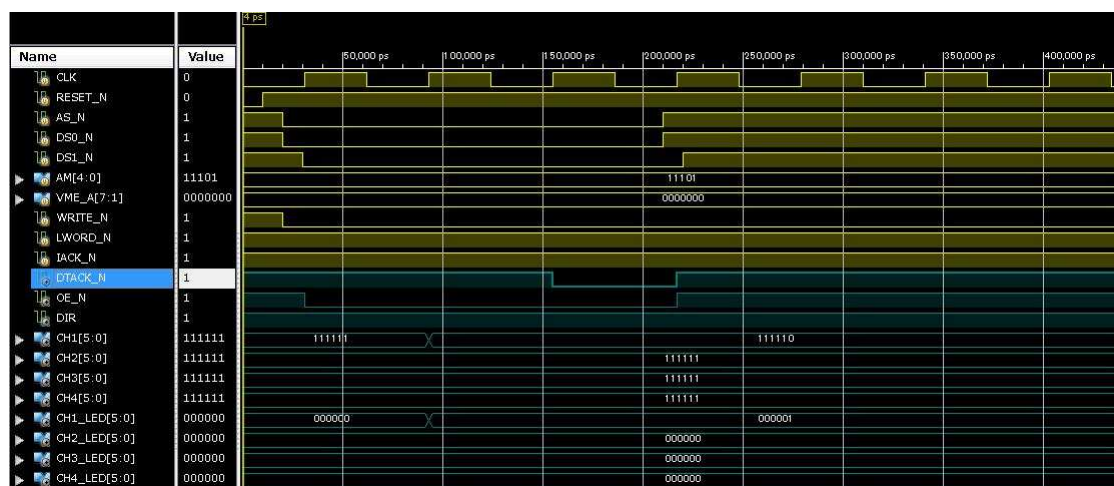


図 4.16: ISE による Behavioral Simulation

図は典型的な Write Cycle をシュミレートしたものである。黄色で示されたシグナルはバスによってドライブされるシグナルで、水色のシグナルは Delay Module の CPLD からの出力を表している。書き込みサイクルでは、まず始めに VME のマスターは AS、DS をアサートし、バス上に有効なアドレスとデータを置く。スレーブはアドレスをデコードして自分が呼ばれているときはバスのデータを内部に取り込みレジスタ等に値を保持する。スレーブはデータ転送が終わったら、DTACK をアサートしなければいけない。DTACK がアサートされるとマスターはデータ転送が終了したと理解し、AS、DS をディアサートする。このシュミレーションでは Channel 1 のレジスタに 0x1 のデータの書き込みを行った。CH1 と CH1_LED という信号線の値が変わり、その後に DTACK がアサートされデータ転送が終了するのがわかる (ただし、CH1 には内部のレジスタが反転してつながれているため、値も反転している)。

4.16 図は Delay Module のロジックを Behavioral Simulation にかけた結果を表している。Delay Module 内部のレジスタに値が書き込まれ、外部出力の CH1 と CH1_LED の値が変化した後にスレーブによって DTACK がアサートされデータ転送が終了していく様子が見える。このようにしてシュミレーションで意図したように動いているかを確認した後、試作機にダウンロードして検証を行い、動作に不具合がないことを確認した。

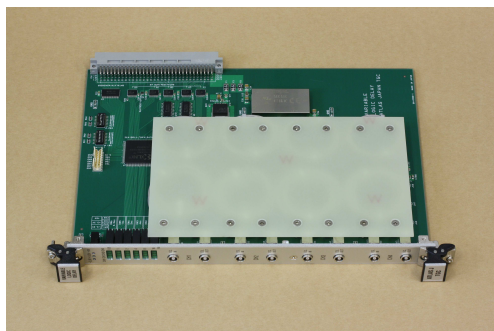


図 4.17: VME Variable Logic Delay



図 4.18: VME Variable Logic Delay のフロントパネル

4.17 図は完成した Delay Module の写真である。フロントパネルには 4 Ch × 6 の LED を搭載し、設定値の確認ができるようにした。

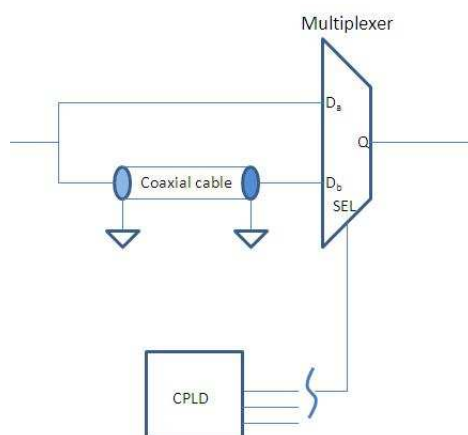


図 4.19: Delay Module の動作原理

4.19 図は Delay Module の動作原理を簡潔に表した図である。Delay Line(同軸ケーブル)、または基板上のパターンを通ってくる信号線のどちらを使用するかを ECL の Multiplexer で選択する。2 種類の入力 (D_a 、 D_b) のうちどちらを用いるかは SEL 線の High、Low で選択できる。High で D_a 、Low で D_b である。SEL 線の High、Low は CPLD でコントロールする。これが Delay Module の基本構成である。

同軸ケーブルは 10 cm、20 cm、40 cm、80 cm、1.6 m、3.2 m の 6 種類の長さがあり、どの Delay Line を有効にするのかを CPLD で操作することで、0 ~ 31.5 ns の範囲で値が設定できる。4.19 図には示されていないが、巻末の回路図を見るとわかるように反転出力と非反転出力を交互に使用する

ような構成となっている。また Multiplexer と CPLD の間には信号レベルを変換する IC が一つ挿入されている。

Delay Module の最終的な仕様を以下にまとめる。Delay Module は全部で 10 枚製作した。

- VME 6U Module(ただし J1 コネクタのみ実装)。
- A24D16 のデータ転送に対応、ただしインタラプト、ブロック転送には未対応。
- 遅延には潤工社製 DFS020 同軸ケーブルを使用。特性インピーダンスは 50Ω 。耐熱、耐寒性にすぐれている。
- 4 Input、4 Output。
 - － 1、2 Channel は NECL AC Couple
 - － 3、4 Channel は NECL DC Couple
- バスコントローラとして Xilinx 社製 CPLD、XC95144XL を使用。
- 1 ステップが 0.5ns で 64 ステップ可変 ($0 \sim 31.5\text{ns}$ の範囲で値を設定可能)。

4.4.3 VME Variable Logic Delay の精密測定

各 Step の精密測定

製作した 10 枚の Delay Module それぞれに対して Delay Step の精度検証を行った。ただし、製造された時期に応じてテストされた環境が異なる。試作機も含め 6 台は CERN のテストベンチにて検証を行った。後から作られた 4 台は KEK 回路室のテストベンチにて検証を行った。後者の 4 台については高分解能の測定器を用いて検証することができたので、以下でその結果について考察する。

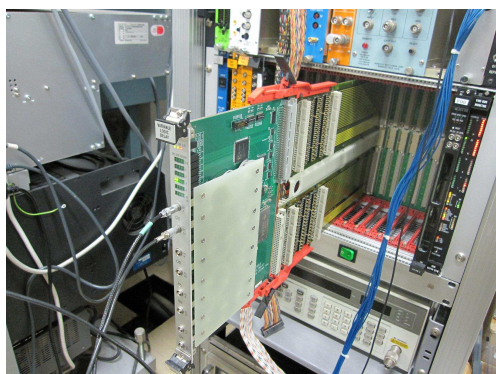


図 4.20: Delay Module の測定風景 1

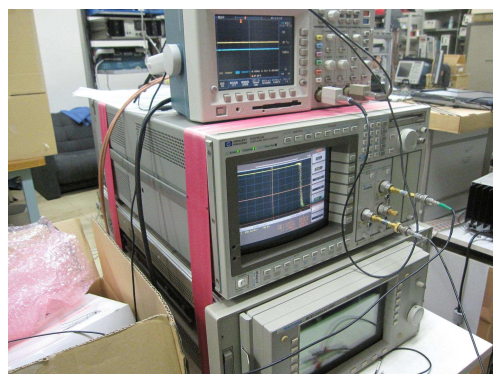


図 4.21: Delay Module の測定風景 2

4.20、4.21 図は KEK 回路室でのテストの様子である。また、4.22 図に測定の際のセットアップを示す。この測定には Hewlett Packard 製の Pulse Generator と Digitizing Oscilloscope を使用し

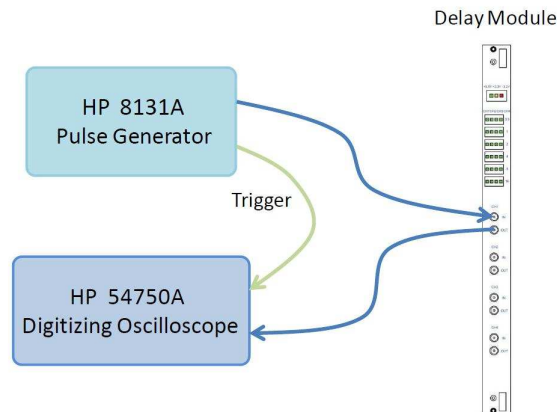


図 4.22: Delay Module の各 Step 測定の際のセットアップ図
HP 54750A は外部トリガーを必要とする。

た。Pulse Generator の主な性能としては Logic Low から High への遷移にかかる時間が 200ps 以内で、500 MHz(Transducer Mode では 1 GHz) までの波形を生成することが可能である。Digitizing Oscilloscope は最大で 20 GHz または 50 GHz と広帯域に対応したものを使用した。2 つのマーカーを用いたタイム・インターバル測定の確度は $10 \text{ ps} \pm 0.1 \% \text{ of reading}$ 以下である。以上の機器を用いて、4 枚の DelayModule $\times$ 4 Channel $\times$ 7 点 分測定した。Delay Module 内には全部で 6 種類 (0.5 ns、1 ns、2 ns、4 ns、8 ns、16 ns) のケーブルがあるが、その 6 点の測定の他にすべてのケーブルを通る 31.5 ns の Delay をかけたときの 7 点分測定を行った。Delay Step の測定の他にも、回路のオフセット (Delay Step が 0 ns のときに入力から入って出力されるまでにどのくらいかかるか) も測定した。

表 4.3 に Delay Step の精度の測定結果を載せる。また、表 4.3 の中の Channel 1 にのみ注目し、Module 4 枚分の値をプロットしたのが 4.23 図である。測定の結果としては Step の精度は数 10 ps 程度で、回路のオフセットは AC Couple のチャンネル (1 Channel 目、2 Channel 目) で 3.5 ~ 3.6 ns 程度、DC Couple のチャンネル (3 Channel 目、4 Channel 目) で 3.9 ns 程度だった。100 ps のオーダーで値がずれているものは見られなかった。以上の測定より、Delay Module の各 Step が精度良く作られていることを示すことができた。

温度特性の測定

Delay Module は USA15 という地下のカウンティングルームにインストールされる。USA15 中は空調設備が常時稼動しているのに加え、クレート内のファンも常に動いていることから急激な温度変化が起こらないようになっている。しかし、ECL の IC は常時電流が流れるために電力消費が大きく、発熱しやすい。そのため、モジュール単体での信頼性を確かめてみる意味でも温度特性を測定することを考えた。というものの、VME クレートが入る恒温槽を探すのは一苦勞であるし、Delay

表 4.3: Delay Step の測定結果

VME Variable Logic Delay

SN:2EC71C347B (Board 1)

	offset	0.5ns	1ns	2ns	4ns	8ns	16ns	31.5ns
1ch	3.52	0.54	1.02	2.03	4.03	7.97	15.95	31.51
2ch	3.56	0.46	1.07	2.00	4.01	7.93	15.97	31.45
3ch	4.09	0.51	1.05	1.95	4.01	7.92	16.03	31.54
4ch	3.92	0.49	1.06	2.00	4.06	7.99	16.02	31.54

SN:2EC71C345D (Board 2)

	offset	0.5ns	1ns	2ns	4ns	8ns	16ns	31.5ns
1ch	3.58	0.52	1.08	2.03	4.08	8.06	15.98	31.57
2ch	3.66	0.46	1.04	1.98	3.98	7.91	15.96	31.52
3ch	3.89	0.49	0.99	2.04	4.02	8.00	15.96	31.52
4ch	3.92	0.51	1.00	1.99	4.04	7.99	15.96	31.53

SN:2EC71C3471 (Board 3)

	offset	0.5ns	1ns	2ns	4ns	8ns	16ns	31.5ns
1ch	3.61	0.51	1.02	2.03	4.02	7.96	16.01	31.49
2ch	3.65	0.49	0.99	2.03	4.01	7.98	15.96	31.48
3ch	3.93	0.49	0.99	2.01	3.96	7.94	15.96	31.54
4ch	3.91	0.52	1.06	2.01	4.03	7.99	15.93	31.51

SN:2EC71C3467 (Board 4)

	offset	0.5ns	1ns	2ns	4ns	8ns	16ns	31.5ns
1ch	3.59	0.51	1.03	2.04	4.01	8.04	15.97	31.56
2ch	3.61	0.49	1.02	2.04	3.98	7.96	15.95	31.52
3ch	3.96	0.51	0.98	1.99	4.02	7.99	15.99	31.53
4ch	3.92	0.51	1.06	2.04	4.04	8.02	16.01	31.56

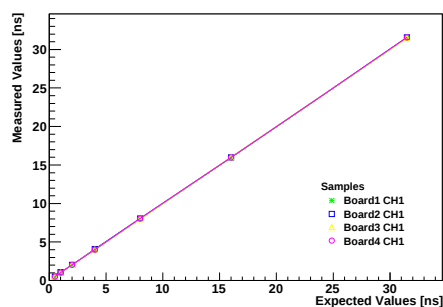


図 4.23: 測定した 4 枚のボードの 1 Channel 目の結果

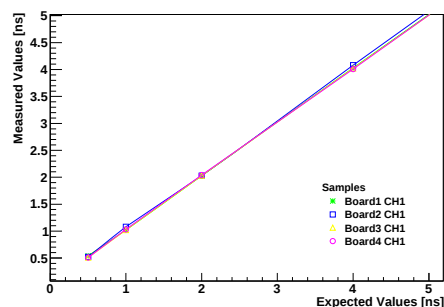


図 4.24: 4.23 図の拡大 1

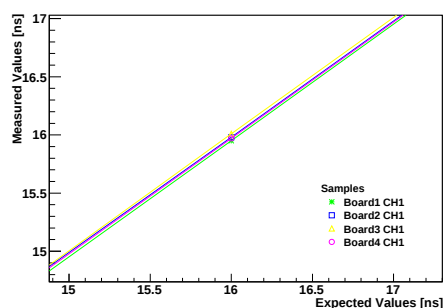


図 4.25: 4.23 図の拡大 2

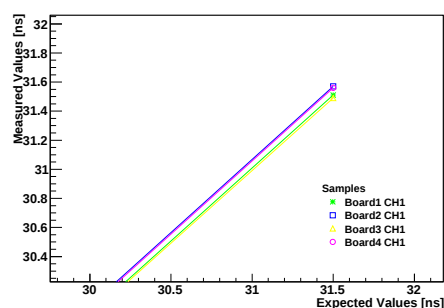


図 4.26: 4.23 図の拡大 3

Module 一台が入る恒温槽があったとしても、電力供給をどうするかなど懸案事項が多いので、今回は身近にあるものを用いて手軽に測定を行うことにした。

温度特性の測定では、ビニール袋とドライヤーを用いて擬似的に温度の高い状態を作ることにした。ビニール袋で Delay Module を包み、ドライヤーで袋内を温めることで Delay Module のみに温度変化が起こる環境を作った。4.27 図に測定の様子を載せる。Delay Module を含む閉じた回路内で発振を起こし、その発振周波数の変化を観測するという手法を取った。

4.29 図は温度特性の測定の際に使用したセットアップ図である。図のように Delay Module の他に ECL-NIM Level Converter、NIM Fan-out モジュールを使用し閉じた経路を作った。NIM Fan-out モジュールは反転出力があるものを使用した。Delay Module の設定は 31.5 ns にし、全てのケーブルが通るようにした。4.28 図の中央の黒いモジュールが使用した NIM Fan-out モジュールである。4.29 図のように奇数回 (図では 3 回) ロジックを反転させることで発振させることができる。信号の測定には Lecroy 社の Wavepro 7300A を使用した。このようなセットアップを組み、Delay Module のみを温度変化させることができる環境を作った。ドライヤーの風が直接当たらない側の基板にプローブを当てて温度の測定を行った。

4.30、4.31 図に測定の結果を示す。測定当時の室温は 26.0 °C ほどであったが、それから 56.0 °C

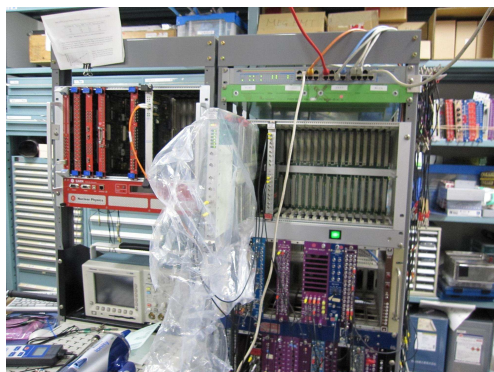


図 4.27: 温度特性測定の様子

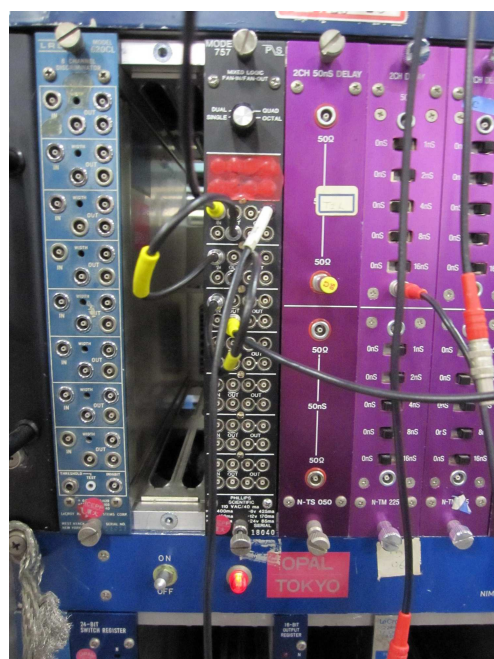


図 4.28: NIM Fan-out モジュール

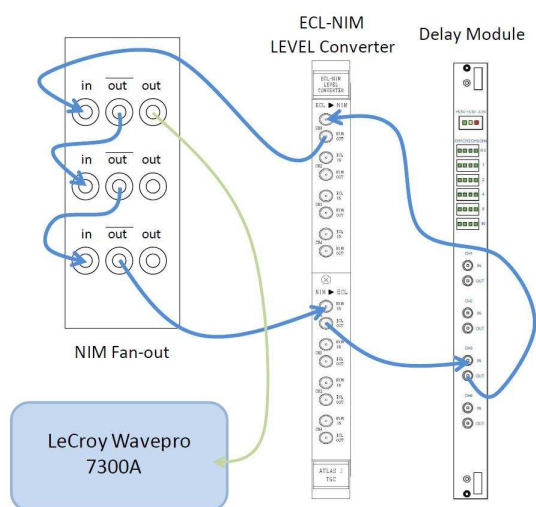


図 4.29: 温度特性測定のセットアップ図

まで温度を上昇させた。波形のアベレージをとったものの周波数を表示させ、変化が現れるかを観察した。4.30 図が測定開始直後のオシロスコープのスクリーンショットで 4.31 図が 56.0 °C 時のオシロスコープのスクリーンショットであるが、結果としてはアベレージの周波数は終始 5.7642 MHz(周期に換算すると 173.48 ns) で、変化は見られなかった。以下で変化がみられなかった理由について考

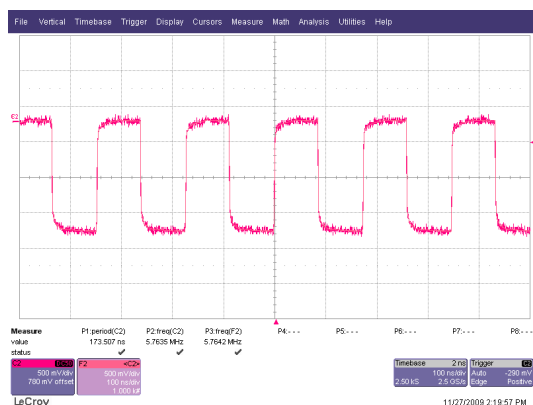


図 4.30: 測定開始直後 (26.0 °C) の波形
波形のアベレージを取ってその値を観察した。図中では P3 に表示されている値である。

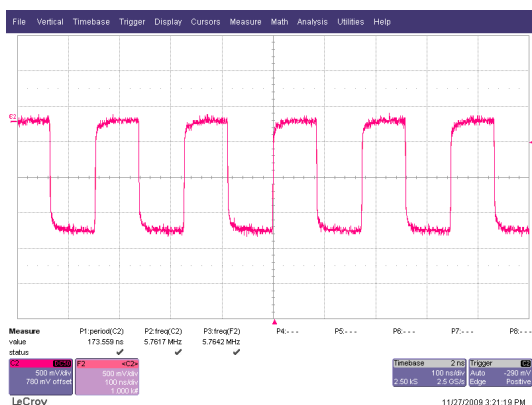


図 4.31: 測定終了時 (56.0 °C) の波形

察してみる。

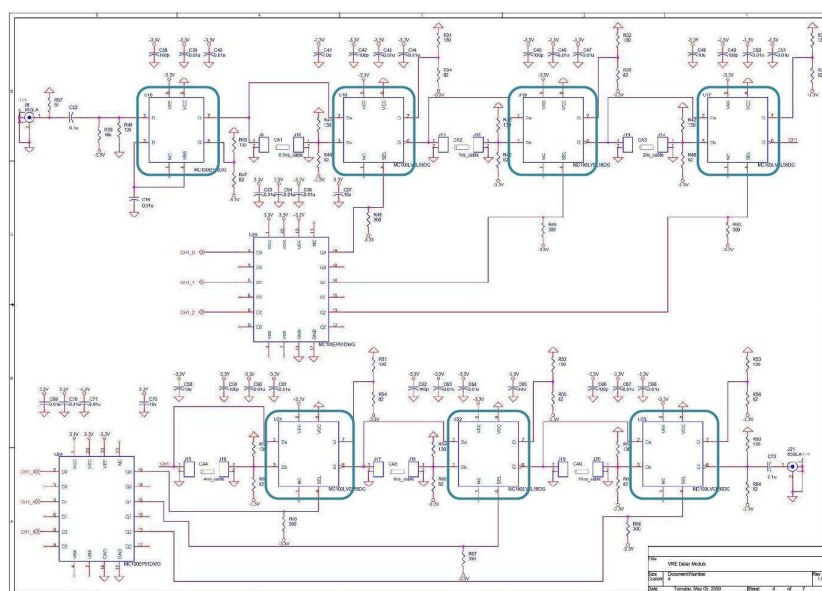


図 4.32: Delay Module の回路の一部

4.32 図はシグナルが通る回路を示している。入力から入って出力までの間に 1 個の MC100EP16DG という IC と 6 個の MC100LVEL58DG という IC の他に 6 種類の長さの違う同軸ケーブル中をシグナルが通る。MC100EP16DG と MC100LVEL58DG はデータシート中に温度による伝播遅延の変化についての記述があるので表 4.4 に載せる。表 4.4 を参考にすると、MC100EP16DG は Typical な Propagation Delay は 25 °C から 85 °C の温度変化で、20 ps 増加し、MC100LVEL58DG は 10 ps 増加

する。同軸ケーブルの伝播遅延はほとんどないと考え無視し、Delay Moduleの85 °C時のPropagation Delayの増加分を見積もると $1 \times 20\text{ps} + 6 \times 10\text{ps} = 80\text{ps}$ となる。今回は26.0 °Cから56.0 °Cまで温度を変化させたので、大体仕様書の半分の温度変化であるから、単純化して考えると40 ps くらいの時間変化に対応する周波数変化があってもおかしくはない。

表 4.4: MC100EP16DG と MC100LVEL58DG の Propagation Delay

MC100EP16DG [19]

	−40 °C			25 °C			85 °C		
	Min	Typical	Max	Min	Typical	Max	Min	Typical	Max
Propagation Delay [ps]	150	220	280	150	220	280	160	240	300

MC100LVEL58DG [18]

	−40 °C			25 °C			85 °C		
	Min	Typical	Max	Min	Typical	Max	Min	Typical	Max
Propagation Delay [ps]	340	435	560	350	440	570	370	450	590

一方、今回使用した LeCroy Wavepro 7300A(以下 Wavepro と略す) の水平軸方向の時間間隔精度は仕様書より、

$$0.06/\text{Sampling Rate} + (5 \text{ ppm of reading}) \quad (\text{RMS})$$

となっている。Wavepro は4 channel 同時入力時には10 GS/sの最大サンプルレート(2 channel 入力時は最大20 GS/s)が可能であるが、測定時の環境では水平軸の設定は100 ns/divであり、サンプリングレートは2.5 GS/sであったので、読み取り値として173.48 nsの値を用いて時間間隔精度を計算すると、

$$0.06/2.5 [\text{GS/s}] + 5 \text{ ppm} \times 173.48 \sim 25 [\text{ps}]$$

程度となる。25 ps 以上の時間変化があれば、周波数の変化が見られるはずだが、実際には周波数に変化はみられなかった。

以上をまとめると、オシロスコープの精度も十分あったが、測定できる範囲でDelay Moduleのステップの精度は変化しなかったと言える。ビニール袋とドライヤーで作った疑似的な恒温環境であり、基板の温度を測定していたので、個々の部品間には温度のむらがあったと考えられる。しかし、現実的に温度上昇が起こる環境を想定してみると、必ずしも一様に温度上昇するとは考えられないので、より現実的な環境であったとも言える。ICの仕様書からの見積もりでは40 ps くらいの変化に対応する周波数変化があってもおかしくはなかったが、簡易的な手法であったこともあり、測定されるほどの変化をしなかった。今回の測定によって、現実的に温度が上昇しうる範囲ではDelay Moduleはまったく影響を受けないことがわかった。

4.4.4 Delay Module のインストール

Delay Module を TTC クレートにインストールするにあたり、どのようにインストールするのが最適かという議論が何度もなされた。TGC のデータテイクのモードとしては Global、Local と呼ばれるものがあるが、Delay Module をインストールしたとしてもこれらのモードの違いによってケーブルの抜き差しがないような配線を考えなければいけなかった。また、Clock Phase Scan も同時に行えるように考えなければいけない。配線は変えることなく、Online Software の設定のみで変更が行えるという思想の元にインストール方法を考えた。

最終的な配線は 4.33 図のように決定した。この図について説明すると、TGC A-side と C-side の TTC クレートは独立に存在するが、片サイドにそれぞれ Delay Module を 3 台ずつインストールすることになった。

それぞれの Delay Module の役割について説明する。Delay2 という名前の Delay Module は A-C 間の位相調整用で、Delay3 が Clock Phase Scan 時に使用するモジュールである。Clock Phase Scan 時に TGC システムが発行する L1A のシグナルを使用するときに問題が生じることがわかった。TGC SL から発行された L1A シグナルは LTP に入り、TTCex にて一度レジスタにラッチされた後に光信号に変換されフロントエンドエレクトロニクスに配布される。しかし、Clock Phase Scan 中は TTCex で用いている Clock の位相も変化していくので、タイミングによってはうまくラッチされない場合が考えられる。つまり、SL が発行する L1A のシグナルも同様に位相を変えなければならない。そのためにインストールされたのが Delay1 である。

しかし、SL の出力は NIM レベルのため Delay Module を使用するためには、NIM-ECL 間のレベル変換モジュールが必要であった。そこで開発されたのが ECL-NIM LEVEL CONVERTER である。LEVEL CONVERTER と Delay Module を併用することで、SL からくる L1A にも Delay をかけることが可能になった。

よって、最終的には各サイドの TTC クレートに LEVEL CONVERTER1 台、Delay Module3 台追加することになった。モジュールのインストール作業は 2009 年の 8 月末に行われた。4.35、4.36 図はインストール前とインストール後の A-side の TTC クレートの写真である。

4.4.5 インストール後のテスト

Delay Module の TTC クレートへのインストール終了後、TGC システム内を流れるクロックに Delay がかけられるかをテストしてみた。テストは USA15 にて行った。

このテストでは A-side のみに Delay をかけ、C-side には Delay をかけなかった。そして A-side の SL からアウトプットされる L1A のシグナル⁵と C-side に流れているクロック両者をオシロスコープで観察した。データソースにはテストパルスを用いた。もし、うまく Delay がかけられれば、両者のシグナルの間に位相の変化が現れるはずである。

4.37、4.38、4.39、4.40 図は A-side に 0 ns、4 ns、8 ns、12 ns の Delay をかけたときのオシロスコープのスクリーンショットである。この測定をしたときにオシロスコープのトリガーに Delay をかけて

⁵片サイドで SL は 36 枚あるが、SL のバックプレーン側には NIM out 端子があり、すべての OR がとられ LTP に供給されている。今回のテストでは OR をとる OR Module からのアウトプットを用いた。

The diagram illustrates the ATLAS Level-1 Trigger System (L1T) architecture. It shows the signal flow from input modules (L1A, L1B, L1C) through various processing modules (Prescaler, Level Conv., Delay1, LTP, TTCvi, Delay2, Delay3, TTCex) to output modules (SL-A, OR Module, Clock Gen., SL-C). The diagram includes timing specifications and component labels.

Input Modules: L1A, L1B, L1C (each with LIA, L1B, L1C, L1D, L1E, L1F, L1G, L1H, L1I, L1J, L1K, L1L, L1M, L1N, L1O, L1P, L1Q, L1R, L1S, L1T, L1U, L1V, L1W, L1X, L1Y, L1Z, L1AA, L1AB, L1AC, L1AD, L1AE, L1AF, L1AG, L1AH, L1AI, L1AJ, L1AK, L1AL, L1AM, L1AN, L1AO, L1AP, L1AQ, L1AR, L1AS, L1AT, L1AU, L1AV, L1AW, L1AX, L1AY, L1AZ, L1BA, L1BB, L1BC, L1BD, L1BE, L1BF, L1BG, L1BH, L1BI, L1BJ, L1BK, L1BL, L1BM, L1BN, L1BO, L1BP, L1BQ, L1BR, L1BS, L1BT, L1BU, L1BV, L1BW, L1BX, L1BY, L1BZ, L1CA, L1CB, L1CC, L1CD, L1CE, L1CF, L1CG, L1CH, L1CI, L1CJ, L1CK, L1CL, L1CM, L1CN, L1CO, L1CP, L1CQ, L1CR, L1CS, L1CT, L1CU, L1CV, L1CW, L1CX, L1CY, L1CZ, L1DA, L1DB, L1DC, L1DD, L1DE, L1DF, L1DG, L1DH, L1DI, L1DJ, L1DK, L1DL, L1DM, L1DN, L1DO, L1DP, L1DQ, L1DR, L1DS, L1DT, L1DU, L1DV, L1DW, L1DX, L1DY, L1DZ, L1EA, L1EB, L1EC, L1ED, L1EE, L1EF, L1EG, L1EH, L1EI, L1EJ, L1EK, L1EL, L1EM, L1EN, L1EO, L1EP, L1EQ, L1ER, L1ES, L1ET, L1EU, L1EV, L1EW, L1EX, L1EY, L1EZ, L1FA, L1FB, L1FC, L1FD, L1FE, L1FF, L1FG, L1FH, L1FI, L1FJ, L1FK, L1FL, L1FM, L1FN, L1FO, L1FP, L1FQ, L1FR, L1FS, L1FT, L1FU, L1FV, L1FW, L1FX, L1FY, L1FZ, L1GA, L1GB, L1GC, L1GD, L1GE, L1GF, L1GG, L1GH, L1GI, L1GJ, L1GK, L1GL, L1GM, L1GN, L1GO, L1GP, L1GQ, L1GR, L1GS, L1GT, L1GU, L1GV, L1GW, L1GX, L1GY, L1GZ, L1HA, L1HB, L1HC, L1HD, L1HE, L1HF, L1HG, L1HH, L1HI, L1HJ, L1HK, L1HL, L1HM, L1HN, L1HO, L1HP, L1HQ, L1HR, L1HS, L1HT, L1HU, L1HV, L1HW, L1HX, L1HY, L1HZ, L1IA, L1IB, L1IC, L1ID, L1IE, L1IF, L1IG, L1IH, L1II, L1IJ, L1IK, L1IL, L1IM, L1IN, L1IO, L1IP, L1IQ, L1IR, L1IS, L1IT, L1IU, L1IV, L1IW, L1IX, L1IY, L1IZ, L1JA, L1JB, L1JC, L1JD, L1JE, L1JF, L1JG, L1JH, L1JI, L1JJ, L1JK, L1JL, L1JM, L1JN, L1JO, L1JP, L1JQ, L1JR, L1JS, L1JT, L1JU, L1JV, L1JW, L1JX, L1JY, L1JZ, L1KA, L1KB, L1KC, L1KD, L1KE, L1KF, L1KG, L1KH, L1KI, L1KJ, L1KK, L1KL, L1KM, L1KN, L1KO, L1KP, L1KQ, L1KR, L1KS, L1KT, L1KU, L1KV, L1KW, L1KX, L1KY, L1KZ, L1LA, L1LB, L1LC, L1LD, L1LE, L1LF, L1LG, L1LH, L1LI, L1LJ, L1LK, L1LL, L1LM, L1LN, L1LO, L1LP, L1LQ, L1LR, L1LS, L1LT, L1LU, L1LV, L1LW, L1LX, L1LY, L1LZ, L1MA, L1MB, L1MC, L1MD, L1ME, L1MF, L1MG, L1MH, L1MI, L1MJ, L1MK, L1ML, L1MN, L1MO, L1MP, L1MQ, L1MR, L1MS, L1MT, L1MU, L1MV, L1MW, L1MX, L1MY, L1MZ, L1NA, L1NB, L1NC, L1ND, L1NE, L1NF, L1NG, L1NH, L1NI, L1NJ, L1NK, L1NL, L1NM, L1NN, L1NO, L1NP, L1NQ, L1NR, L1NS, L1NT, L1NU, L1NV, L1NW, L1NX, L1NY, L1NZ, L1OA, L1OB, L1OC, L1OD, L1OE, L1OF, L1OG, L1OH, L1OI, L1OJ, L1OK, L1OL, L1OM, L1ON, L1OO, L1OP, L1OQ, L1OR, L1OS, L1OT, L1OU, L1OV, L1OW, L1OX, L1OY, L1OZ, L1PA, L1PB, L1PC, L1PD, L1PE, L1PF, L1PG, L1PH, L1PI, L1PJ, L1PK, L1PL, L1PM, L1PN, L1PO, L1PP, L1PQ, L1PR, L1PS, L1PT, L1PU, L1PV, L1PW, L1PX, L1PY, L1PZ, L1QA, L1QB, L1QC, L1QD, L1QE, L1QF, L1QG, L1QH, L1QI, L1QJ, L1QK, L1QL, L1QM, L1QN, L1QO, L1QP, L1QQ, L1QR, L1QS, L1QT, L1QU, L1QV, L1QW, L1QX, L1QY, L1QZ, L1RA, L1RB, L1RC, L1RD, L1RE, L1RF, L1RG, L1RH, L1RI, L1RJ, L1RK, L1RL, L1RM, L1RN, L1RO, L1RP, L1RQ, L1RR, L1RS, L1RT, L1RU, L1RV, L1RW, L1RX, L1RY, L1RZ, L1SA, L1SB, L1SC, L1SD, L1SE, L1SF, L1SG, L1SH, L1SI, L1SJ, L1SK, L1SL, L1SM, L1SN, L1SO, L1SP, L1SQ, L1SR, L1SS, L1ST, L1SU, L1SV, L1SW, L1SX, L1SY, L1SZ, L1TA, L1TB, L1TC, L1TD, L1TE, L1TF, L1TG, L1TH, L1TI, L1TJ, L1TK, L1TL, L1TM, L1TN, L1TO, L1TP, L1TQ, L1TR, L1TS, L1TT, L1TU, L1TV, L1TW, L1TX, L1TY, L1TZ, L1UA, L1UB, L1UC, L1UD, L1UE, L1UF, L1UG, L1UH, L1UI, L1UJ, L1UK, L1UL, L1UM, L1UN, L1UO, L1UP, L1UQ, L1UR, L1US, L1UT, L1UU, L1UV, L1UW, L1UX, L1UY, L1UZ, L1VA, L1VB, L1VC, L1VD, L1VE, L1VF, L1VG, L1VH, L1VI, L1VJ, L1VK, L1VL, L1VM, L1VN, L1VO, L1VP, L1VQ, L1VR, L1VS, L1VT, L1VU, L1VV, L1VW, L1VX, L1VY, L1VZ, L1WA, L1WB, L1WC, L1WD, L1WE, L1WF, L1WG, L1WH, L1WI, L1WJ, L1WK, L1WL, L1WM, L1WN, L1WO, L1WP, L1WQ, L1WR, L1WS, L1WT, L1WU, L1WV, L1WW, L1WX, L1WY, L1WZ, L1XA, L1XB, L1XC, L1XD, L1XE, L1XF, L1XG, L1XH, L1XI, L1XJ, L1XK, L1XL, L1XM, L1XN, L1XO, L1XP, L1XQ, L1XR, L1XS, L1XT, L1XU, L1XV, L1XW, L1XX, L1XY, L1XZ, L1YA, L1YB, L1YC, L1YD, L1YE, L1YF, L1YG, L1YH, L1YI, L1YJ, L1YK, L1YL, L1YM, L1YN, L1YO, L1YP, L1YQ, L1YR, L1YS, L1YT, L1YU, L1YV, L1YW, L1YX, L1YY, L1YZ, L1ZA, L1ZB, L1ZC, L1ZD, L1ZE, L1ZF, L1ZG, L1ZH, L1ZI, L1ZJ, L1ZK, L1ZL, L1ZM, L1ZN, L1ZO, L1ZP, L1ZQ, L1ZR, L1ZS, L1ZT, L1ZU, L1ZV, L1ZW, L1ZX, L1ZY, L1ZZ, L1AA, L1AB, L1AC, L1AD, L1AE, L1AF, L1AG, L1AH, L1AI, L1AJ, L1AK, L1AL, L1AM, L1AN, L1AO, L1AP, L1AQ, L1AR, L1AS, L1AT, L1AU, L1AV, L1AW, L1AX, L1AY, L1AZ, L1BA, L1BB, L1BC, L1BD, L1BE, L1BF, L1BG, L1BH, L1BI, L1BJ, L1BK, L1BL, L1BM, L1BN, L1BO, L1BP, L1BQ, L1BR, L1BS, L1BT, L1BU, L1BV, L1BW, L1BX, L1BY, L1BZ, L1CA, L1CB, L1CC, L1CD, L1CE, L1CF, L1CG, L1CH, L1CI, L1CJ, L1CK, L1CL, L1CM, L1CN, L1CO, L1CP, L1CQ, L1CR, L1CS, L1CT, L1CU, L1CV, L1CW, L1CX, L1CY, L1CZ, L1DA, L1DB, L1DC, L1DD, L1DE, L1DF, L1DG, L1DH, L1DI, L1DJ, L1DK, L1DL, L1DM, L1DN, L1DO, L1DP, L1DQ, L1DR, L1DS, L1DT, L1DU, L1DV, L1DW, L1DX, L1DY, L1DZ, L1EA, L1EB, L1EC, L1ED, L1EE, L1EF, L1EG, L1EH, L1EI, L1EJ, L1EK, L1EL, L1EM, L1EN, L1EO, L1EP, L1EQ, L1ER, L1ES, L1ET, L1EU, L1EV, L1EW, L1EX, L1EY, L1EZ, L1FA, L1FB, L1FC, L1FD, L1FE, L1FF, L1FG, L1FH, L1FI, L1FJ, L1FK, L1FL, L1FM, L1FN, L1FO, L1FP, L1FQ, L1FR, L1FS, L1FT, L1FU, L1FV, L1FW, L1FX, L1FY, L1FZ, L1GA, L1GB, L1GC, L1GD, L1GE, L1GF, L1GG, L1GH, L1GI, L1GJ, L1GK, L1GL, L1GM, L1GN, L1GO, L1GP, L1GQ, L1GR, L1GS, L1GT, L1GU, L1GV, L1GW, L1GX, L1GY, L1GZ, L1HA, L1HB, L1HC, L1HD, L1HE, L1HF, L1HG, L1HH, L1HI, L1HJ, L1HK, L1HL, L1HM, L1HN, L1HO, L1HP, L1HQ, L1HR, L1HS, L1HT, L1HU, L1HV, L1HW, L1HX, L1HY, L1HZ, L1IA, L1IB, L1IC, L1ID, L1IE, L1IF, L1IG, L1IH, L1II, L1IJ, L1IK, L1IL, L1IM, L1IN, L1IO, L1IP, L1IQ, L1IR, L1IS, L1IT, L1IU, L1IV, L1IW, L1IX, L1IY, L1IZ, L1JA, L1JB

89

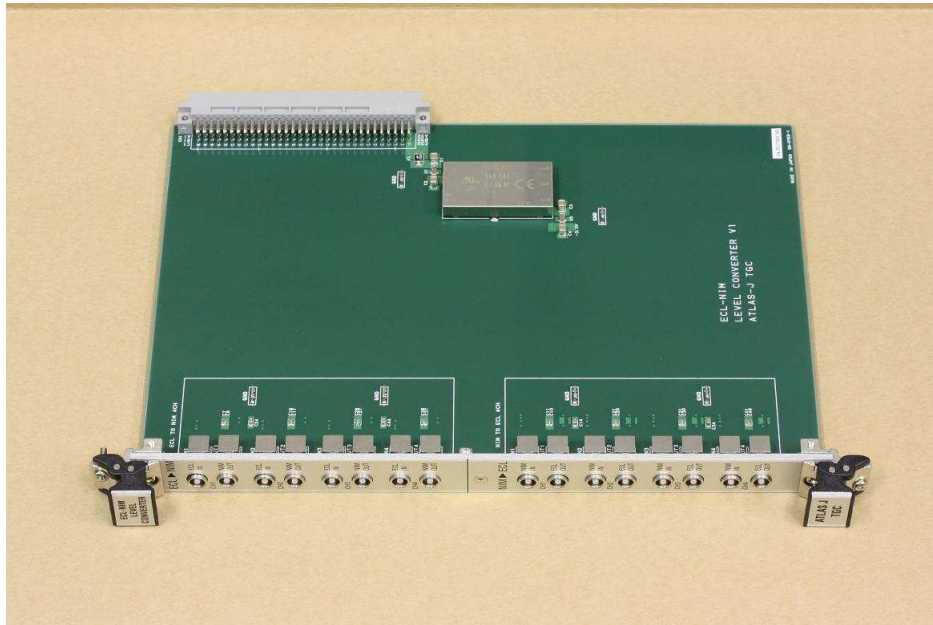


図 4.34: ECL-NIM LEVEL CONVERTER

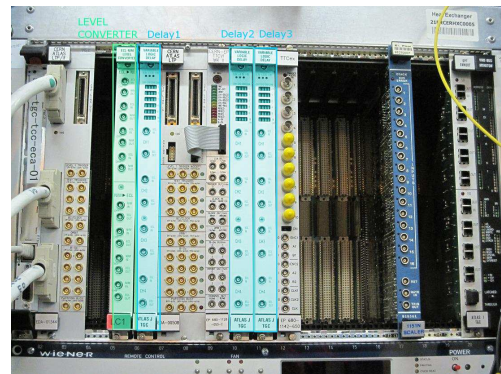
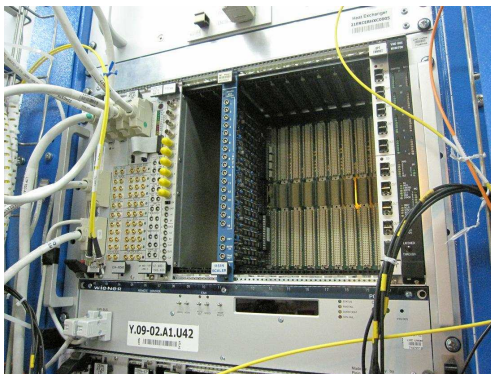


図 4.35: インストール前の A-side TTC クレート 図 4.36: インストール後の A-side TTC クレート

いる A-side の L1A のシグナルを用いていたので、逆に C-side のクロックが進んでいるように見えているが、Delay Module に設定した値の分だけ進んでいるので、A-side に正しく Delay がかけられていることがわかる。

この結果から Delay Module が TGC システム内でも正しく動作し、Delay がかけられていることが示された。

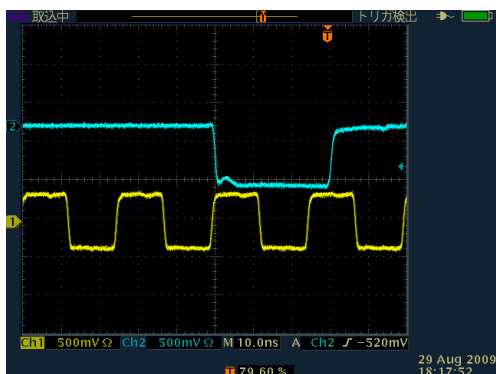


図 4.37: 0ns Delay

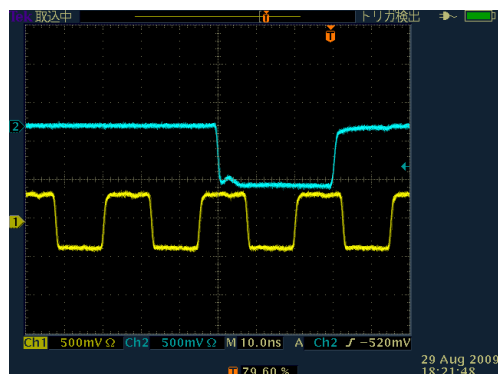


図 4.38: 4ns Delay

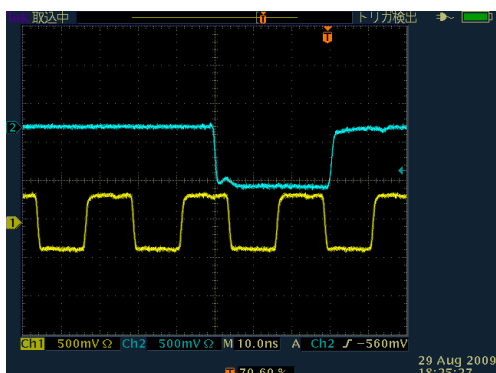


図 4.39: 8ns Delay

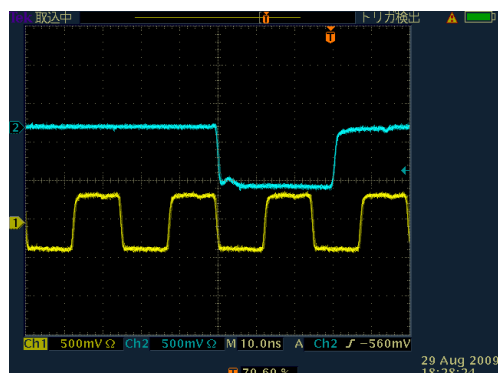


図 4.40: 12ns Delay

上の水色のシグナルが A-side で発行された L1A のシグナルである。下の黄色のシグナルが C-side に流れているクロックである。Delay は A-side にかけたが、オシロスコープのトリガーに A-side の L1A のシグナルを用いたので、C-side のクロックが進んでいるように見える。

4.4.6 Delay Module 用 Online Software の開発

ATLAS 検出器内のすべてのサブディテクターはデータテイクを始める前に同期してコンフィギュレーションされる。それを担うのが Onlins Software で、TDAQ と呼ばれるシステムである。

TTC クレートにインストールされた Delay Module は Global、Stand Alone、さらに Clock Phase Scan などのそれぞれの Run に合わせて他のハードウェア同様にコンフィギュアされる (Delay 値が設定される) 必要がある。そこで、Delay Module 専用の Online Software パッケージを用意することになった。このパッケージに求められることは、

1. xml ファイルから設定値を読み出しコンフィギュレーションを行う
2. 設定した値を IS(Inforamtion Server)に残す

の 2 つである。また、これらの機能が TDAQ パネル (GUI) と同期して動作しなければいけない。

Online Software について

Online Software はフレームワークとなる TDAQ の環境の上に構築される。Online Software は 4.41 図に示されるような各コンポーネントからなる。

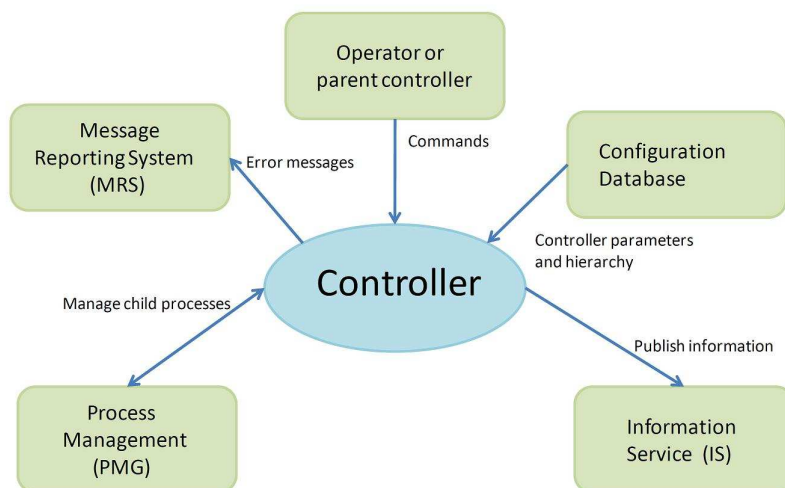


図 4.41: Offline Software のコンポーネント

最新の TDAQ のリリースは tdaq-02-00-03 で、Delay Module のパッケージ、TGCDelayModule もこのリリースを利用して開発された。

TDAQ システムを通してトリガーの制御、リードアウトするセクターの制御が可能である。これらの操作は全て GUI を通して制御可能である。実際に Run を始める際には、Run Controller がコンフィギュレーションのステートを管理し、各検出器間で同期をとって、ステートが遷移していく。

4.42 図は一般的なステートの遷移の例を表している。ステートが遷移する間に各ハードウェアがコンフィギュレーションされる。

TGCDelayModule パッケージの開発

TGCDelayModule パッケージは以下のような構成になっている。

cmt

requirements ファイルが置かれる

hw

どの Hardware(計算機、または SBC) を使うかを記述する xml ファイルが置かれる

partitions

このパッケージがどういう名前のパーティションかを記述する xml ファイルが置かれる

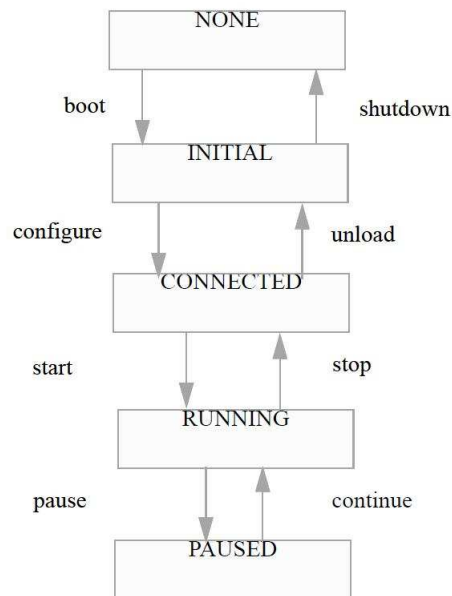


図 4.42: ステートマシンのダイアグラム

四角で囲まれたものが状態で、その脇に添えられた文字がコマンドを表している。例えば、ユーザーが boot コマンドを発行すれば (GUI で”boot”というボタンを押せば)、状態が”NONE”から”INITIAL”に遷移する [5]。

schema

クラスが記述された xml ファイルが置かれる

segments

パッケージ内がどのようなセグメントに分けられるかが記述されている

src

C++のソースコードが置かれる

sw

Software に関する設定が書かれた xml ファイルが置かれる

TGCDelayModule

ヘッダーファイルが置かれる

Delay Module に値を設定したり、IS への設定値の publish は C++のソースコードの中に記述される。Delay Module は各サイドの TTC クレートに 3 台ずつインストールされている。そこでソフトウェアでもその構造を継承し、TGCDelayBoardSet の中に 3 つの TGCDelayBoard があるという構成にした。TGCDelayBoardSet の上位には TGCDelayModule-SideA、TGCDelayModule-SideC

というクレートに対応するものがあり、またその上位には RCD(ROD Crate DAQ) と呼ばれるものがある。

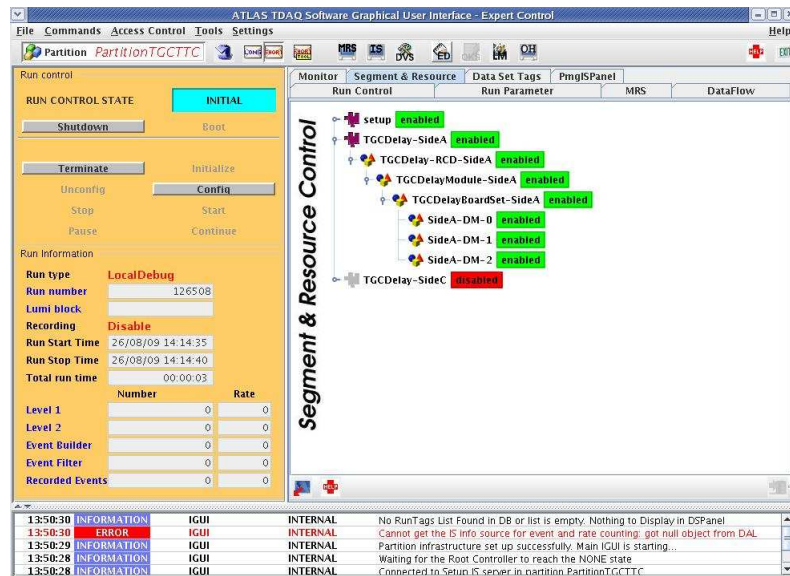


図 4.43: TGCDelayModule の Segments&Resources

4.43 図は TGCDelayModule パッケージがどのようなセグメントに分かれるかを示している。

RCA(Run Control Application) は Run のステートの遷移に合わせて Delay Module をコンフィギュレーションする。TGCDelayModule パッケージでは setup() という関数中で ("BOOT" コマンドが発行されたときに) 各モジュールに値を設定するようになっている。また Oracle Database からコンフィギュレーションのパラメータを取得しコンフィギュレーションすることが可能になったので、他のエレクトロニクスとの同期をとるために Barrier という機能も実装した。

setup() が呼ばれた直後に Barrier は立ち上がる。Delay Module への値の設定は通常通り行われるが、設定終了後指定の時間だけ待つという動作が Barrier によってなされる。この待つ時間を各モジュール間で調整することで、同期してコンフィギュレーションすることを可能にしている。

開発されたソフトウェアのテストは local-daq という予備のクレートを用いて行った。このクレートに Delay Module はささっていないかったが、エラーが出ないような対処を行い、ステートの遷移がエラーなく行えるかを試した。

4.45 図はソフトウェアの設定の手順に間違いがなく "RUNNING" の状態まで行ったときのスクリーンショットである。

また、もう一つの要求されていた機能である IS への設定値の publish についても同様の手法で動作確認がなされている。

4.46 図は IS Monitor のスクリーンショットである。このテストを行ったときはモジュールがクレートにささっていないだったので、値は ffff が入っている。しかし、実際にはモジュールからリードした値が publish される予定である。

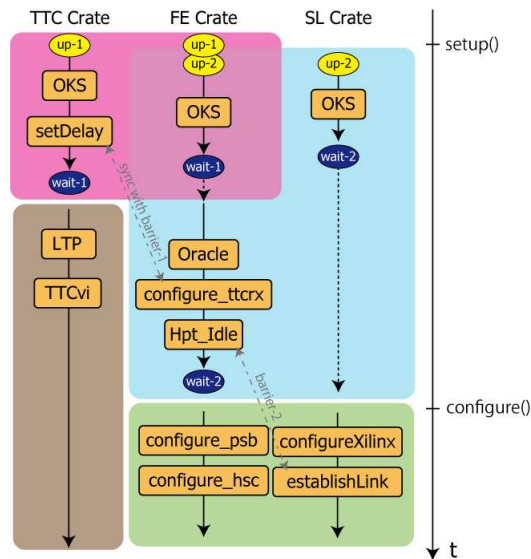


図 4.44: TGC Onlins Software の Barrier 機能

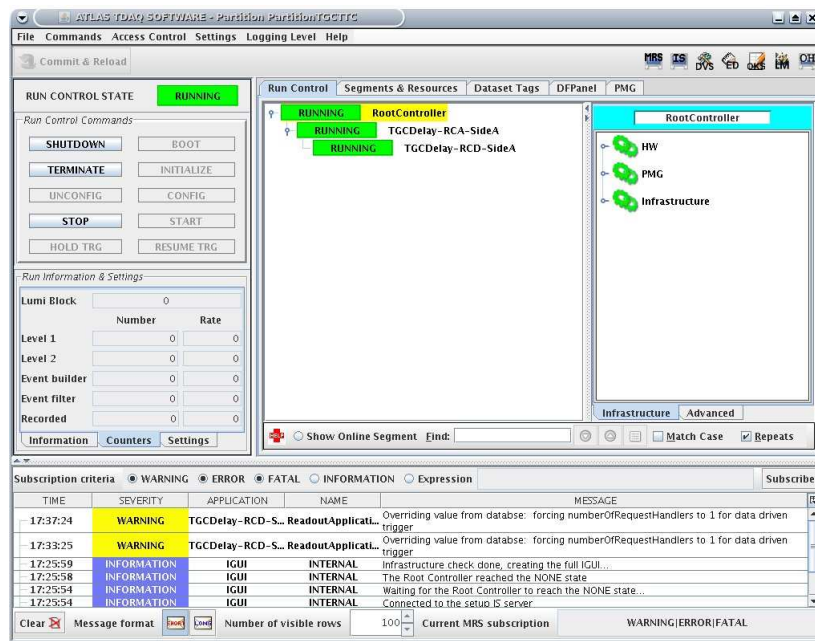


図 4.45: Delay Module ソフトウェアの TDAQ パネル

これらのテストを終えた TGC Delay Module パッケージは Global と Local のソフトウェアパッケージにインストールされ、現在は問題なく動作している。

Partition 'PartitionTGCTTC', server 'Monitoring'			
Name	Type	Modified	Description
TGC_DelayModule1	TGCDelay_IS	9/10/09 15:43:09,552918	
Conductor	ConductorInfo	9/10/09 15:50:38,326137	contains information about
TGC_DelayModule2	TGCDelay_IS	9/10/09 15:43:09,553943	
TGC_DelayModule3	TGCDelay_IS	9/10/09 15:43:09,554815	
/Sampler/TGCDelay-RCB-SideA	SamplerInfo	9/10/09 15:43:19,567142	Contains information about
Value	Type	Name	Description
0000ffff	String		
0000ffff	String		
0000ffff	String		
0000ffff	String		

図 4.46: IS Monitor

4.4.7 Clock Phase Scan の準備状況

現在までに Delay Module の開発をすすめ、完成品のテストを行い、それを動かすためのソフトウェアの開発を行ってきた。モジュール自体も精度よく作られていて、Online Software も安定して動作している。2009 年の時点で Clock Phase Scan に必要なインフラは全て揃ったと言える。また、LHC は 2009 年 11 月より運転を再開し、11 月 24 日には ATLAS 検出器で First Collision を向かえた。あとは TGC 単独でデータテイクを行う時間を得て測定を行うのみの状況となっている。

計算機センターである。Tier0 では Raw データを保存し、Raw データをトラックやクラスターの情報に Reconstruction し、ESD(Event Summery Data) を生成する。Raw データと ESD は Tier1 に送られ、Tier1 は ESD を元に AOD(Analysis Object Data) を生成する。

以上が検出器からの生の情報が物理解析を行えるような形になるまでの一連の流れであるが、この一連の流れの中で使われるのがオフラインソフトウェアと呼ばれるもので、それは Athena という C++ と python をベースにしたフレームワーク上で開発されたものである。4.47 図で DAQ を担当するのはオンラインソフトウェアと呼ばれるもので、オフラインソフトウェアと区別される。

TGC のオフラインソフトウェアには BS を ESD にリコンストラクションするまでに使用される一連のソフトウェアと DQA(Data Quality Assessment) などがある。

4.5.2 TGC オフラインデコードソフトウェアの概要

TGC オフラインソフトウェアの中で BS をデコードするのに使用されるソフトウェアは

/MuonSpectrometer/MuonCnv/MuonTGC_CnvTools

パッケージである。このソフトウェアによって BS→RDO→PRD の変換が行われる。RDO(Raw Data Object) にはハードウェアのレベルの情報をオブジェクト化したものである。TGC の場合はエレクトロニクスの ID やチャンネルの情報などのハードウェア固有のものが入っている。RDO の情報から抽象化を進めたものが PRD(Prepared Raw Data) である。ESD にはこの PRD レベルの情報が含まれる。TGC には PRD が大きく分けて 2 種類あり、TgcPrepData と TgcCoinData と呼ばれる。TgcPrepData は Hit の位置情報が含まれていて、どのチャンネルかを一意に特定できる。TgcCoinData には Tracklet(Low-Pt Coincidence)、HiPt(High-Pt Coincidence)、SL の 3 つのトリガー情報が含まれている。

TGC の ROD の出力フォーマットは 2 種類あるために、BS→RDO のコンバートを行うソフトも 2 種類あり、それぞれ TGC_RodDecoderRawdata (Rawdata フォーマット) と TGC_RodDecoderReadout (Readout フォーマット) と呼ばれる。RDO→PRD の変換を行うのは TgcRdoToPrepDataTool と呼ばれる。TgcRdoToPrepDataTool 内では Hit、Tracklet、HiPt、SL のそれぞれにデコードを行うメソッドが用意されている。

RDO→PRD の変換時には Cabling Service が使用される。Cabling Service は実際のエレクトロニクスの配線をデータベース化し、エレクトロニクスの ID から Offline ID を取得する、または Offline ID からエレクトロニクスの ID を取得するということを可能としている。Offline ID は ATLAS 検出器の全チャンネルに割り振られていて、Offline ID を指定することで、一意にどの検出器のどのチャンネルであるかということ特定できるようになっている。つまり、この Cabling Service を用いることでチャンネルとエレクトロニクスの関係を瞬時に得ることができるのである。

4.5.3 TrigMuonCoinHierarchy の開発

TGC のトリガーは階層構造になっている。4.48 図はそれを表したものである。

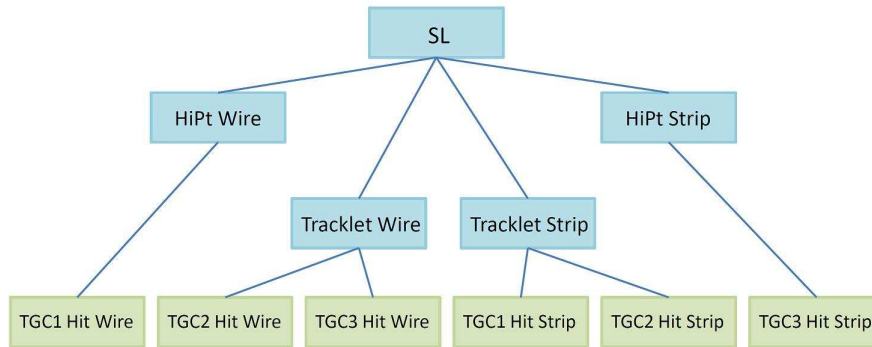


図 4.48: TGC のトリガーの階層構造

Hit は Wire か Strip か、どの Station のヒットだったかという風に大きく分けることが可能である。Tracklet は TGC2 と TGC3 の 2-station コインシデンスを満たし、SLB から出力されたデータをデコードした ESD 上でのデータオブジェクトである。HiPt は Tracklet の条件に加え、TGC1 の Hit を要求し、HPT でのコインシデンス処理を満たしたデータをデコードした ESD 上のデータオブジェクトである⁶。

ここまでの処理は Wire と Strip で独立して行っている。ハードウェアとしての SL は $R\text{-}\phi$ コインシデンスを取り、ミューオンのトラックを構成し、Pt Threshold をつけてトリガーを発行している。SL で処理された結果として残ったデータは SL 上の SLB からリードアウトされ、デコードがなされた後に、“SL”という名前のデータオブジェクトになる。

頂点に位置する SL の方から考えてみると、イベント中に SL データオブジェクトが存在すると、必ず SL に対応した Tracklet は存在するはずである。また、時には対応した HiPt オブジェクトが存在する場合もある。Tracklet が存在すると、必ずそれに対応した TGC2 と TGC3 の Hit が存在するはずである。HiPt が存在するときは、必ずそれに対応した TGC1 の Hit が存在するはずである。以上のように SL を起点に考えてトリガーに対応する Hit を探すこともできる。

このトリガーとヒットの階層関係を利用したのが TrigMuonCoinHierarchy パッケージである。4.48 図のような階層構造が見つかったときに、TgcPrepData(Hit) と TgcCoinData(Coincidence) の両者のポインタを保持し、Hit、Coincidence、Track などからアクセスできるようになっている。このパッケージを用いることで Track からそのもとになった Hit が生成したトリガー情報にアクセスすることを可能にすることに加え、トリガーが存在するときにそれに対応する Hit が存在するかをチェックすることで、ソフトウェアのデコーダーのデバッグやハードウェアの不具合を知ることができる。

⁶HPT はデータをリードアウトする機能が備わっていないので、正確には SL に送られて SL の SLB からリードアウトされたデータをデコードしたものである

TrigMuonCoinHierarchy パッケージの概要

TrigMuonCoinHierarchy パッケージは以下のような各構成要素からなっている。

TgcCoinHierarchy class

階層構造を保ったまま PRDs (TgcPrepData、TgcCoinData) のポインタを保持するためのメソッドを記述するクラス。

TgcCoinHierarchySvc class

TgcCoinHierarchy オブジェクトを保持する。また、クライアントのパッケージに TgcCoinHierarchy のポインタを供給するためのメソッドを記述するクラス。

TgcCoinHierarchyClassifyTool class

TGCCoinHierarchyClassifyTool は TgcCoinHierarchyTriggerSector を用いて TgcPrepData、TgcCoinData をそれぞれのトリガーセクターにクラス分けする。

TgcCoinHierarchyTriggerSector class

一つのトリガーセクター内にある複数の TgcPrepData と TgcCoinData を保持する。TgcCoinHierarchyClassifyTool によって呼び出され、TgcCoinHierarchy を構成するために使われる。

TgcCoinHierarchyFindTool class

必要十分条件を満たすトリガーの階層関係を見つけるためのクラス。TgcCoinData に対応する TgcPrepData を見つける。

TgcCoinHierarchyTestAlg class

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を含んでいる。

4.49 図は TrigMuonCoinHierarchy パッケージ内の各クラスの依存関係を図で示したものである。TgcCoinHierarchy パッケージ部分の基本構成は ICEPP の織田氏によって開発された。私は上記の中で TgcCoinHierarchyFindTool の改良に携わった。

TgcCoinHierarchyFindTool の改良

TgcCoinData (トリガー) に対応する TgcPrepData (Hit) を見つける、または TgcPrepData (Hit) が作りだした TgcCoinData (トリガー) を見つけるためには、Cabling Service を用いて、両者がエレクトロニクスの ID 上の情報で一致するか、つまり RDO レベルの情報で一致しているかを確認する必要がある。というのも、RDO→PRD の変換のときに、一つの RDO をもとに TgcPrepData という PRD と TgcCoinData という PRD に分けて変換するという処理がなされている。もともとは両者は同一の情報から生成されたものである。Cabling Service を TgcCoinHierarchyFindTool に導入し、TgcCoinData と TgcPrepData を RDO レベルの情報で比較することで、必要十分な条件を満たす対応関係を探すことができる。

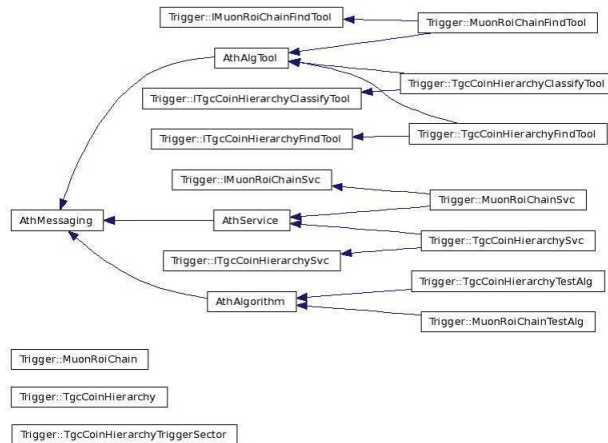


図 4.49: TrigmMuonCoinHierarchy の Class の階層図 [21]

TgcCoinHierarchyFindTool に Cabling Service を導入するにあたり、MuonTgc_CnvTools 内の TgcRdoToPrepDataTool パッケージ実装されている手法を参考にすることにした。TgcRdoToPrepDataTool では Cabling Service を用いて RDO→PRD の変換を行っている。つまり TgcRdoToPrepDataTool に実装されている手法の逆をすることで、PRD→RDO の変換が行える (正確に言えば PRD→RDO の逆変換は完全にはできない。理由については後ほど言及する。)。そこで、TgcRdoToPrepDataTool に実装されている Cabling Service の使用手順を逆にし、使用している関数などもそっくりそのまま使用した⁷。

このような手法をとることで、TgcCoinHierarchyFindTool の開発を進めると同時にデコードソフトウェアの TgcRdoToPrepDataTool のデバッグにも役立てることができる。対応関係がとれない場合が多ければ、TgcRdoToPrepDataTool 内のデコード部分に誤りがあるということである (ハードウェアの不具合がない限り、TgcCoinData と TgcPrepData の対応関係は 100% とれるはずである)。以上のように TgcCoinHierarchyFindTool の開発をデコードソフトウェアのデバッグにも役立てることができた。

以下では、TgcCoinHierarchyFindTool に実装したアルゴリズムについて解説する。

4.50 図は TgcCoinHierarchyFindTool のメインとなるアルゴリズムを模式的に表したものである。一番最初に呼び出されるのは find() で find() の中には findCoincidence() と findHit() の二つの関数がある。

まず findCoincidence() が実行される。findCoincidence() ではトリガーセクター内に見つかった Coincidence の情報 (TgcCoinData) をもとに、TgcCoinHierarchy オブジェクトのベクターを作り、下位の階層のトリガーが見つかるかを調べ、見つかった場合はそのポインタを TgcCoinHierarchy オブジェクトに保持するという操作を行っている。SL、HiPt、Tracklet ごとに for ループが回される

⁷通常であれば TgcRdoToPrepDataTool のヘッダーをインクルードすることで、TgcRdoToPrepDataTool 内で使用している関数を使うことができるが、MuonTgc_CnvTools は特殊で、外部のパッケージからの関数の参照ができなくなっている。そこで、TgcCoinHierarchyFindTool には文字通りそっくりそのまま同じ関数をコピーした。

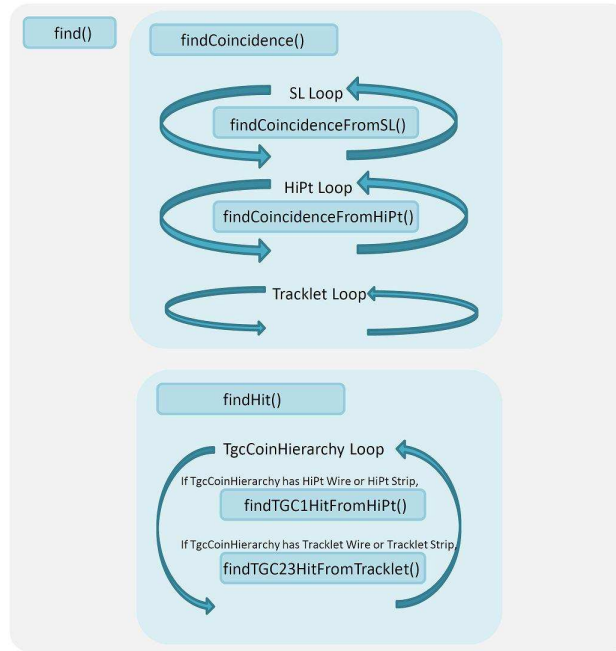


図 4.50: TgcCoinHierarchyFindTool のアルゴリズム
各関数の引数は省略して示している。

が、SL、HiPt のループ中ではそれぞれ `findCoincidenceFromSL()`、`findCoincidenceFromHiPt()` という関数が呼ばれ、下位のトリガー階層が見つかるかチェックを行っている。SL、HiPt、Tracklet 間の対応関係のチェックには `trackletId` と呼ばれる変数を用いている。`trackletId` は同じトリガーの情報であれば、SL、HiPt、Tracklet 間で同じ値を持つという性質を利用している。

次に `findHit()` が呼び出される。`findCoincidence()` 中で作りだされた `TgcCoinHierarchy` のベクターのループを回し、コインシデンスに対応する Hit が見つかる。HiPt Wire または HiPt Strip があるときは `findTGC1HitFromHiPt()` 内で対応付けが行われる。Tracklet Wire、Tracklet Strip の場合は `findTGC23HitFromTracklet()` 内で対応付けを行う。この `findTGC1HitFromHiPt()` と `findTGC23HitFromTracklet()` 内に Cabling 関数を実装した。以下に `findTGC23HitFromTracklet()` と `findTGC1HitFromHiPt()` の処理の詳細を載せる。

findTGC23HitFromTracklet()

1. Tracklet が持つ `OfflineId` から各種のエレクトロニクス ID に変換を行う。このとき、Cabling 関数中の `getLowPtCoincidenceIDfromOfflineID()` を使用する。
2. Hit が持つ `OfflineId` から各種のエレクトロニクス ID に変換を行う。同様に `getLowPtCoincidenceIDfromOfflineID()` を使用する。
3. 両者のエレクトロニクスの ID が一致するかをチェックする。

findTGC1HitFromHiPt()

1. getHiPtIds() という関数内で、HiPt の OfflineId をもとに Cabling 関数の getReadoutIDfromOfflineID()、getSLBIDfromReadoutID() を使用する。
2. getHighPtIDfromOfflineID() を実行する。
3. getBitPosOutWire() (またはgetBitPosOutStrip()) を実行し、TGC3 上の 2 つの bitpos(bit position) を取得する。そして、TGC3 上の 2 つの bitpos から getBitPosInWire() (または getBitPosInStrip()) を実行し、TGC3 上の bipos から HPT のコインシデンスマトリックスで得られた delta の値を考慮してプロジェクションした TGC1 上の 2 つの bitpos を取得する。また、bitpos をもとに TGC1 上での SLB の channel 番号に変換する。
4. Hit の OfflineId をもとに、getReadoutIDfromOfflineID() → getSLBIDfromReadoutID() という流れで Cabling 関数を実行する。
5. 得られた Hit の bitpos をもとに TGC1 の SLB 上での channel 番号に変換する。
6. HiPt の OfflineId から求められた SLB 上での channel の範囲の中に Hit の OfflineId から求めた SLB 上での channel が位置するかを比較する。

SLB の出力するデータは 200bit あるが (SLB の出力するデータフォーマットを付録 C に載せる)、最初の 40bit はトリガーの情報を含んでいる (4.51 図)。残りの 160bit は Hit の位置を表すが、これを bitpos と呼んでいる (4.52 図、各 SLB の bitpos フォーマットについては付録 D に載せる)。

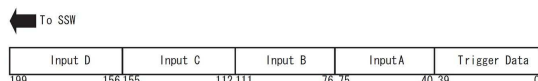


図 4.51: SLB が出力する 200bit のデータ

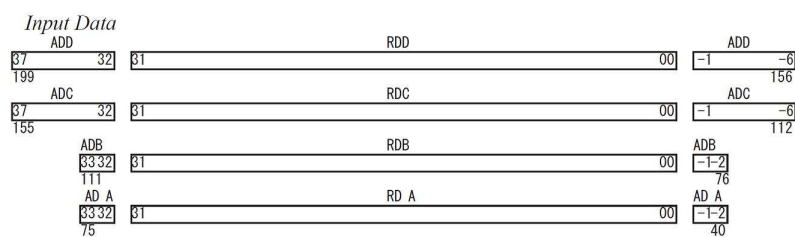


図 4.52: SLB が出力する 200bit のデータ中の Hit 情報

データフォーマットは全ての SLB で共通であるが、Hit 情報の中でどの subMatrix からの出力であるかというのはインストールされる場所で異なってくる。4.53 図は Triplet Wire の bipos がどのようにアサインされているかを示している。

	adjacent				subMatrix=0				subMatrix=1				subMatrix=2				adjacent																											
Layer 1 (C-INPUT)	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199
Layer 2 (B-INPUT)	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155
Layer 3 (A-INPUT)							42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75				

図 4.53: Triplet Wire SLB の bitpos

両端の赤で囲まれた部分は Adjacent チャンネルを示している。この部分は隣の SLB と共有しているチャンネルである。

	Triplet Wire SLB channel																																			
Layer 1 (C-INPUT)	-6	-3	0	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63	66	69	72	75	78	81	84	87	90	93	96	99
Layer 2 (B-INPUT)	-5	-2	1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46	49	52	55	58	61	64	67	70	73	76	79	82	85	88	91	94	97	100
Layer 3 (A-INPUT)	-4	-1	2	5	8	11	14	17	20	23	26	29	32	35	38	41	44	47	50	53	56	59	62	65	68	71	74	77	80	83	86	89	92	95	98	101

図 4.54: Triplet Wire の SLB Channel

これを SLB 上での channel として順番に並べると 4.54 図のようになる。

Triplet Wire の場合はこの SLB Channel を R が大きい方から小さい方へ一列に並べたものを特に RDO Channel と呼ぶときがある。TGC1 での Hit と HiPT Wire の対応関係を見つけるとときにはこの RDO Channel を使用する。TGC3 から TGC1 への bitpos のプロジェクションを模式図で示したものが 4.55 図である。TgcRdoToPrepDataTool においてプロジェクションの元となる TGC3 上の bitpos は ROI の上端と下端の channel を指定するようになっているので、そのやり方を踏襲した。このプロジェクションした RDO Channel の範囲内に Hit の RDO Channel がおさまれば、対応がとれたということになる。

Triplet Strip の場合の bitpos は 4.56 図のようになる。Triplet Strip の SLB には二つのチェンバーからの入力がある。そのため bitpos から SLB Channel に変換するときは一つのチェンバーからの入力を一列に並べる。strip での SLB Channel は 4.57 図のようになる。TGC1 での Strip と HiPt Strip の対応関係を見つけるとときには SLB Channel を使用する。

TrigMuonCoinHierarchy を用いたデータ解析

開発を進める上で、2009 年 8 月のコンバインドラン、Run124887 の宇宙線データを使用してきた。Run124887 を開発初期段階の TrigMuonCoinHierarchy を用いてプロセスした結果と、最近の TrigMuonCoinHierarchy パッケージを用いてプロセスした結果を比較してみる。9 月 19 日に Tag 付けされた TrigMuonCoinHierarchy-00-00-10-07 と 10 月 28 日に Tag 付けされた TrigMuonCoinHierarchy-00-00-19 を使用した。TrigMuonCoinHierarchy を使用するとトリガーのヒエラルキー間での対応がとれる割合を root ファイルとして出力してくれる。

4.58 図と 4.59 図の中で SL→HiPt Strip の対応がとれる確率が良くないが、これはこの Run のと


```

*****
SL->Tracklet Wire           : 628395 / 628490 = 99.9849%
SL->Tracklet Strip          : 628394 / 628490 = 99.9847%
SL (PT=4 or 5)->HiPt Wire  : 72133 / 72133 = 100.0000%
SL (PT=4 or 5)->HiPt Strip : 60566 / 72133 = 83.9643%
HiPt Wire->Tracklet Wire    : 78432 / 78435 = 99.9962%
HiPt Wire->TGC1 Wire Hits   : 78332 / 78435 = 99.8687%
HiPt Strip->Tracklet Strip   : 266521 / 266542 = 99.9921%
HiPt Strip->TGC1 Strip Hits  : 263486 / 266542 = 98.8535%
Tracklet Wire->TGC2 and 3 Wire Hits : 709124 / 709193 = 99.9903%
Tracklet Strip->TGC2 and 3 Strip Hits : 823589 / 824208 = 99.9249%
*****

```

図 4.58: TrigMuonCoinHierarchy-00-00-10-07 の結果

```

*****
SL->Tracklet Wire           : 628395 / 628490 = 99.9849%
SL->Tracklet Strip          : 628394 / 628490 = 99.9847%
SL (PT=4 or 5)->HiPt Wire  : 72133 / 72133 = 100.0000%
SL (PT=4 or 5)->HiPt Strip : 60566 / 72133 = 83.9643%
HiPt Wire->Tracklet Wire    : 78432 / 78435 = 99.9962%
HiPt Wire->TGC1 Wire Hits   : 78332 / 78435 = 99.8687%
HiPt Strip->Tracklet Strip   : 266521 / 266542 = 99.9921%
HiPt Strip->TGC1 Strip Hits  : 265960 / 266542 = 99.7816%
Tracklet Wire->TGC2 and 3 Wire Hits : 709124 / 709193 = 99.9903%
Tracklet Strip->TGC2 and 3 Strip Hits : 823589 / 824208 = 99.9249%
*****

```

図 4.59: TrigMuonCoinHierarchy-00-00-19 の結果

とれる割合が高くなった。HiPt Strip と TGC1 Hit Strip の間の対応関係が悪いのは HPT Endcap Strip ボードの Chip1 にあるハードウェアのバグのためである。このバグによって TGC1 の HPT ボードでコインシデンスのとれたチェンバーがどの入力からきたものかが分からなくなっている。これは言い換えると HPT Endcap Strip ボード Chip1 を通ったデータではシグナルが T6 チェンバーから来たのか、T8 チェンバーから来たのかが分からなくなっている。Offline ソフトウェアではこのバグに対処するため両方のチェンバーに対応する RDO を作るという操作を加えている。このために HiPt Strip オブジェクトが見つかっても対応する Hit がみつからないという事態が起こる。TrigMuonCoinHierarchy-00-00-19 では TgcCoinHierarchyFindTool の中で HiPt Strip に対応する Hit を見つけるときに、HiPt オブジェクトの TGC3 上でのチェンバーの位置と TGC1 上での Hit の

表 4.5: Run124887 の時点でのトリガー条件

From 9th September, 2008	TGC 1		TGC 2 and 3	TGC 2 and 3
	Wire	Strip	Wire	Strip
MU0_TGC_HALO (PT=1)	not required	not required	3/4	3/4
MU0_TGC (PT=4)	$\frac{2}{3}$ and $10 < \delta R \leq 15$			
MU6_TGC (PT=5)	$\frac{2}{3}$ and $ \delta R \leq 10$			

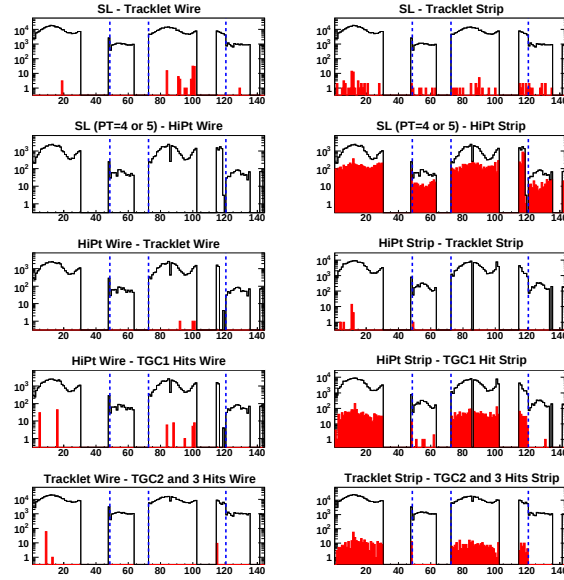


図 4.60: Trigger Sector 別にみた対応関係の結果 (TrigMuonCoinHierarchy-00-00-10-07)

黒色で示されているのが Coincidence と Hit の対応がとれたエントリで、赤色は対応がとれなかったもののエントリである。横軸は TGC の Trigger Sector(Endcap で 48、Forward で 24、A-side と C-side があるので合計で 144 ある) を一列に並べたものである。

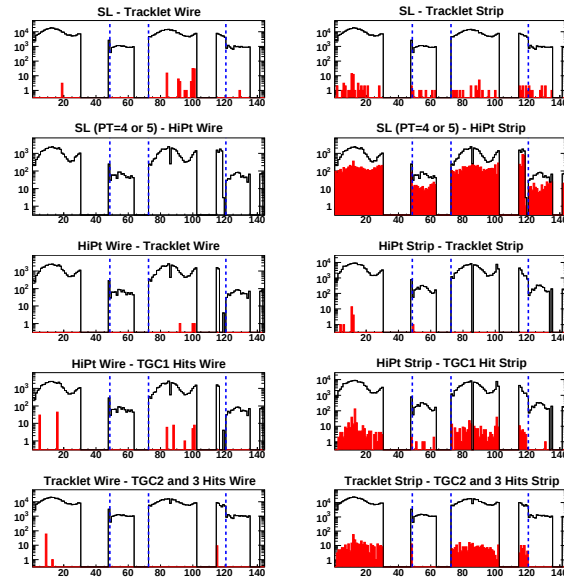


図 4.61: Trigger Sector 別にみた対応関係の結果 (TrigMuonCoinHierarchy-00-00-19)

チェンバーの位置の必要最小限の組み合わせのみに対応する Hit を見つけるという処理が追加されている。さらに TgcCoinHierarchyTestAlg 中では HiPt Strip オブジェクトに対応する Hit が見つからなかったときに、そのパートナーとなる HiPt Strip をもう一度探し、Hit の対応関係とるという処理が追加されている。以上の理由から TrigMuonCoinHierarchy-00-00-19 で HiPt Strip→TGC1 Hit Strip の対応関係が改善している。4.60 図と 4.61 図を見比べてみても、4.61 図の中の HiPt Strip→TGC1 Hit Strip のプロットで赤色のエントリが少なくなっていることから、対応関係が改善していることがわかる。

しかし、4.61 図においても Strip のトリガー階層間で対応関係がとれていないところがあり、しかも特定のトリガーセクターが悪いというわけではないので、何かしらの問題がまだ残されていると考えられる。最新の TrigMuonCoinHierarchy パッケージ (2009 年 12 月 21 日にリリースされた TrigMuonCoinHierarchy-00-01-03) を使い、2009 年ビーム衝突前に取得された TGC 単独の宇宙線 Run(run137898) とモンテカルロデータでの対応関係も調べてみた。

```
*****
6569908 events have been processed.
SL->Tracklet Wire           : 603838 / 603846 = 99.9987%
SL->Tracklet Strip          : 602866 / 603846 = 99.8377%
SL (PT=1, 2, 3 or 6)->HiPt Wire : 109407 / 109407 = 100.0000%
SL (PT=1, 2, 3 or 6)->HiPt Strip : 109029 / 109407 = 99.6545%
HiPt Wire->Tracklet Wire    : 142849 / 142849 = 100.0000%
HiPt Wire->TGC1 Wire Hits   : 142374 / 142849 = 99.6675%
HiPt Strip->Tracklet Strip   : 363240 / 363303 = 99.9827%
HiPt Strip->TGC1 Strip Hit  : 362674 / 363303 = 99.8269%
Tracklet Wire->TGC2 and 3 Wire Hits : 742127 / 742270 = 99.9807%
Tracklet Strip->TGC2 and 3 Strip Hits : 1827437 / 1827437 = 100.0000%
*****
```

図 4.62: run137898 の対応関係の結果

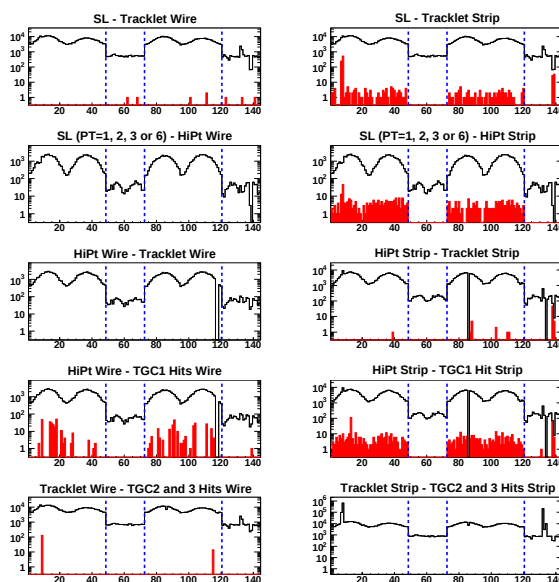


図 4.63: run137898 のトリガーセクター別に見た対応関係の結果

```

*****
51972 events have been processed.
SL->Tracklet Wire           : 41493 / 41493 = 100.0000%
SL->Tracklet Strip          : 41492 / 41493 = 99.9976%
SL (PT=1, 2, 3 or 6)->HiPt Wire : 36207 / 36207 = 100.0000%
SL (PT=1, 2, 3 or 6)->HiPt Strip : 36206 / 36207 = 99.9972%
HiPt Wire->Tracklet Wire    : 42232 / 42232 = 100.0000%
HiPt Wire->TGC1 Wire Hits   : 42232 / 42232 = 100.0000%
HiPt Strip->Tracklet Strip   : 42559 / 42560 = 99.9977%
HiPt Strip->TGC1 Strip Hit   : 42557 / 42560 = 99.9930%
Tracklet Wire->TGC2 and 3 Wire Hits : 43526 / 43526 = 100.0000%
Tracklet Strip->TGC2 and 3 Strip Hits : 44118 / 44145 = 99.9388%
*****

```

図 4.64: モンテカルロで見た対応関係の結果

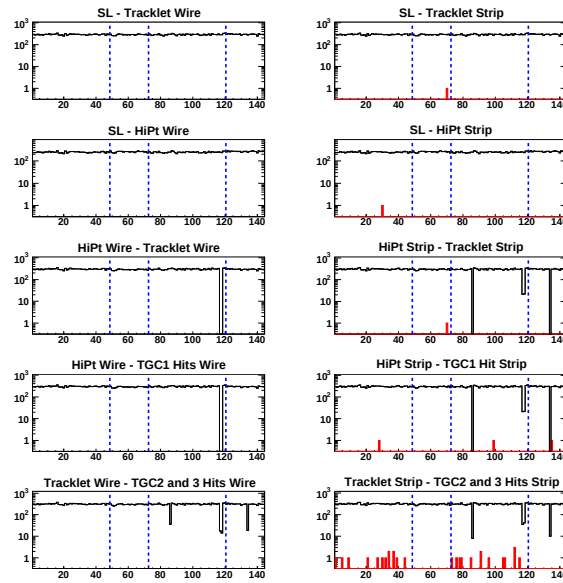


図 4.65: モンテカルロデータをトリガーセクター別に見たときの対応関係の結果

4.62 図は run137898 の対応関係の結果で、4.63 図はそれをトリガーセクター別にみたものである。このデータでも Strip の方で対応がとれないものがみられる。特定のセクターで対応がとれない数が多いものはハードウェアの不具合によるものと考えられる。Strip で対応がとれないものはやはりソフトウェアのバグがまだ残っていると思われる。

また 4.64 図、4.65 図は mc09_valid.106051.PythiaZmumu_1Lepton.reco.ESD⁸を使用したものだが、Strip の方で対応がとれないものがいくつか見られる。しかし、モンテカルロデータでは対応がとれない数はどれも 1、2 など少数である。シミュレーションでトリガーをシュミレートしているのは TrigT1TGC というパッケージであるが、このパッケージに問題があるのか、RDO から PRD に変換する TgcRdoToPrepDataTool に問題があるのかはさらなる調査が必要である。

⁸Pythia というジェネレータを使用し、 $Z \rightarrow \mu\mu$ と崩壊するイベントをシュミレートしたものである。さらに Muon Spectrometer のアクセプタンス外のは除き、少なくとも 1 つのレプトンはリコンストラクションが可能なようにジェネレートされている。

4.6 まとめ

TrigMuonCoinHierarchy パッケージを用いて Coincidence と Hit の対応関係は実データ、モンテカルロともに 99 % 以上対応することを確認できた。特に、Wire の方では対応関係がほぼ 100 % とれるので、TrigMuonCoinHierarchy パッケージを用いて問題のあるハードウェアを見つけることも可能である。ただし Strip の方では SL→Tracklet Strip、SL→HiPt Strip、HiPt Strip→TGC1 Hit Strip などの場所で全体的に対応がとれていない。これがまだ問題として残されている。ソフトウェアのバグによるものと考えられるが、どの部分が問題なのかをさらに追及する必要がある。

第5章 LHC アップグレードに向けたTGC読み出し回路の研究開発

5.1 Super LHC 計画

2009 年 11 月に LHC は安定稼働を迎えたが、LHC のアップグレードに向けた議論もなされている。現在のデザインルミノシティである $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ から一桁上昇させ、 $10^{35} \text{ cm}^{-2} \text{ s}^{-1}$ にしようというのが目的であり、Super LHC 計画 (以後 SLHC 計画とする) と呼ばれる。以前は Center-of-mass エネルギーの上昇も議論されていたが、これは超伝導磁石の大部分を交換することになるので現実的ではなく、今ではルミノシティの上昇を前提にした上で様々な議論がなされている。LHC のアップグレードによって物理のポテンシャルを拡張することが可能となる。

5.1.1 物理的なモチベーション

積分ルミノシティの大きさが最大一桁上昇した場合、新粒子の質量について 20 ~ 30 % ほど発見可能領域が大きくなり、さらに精密測定まで可能にする。SLHC 計画としては各実験において、 3000 fb^{-1} の積分ルミノシティを記録することを目指している。これによって超対称性粒子、MSSM Higgs boson、新種の重いゲージボソンなどの discovery potential が高まることが期待されている。また、統計が 10 倍になることによって標準理論内の、または標準理論を超えた現象の精密測定も可能にする。Higgs とフェルミオン、ボソンとのカップリング、top の (フレーバーの変わる中性カレントの過程を経た) 稀崩壊、ゲージボソンの 3 重または 4 重カップリング、超対称性理論を形成するパラメータなどの測定を可能にする。さらに通常の LHC では制限される、Higgs の pair production を探すことで、Higgs のセルフカップリングを観測できるかもしれない。

以上のように LHC のアップグレードによって非常に興味深い物理現象を測定する機会が得られる。

5.1.2 加速器のシナリオ

LHC 側のアップグレードのシナリオとして様々なケースが考えられているが、検出器にとって重要なパラメータとなるバンチ衝突間隔について分けると、3 つの場合が考えられる。現行のバンチ衝突間隔 (25 ns) を維持するシナリオ (Scenario A)、現行より短い衝突間隔にするシナリオ (Scenario B)、現行より長い衝突間隔にするシナリオ (Scenario C) の三種類である。

Scenario A

このプランでは現行の 25 ns の衝突間隔を維持し、その他の部分を変更することでルミノシティの上昇を図る。この場合に考えられることは、

1. マシンのアップグレードなしにルミノシティを上昇させる
2. IR(Interaction Region) をアップグレードする

の二つである。

1 の場合はバンチあたりの陽子の個数を増やす。この選択をした場合、2 倍より若干多い程度のルミノシティの上昇を得られるが、目標とする 10 倍には及ばない。

2 の場合は新しい Focus 用の 4 重極電磁石を IR にインストールし、ビームの横方向のサイズを小さくする。また、この新しくインストールされるマグネットは現行の位置よりも IP に近いところ (IP から数 m のところ) にインストールされる必要がある。この選択をした場合はさらにバンチの長さを半分にするために RF システムの変更もしなければいけない。この場合ではデザインルミノシティから約 5 倍程度の上昇が見越されている。

IR のアップグレードに合わせて、Piwinski Parameter¹を大きくする装置も加えることで、7.2 倍のルミノシティを得ることもできる。

Scenario B

これは現在のバンチ衝突 25 ns より衝突間隔を短くするというシナリオであるバンチ間隔としては 5 の倍数の 10、15 ns や 12.5 ns という値が議論されている。7.7 から 11.5 倍のピークルミノシティの上昇が期待される。しかし、衝突間隔を短くするためには加速器の諸々の構成要素の交換などで費用がかさむのに加え、バンチ間の間隔が短くなることによって電子雲効果の影響が顕著になる。シンクロトロン放射やハロー陽子がビームパイプの壁面に衝突すると電子を引き剥がす。この電子は加速され電子自身がまた壁面にぶつかることで連鎖的に電子を生成していくが、この反応によって生じるエネルギーがマグネットの温度を上昇させ、クエンチが起きる。またそれらの電子の存在がビームに干渉し、バンチ構造が崩壊する可能性がある。これが電子雲効果である。以上のような理由からこのシナリオが選択される可能性は低い。

Scenario C

これは現在のバンチ衝突間隔 25 ns より長いバンチ衝突間隔の 50 ns や 75 ns を採用するというものである。バンチ衝突間隔が長くなれば電子雲効果を避けることができ、加速器を放射線から防護する観点から見ても利益が大きい。また、縦方向のバンチの形を正規分布のようなかたちから長方形的なフラットな形にすることで $\sqrt{2}$ 倍のルミノシティが得られる。このシナリオでは 9 倍に近いルミノシティの上昇が得られる。

以上のような可能性が考えられていたが、現在では Scenario A(現行と同じバンチ衝突間隔) と Scenario C(50 ns のバンチ衝突間隔) が主に検討されている。アップグレードは phase1、phase2 と

¹Piwinski Parameter $\phi = \theta_c \sigma_z / 2\sigma_x$ である。ここで、 θ_c は Crossing Angle、 σ_z はバンチの長さ、 σ_x はバンチの横方向の広がりである。Piwinski Angle と呼ばれる。

二つの phase で進むことが検討されている。phase1 では新しい直線加速器、Linac4 をインジェクターとして利用するのに加え、IR の 4 重極マグネットを新しい NbTi トリプレットに交換する予定である。phase2 では老朽化している Linac 以外のインジェクターの新調に加え、IR の完全な再設計が計画されている。IR のデザインには今のところ 4 つの計画がある。

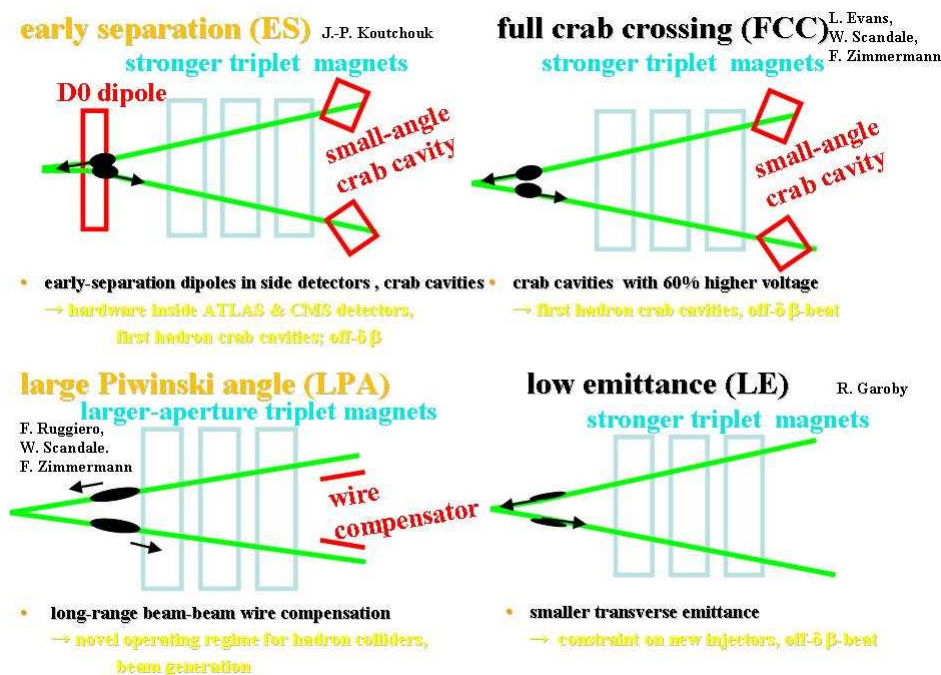


図 5.1: IR のアップグレードプラン [25]

5.1 図は phase2 の IR のアップグレードで期待されている 4 つのプランを示している。4 つのプランは

- Early Separation (ES)
- Full Crab Crossing (FCC)
- Large Piwinski Angle (LPA)
- Low Emittance (LE)

の 4 つである。

ES では新しい 2 重極マグネットを IP をはさんで両サイドにインストールする。これによって 2 重極マグネットの位置までバンチの分離を保つことができ、チューンシフトを減少させることができる。Small-angle crab cavities は 4 重極マグネットの外に置かれ、これによって衝突地点での完全なバンチのオーバーラップを可能にする。

FCCはcrab cavityに全てを委ねるプランである。Crab cavityによってバンチのオーバーラップを最大限にする。

LEプランではバンチ横方向の広がりを抑え、かなり薄くすることでジオメトリのロスを最小限にし、ルミノシティの上昇を得る。

LPAにすることでより強力なビームにすることができる。このプランではバンチの衝突間隔を50 nsにし、衝突角を大きくし、バンチを平たい形にすることでチューンシフトを制限することが求められる。チューンの広がりを小さくするために、それを補うためのワイヤーを必要とする。また LHCbでの衝突のために25 nsの間隔で主要なバンチの間に小さなバンチが挿入される(スーパーバンチ構成という)。

表 5.1 に 4 つのプランごとの主要な加速器のパラメータを載せる。

表 5.1: Main parameters of the proposed schema for the IRs

Parameter	Nominal	Ultimate	ES	FCC	LE	LPA
transverse emittance [μm]	3.75	3.75	3.75	3.75	1	3.75
N_b [10^{11}]	1.15	1.7	1.7	1.7	1.7	4.9
bunch spacing [ns]	25	25	25	25	25	50
longitudinal profile	Gauss	Gauss	Gauss	Gauss	Gauss	Flat
rms bunch length [cm]	7.55	7.55	7.55	7.55	7.55	11.8
β^* [m]	0.55	0.5	0.08	0.08	0.1	0.25
θ_c [μrad]	285	315	0	0	311	381
Piwinski Parameter $\phi = \theta_c \sigma_z / 2 \times \sigma_x^*$	0.64	0.75	0	0	3.2	2
peak L [$10^{34} \text{cm}^{-2} \text{s}^{-1}$]	1	2.3	15.5	15.5	16.3	10.7
peak events/crossing	19	44	294	294	309	403

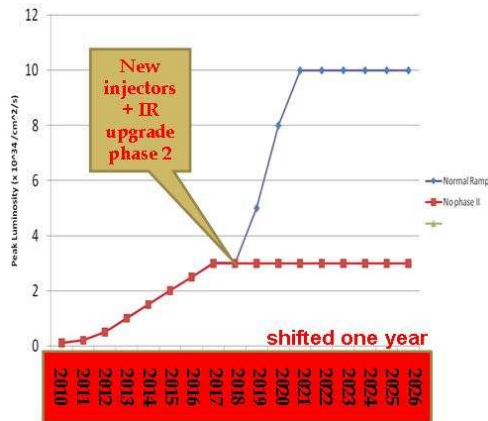


図 5.2: Peak Luminosity [25]

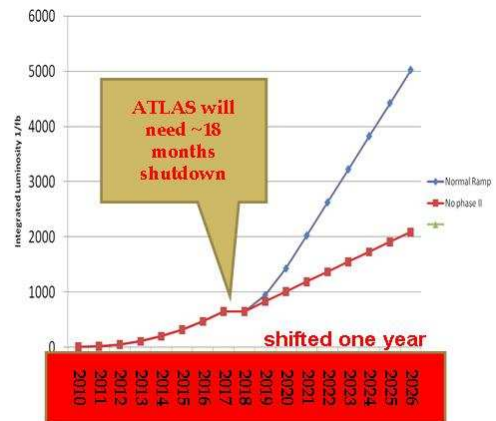


図 5.3: Integrate Luminosity [25]

5.2、5.3 図にピークルミノシティと積分ルミノシティでみる計画進捗の予想図を載せる。

Crab Cavity

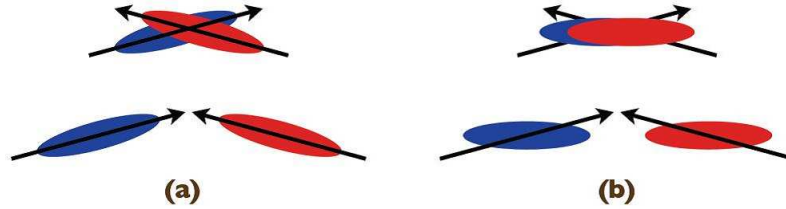


図 5.4: IP における (a)head-on 衝突と (b)crab crossing を持った衝突 [16]

5.4 図は通常の head-on 衝突の場合と crab-crossing を持った衝突の模式図である。Crab crossing がある方が明らかにバンチの重なりが大きくなることによって、ルミノシティの上昇が図れる。Crab RF cavity を IR にインストールすることで、crab crossing を持った衝突が可能となる。現在のところ Crab cavity が実用化されているのは、日本の KEK(高エネルギー加速器研究機構) に建設されている KEKB のみである。

5.1.3 ATLAS MuonSpectrometer のアップグレード

Muon Spectrometer はバックグラウンドに対してある程度のセーフティーファクターを持って建設されているが、LHC のアップグレード時に最も懸念されるのはバックグラウンドに対する対処である。sLHC の運転時にはバックグラウンドも 10 倍になると考えられている。バックグラウンドを減少させるのに最も効果的なのはビームパイプをベリリウム製のものに交換することで、これによってバックグラウンドを $1/2$ または $1/3$ に減少させることができる。

バックグラウンドが上昇すると、MDT チェンバーにおいては特に中心のワイヤーから遠いところで、陽イオンのために分解能の減少が起こる。 $10 \text{ cm}^{-2} \text{ s}^{-1}$ のルミノシティ時には、不定性はあるものの、 η が大きくなるエンドキャップのインナーステーションの領域で 1 kHz/cm^2 が予想されていて、このときには通常の分解能より 1.5 ~ 2 倍悪くなる (5.5 図参照)。ほとんどの領域では $\sim 100 \text{ Hz/cm}^2$ 程度での運転が予想されるため、分解能を悪くする効果は限定的である。MDT の最大ドリフト時間は約 700 ns 程度であるが、MDT のパフォーマンスはバンチ衝突の間隔にはあまり影響を受けない。しかし、MDT の最大ドリフト時間より ATLAS で用いるバンチ間隔が長くなってしまいうようなスーパーバンチ構成のときは、検出器の実効 Occupancy は上昇するものの、パターン認識とトラッキングの効率は悪くなる。

エンドキャップの η の大きい領域は MDT ではなく CSC が用いられている。CSC の電荷収集の時間は 40 MHz のバンチ衝突間隔に合うようにデザインされているため、ハイルミノシティ環境下と早いバンチ衝突間隔でのパフォーマンスの評価のためにはさらなる研究が必要である。

RPC チェンバーのアバランシェ信号の応答速度は 2.3 より、読み出しストリップ中での信号の伝播時間も含めて約 10 ns と早いので、バンチ衝突間隔が短くなったとしても運転可能である。

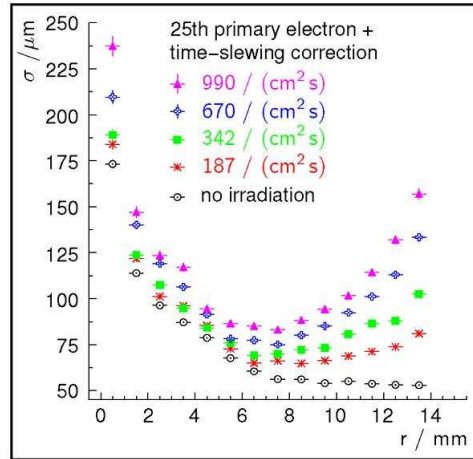


図 5.5: 様々なバックグラウンド環境下でみた MDT の分解能
横軸が中心からの距離、縦軸が位置分解能である。

TGC チェンバーの応答時間は現在の LHC の衝突間隔に合っている。97% の収集効率を得られる応答時間は粒子の入射角により異なるが、10 から 16 ns の間である。バンチ衝突の間隔が変わらなければ、チェンバーの交換は不要である。しかし、インナー領域の TGC(EIFI) は交換が必要となる。

MuonSpectrometer の読み出し、トリガーエレクトロニクスは 40 MHz での動作を前提に作られているため、衝突間隔が短くなる場合、40 MHz よりも速くなる場合、エレクトロニクスの交換をしなければいけない。衝突間隔が変わらない場合でも L1 Latency が変わる場合はエレクトロニクスの交換が必要になる。

5.2 TGC 読み出し系のアップグレード

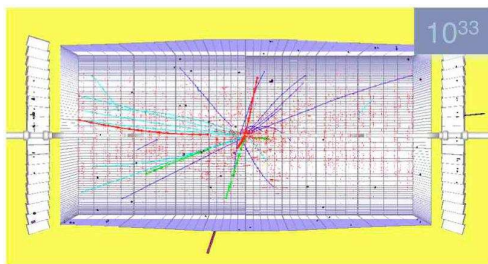


図 5.6: (a)

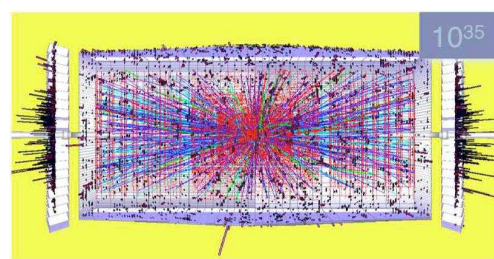


図 5.7: (b)

10^{33} のルミノシティのときにシュミレートしたイベントディスプレイ (a) と、 10^{35} のルミノシティのときにシュミレートしたイベントディスプレイ (b)

5.6、5.7 図はルミノシティが上昇したときの劇的な変化を表している。イベントのパイルアップが

増加するのに加え、トラックとヒット数が増加する。一方で、Level-1 rate がどうなるかというと、

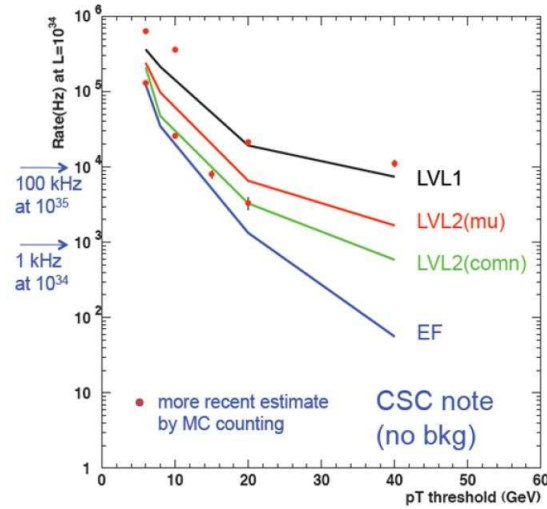


図 5.8: $10^{34} \text{ cm}^{-2}\text{s}^{-1}$ で期待されている inclusive Level-1 rate

5.8 図はルミノシティ $10^{34} \text{ cm}^{-2}\text{s}^{-1}$ のときの inclusive Level-1 rate を示したものであるが、ルミノシティが $10^{35} \text{ cm}^{-2}\text{s}^{-1}$ のときは単純に縦軸の値を 10 倍してこのグラフをみると、MU40 の包括的なトリガーレートは 150 kHz 程度となる。このような理由から SLHC での Level-1 Trigger rate は 150 kHz と見積もられている。また、SLHC では Level-1.5 のハードウェアトリガーを導入してさらにイベントのリジェクションを行う動きもある。以上の事項が TGC の読み出し系に及ぼす影響について考えてみる。

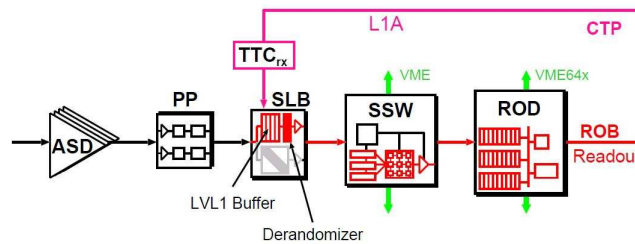


図 5.9: 現在の TGC リードアウトシステム

5.9 図は現行の TGC リードアウトシステム系の概略図である。現在の SLB は Level-1 Buffer の深さとしては、128 clock 分 ($128 \times 25 \text{ ns} = 3.2 \mu\text{s}$) ある。現在の Level-1 Latency は $2.5 \mu\text{s}$ であるので、Level-1 Buffer は十分なデプスが確保されている。一つの L1A についてデータを出力するまでにかかる時間は 230 clock である。このときの load が何% になるかを計算してみると、 $230 \times 100 \text{ kHz} / 40 \text{ MHz} = 57.5\%$ とまだまだ余裕がある。それでは Level-1 Trigger rate が 150 kHz のときの load を計算してみると、

230 kHz/40 MHz ~ 86.3% となる。

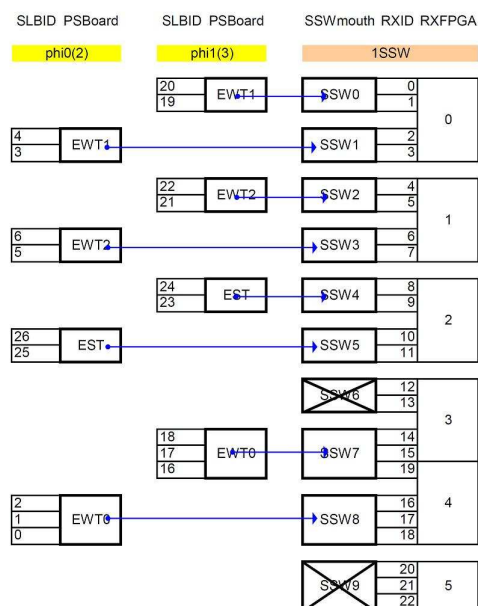


図 5.10: Triplet PSB と SSW のコネクション図

SSWには10口の入力があり、それぞれPSBからの出力を受けるが、現在全ての口を使用していない。一つのL1Aのシグナルがあったときに、SSW内でヘッダーとトレーラーをRXFPGAからTXFPGAへ転送するのに4clockかかり、8-bitのデータをRXからTXへ転送するのに1clockかかる。さらにRXFIFOの読み出しをスイッチするのに7clock要する。現在のPSBとSSWのコネクションからすると、最大で18個のRXFIFOに入力のある場合がある。それはEndcap TripletのPSB(EWT、EWS)と接続するSSWのときである(5.10図参照、図のRXIDの個数がRXFIFOの個数に一致する)。18個のRXFIFOにヒットデータがなかった場合に処理にかかるクロック数は $(4 + 7) \times 18 = 198$ clockで、このときのloadは $198 \times 100 \text{ kHz} / 40 \text{ MHz} \sim 50\%$ である。RXFIFOそれぞれに1hitのデータがあった場合、処理にかかるクロック数は $(4 + 1 + 7) \times 18 = 216$ clockで、そのときのloadは $216 \times 100 \text{ kHz} / 40 \text{ MHz} = 54\%$ となる。それではそれぞれのSLBにトラックが存在した場合、その処理にかかるクロック数は $(4 + 5 + 7) \times 18 = 288$ clockで、そのときのloadは $288 \times 100 \text{ kHz} / 40 \text{ MHz} = 72\%$ となる。この場合150 kHzのLevel-1 Trigger rateのときには108%となってしまうので、SSWの数を増やしてPSBとSSWのコネクションを考え直す必要がある。

5.11図は新しいTGCの読み出し系の一つの場合を示したものである。Level-1 Latencyが変わらなければSLBとSSWはそのまま使用することが可能である。しかし、SSWとPSBのコネクションに関してはまた新たに考える必要性はある。Level-1.5 Triggerが導入されるのであれば、それはRODにシグナルとして供給されると考えられるため、RODは新たに開発する必要がある。またこの新RODには、Level-1.5を受け取る機能の他に、運転させる上である程度の柔軟性も求められるため、Micro Processorを搭載し、CやC++などのプログラムでも操作可能であることが望まれる。

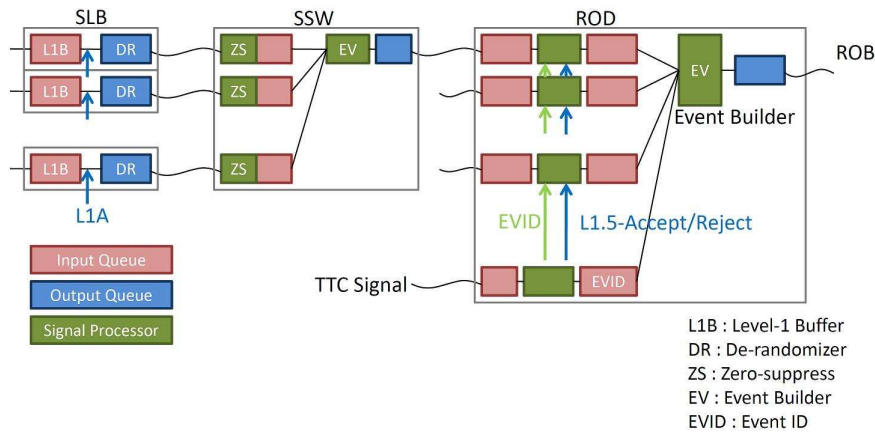


図 5.11: 新しい TGC 読み出し系の案

5.3 新 ROD のデザイン

ROD の基本機能は、複数の SSW からくるデータを受け取り、イベントビルディングを行い ROS にデータを送ることである。現行の ROD は 10 個の SSW からの入力を受けるが、これは 1/12 セクター内の SSW 全てからの入力に相当する。ROD と SSW 間の接続には G-Link が使用されている。ROD はこの SSW10 個からの入力をまとめ、ROS にデータを送る。ROD-ROS 間の接続には S-Link が使用される。

以上の機能、構成は新 ROD においても受け継がれるものと思われる。これを踏まえた上で今現在考えている新 ROD のデザインを示したのが 5.12 図である。

5.12 図にあるように、SSW からの入力を受ける G-Link Rx Chip を 10 個、大規模な FPGA を 1 個、S-Link Tx Chip を 1 個搭載する。1 個の大規模な FPGA が SSW からのデータを集め、イベントビルディングを行い S-Link Tx chip にデータを送るロジック (Read out driver logic) を担当する。また、この FPGA に Micro Processor を搭載することを検討している。Micro Processor は SSW から送られてくるデータが壊れていないか、もしくは大量のヒットデータが送られてきたときに対処するなどのエラーハンドラーの役割をする。通常の Read out driver logic に加え Micro Processor を一つの FPGA に実装させたいと考えているので、FPGA としてはリソースが膨大で、IO ピンの数が多いものが求められる。現在の時点で使用を検討しているのは、2009 年に発表された Xilinx 社の Spartan 6 デバイスである。

5.13 図は現在検討している FPGA 内部のロジックのダイアグラムを示している。5.13 図にあるように CPU コアを搭載することで、CPU コアを介して Ethernet や RS232C のようなコンソールインターフェースを利用することが可能となり、フレキシブルな運転が可能になる。また、CPU コアを介してエラーがあったときに、そのデータを Dual Port Memory に格納し、VME バスからの読み出しができるような機能を加えることも考えている。

5.12 図には Level-1.5 Trigger をどこで受けるかは示されていないが、どのように受けとるのかは今後検討する必要がある。

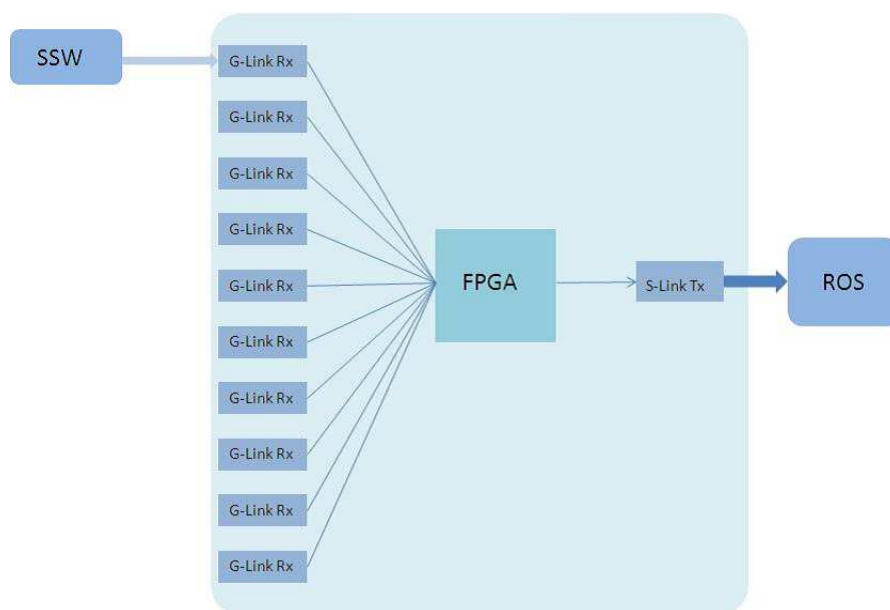


図 5.12: 新 ROD のデザイン

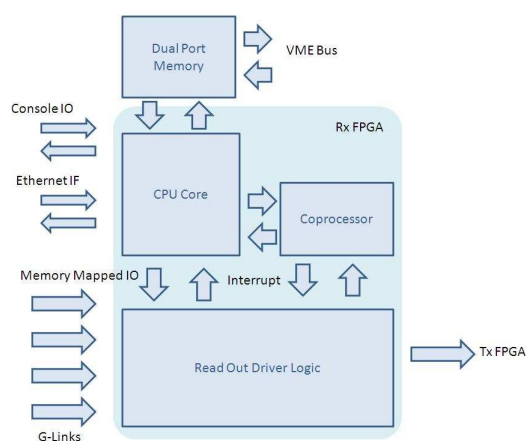


図 5.13: FPGA 内部のブロックダイアグラム

5.4 CPU コアの評価

今回検証した CPU コアはソフト CPU コアと呼ばれるもので、FPGA を製造する際に埋め込まれる CPU コア (ハード CPU コア、Power PC など) ではなく、専用のソフトウェアを用い通常のロジックブロックを使用して生成されるものである。この CPU コアは MicroBlaze と呼ばれ、32bit の RISC

型プロセッサ²である。ボード、MicroBlaze を載せるデバイスにもよるものの、最大 150 MHz での動作が確認されている。CPU コアを検証する評価ボードとしては Spartan 3E が搭載された Xilinx 社製 Spartan 3E Starter Kit を使用した。また OS を搭載した開発事例もあり、 μ ITRON、uClinux などの組み込みシステム向け OS が搭載可能なことが確認されている。

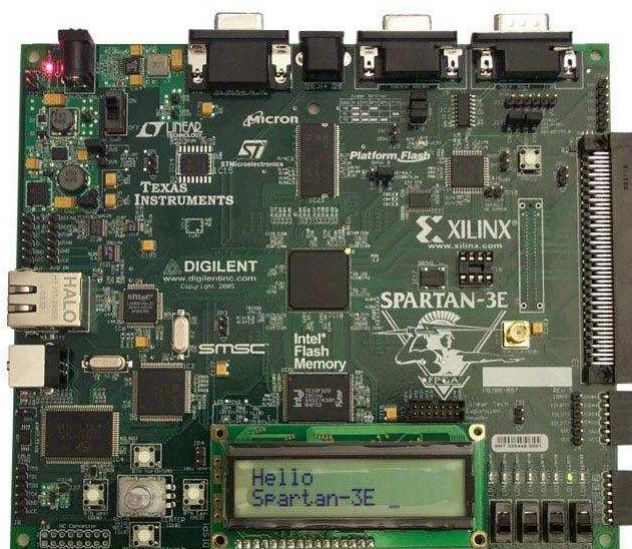


図 5.14: Spartan 3E Starter Kit

5.14 図が Spartan 3E Starter Kit である。このボードには 64MB の DDR SDRAM、16MB のパラレル Intel NOR Flash PROM、9pin RS-232 が 2 機、10/100 イーサネット物理レイヤインターフェース、その他にも各種のデバイス、スイッチ類が搭載されている。

CPU コアの生成には Xilinx Platform Studio(XPS) を使用した。XPS には Base System Builder(BSB) という機能があり、MicroBlaze システムを簡単に作成できる (5.15 図参照)。また、今回は Xilinx 社製のボードを使用したため、ボードのコンフィギュレーションがソフトウェアに登録されているため、簡単に各機能を追加できる (5.16 図、5.17 図参照)。各 IO、Peripheral のアドレスも BSB が自動生成してくれる。あとはユーザーがそれらを用いてプログラムを作成することで、簡単に評価が行えるようになっている。

5.19 図は基本的な IO、Peripheral を搭載した MicroBlaze のダイアグラムである。このシステムには外部メモリ (DDR SDRAM) は搭載されていない。プログラムは全て 5.19 図中の上部にある BRAM で実行される。各種 IO や Peripheral は図中の下部にあるが、Processor とは Processor Local Bus(PLB) を介してつながっている。MicroBlaze の評価をする上で RS232_DCE コネクタを標準入出力として使用した。RS232_DCE とデスクトップコンピュータのシリアルポートをシリアルケーブルを介してつなぎ、TeraTerm をコンソールとして使用した (5.21 図)。5.20 図に開発風景の写真を載

²Reduced Instruction Set Computer。命令の種類を減らし、回路を単純化して演算速度の向上を図る設計手法に基づいて設計されたプロセッサ。

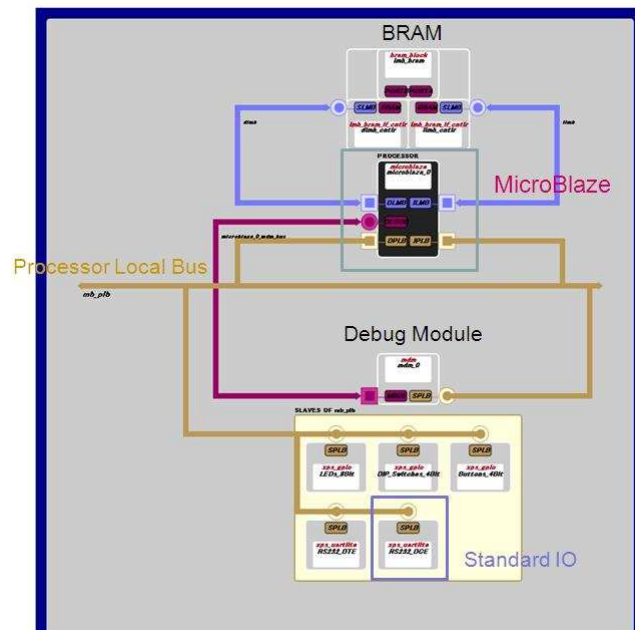


図 5.19: MicroBlaze のブロックダイアグラム



図 5.20: 開発風景



図 5.21: TeraTerm

ていることの確認にもなる。

ソースコード 5.1: tutorial.c

```

1 #include "xparameters.h"
2
3 #include "stdio.h"
4
5 int main(void)
6 {
7     print("-- Entering main() --\r\n");
8
9     int num=0;
10    int i, wait;
11
12    for(i=0; i<10; i++) {
13        xil_printf("num = %d\r\n", num);
14
15        num++;
16
17        for(wait=0; wait<5000000; wait++);
18    }
19
20    print("-- Exiting main() --\r\n");
21    return 0;
22 }
23

```

ソースコード 5.1 が今回作成したソースコードである。5.1 内では printf ではなく、Xilinx から提供されているライブラリにある xil_printf を使用している。普通の printf ではメモリの消費量が大きいため、xil_printf の使用が推奨されている。

```

-- Entering main() --
num = 0
num = 1
num = 2
num = 3
num = 4
num = 5
num = 6
num = 7
num = 8
num = 9
-- Exiting main() --

```

図 5.22: tutorial.c の出力

5.22 図は TeraTerm に表示された tutorial.c の出力結果である。標準出力が正しく機能していることが確認できた。

5.4.2 IO の操作

次に検証したのはプログラムからの IO の操作である。Spartan 3E Starter Board には 8 つの LED が搭載されているので、プログラムからこの LED の操作ができるかを試してみた。MicroBlaze から IO を操作するには XGpio オブジェクトを使用する。BSB によって LED にベースアドレスが

与えられているので、ベースアドレスを元にポインタに LED を操作するためのアドレスを渡し、XGpio_DiscreteWrite() 関数と XGpio_DiscreteRead() 関数を使用して読み書きを行う。

ソースコード 5.2: led_ctrl.c

```
1 #include "xparameters.h"
2 #include "xgpio.h"
3 #include "stdio.h"
4 #include "xio.h"
5 #include "xstatus.h"
6
7 int main(void) {
8
9     print("-- Entering main() --\r\n");
10
11     xil_printf("led_ctrl.c start!! \r\n");
12
13     int i;
14     volatile int wait;
15
16     XStatus xs;
17     XGpio led;
18     xs = XGpio_Initialize(&led, XPAR_LEDS_SBIT_DEVICE_ID);
19
20     i = 0x55;
21
22     while(1) {
23         i = ~i;
24
25         for(wait=0; wait<10000000; wait++);
26
27         xil_printf("Execute DiscreteWrite \r\n");
28         XGpio_DiscreteWrite(&led, 1, i);
29     }
30
31     print("-- Exiting main() --\r\n");
32     return 0;
33 }
34
35 }
```

ソースコード 5.2 が LED の操作のために作成したプログラムである。8 個の LED のうち一つ飛ばしで 4 つが交互に点灯するのを無限に繰り返すプログラムである。また人の目で見えて交互に点灯しているのがわかるように、for 文を使用した wait 処理を行っている。

5.23 図は led_ctrl.c を実行したときの TeraTerm への出力で、5.24 図、5.25 図は LED が実際に交互に点灯しているのを写したものである。以上のように IO の操作が行えることも確認できた。

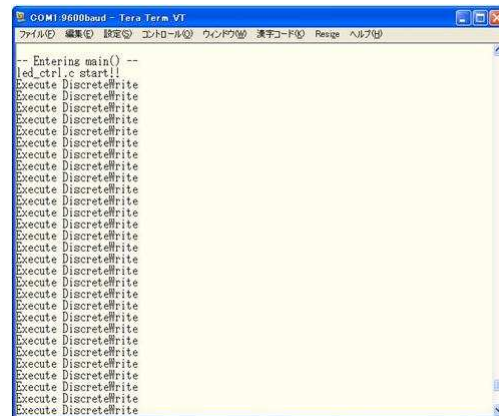


図 5.23: led_ctrl.c の出力

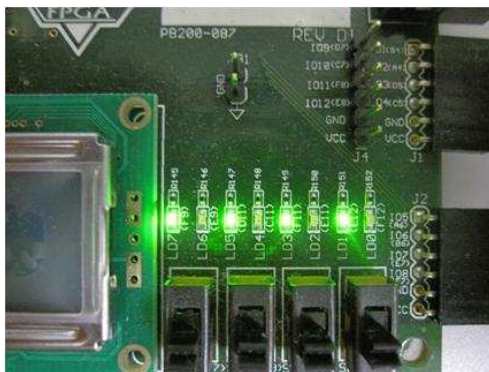


图 5.24: (a)

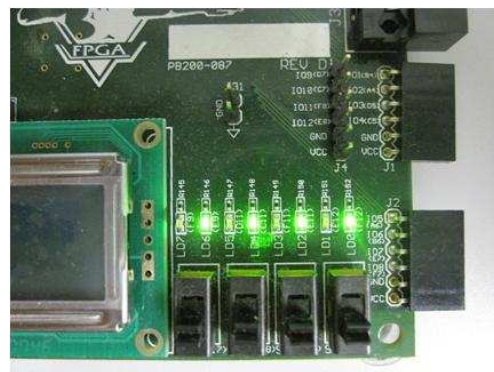


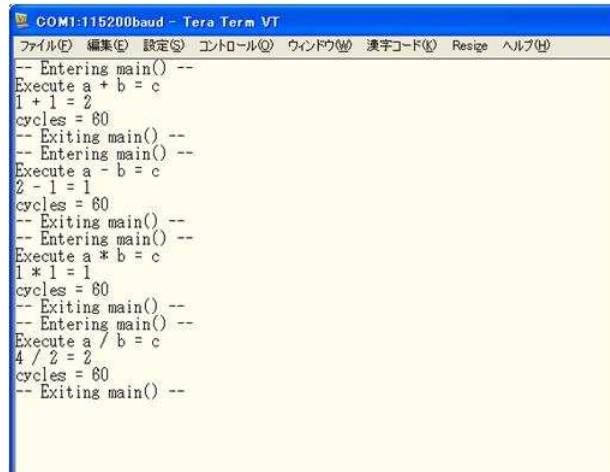
図 5.25: (b)

(a)→(b)→(a)→... のように LED が交互に点灯する

5.4.3 性能の評価

MicroBlazeにはTimer機能を追加することができ、処理にかかったクロック数を計測することができる。ここではMicroBlazeに計算をさせ、それにどれくらいの処理がかかるのかを計測し、一般的なCPUに処理させたときと比較をしてみた。

まず、MicroBlaze に簡単な四則演算をさせてみた。5.26 図がその結果である。簡単な四則演算では四則間では違いがみられず、60 clock かかった。この MicroBlaze は 50 MHz で動作しているので、 $60 \times 20 \text{ ns} = 1.2 \mu\text{s}$ に相当する。一般的な CPU ではこれらの計測はできなかったので、10,000,000 回の空の for loop を処理させ、MicroBlaze と一般的な CPU (今回は Pentium D、2.80 GHz を使用した) と比較を行った。



```
COM1:115200baud - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(C) ウィンドウ(W) 渡字コード(O) Resize ヘルプ(H)
-- Entering main() --
Execute a + b = c
1 + 1 = 2
cycles = 60
-- Exiting main() --
-- Entering main() --
Execute a - b = c
2 - 1 = 1
cycles = 60
-- Exiting main() --
-- Entering main() --
Execute a * b = c
1 * 1 = 1
cycles = 60
-- Exiting main() --
-- Entering main() --
Execute a / b = c
4 / 2 = 2
cycles = 60
-- Exiting main() --
```

図 5.26: 簡単な四則演算の計測結果

ソースコード 5.3: forloops

```
1 #include "xparameters.h"
2 #include "xtmrctr.h"
3
4 #define TMRCTR_DEVICE_ID XPAR_TMRCTR_0_DEVICE_ID
5
6 #define TIMER_COUNTER_0 0
7
8 XTmrCtr TimerCounter;
9
10 int main(void)
11 {
12     print("-- Entering main() --\r\n");
13
14     int cycles;
15     volatile int i;
16
17     XStatus Status;
18     XTmrCtr *TmrCtrInstancePtr = &TimerCounter;
19     Xuint8 TmrCtrNumber = TIMER_COUNTER_0;
20
21     Status = XTmrCtr_Initialize(TmrCtrInstancePtr, TMRCTR_DEVICE_ID);
22     if( Status != XST_SUCCESS ) {
23         return XST_FAILURE;
24     }
25
26     XTmrCtr_SetResetValue(TmrCtrInstancePtr, TmrCtrNumber, 0x00000000);
27
28     XTmrCtr_Reset(TmrCtrInstancePtr, TmrCtrNumber);
29
30     XTmrCtr_Start(TmrCtrInstancePtr, TmrCtrNumber);
31
32     for(i=0; i<10000000; i++);
```

```

33
34     XTmrCtr_Stop( TmrCtrInstancePtr , TmrCtrNumber );
35
36     cycles = XTmrCtr_GetValue( TmrCtrInstancePtr , TmrCtrNumber );
37
38     xil_printf("Execute 10000000 of for-loops \r\n");
39     xil_printf("cycles = %d \r\n", cycles);
40
41
42     print("— Exiting main() --\r\n");
43     return 0;
44 }

```

ソースコード 5.3 は Timer のインプリメンテーションを行ったプログラムである。Timer に 10,000,000 回の空の for loop を回している。

```

COM1:115200baud - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(C) ウィンドウ(W) 漢字コード(K)
-- Entering main() --
Execute 10000000 of for-loops
cycles = 130000066
-- Exiting main() --

```

図 5.27: forloops.c の出力結果

5.27 図は 5.3 の出力結果である。MicroBlaze の処理では 130,000,066 clock かかっている。これは時間に換算すると、 $130000066 \times 20 \text{ ns} \sim 2.6 \text{ s}$ に相当する。

同様の処理を一般的な CPU で計算させてみた。

5.28 図がその実行結果である。C 言語のプログラムによる時間情報の取得には clock_gettime 関数を使用した。clock_gettime で得られる値には若干のふらつきがあるので、10,000,000 回の for loop にかかる時間の計測を 100 回繰り返し、標準誤差付きで実行時間を求めるプログラムを作成した。5.28 図の結果をみると、count の値が 98 になっているが、この値は 100 回の繰り返しのうち有意な値が取得できた回数を表している。ごく稀に有意な値が取得できないことがあるため、そのような場合は除外するようなプログラムになっている。有意な値が得られない回数は 100 回中、2 回もしくは 3 回とそれほど多くないため、今回はこの原因を探ることはしなかった。

また今回使用した clock_gettime 関数は最大ではナノ秒単位の精度が得られるはずだが、マイクロ秒精度の計測が可能な gettimeofday 関数での結果とほぼ変わらなかったため、マイクロ秒以下は切り捨てて考えると、プログラムによって得られた実行時間は

$$25.294 \pm 0.074 \text{ ms}$$

となる。

```

yohsuke@sakamoto-lab-desktop: ~/work
ファイル(E) 編集(E) 表示(V) 端末(T) ヘルプ(H)
Execute 100000000 of for-loops
Processing Time = 25207154.000000 nsec.
Execute 100000000 of for-loops
Processing Time = 25257112.000000 nsec.
Execute 100000000 of for-loops
Processing Time = 25150532.000000 nsec.
Execute 100000000 of for-loops
Processing Time = 25153860.000000 nsec.
Execute 100000000 of for-loops
Processing Time = 25165114.000000 nsec.
Execute 100000000 of for-loops
Processing Time = 25116344.000000 nsec.
Execute 100000000 of for-loops
Processing Time = 25275506.000000 nsec.
Execute 100000000 of for-loops
Processing Time = 25288324.000000 nsec.
Execute 100000000 of for-loops
Processing Time = 25184184.000000 nsec.
*****
mean = 25294772.000000 nsec.
count = 98
standard deviation = 739291.125000
standard error = 74679.679688
yohsuke@sakamoto-lab-desktop: ~/work$

```

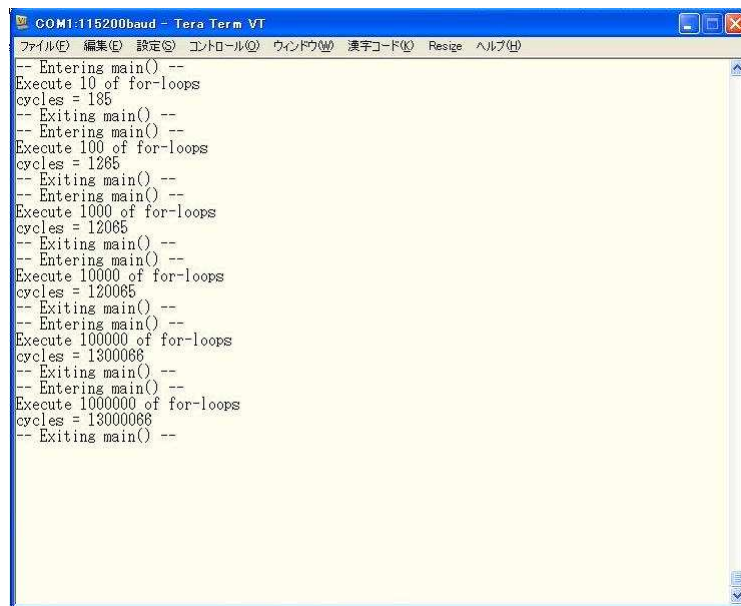
図 5.28: 一般的な CPU(Pentium D) での実行結果

	MicroBlaze	Pentium D
Clock Frequency	50 MHz	2.8 GHz
Processing time (sec)	2.6 s	25.294 ± 0.074 ms
Processing time (clock)	130,000,066	~ 70,000,000
Processing time of 1 for-loop (clock)	~ 13	~ 7

表 5.2: MicroBlaze と Pentium D での実行結果

表 5.2 は両者の計測結果をまとめたものである。2.8GHz のクロックの 1 サイクルはだいたい 0.36 ns であるから、Pentium D で計算に要したクロック数を見積もると ~ 70,000,000 である。MicroBlaze では for loop の 1 loop に 13 クロックかかっていたが、Pentium D では 7 クロックほどで Pentium D の方が 2 倍程度処理が早い。実行時間を比較すると Pentium D の方が 100 倍程度早い、Pentium D の方が MicroBlaze より 50 倍ほどクロック周波数が早く、for loop の処理も 2 倍ほど早いことから理に合った値になっている。

この結果から一般的な PC に搭載されている CPU と比較すると、MicroBlaze の計算処理能力はそれほど高くはないので、for 文や if 文を多用するような複雑なプログラムの実行はなるべく避けるべきであることがわかった。MicroBlaze で使用するクロック周波数を高くすることで、処理時間の短縮は可能だが、安定して動作する周波数がどのくらいであるかは別途調査が必要である。



```
COM1-115200baud - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(C) ウィンドウ(W) 漢字コード(C) Resize ヘルプ(H)

-- Entering main() --
Execute 10 of for-loops
cycles = 185
-- Exiting main() --
-- Entering main() --
Execute 100 of for-loops
cycles = 1265
-- Exiting main() --
-- Entering main() --
Execute 1000 of for-loops
cycles = 12065
-- Exiting main() --
-- Entering main() --
Execute 10000 of for-loops
cycles = 120065
-- Exiting main() --
-- Entering main() --
Execute 100000 of for-loops
cycles = 1300066
-- Exiting main() --
-- Entering main() --
Execute 1000000 of for-loops
cycles = 13000066
-- Exiting main() --
```

図 5.29: for loop の回数を変えて実行した結果

for loop の回数	処理にかかったクロック数
10	185
100	1265
1000	12065
10000	120065
100000	1300066
1000000	13000066

表 5.3: for loop の回数と処理にかかったクロック数の関係

また、for loop の回数を一桁ずつ変えて実行結果を調べてみた。5.29 図がその実行結果であり、表 5.3 はそれを表にまとめたものである。10~10,000 回までの値をみると、オーバーヘッドになっているのは 65 clock 分で、純粋な for loop の演算には 10 回だと 120 clock かかっており、100 回だと 1200 clock という感じにちょうど 10 倍になっている。10,000 回と 100,000 回のところはちょうど 10 倍ということにはなっていない。10,000 と 100,000 の間のどこかで内部処理に変化があり遅くなるのだと考えられる。100,000 回以降はまたちょうど 10 倍ずつクロック数が増加していく。

5.4.4 外部メモリの操作

Spartan 3E Starter Kit には外部メモリとして 64MB の DDR SDRAM が搭載されている。この DDR SDRAM はプログラムを実行するメモリとして使用することも可能だし、実行プログラムを格納し、ここから BRAM にプログラムをロードして BRAM でプログラムを実行するという使い方もできる。またメモリであるので、ある値を保持させるという使い方もできる。新 ROD では、エラーがあったときにそれがどこで起きたのかという情報を外部メモリに保持させる機能を載せることを検討している。つまり、CPU コアから外部メモリの読み書きを行いたいということである。そこで、MicroBlaze が DDR SDRAM の読み書きを行うプログラムを作成し、実際それにどれくらいの時間がかかるのかを Timer を用いて計測を行った。

ソースコード 5.4: MemoryTest.c

```
1 #include "xparameters.h"
2 #include "xstatus.h"
3 #include "stdio.h"
4 #include "xtmrctr.h"
5
6 int MemoryTest(Xuint32 *Address, Xuint32 Value);
7
8 #define TMRCTR_DEVICE_ID XPAR_TMRCTR_0_DEVICE_ID
9 #define TIMER_COUNTER_0 0
10 XTmrCtr TimerCounter;
11
12 int main(void)
13 {
14
15     print("-- Entering main() --\r\n");
16
17     XStatus status;
18     int cycles;
19
20     print("Starting MemoryTest for DDRSDRAM:\r\n");
21     print("Runing 32-bit Memory Test...\r\n");
22
23     //add timer
24     XTmrCtr *TmrCtrInstancePtr = &TimerCounter;
25     Xuint8 TmrCtrNumber = TIMER_COUNTER_0;
26
27     status = XTmrCtr_Initialize(TmrCtrInstancePtr, TMRCTR_DEVICE_ID);
28     if( status != XST_SUCCESS ) {
29         return XST_FAILURE;
30     }
31
32     XTmrCtr_SetResetValue(TmrCtrInstancePtr, TmrCtrNumber, 0x00000000);
33
34     XTmrCtr_Reset(TmrCtrInstancePtr, TmrCtrNumber);
35
36     XTmrCtr_Start(TmrCtrInstancePtr, TmrCtrNumber);
37
38     status = MemoryTest((Xuint32*)XPAR_DDR_SDRAM_MPMC_BASEADDR, 0x00000001);
39
```

```

40 XTmrCtr_Stop(TmrCtrInstancePtr, TmrCtrNumber);
41 cycles = XTmrCtr_GetValue(TmrCtrInstancePtr, TmrCtrNumber);
42
43 if(status == XST_SUCCESS) {
44     print("PASSED!\r\n");
45 } else {
46     print("FAILED!\r\n");
47 }
48
49 xil_printf("cycles = %d \r\n", cycles);
50
51 print("-- Exiting main() --\r\n");
52 return 0;
53
54 }
55
56 int MemoryTest(Xuint32 *Address, Xuint32 Value)
57 {
58     int i;
59     Xuint32 FirstValue = Value;
60     Xuint32 ReadValue;
61
62     Address[0L] = Value;
63
64     ReadValue = Address[0L];
65
66     if(ReadValue != FirstValue) {
67         return XST_FAILURE;
68     }
69
70     return XST_SUCCESS;
71 }
72

```

ソースコード 5.4 が DDR SDRAM の読み書きを行うプログラムである。DDR SDRAM もベースアドレスが BSB でプロジェクトを作成したときに生成されているので、そのアドレスをポインタとして使用し、値を代入したり、ポインタがポイントするアドレスが保持している値を読み出すことで、DDR SDRAM の読み書きが行える。これは VME をコントロールするプログラムとほぼ同じで、VME ソフトウェアの方では、PC のメモリ空間上にモジュールのベースアドレスをマップし、ポインタの操作でレジスタの読み書きなどを行っている。ソースコード中でこの読み書きをおこなっているのは、MemoryTest という関数である。今回は Timer でこの MemoryTest 関数を実行するのに何 clock かかるかを計測した。

また比較する対象として、VME の汎用モジュールである PT5 を使用した。PT5 はボード上に Dual Port Memory(DPM) を 2 機搭載している (5.30 参照)。DPM 一つが 64K のアドレス空間 × 16 bit データの大きさを持っている。つまり 2 つ同時にアクセスすると 64K アドレス空間 × 32 bit データのメモリにアクセスしているように見える。PT5 は VME バスから 2 つの DPM に同時にアクセス可能なように作られている。PC から bit3 を経由して PT5 上の DPM にアクセスするのにかかる時間と MicroBlaze で DDR SDRAM にアクセスするのにかかる時間を比較した。

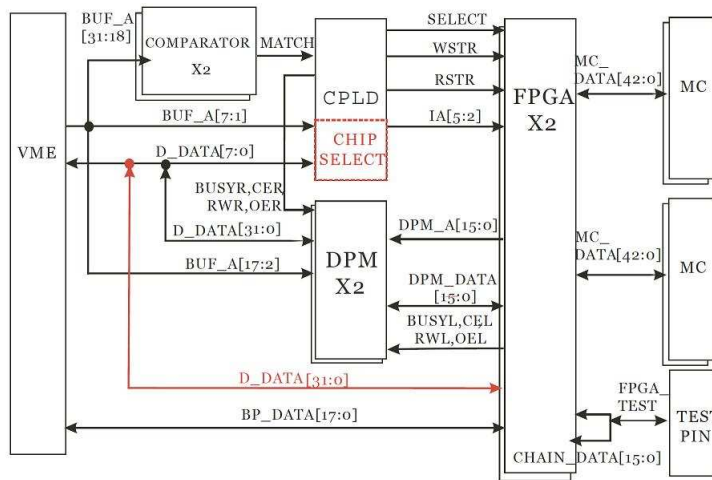


図 5.30: PT5 のブロックダイアグラム

```
COM1:115200baud - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(C) ウィン
-- Entering main() --
Starting MemoryTest for DDR_SDRAM:
Runing 32-bit Memory Test...
PASSED!
cycles = 81
-- Exiting main() --
```

図 5.31: MemoryTest.c の実行結果

```
yohsuke@misuzu:~/vme_software/PT5/bin
[yohsuke@misuzu bin]$ ./DpmTest2
DpmTest Success!
read_value = 0x10101010
Timer Result>>time = 6 usec
[yohsuke@misuzu bin]$
```

図 5.32: DPM アクセスの実行結果

5.31、5.32 図は一回のメモリの読み書きにどれだけ時間がかかるかを計測した結果である。MicroBlaze の方は 81 clock であるから、かかった時間を計算すると、 $81 \times 20 \text{ ns} = 1.62 \mu\text{s}$ である。DPM へのアクセスの方は $6 \mu\text{s}$ であるので、MicroBlaze からオンボード上のメモリにアクセスする方がかなり早かった。MicroBlaze の計測では、メモリの読み書きを一回行う関数自体が実行される時間を Timer を用いて測っているのので、実際の DDR SDRAM へのアクセス以外のオーバーヘッドもかなりのクロック数あると考えられる。そこで、オフセットが 0x1、0x2 のアドレスへの読み書きを追加し、DDR SDRAM へのアクセスに実際にかかっている時間を調べてみた。

5.33 図がその計測の結果である。93 cycles という結果は 5.31 図で用いたプログラムにオフセット 0x1 への読み書きを一つ加えたもので、105 cycles という結果はオフセット 0x2 への読み書きも加えたものである。5.31 図の結果も合わせて考えると、一回の DDR SDRAM の読み書きの操作に 12 clock かかるという結論になる。これは時間に換算すると 240 ns となるのでかなり高速にデータの読み書きが行えている。MicroBlaze を導入することで、オンボード上のメモリの読み書きが VME アクセスに比べ高速に行えることがわかった。

```

COM1:115200baud - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W)
-- Entering main() --
Starting MemoryTest for DDR_SDRAM:
Runing 32-bit Memory Test...
PASSED!
cycles = 93
-- Exiting main() --
-- Entering main() --
Starting MemoryTest for DDR_SDRAM:
Runing 32-bit Memory Test...
PASSED!
cycles = 105
-- Exiting main() --

```

図 5.33: DDR SDRAM へのアクセス時間の計測

5.4.5 MicroBlaze のリソースの使用量

XPS Synthesis Summary						
Report	Generated	Flip Flops Used	LUTs Used	BRAMS Used	Errors	
system	木 1 21 20:39:08 2010	4993	5446	11		0
microblaze_0_wrapper	木 1 21 20:37:53 2010	966	1456			0
xps_intc_0_wrapper	金 11 20 14:17:20 2009	120	88			0
proc_sys_reset_0_wrapper	金 11 20 14:17:01 2009	67	51			0
mdm_0_wrapper	金 11 20 14:16:54 2009	119	142			0
clock_generator_0_wrapper	金 11 20 14:16:40 2009	10	6			0
xps_timer_0_wrapper	金 11 20 14:16:30 2009	359	381			0
ethernet_mac_wrapper	金 11 20 14:16:04 2009	713	999	2		0
ddr_sdram_wrapper	金 11 20 14:15:15 2009	1495	897	5		0
flash_wrapper	金 11 20 14:10:55 2009	468	401			0
buttons_4bit_wrapper	金 11 20 14:10:26 2009	97	59			0
leds_8bit_wrapper	金 11 20 14:10:09 2009	125	71			0
rs232_dce_wrapper	金 11 20 14:09:52 2009	142	130			0
rs232_dte_wrapper	金 11 20 14:09:30 2009	142	130			0
lmb_bram_wrapper	金 11 20 14:09:08 2009			4		0
ilmb_cntlr_wrapper	金 11 20 14:09:01 2009	2	6			0
dlmb_cntlr_wrapper	金 11 20 14:08:55 2009	2	6			0
dlmb_wrapper	金 11 20 14:08:49 2009	1	1			0
ilmb_wrapper	金 11 20 14:08:43 2009	1	1			0
mb_plb_wrapper	金 11 20 14:08:36 2009	164	621			0

図 5.34: Synthesis Summary

5.34 図は 8KB の BRAM、DDR SDRAM コントローラ、Flash コントローラ、各種 IO を組み込んだ MicroBlaze システムを合成 (Synthesis) した後のリソースの使用量を表示したものである。Spartan 3E デバイスは一つの Slice の中に 2 つの Flip Flop と 2 つの LUT (Look Up Table) がある構造となっている。5.34 図は各 Peripheral と MicroBlaze にどれだけの Flip Flop と LUT が使われているかを表している。“system” という行が全体の使用量を表している。5.34 図を見ると、Flip Flop と LUT の消費量が多いのは、MicroBlaze と DDR SDRAM のコントローラであることがわかる。それぞれの名前には “wrapper” と付け加えられているが、個々の Peripheral や MicroBlaze の回路はブラックボックスとなっていて、それを包みこみ、ユーザーにそのモジュールの入出力を教えるのが wrapper ファイルである。Synthesis Summary で表示されているリソースの使用量は、この wrapper ファイルとその下位にあるブラックボックスモジュールを合成したときに使用するリソースの量である。

Device Utilization Summary					
Logic Utilization	Used	Available	Utilization	Notes	
Number of Slice Flip Flops	3,959	9,312	42%		
Number of 4 input LUTs	4,316	9,312	46%		
Number of occupied Slices	3,556	4,656	76%		
Number of Slices containing only related logic	3,556	3,556	100%		
Number of Slices containing unrelated logic	0	3,556	0%		
Total Number of 4 input LUTs	4,483	9,312	48%		
Number used as logic	3,766				
Number used as a route-thru	167				
Number used as 16x1 RAMs	4				
Number used for Dual Port RAMs	342				
Number used as Shift registers	204				
Number of bonded IOBs	113	232	48%		
IOB Flip Flops	98				
IOB Master Pads	1				
IOB Slave Pads	1				
Number of ODDR2s used	22				
Number of RAMB16s	11	20	55%		
Number of BUFGMUXs	5	24	20%		
Number of DCMs	2	4	50%		
Number of BSCANs	1	1	100%		
Number of MULT18X18SDOs	3	20	15%		
Average Fanout of Non-Clock Nets	3.29				

図 5.35: Device Utilization Summary

5.35 図は”配置配線後に”どれだけこのシステムが FPGA リソースを使用しているかを表している。配置配線後なので、各 Peripheral 間で重複していた回路の使用分が除かれて、Synthesis Summary のときよりも Flip Flop と LUT の使用量が少なくなっている。そのためこちらの方が実際のリソース使用量を表している。5.35 図をみると、Flip Flop の使用量は 3,959 個、LUT のトータルの使用量は 4,483 個である。

表 5.4: Spartan-6 FPGA Feature Summary (抜粋) [24]

Device	Logic Cells	Configurable Logic Blocks(CLBs)			Max User I/O
		Slices	Flip-Flops	LUTs	
XC6LX25	24,051	3,750	30,064	15,000	266
XC6LX45	43,661	6,822	54,576	27,288	358
XC6LX75	74,637	11,662	93,296	46,488	408
XC6LX100	101,261	15,822	126,576	63,288	480
XC6LX150	147,443	23,038	184,304	92,152	576

Spartan-6 Family の Slice は、Slice 一つの中に 4 つの LUT と 8 つの Flip Flop がある構造となっている。LUT の数は単純に Slice の数 × 4 をして求めた。

Spartan-6 デバイスでこのリソース使用量を評価してみる。表 5.4 は Spartan-6 Family の特徴をまとめた表である。中規模から大規模のデバイスを選んで表にした。Spartan-6 においても同様の Flip Flop と LUT を消費すると単純化して考えると、XC6L25 デバイスでは Flip Flop の使用量は 13 % 程度、LUT の使用量は約 30 % と十分リソースに余裕がある。一番は大きい XC6L150 デバイスでは Flip Flop の使用量は 2 % 程度、LUT の使用量は 5 % 程度とほとんど使用しない。

以上から Spartan-6 デバイスを用いれば、ユーザーロジックを加えても十分にリソースに余裕を

持った設計を行うことができ、さらに CPU コアを複数搭載させることも可能なことが分かる。

5.5 TGC 読み出し回路の研究開発のまとめ

今回は MicroBlaze からの IO の操作、Timer 機能を用いた処理時間の計測、オンボード上のメモリへのアクセスなど行い、性能の検証を行った。まず IO の操作であるが、C 言語プログラムによるボード上の IO の操作を行うことができた。これを応用すると、プログラムによってシグナルを検出し、その状態に応じて LED を点灯させるといったことが可能である。Timer 機能を用いた処理時間の計測では、簡単な四則演算と空の for loop の処理にかかるクロックサイクル数の計測を行った。デスクトップマシンなどに搭載されている CPU と比較すると処理時間の差は大きい、MicroBlaze はクロックが低速なことから RISC 型で CPU の命令セットも市販のものとは違うことなどを加味するとまずまずの結果であると言える。やはり、安価に CPU を搭載したシステムを設計できるのが MicroBlaze の魅力であると考えられる。オンボード上のメモリへのアクセスは MicroBlaze の方が VME アクセスを用いてメモリにアクセスするよりも早いという結果が得られた。これが今回一番重視していた項目で、メモリへのアクセス時間を短縮できることが MicroBlaze を搭載するメリットである。高速にオンボード上のメモリにアクセスできることから、メモリをエラーデータやその他の情報の記録領域として使用した、よりフレキシブルなシステムの構築ができる。

今回 Xilinx 社の MicroBlaze の基本的な機能、性能を評価してみて、新 ROD で十分採用できるということが確かめられた。まだ残されている課題としては、今回は検証できなかった Ethernet のインターフェースの使用と、OS が動くかという二点がある。これらの使用が可能になればより柔軟な運転が可能になる。また、MicroBlaze のリソースの使用量も知ることができ、Spartan-6 デバイスを採用する場合は、リソースに十分な余裕を持たせた設計も行えることがわかった。

現在のところ新 ROD のデザインはある程度固まってきつつある。新 ROD の開発の次の段階としては、プロトタイプを試作があるが、使用を検討している Spartan 6 を搭載した試作機を作り、より実践的な状況で MicroBlaze が動作するかを確認しなければいけない。さらに MicroBlaze に持たせる機能、どのようなプログラムを実行させるのかなどの議論がこれから必要になると考えられる。

第6章 まとめと今後について

4章で述べたように、私はDelay Moduleの開発を通して、Level-1 トリガーを担当する TGC システムで重要なパラメータとなるクロックの位相調整に携わり、オフラインソフトウェアの開発ではデコードソフトウェアのデバッグとエレクトロニクスの不具合の発見が可能な TrigMuonCoinHierarchy パッケージの開発に携わった。これらのハードとソフトの面で TGC システムのコミッショニングを進め、2008 年の時点と比べてより完成度の高いシステムを構築することに貢献できたと実感している。

2009 年 11 月に LHC は再稼働を始め、12 月には世界最高エネルギーでの衝突を迎えた。TGC システムもビーム周回に合わせて、バンチ衝突のタイミングを調整してきた。あとは TGC システム内のクロックと衝突の間の位相調整 (Clock Phase Scan) を行うのみである。Clock Phase Scan に必要なものは全てそろっているのも、あとは来年の早い時期に TGC 単独でのデータ取得の機会をいただき、実行に移すことを計画している。TGC のタイミング調整は最終段階にあると言える。

オフラインソフトウェアの開発を通して、デコーダーのデバッグを行うことができた。また、Hit データ (TgcPrepData) と Coincidence データ (TgcCoinData) の対応関係からは、SL-HiPt Strip、HiPt Strip-TGC1 Hit Strip 間で一様に対応がとれていないということがわかった。この点についてはさらなる追及が必要である。TrigMuonCoinHierarchy は実データ、モンテカルロデータ両方で使用が可能なことが確かめられていて、どちらにおいても全体的に対応関係がとれる確率は 99 % 以上ある。

また、我々は同時に LHC のアップグレードに向けた TGC 読み出し回路の研究開発を行ってきた。5 章では新しい ROD に搭載させることを考えている CPU コア (MicroBlaze) の評価について述べた。各機能の動作と Timer 機能を用いた実行時間の計測を行い、MicroBlaze の性能について知ることができた。何よりも、各機能の検証を通して、CPU コアの利便性を実感することができた。また MicroBlaze を搭載したシステムがどれだけのリソースを使用するかについても検証し、Spartan-6 デバイスでは十分に余裕を持った設計が可能なが確かめられた。あとはこの技術をどのようにして実際の ROD というエレクトロニクスに落とし込んでいくかというのが残された課題である。

付 録 A Delay Module の回路図

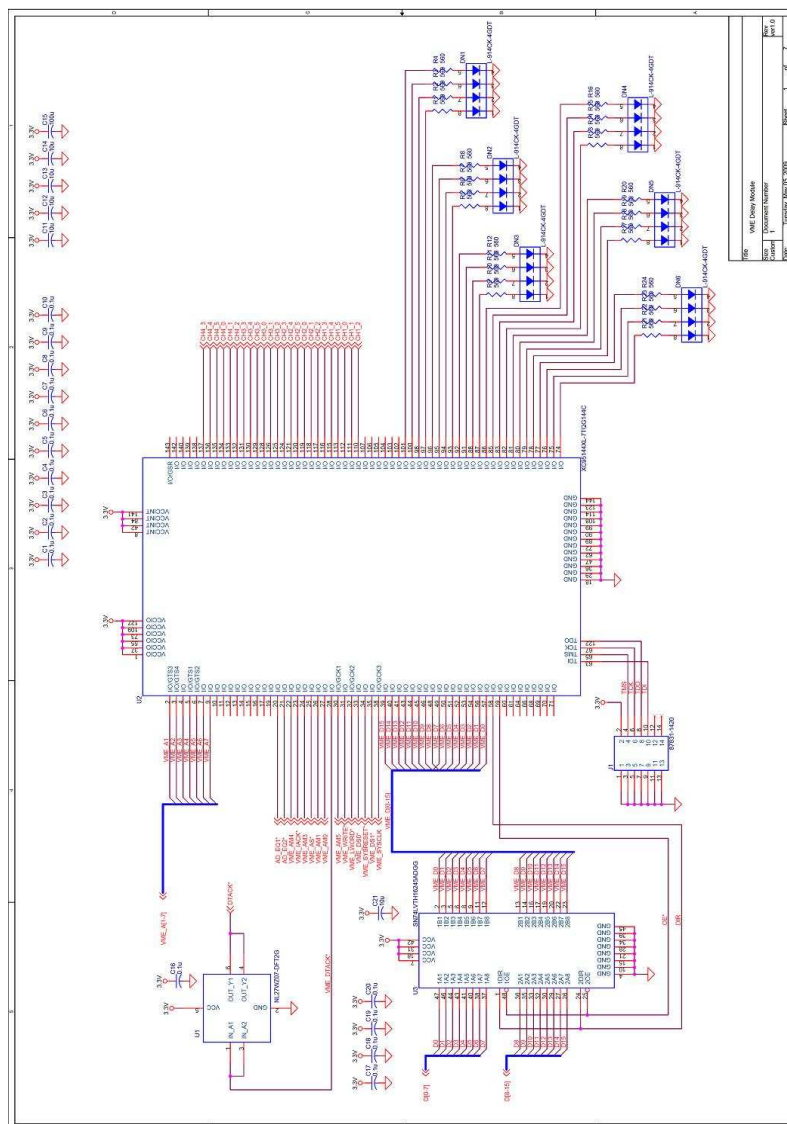


図 A.1: Delay Module の回路図 (1)

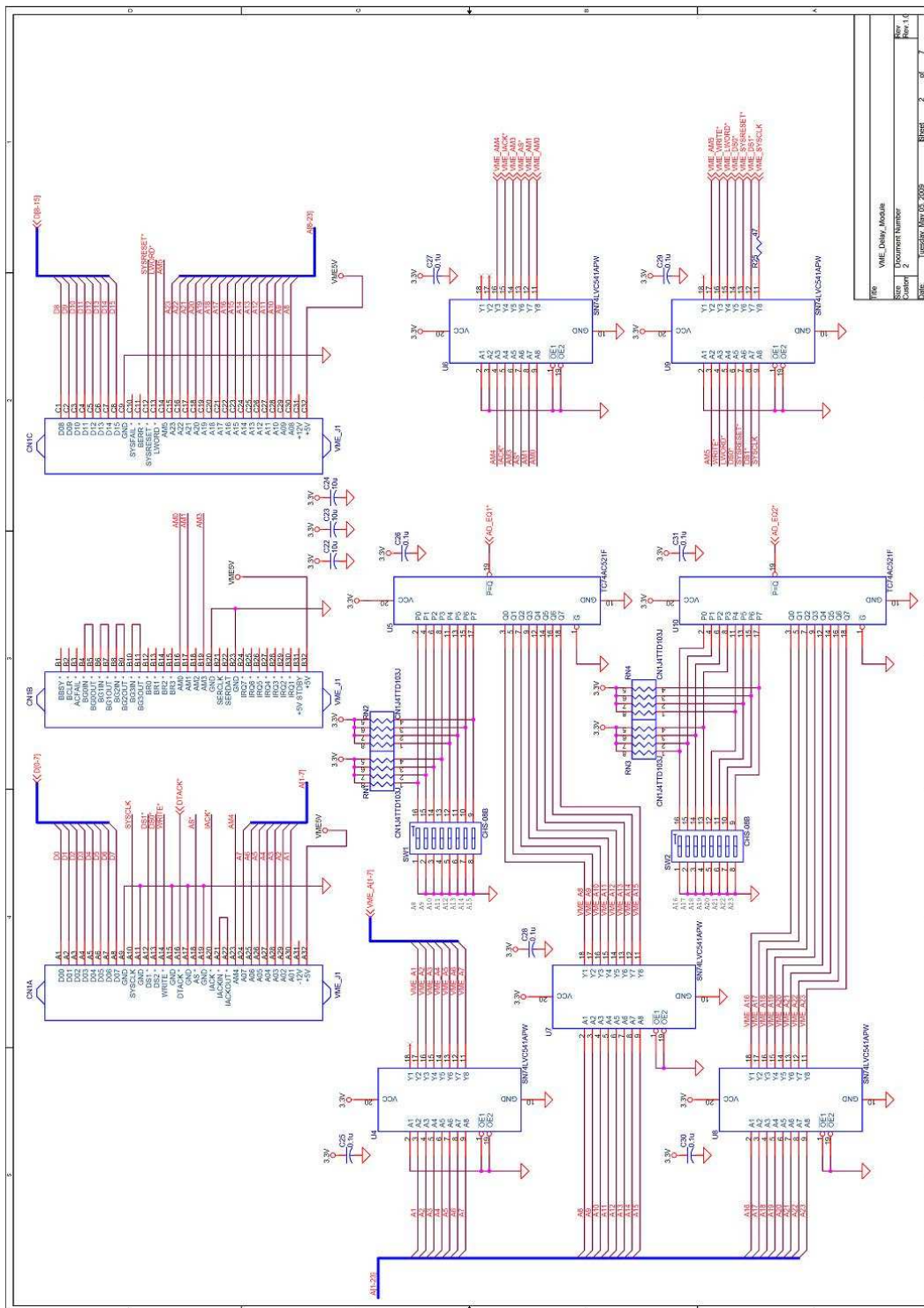


図 A.2: Delay Module の回路図 (2)

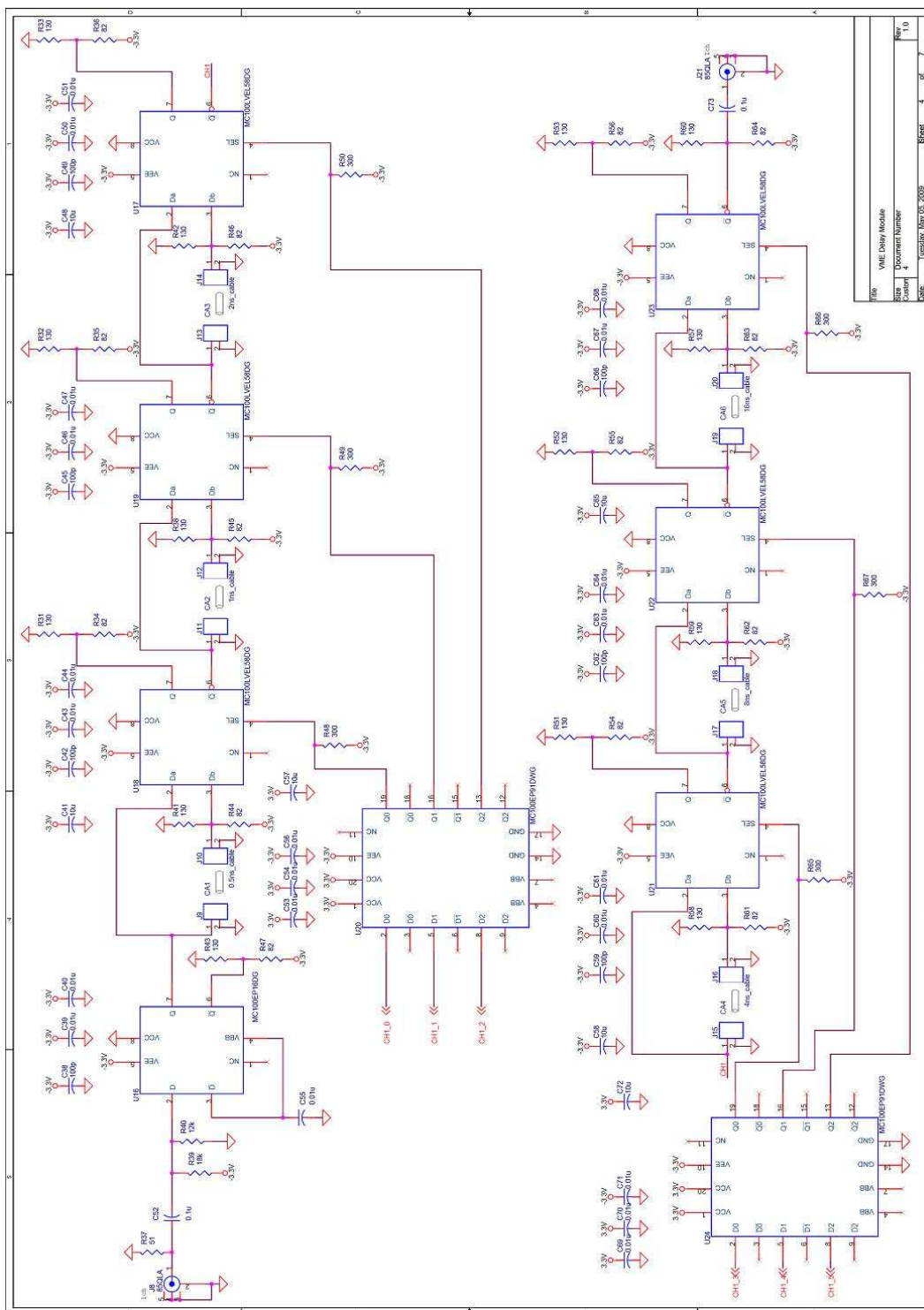


図 A.4: Delay Module の回路図 (4)

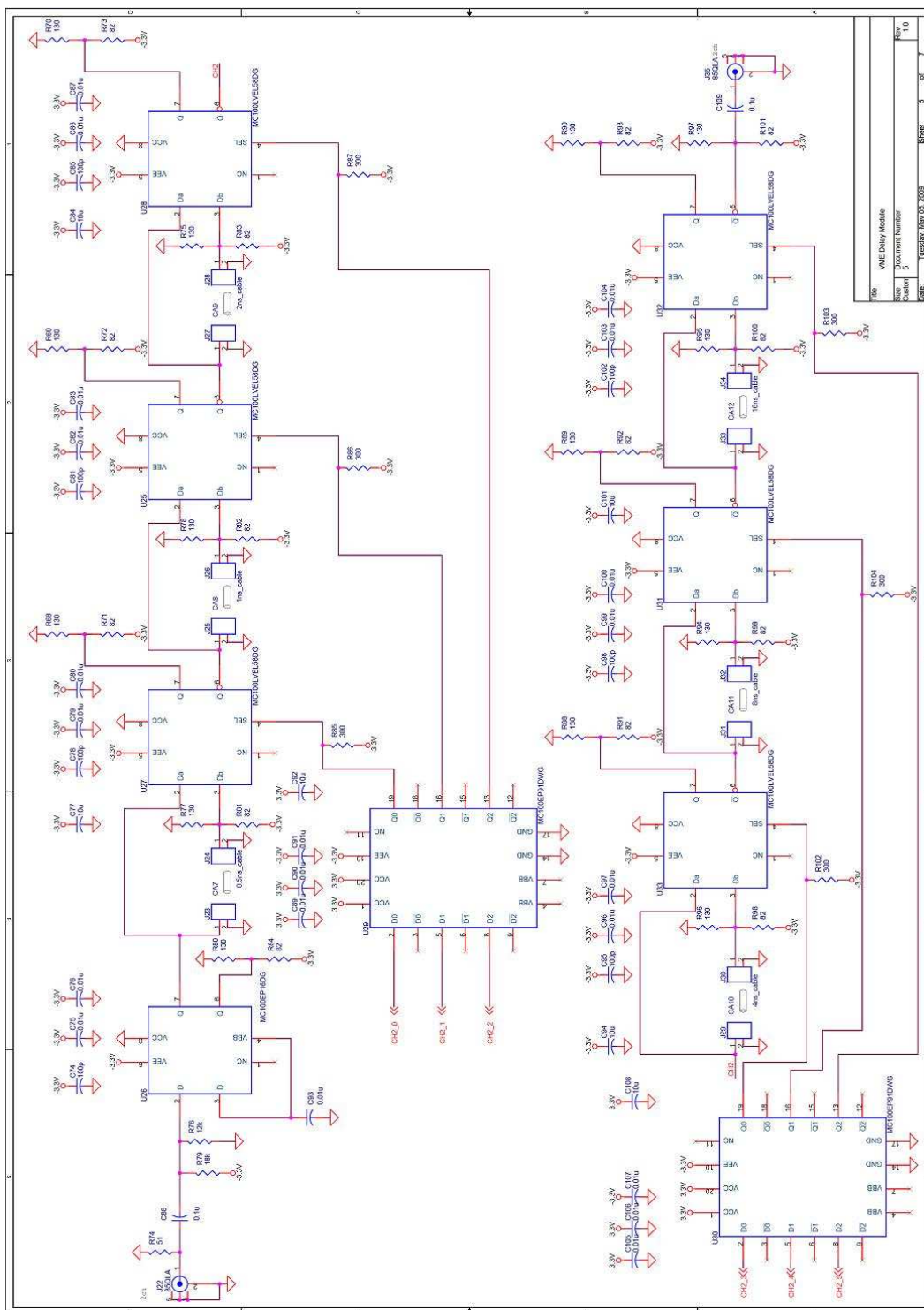


図 A.5: Delay Module の回路図 (5)

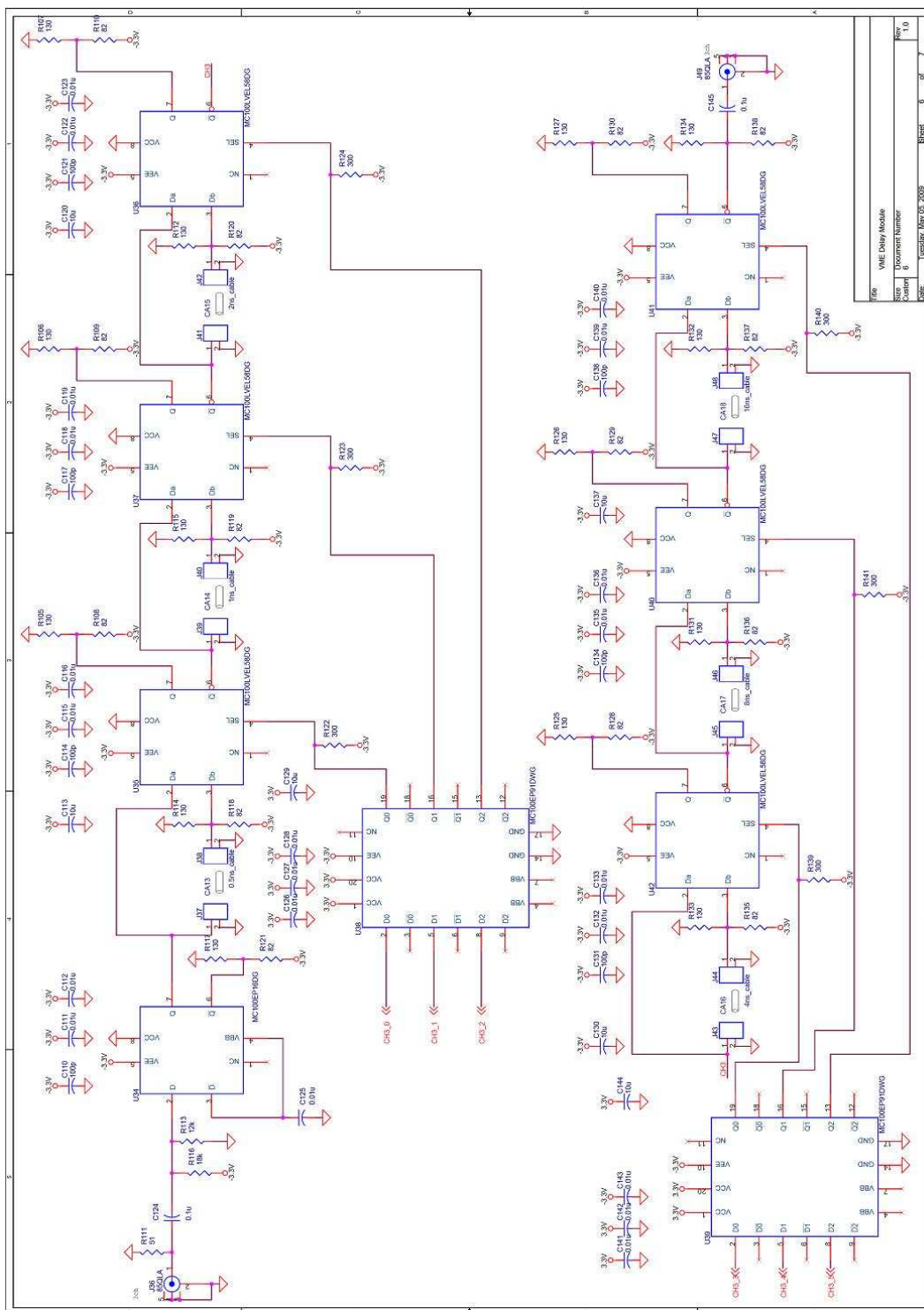


図 A.6: Delay Module の回路図 (6)

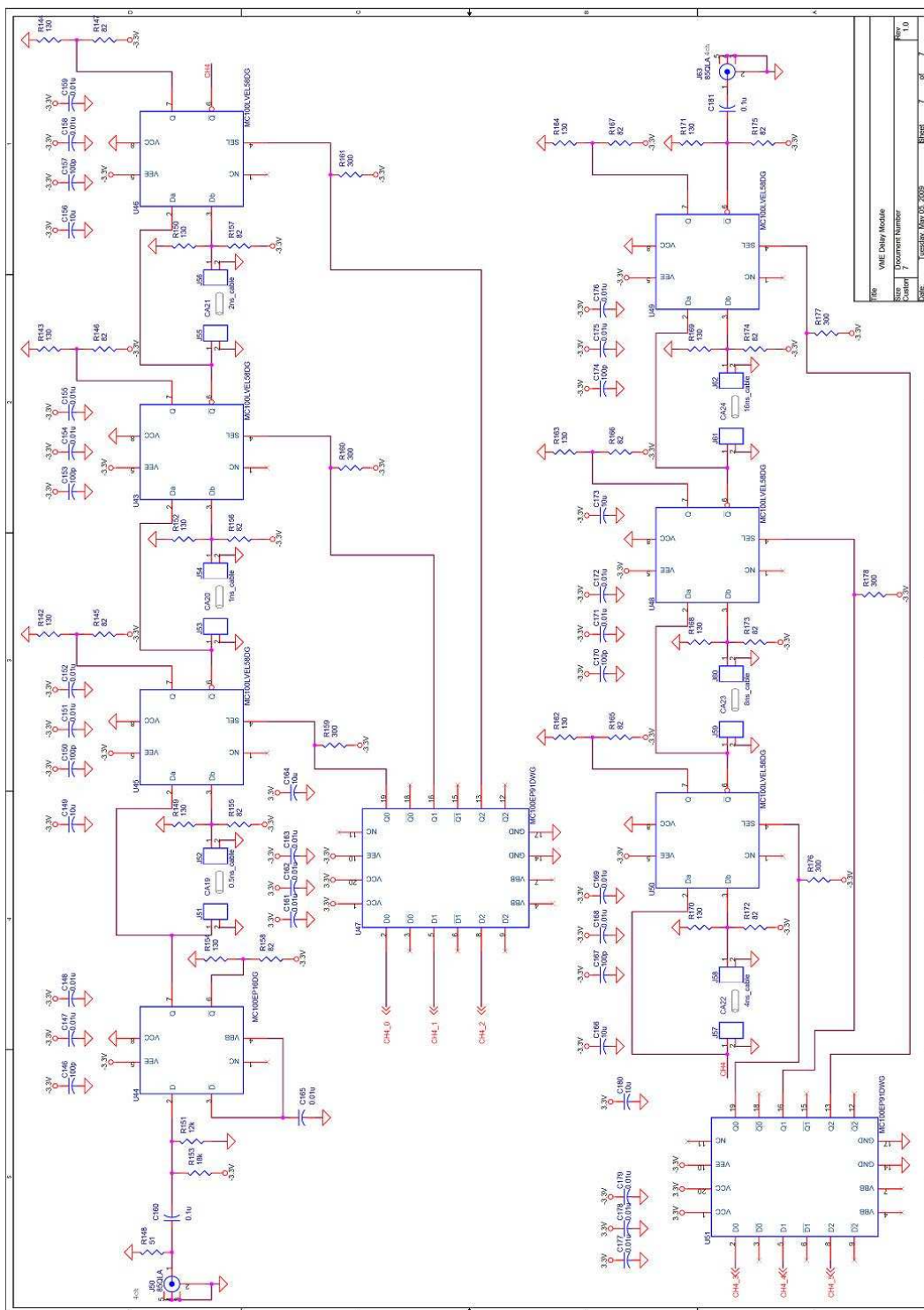


図 A.7: Delay Module の回路図 (7)

付 録 B PS Board と SSW の接続

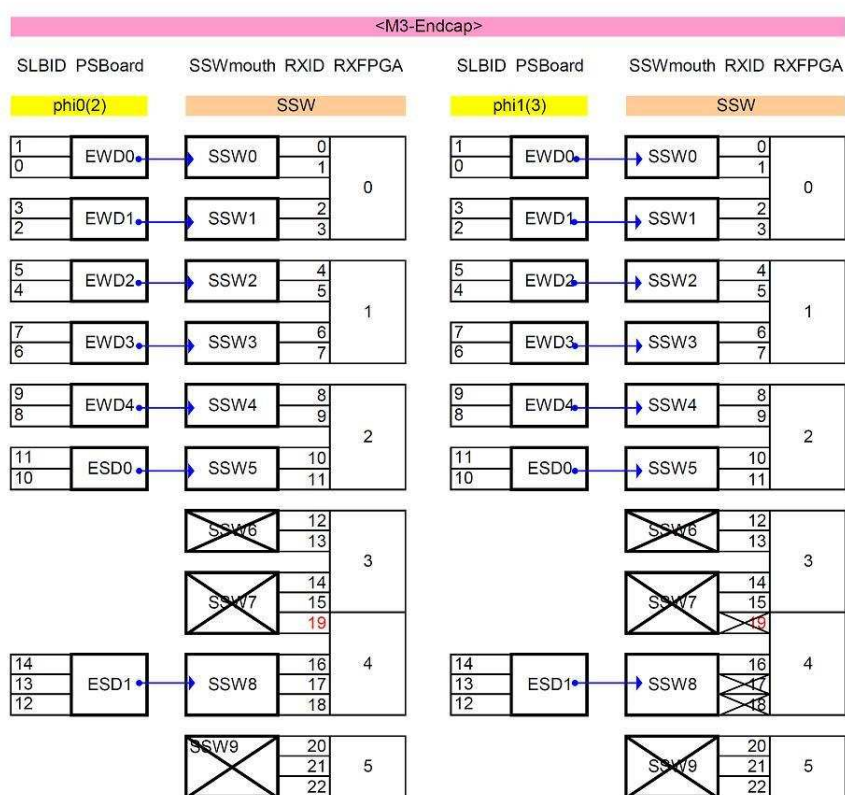


図 B.1: Doublet PSB と SSW の接続 (1) [14]

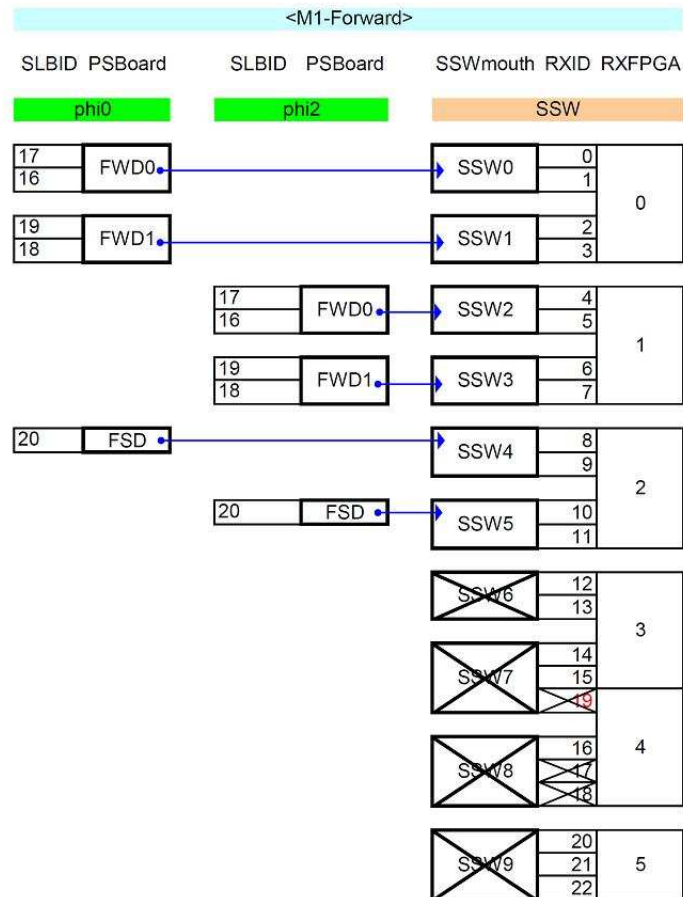


図 B.2: Doublet PSB と SSW の接続 (2) [14]

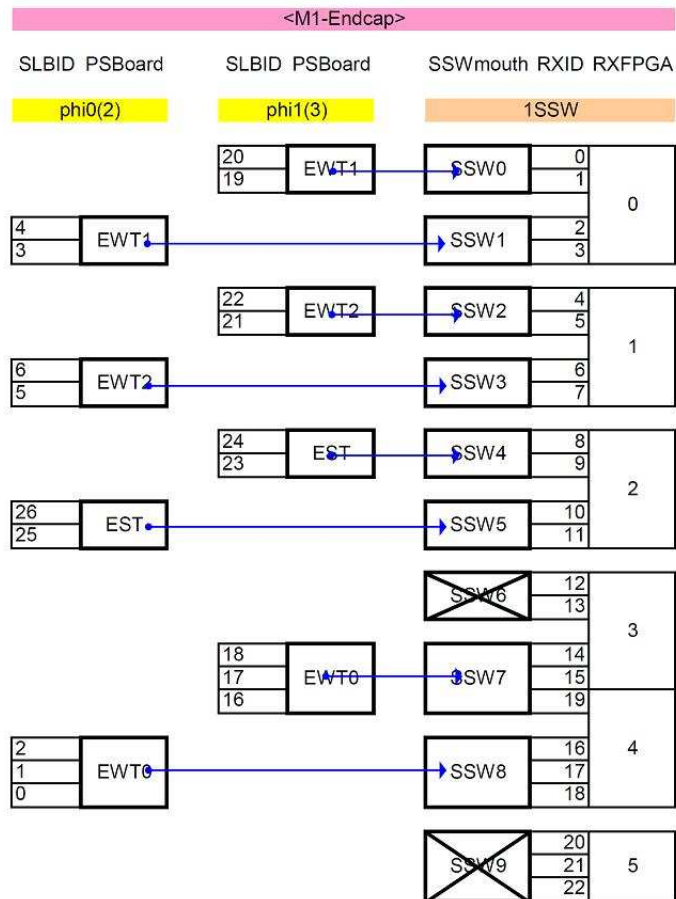


図 B.3: Triplet PSB と SSW の接続 (1) [14]

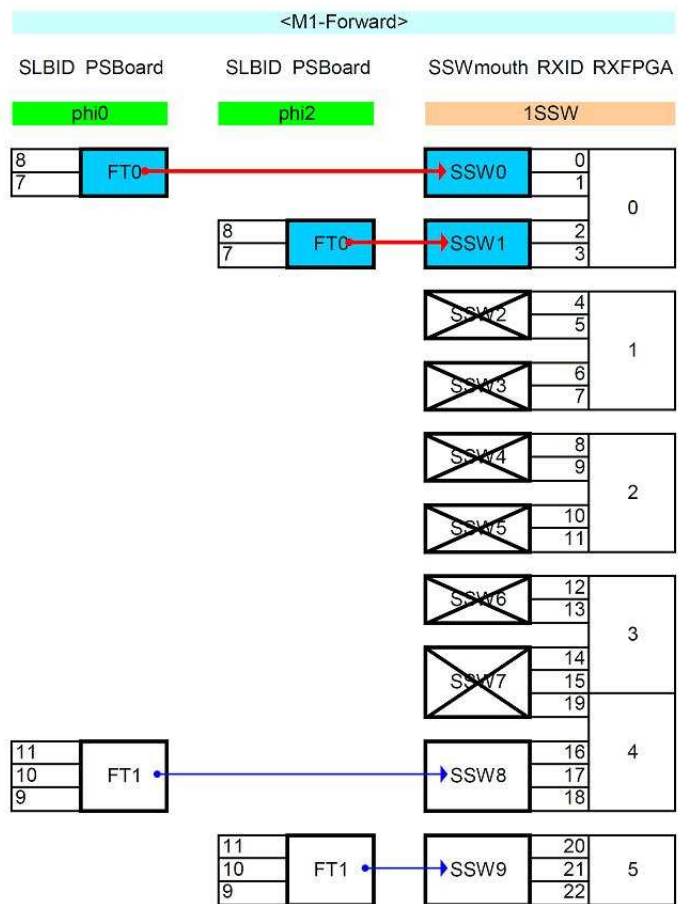


図 B.4: Triplet PSB と SSW の接続 (2) [14]

付 録 C SLB データフォーマット

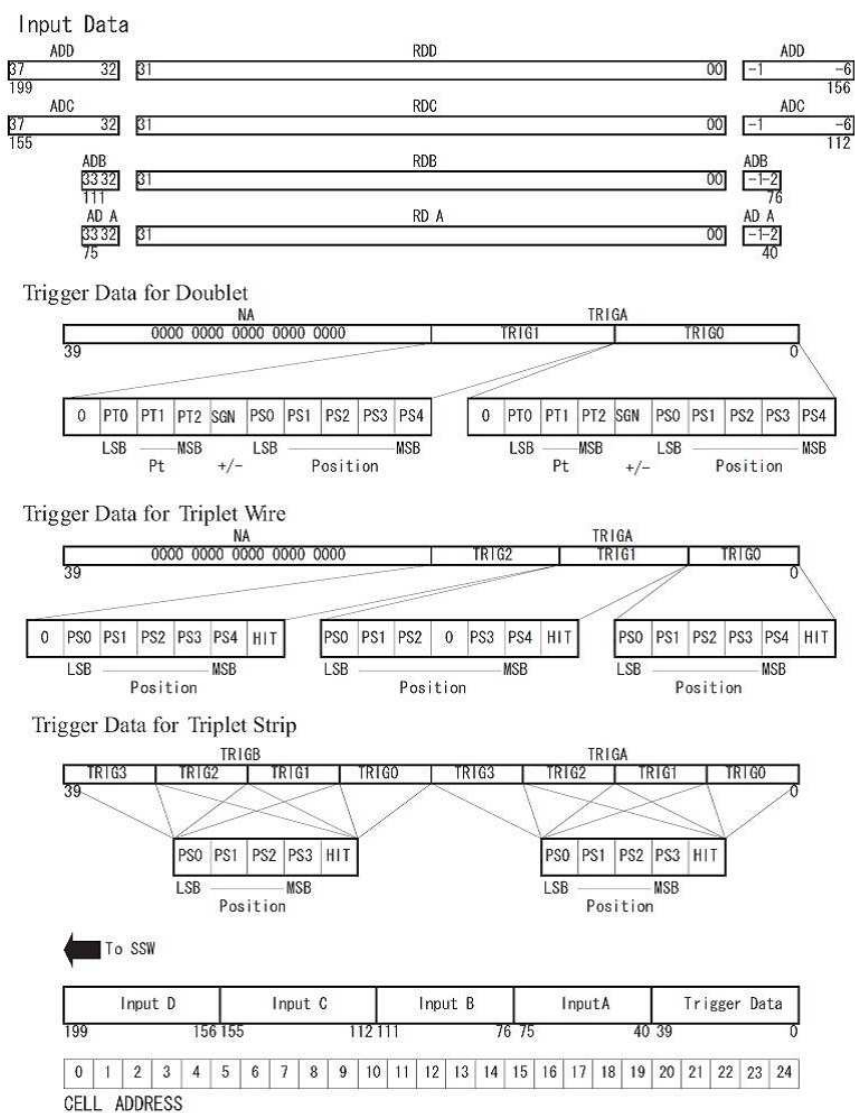


図 C.1: SLB のデータフォーマット [30]

付 録D bitpos

Triplet Wire

	adjacent										subMatrix=0										subMatrix=1										subMatrix=2										adjacent									
Layer 1 (C-INPUT)	158	157	38	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199						
Layer 2 (B-INPUT)	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155						
Layer 3 (A-INPUT)	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85				

	adjacent										subMatrix=0										subMatrix=1										subMatrix=2										adjacent									
Layer 1 (C-INPUT)	-6	-3	0	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63	66	69	72	75	78	81	84	87	90	93	96	99	102	105	108	111	114	117	120	123	126	129				
Layer 2 (B-INPUT)	-5	-2	1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46	49	52	55	58	61	64	67	70	73	76	79	82	85	88	91	94	97	100	103	106	109	112	115	118	121	124	127	130				
Layer 3 (A-INPUT)	-4	-1	2	5	8	11	14	17	20	23	26	29	32	35	38	41	44	47	50	53	56	59	62	65	68	71	74	77	80	83	86	89	92	95	98	101	104	107	110	113	116	119	122	125	128	131				

	adjacent										subMatrix=0										subMatrix=1										subMatrix=2										adjacent									
Layer 1 (C-INPUT)	-6	-3	0	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63	66	69	72	75	78	81	84	87	90	93	96	99	102	105	108	111	114	117	120	123	126	129				
Layer 2 (B-INPUT)	-5	-2	1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46	49	52	55	58	61	64	67	70	73	76	79	82	85	88	91	94	97	100	103	106	109	112	115	118	121	124	127	130				
Layer 3 (A-INPUT)	-4	-1	2	5	8	11	14	17	20	23	26	29	32	35	38	41	44	47	50	53	56	59	62	65	68	71	74	77	80	83	86	89	92	95	98	101	104	107	110	113	116	119	122	125	128	131				

	adjacent										subMatrix=0										subMatrix=1										subMatrix=2										adjacent									
Layer 1 (C-INPUT)	-6	-3	0	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63	66	69	72	75	78	81	84	87	90	93	96	99	102	105	108	111	114	117	120	123	126	129				
Layer 2 (B-INPUT)	-5	-2	1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46	49	52	55	58	61	64	67	70	73	76	79	82	85	88	91	94	97	100	103	106	109	112	115	118	121	124	127	130				
Layer 3 (A-INPUT)	-4	-1	2	5	8	11	14	17	20	23	26	29	32	35	38	41	44	47	50	53	56	59	62	65	68	71	74	77	80	83	86	89	92	95	98	101	104	107	110	113	116	119	122	125	128	131				

図 D.1: Triplet Wire の bitpos

Endcap Strip Triplet

FORWARD chambers

	subMatrix = 0																subMatrix = 1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
T7 layer2 (D-input)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				

BACKWARD chambers

	subMatrix = 0																subMatrix = 1															
T3 layer1 (D-input)	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	191	192	193	
T3 layer2 (C-input)	116	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149
T6 layer1 (B-input)	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109
T6 layer2 (A-input)	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	72	73	

slbchannel

0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47	49	51	53	55	57	59	61	63

slbchannel in TGC[0]

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

slbchannel in TGC[1]

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

No Entries are expected

slbchannel

0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47	49	51	53	55	57	59	61	63

slbchannel in TGC3

0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47	49	51	53	55	57	59	61	63

⊠ D.2: Triplet Strip の bitpos

付 録 E 放射線環境下に耐えうる VME コントローラと PCI インタフェースの開発研究

E.1 序論



図 E.1: CCI と HSC

ATLAS 前後方ミューオントリガーシステム (TGC システム) では放射線環境下の実験ホールに VME クレートを配置し、放射線耐性を持ったモジュールがインストールされている。クレート中のモジュールとさらにフロントエンドに位置するモジュールのコンフィギュレーションを行うために、光ファイバーを用いた遠隔操作システムが構築されている。その遠隔操作システムは、放射線耐性を持ち、実験ホールのクレートにインストールされる VME マスタモジュール (HSC、High- p_T Star switch Control board) と USA15 と呼ばれるカウンティングホール中のクレートにインストールされる VME スレーブモジュール (CCI, Control Configure Interface board) と、両者を接続する G-Link 規格のオプティカルファイバーからなる。E.1 図が CCI と HSC、また両者の接続に用いる G-Link ケーブルである。CCI がインストールされるクレートのマスタは SBC (Single Board Computer) である。SBC は Ethernet インターフェースを搭載し、LAN (Local Area Network) ケーブルでネットワークに接続しているので、ユーザーは外部の PC から SBC にログインし、CCI 中のレジスタの値を読み書きすることで、HSC と通信が可能になっている。E.2 図は ATLAS 前後方ミューオントリガーのコントロールシステムの全体像表している。

我々はこの ATLAS 前後方ミューオントリガーシステムで使用している遠隔コントロールシステム

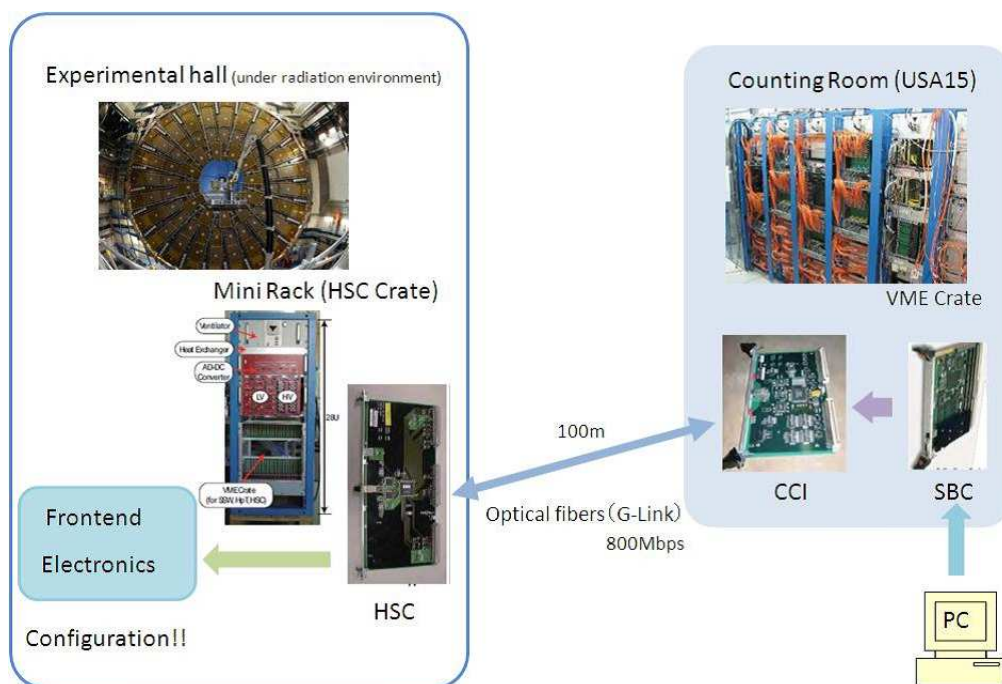


図 E.2: CCI と HSC を用いた ATLAS 前後方ミューオントリガーのコントロールシステム

を基にした新たな遠隔コントロールシステムの開発を行っている。新たなコントロールシステムは 9U サイズの HSC を 6U サイズにコンパクト化した放射線耐性を持つ VME マスタモジュール (RTC, Radiation Tolerant Controller) と CCI の機能を備えた PCI¹カード (PCI-CCI) から構成され、両者の通信には G-Link を使用する。大型実験以外では 6U サイズの VME モジュールが一般的であり、SBC は主に欧州で使用されることが多く、国内ではほとんど使用されることはないので、HSC のダウンサイジングを図り、デスクトップコンピュータから直接コントロールができるように CCI は PCI カードに変更することに決めた。これの使用の変更によりより汎用性を持った放射線耐性のあるコントロールシステムにすることができる。以下では、ATLAS 前後方ミューオントリガーシステムの各構成要素の説明を行った後に、この新規コントロールシステムの開発計画と現在の状況についてまとめる。

E.2 ATLAS 前後方ミューオントリガーのコントロールシステムの構成要素

E.2.1 HSC

HSC は 9U の VME マスタモジュールである。フロントパネルには G-Link のコネクタと ATLAS 実験においてタイミング関連のシグナルを分配する TTC(Timing Trigger and Control) システムからのクロックの入力を搭載する。また、バックプレーンの J3 コネクタは JTAG に使用する。同じレートにインストールされる SSW(Start Switch) というスレーブモジュールを介して JTAG 通信を行うことで、さらにフロントエンドにあるエレクトロニクスをコンフィグできるようになっている。RTC ではこの JTAG の機能を削除し、6U にダウンサイジングする。

HSC の放射線耐性

HSC では耐放射線性を持たせるために Anti-Fuse FPGA が用いられている。Anti-Fuse FPGA は再書き込みが不可能な FPGA で、Fuse 方式²を採用し、一回だけ書き込みが可能である。FPGA の Configuration memory や Flash memory に似た構造の CPLD は放射線の影響を受けやすい。ASIC(Application Specific Integrated Circuit) も放射線環境下で使われるが、製作に時間とお金がかかり過ぎてしまう。Anti-Fuse FPGA を用いることで、製作期間の短縮とコストの削減が図れる。

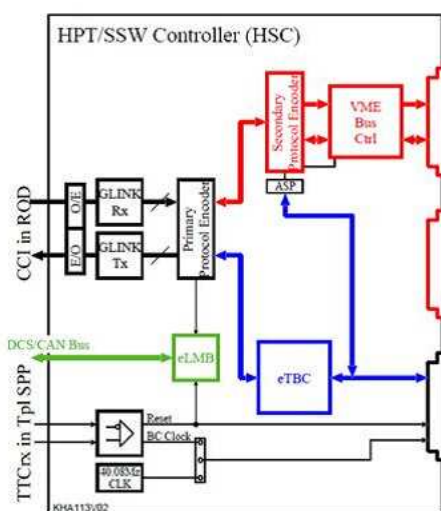


図 E.3: HSC のブロックダイアグラム

¹Peripheral Component Interconnect bus。プロセッサと周辺機器間の通信を行うためのバスアーキテクチャの一つ。現在でもデスクトップ型のワークステーションなどに搭載されている。PCI の後継規格として PCI Express がある。PCI プロトコルについては本文中で言及する。

²高電圧をかけ、ヒューズを導通させることで回路を構成する。

CPLD や FPGA などの半導体デバイスへの放射線の影響としては蓄積効果 (TID, Total Ionizing Doze) と確率的現象 (SEE, Single Event Effect) が知られている。TID は放射線によって生成される、電子・ホール対による蓄積ダメージの結果として、リーク電流の増加やトランジスタの動作速度の低下がもたらされる現象である。消費電流の増加が起こり、最終的にそのデバイスは動作しなくなる。SEE は荷電粒子が局所的に発生させる電子・ホールペアによって引き起こされる確率的な現象である。SEE はさらに SEU(Single Event Upset) と SEL(Single Event Latch-up) に区別される。SEU はレジスタやメモリのビットの反転が起こる現象で、SEL はラッチアップの状態 (常に電流が流れる状態) になってしまう現象である。

HSC に使われている Actel 社製の Anti-Fuse FPGA と G-Link Tx/Rx chip、OE/EO converter に対して放射線耐性を検査する試験が 2004 年に東北大の CYRIC にて行われている。このときは各デバイスに 70 MeV の陽子ビームを照射して検査が行われた。Anti-Fuse FPGA の放射線耐性の結果としては TID は 100 krad まで耐えられることがわかり、SEE は観測されず、SEU 断面積として $1.5 \times 10^{15} \text{ cm}^2/\text{bit}$ 以下という値を得ることができた。また、G-Link Tx/Rx chip、OE/EO Converter も放射線環境下で使用できることが確認された。

このように Anti-Fuse FPGA は放射線耐性のあるデバイスであるが、そのロジックの内部にはレジスタ (Flip Flop) などの記憶素子が作られるため、その Flip Flop が SEU の影響を受ける可能性がある。その対策としては Flip Flop を 3 個使用した多数決回路が用いられる。これによって、リソースは 3 倍多く使用するものの、一つの Flip Flop が影響を受けたとしても正しい値を保持することが可能となる。Actel のコンパイラには自動で多数決論理回路を作る機能があり、どのレジスタに使用するかを指定するだけで、自動生成してくれる。

以上のようにして、HSC は放射線耐性に優れた VME マスタモジュールとなっている。

E.2.2 CCI

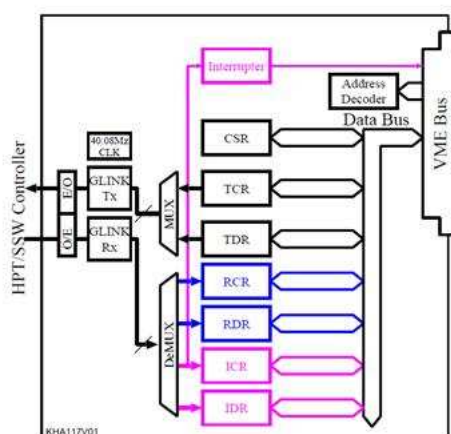


図 E.4: CCI のブロックダイアグラム

CCI はカウンティングルームにインストールされる VME スレーブモジュールである。E.4 図は CCI のブロックダイアグラムである。6U のサイズで、フロントパネルには G-Link のコネクタが実装されている。CCI は VME の割り込み (Interrupt) に対応していて、HSC からの割り込み信号を受け取り、割り込み要求をする。CCI には 7 つのレジスタが搭載されていて、そのレジスタの読み書きを通して HSC との通信を行う。

E.2.3 G-Link

G-Link はオプティカルファイバーを使用したシリアル通信規格である。通信速度は 200 Mbit/s から 1,120 Mbit/s の間で設定可能で、それに応じて Baud rate は 260 MBaud から 1400 MBaud の値をとる。基本的に 16bit のデータ幅でコントロールワードもしくはデータワードを送信する。32bit 通信のときは 17 番目の bit がワードの偶奇を表すために付加される。4bit のエンコーディングビットを付加して合計 20bit のデータを転送する。Agilent HDMP-1032/1034 というチップセットがパラレル-シリアル変換を行う。

E.3 開発計画

E.3.1 RTC

RTC は HSC から JTAG の機能をはずし、6U サイズのモジュールへとダウンサイジングを図ることで、汎用性を持たせる。また HSC 製作のときに試験された Anti-Fuse FPGA、G-Link chip set、OE/EO Converter をそのまま使用することで、HSC と変わらない放射線耐性を持たせる。

E.3.2 PCI-CCI

CCI は VME バスから PCI バスに準拠した PCI カードへと変更を行う。PCI カードにすることで、SBC を介さずに直接 HSC との通信が可能となる。PCI カードにするにあたり、VHDL で書かれているロジックを Verilog に変更する。また PCI-CCI が使われるマシンの環境としては、高エネルギーの分野で使われる機会が多い Linux OS の環境を想定して、ドライバの開発を行う。

PCI バス

PCI バスは 32bit または 64bit のバス幅を持ち、33 MHz または 64 MHz のバスクロックで動作する。またほとんどの信号線が PCI バスクロックに同期して動作する (VME バスは非同期である)。アドレスとデータを同じ線で時分割 (multiplex) してデータ転送を行う (VME では別々の線になっている)。また PCI カードには PC からアクセスできる領域として、最大 4GB のアドレス空間と、64kB の I/O 空間とコンフィギュレーション空間がある。コンフィギュレーション空間はディップスイッチをレジスタにしたようなもので、これによって Plug & Play が可能になっている。

PCIバスに準拠したモジュール開発のために、Xilinx社製のIP(Intellectual Property)コアを使用した。IPコアはユーザーが使いやすい形でPCIバスの信号を提供し、開発を容易にする。IPコアの評価のためにPCI評価ボードを購入し、IPコアの評価を行った。E.5図がそのPCI評価ボードである。



図 E.5: PCI 評価ボード

また、PCIカードのドライバも開発しなければならないが、ドライバの開発時間の短縮のために、Jungo社のWinDriverというPCIドライバ自動作成ソフトを購入した。WinDriverはドライバソフトを自動生成してくれるだけでなく、簡単なレジスタの読み書きもWinDriver上で行うことができる。E.6図がWinDriverのスクリーンショットである。

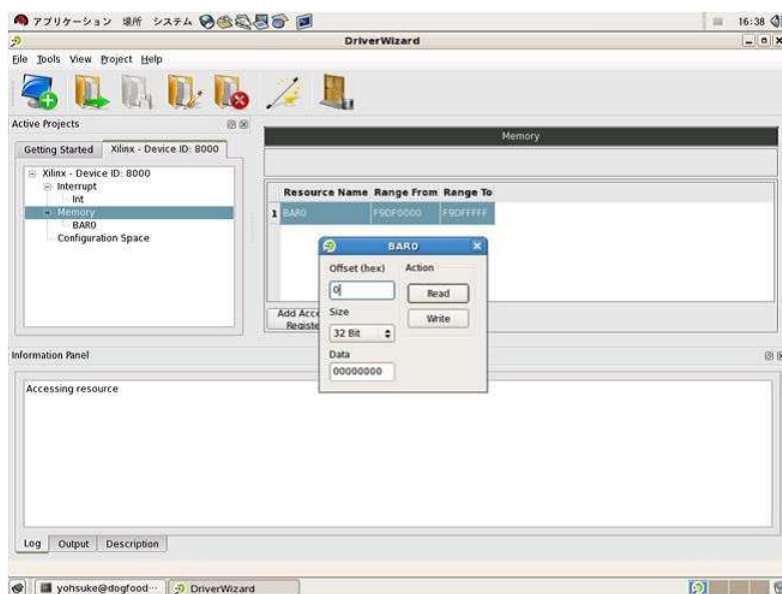


図 E.6: WinDriver のスクリーンショット

E.4 開発状況



図 E.7: PCI-CCI の写真

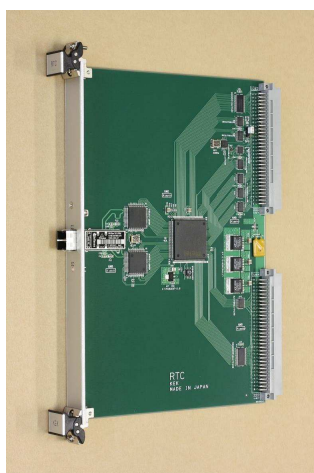


図 E.8: RTC の写真

現在、RTC、PCI-CCI の両者ともに回路図を作り、製作をするところまで完成している。RTC、PCI-CCI 共に 2 枚ずつ製作した。E.7 図が PCI-CCI の写真、E.8 図が RTC の写真である。

Actel 社の統合開発環境ソフトが高額だったということもあり、RTC には HSC と同様のロジックを焼いた Anti-Fuse FPGA を搭載した。PCI-CCI のロジックは現在開発中である。両者とも動作試験はまだ行っていないので、PCI-CCI のロジック開発が終了した後に、動作試験を行うのが課題として残されている。

参考文献

- [1] Atlas. "Approved Trigger Plots". <https://twiki.cern.ch/twiki/bin/view/Atlas/ApprovedPlotsTrigger>.
- [2] CERN. "Timing, Trigger and Control (TTC) Systems for the LHC". <http://ttc.web.cern.ch/TTC/>.
- [3] CERN. "CERN Document Server, CERN Photo Lab". <http://cdsweb.cern.ch/>.
- [4] CERN. "CERN Press Release Incident in LHC sector 3-4". <http://press.web.cern.ch/press/PressReleases/Releases2008/PR09.08E.html>.
- [5] CERN. "Online Software Training Version 5.0".
- [6] ATLAS Muon Collaboration. ATLAS muon spectrometer: Technical Design Report. Technical Design Report ATLAS. CERN, Geneva, 1997. distribution.
- [7] The CDF Collaboration, the D0 Collaboration, the Tevatron New Physics, and Higgs Working Group. Combined CDF and D0 Upper Limits on Standard Model Higgs- Boson Production with 2.1 - 5.4 fb⁻¹ of Data. 2009.
- [8] Peter Cwetanski, Risto Orava, and Fido Dittus. Straw Performance Studies and Quality Assurance for the ATLAS Transition Radiation Tracker. [oai:cds.cern.ch:962570](http://cds.cern.ch/record/962570). PhD thesis, Helsinki Univ., Helsinki, 2006. Presented on 22 Jun 2006.
- [9] Aad et al. The atlas experiment at the cern large hadron collider. J. Instrum., Vol. 3, p. S08003, 2008.
- [10] Suzuki et al. Precise timing adjustment for the atlas level1 endcap muon trigger system. Technical Report ATL-DAQ-PROC-2009-035. ATL-COM-DAQ-2009-125, CERN, Geneva, Nov 2009.
- [11] ATLAS Experiment. "ATLAS Photos". <http://www.atlas.ch/photos/index.html>.
- [12] CMS Experiment. "The Compact Muon Solenoid Experiment". <http://cms.web.cern.ch/cms/index.html>.
- [13] ATLAS Level-1 Trigger Group. ATLAS level-1 trigger: Technical Design Report. Technical Design Report ATLAS. CERN, Geneva, 1998.

- [14] ATLAS TGC group. "TGC Document". <https://twiki.cern.ch/twiki/pub/Main/TgcDocument>.
- [15] K. クラインクネヒト. 粒子加速器 -放射線計測の基礎と応用-. 培風館, 1987.
- [16] Michelangelo L Mangano. The super-lhc. oai:cds.cern.ch:1210230. Technical Report arXiv:0910.0030. CERN-PH-TH-2009-164, Oct 2009. Comments: To appear in Contemporary Physics.
- [17] Hiroshi Nomoto. "Star Switch Spec". https://twiki.cern.ch/twiki/pub/Main/TgcDocument/SSW_spec_v20090911.pdf.
- [18] On Semiconductor. "MC100LVEL58 3.3V ECL 2:1 Multiplexer".
- [19] On Semiconductor. "MC10EP16, MC100EP16 3.3V/5V ECL Differential Receiver/Driver".
- [20] Michael Spira. Qcd effects in higgs physics. Fortschr. Phys., Vol. 46, No. hep-ph/9705337. CERN-TH-97-068, pp. 203–284. 100 p, May 1997.
- [21] Oda Susumu. "TrigMuonCoinHierarchy Doxygen". <http://atlas-computing.web.cern.ch/atlas-computing/links/nightlyDocDirectory/TrigMuonCoinHierarchy/html/index.html>.
- [22] VITA. "American National Standard for VME64".
- [23] W.R.Leo. Techniques for Nuclear and Particle Physics Experiments 2nd Edition. Springer-Verlag, 1994.
- [24] Xilinx. "Spartan-6 Familiy Overview".
- [25] Frank Zimmermann. "LHC Upgrade". LARP CM12, 9 April 2009.
- [26] 上田郁夫. "ATLAS コンピューティングとグリッド" 日本物理学会, 2008.
- [27] 神前純一. "Higgs 入門@LHC", 2005.
- [28] 浅井祥仁. 高エネルギーニュース vol.24 no.4 lhc で期待されている物理, 2006.
- [29] 鈴木友. Atlas muon trigger system の commissioning~ タイミング調整とその検証. Master's thesis, 総研大, 2009.
- [30] 桑原隆志. Atlas 前後方ミューオントリガーシステムの構築. Master's thesis, 東京大学, 2009.
- [31] 平山翔. Atlas 実験前後方レベル 1 ミューオントリガーシステムの構築と検証. Master's thesis, 東京大学, 2009.

謝辞

この二年間の研究生活におきまして、本研究を行う貴重な機会と惜しみのない指導と助言を与えてくださった指導教官の坂本宏^a教授には深く感謝致します。本研究の Delay Module 開発においては佐々木修氏^b、池野正弘氏^b、(有) ジー・エヌ・ディーの皆様にご指導と多くの助言をいただき、完成させることができました。ありがとうございます。また、川本辰男氏^a、石野雅也氏^b、織田勸氏^aには CERN 滞在中の生活と研究の双方においてにご指導していただき、またときにはご迷惑をおかけすることもあり、深く感謝しています。

TGC 日本グループの岩崎博行氏^b、蔵重久弥氏^c、松下崇氏^c、越智敦彦氏^c、石川明正氏^c、戸本誠氏^d、杉本拓也氏^d、竹下徹氏^e、福永力氏^f、菅谷頼仁氏^gからも多くのご指導をいただきました。様々な機会に貴重なご意見とご指導を頂きました近藤敬比古氏^a、小林富雄氏^a、浅井祥仁氏^h、真下哲郎氏^a、田中純一氏^a、上田郁夫氏^a、金谷奈央子氏^a、磯部忠昭ⁱ氏、他 ATLAS 日本グループの方々にも深く感謝致します。

TGC 日本グループで共に研究に励み、多くの助言を頂いた久保田隆至氏^a、結束晃平氏^a、神谷隆之氏^a、二ノ宮陽一氏^a、奥村恭幸氏^d、高橋悠太氏^d、長谷川慧氏^d、伊藤悠貴氏^d、岸木俊一氏^d、早川俊氏^e、西山知徳氏^e、吹田航一氏^e、徳永香氏^e、鈴木友氏^jにもお礼を述べたい思います。また同期の石田明氏^a、岡村淳一氏^a、高橋武士氏^a、田川智博氏^a、武市祥史氏^a、橋本マーク無三四氏^a、村瀬拓郎氏^aの存在は研究を進める上で励みになりました。秘書の安蒜律子氏^a、片岡直子氏^a、塩田雅子氏^a、湯野栄子氏には出張手配や事務手続き等で大変お世話になりました。ご指導頂いた ICEPP の方々にも感謝致します。

上記の方々のご協力のもとに、私は研究に励むことができました。心より感謝を申し上げます。

最後に、大学院卒業まで私を支援してくれた両親に感謝します。

所属:

東京大学素粒子国際研究センター (ICEPP)^a

高エネルギー加速器研究機構 (KEK)^b

神戸大学自然科学研究科^c

名古屋大学理学系研究科^d

信州大学理学部^e

東京都立大学理学系研究科^f

大阪大学理学部^g

東京大学理学系研究科^h

東京大学原子核研究センターⁱ

総合研究大学院大学^j

