

修士学位論文

高輝度LHC-ATLAS実験に向けた
ミューオントリガー論理回路の実装及び性能評価
— トリガー系の統合と試験システムの構築 —

(Implementation and performance evaluation of the muon trigger logic
for the High-Luminosity LHC-ATLAS experiment
— Integration of the trigger logic and development of its validation framework —)

東京大学大学院
理学系研究科 物理学専攻
奥村研究室

成川 佳史

2024 年 1 月

概要

Large Hadron Collider (LHC) は、欧州原子核機構 (CERN) に建設された世界最高エネルギーの陽子陽子衝突型加速器である。LHC の衝突点の 1 つに設置された ATLAS 検出器では、陽子陽子衝突で生成された粒子を観測することで、素粒子標準模型の精密測定や、標準模型を超えた新物理の探索を行っている。2029 年から始まる高輝度 LHC-ATLAS 実験では、高統計量を活かした物理探索を推し進めるため、最高瞬間ルミノシティが現行の約 3 倍の $7.5 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ に増強される。加速器の高輝度化による衝突レートの増加に対応するため、ATLAS の Trigger and Data Acquisition (TDAQ) システムもアップグレードされ、初段トリガーレートは 1 MHz に、初段トリガーレイテンシーは $10 \mu\text{s}$ に拡張される。これに伴い、ATLAS 検出器のエンドキャップ部分に位置する Thin Gap Chamber (TGC) 検出器は、陽子バンチ衝突点から飛来するミューオンを捉えるための読み出し・トリガーエレクトロニクスを一新する。主な変更点は検出器からのすべてのヒット信号をヒットの有無にかかわらず、前段回路から後段回路へ転送する点にあり、より柔軟かつ包括的なトリガー判定が可能となる。この実現のために、前段回路は新たに FPGA を搭載したものに、後段回路は大規模 FPGA と System on Chip (SoC) を搭載したものに置き換えられる。

本研究では、後段回路上に実装されるトリガー論理回路の統合を完了し、その性能評価のための試験システムを開発した。エンドキャップ部ミューオントリガーは後段回路上に大規模論理回路として実装され、6 段階のトリガーマジュールをパイプライン的に接続することで実現される。各トリガーマジュール間の配線、トリガー動作に必要な周辺機能の実装、リソース使用量・タイミング制約を意識したロジック最適化を進め、トリガー回路を後段回路の全体ファームウェアへ統合した。次に、統合したトリガー回路が実機上で正常に動作していること、また各トリガーマジュールが期待されるトリガー性能を有していることを検証するため、SoC を活用した次世代的なシングルボード試験システムを開発した。これにより、シミュレーションデータや実データなどの任意のデータセットに対するトリガー応答を、ハードウェア上で動作するトリガー回路を直接用いて検証することが可能となった。また、この試験システムとソフトウェアシミュレーターを合わせることで、トリガー回路に対する詳細な調査およびデバッグが可能となり、トリガー開発における盤石な開発基盤を確立した。さらに、本システムは開発段階だけでなく、本番運用時のコミッショニングやアップグレードにも活用される予定で、精度の高いミューオントリガーの実現に貢献していく。

また、本研究では、多くの要素技術を共有する形で前段回路の品質保証試験に向けた、コンパクト DAQ システムを開発した。はじめに、前段回路に搭載された素子とインターフェイスを網羅的に試験可能なセットアップとして、SoC デバイスを起点とした試験システムを考案した。このシステムでは SoC デバイス上に起動した Linux を起点に、すべての試験を完結させる。次に、試験の実現に必要なファームウェアとアプリケーションを開発し、テストベンチを用いた動作検証を完了させた。最後に、品質保証試験の高速化を目的とした、システムの並列化を達成した。このシステムは、2024 年から開始される 1400 枚以上のボードの量産試験で使用される。また、その高い拡張性とコンパクトさから、TGC 検出器の性能評価やエレクトロニクスの試験に幅広く利用される予定で、TGC システムの安定した運用を支える重要なインフラとして機能する。

本研究で確立した、高速 FPGA や SoC を活用した次世代型の検証モデルや、汎用的な読み出しシステムは、ATLAS 実験などの高エネルギー物理実験のみに留まらず、様々なエレクトロニクスシステムに幅広く応用されることが期待される。

目次

第 1 章	序論	1
1.1	素粒子標準模型	1
1.2	LHC-ATLAS 実験	2
1.3	高輝度 LHC-ATLAS 実験に向けた Phase-II アップグレード	4
1.4	本論文の目的・内容と構成	5
第 2 章	高輝度 LHC-ATLAS 実験に向けた TGC 検出器システムのアップグレード	7
2.1	LHC-ATLAS 実験における TGC 検出器	7
2.2	TDAQ システムと PhaseII アップグレード	11
2.3	TGC 検出器の読み出し、トリガー、電気回路システムと PhaseII アップグレード	15
第 3 章	高輝度 LHC-ATLAS 実験に向けたエンドキャップ部ミュオントリガー回路の統合	37
3.1	高輝度 LHC-ATLAS 実験におけるトリガーロジックの概要と論理回路実装	37
3.2	トリガー論理回路の統合	54
第 4 章	高輝度 LHC-ATLAS 実験に向けたエンドキャップ部ミュオントリガー回路の性能評価	61
4.1	トリガー論理回路試験システムの開発	61
4.2	無限運動量飛跡を用いたトリガー回路の動作検証	69
4.3	シミュレーションデータを用いたトリガー回路の性能評価	73
第 5 章	PS board 品質保証試験に向けたコンパクト DAQ システムの開発	81
5.1	PS board 品質保証試験の設計	81
5.2	コンパクト DAQ システムの機能開発	88
5.3	PS board 品質保証試験のデモンストレーション	101
5.4	コンパクト DAQ システムとしての応用例	105
第 6 章	結論と今後の展望	109
	謝辞	111
	付録	113
A	これまでに行われたトリガーロジックの性能評価研究	113
B	無限運動量飛跡試験で明らかになった問題点とその修正	116
C	MC データを用いた試験で明らかになった問題点とその修正	119
D	タイミングバイオレーションへの対処	121

引用文献

127

目次

1.1	LHC 加速器の概観	2
1.2	ATLAS 検出器の概要	3
1.3	LHC 実験のスケジュール	4
2.1	ATLAS 検出器における座標系	8
2.2	超伝導磁石の配置	9
2.3	高輝度 LHC-ATLAS 実験でのミューオンスペクトロメーターの R - z 平面断面図	10
2.4	TGC チェンバーの断面図	11
2.5	TGC 検出器	12
2.6	Run 3 における TDAQ システムの概要	13
2.7	Run2 でのトリガーマニューの一例	14
2.8	高輝度 LHC-ATLAS 実験における TDAQ システムの概要	15
2.9	高輝度 LHC におけるトリガーマニューの例	16
2.10	TGC におけるトリガーのコンセプト	17
2.11	L1 Muon をパスしたイベント数とオフラインで再構成されたミューオンイベント数の比較	18
2.12	フェイクトリガーの例	18
2.13	Run 3 における TGC TDAQ システム	19
2.14	TGC エレクトロニクスが設置されている場所	20
2.15	Run 3 TGC システムにおける TTC システムの概要	21
2.16	高輝度 LHC-ATLAS 実験における TGC 検出器システムの概要	22
2.17	ASD の概要	23
2.18	PS board の概要	25
2.19	PP ASIC 回路のブロック図	26
2.20	TGC に入射するミューオンの時間分布	27
2.21	陽子パンチ識別回路のタイミングチャート	28
2.22	PS board から SL へ送るデータフォーマット	28
2.23	SL から PS board へ送るコントロールデータのフォーマット	29
2.24	JATHub の写真	29
2.25	JATHub によるリカバリー手続きの概要	30
2.26	SL 第一試作機の写真	31
2.27	SL FPGA に実装されるファームウェアの概要	32
2.28	SLR の概略図	33
2.29	SL FPGA のフロアマップ	34

2.30	Mercury XU5 メザニンカードの構造と SL ボード上における接続関係	35
3.1	トリガー回路の全体像	37
3.3	Station コインシデンスの概要	39
3.4	M1 Triplet におけるコインシデンスロジック	40
3.5	M2・M3 ステーションにおけるコインシデンスロジック	41
3.6	Segment Reconstruction のコンセプト	42
3.7	Wire segment Reconstruction における Unit	43
3.8	Wire segment Reconstruction のブロックダイアグラム	43
3.9	Strip segment Reconstruction における Unit	45
3.10	Strip segment Reconstruction のブロックダイアグラム	45
3.11	Wire Strip Coincidence における Coincidence Window の例	47
3.12	Wire Strip Coincidence 以降の Region の分割	48
3.14	Wire Strip Coincidence ファームウェアの概要	50
3.15	磁場内部の検出器でカバーされる η 、 ϕ 領域	51
3.16	NSW を用いたコインシデンスロジックの概要	52
3.17	Inner Coincidence の概要	52
3.18	Track Selector の概要	54
3.19	8-key sorting network の概要	54
3.20	16-key merging network の概要	55
3.21	SL FPGA 内におけるトリガーロジックの配置	55
4.1	トリガー試験ファームウェアの概要	62
4.2	トリガー読み出し回路の概要	64
4.3	Event Builder で成形されるトリガー出力のフォーマット	66
4.4	SL FPGA から MPSoC へのチップ間通信	66
4.5	アプリケーションのフローチャート	67
4.6	トリガー論理回路検証機構	68
4.7	用意した無限運動量飛跡のデータセット	70
4.8	無限運動量飛跡に対する、Strip Segment Reconstruction の応答	72
4.9	無限運動量飛跡に対する、Wire Segment Reconstruction の応答	73
4.10	無限運動量飛跡に対する、Wire Strip Coincidence の応答	74
4.11	Strip Segment Reconstruction の検出効率	76
4.12	Wire Segment Reconstruction の検出効率	77
4.13	Wire Strip Coincidence の検出効率	79
4.14	Wire Strip Coincidence の p_T ごとの検出効率	80
5.1	PS board 量産のスケジュール	81
5.2	PS board の全体像	82
5.3	PS board QAQC 試験セットアップ	84
5.4	QAQC 用 JATHub を用いた ASD テストパルス試験	85
5.5	JATHub によるクロック位相測定概念図	87
5.6	コンパクト DAQ システムの全体像	88

5.7	Ubuntu の起動シーケンス	89
5.8	光 Ethernet 通信の仕組み	90
5.9	PL から PS へのデータ読み出しシステム概要	91
5.10	調停回路のシーケンス	92
5.11	JATHub コントロール回路	93
5.12	QAQC 用 JATHub に実装されたモニター機構	94
5.13	QAQC 用 JATHub における GTX トランシーバーの概要	96
5.14	Comma detector の概要	97
5.15	読み出し回路の全体像	97
5.16	JATHub からのヒットデータ読み出しフォーマット	99
5.17	PS からのヒットデータ読み出し速度の測定	100
5.18	リードアウト回路におけるバックプレッシャー機能	100
5.20	ディレイカーブ	102
5.21	ノイズスキャンの結果	103
5.22	ASD テストパルスの結果	104
5.23	並列化システムの概要	105
5.24	並列化システムのセットアップ	106
5.25	QAQC 用 JATHub の使用例：EIL4 チェンバー試験	107
5.26	QAQC 用 JATHub の使用例：PS board SEU モニター	108
A.1	ソフトウェアシミュレーターで測定された、Wire Strip Coincidence のトリガー効率	114
A.2	Wire Segment Reconstruction の検出効率	114
B.3	無限運動量飛跡に対する、Strip Segment Reconstruction の応答	116
B.4	無限運動量飛跡に対する、Wire Segment Reconstruction の応答	117
B.5	無限運動量飛跡に対する、Wire Strip Coincidence の応答	118
D.8	シフトレジスタを構成するプリミティブ	122
D.9	SRL を利用する場合と利用しない場合におけるリソース使用状況の比較	123
D.10	Track Selector の最適化前後の比較	124

表目次

1.1	標準模型のフェルミオン	1
1.2	標準模型のボゾン	2
3.1	Wire Station Coincidence におけるコインシデンスパターン	44
3.2	Wire segment のフォーマット	44
3.3	Strip segment Reconstruction における Wire segment のデータフォーマット	46
3.4	Strip segment のフォーマット	46
3.5	Wire Strip Coincidence における飛跡候補のデータフォーマット	49
3.6	Inner Coincidence における飛跡候補のデータフォーマット	53
3.7	Wire Strip Coincidence における信号線	56
3.8	Inner Coincidence における信号線	57
3.9	最適化後のデバイスのリソース使用状況	58
4.1	Wire Strip Coincidence の読み出しフォーマット	63
4.2	各モジュールの出力ビット幅	64
4.3	パケットのフォーマット	65
4.4	無限運動量飛跡に対するトリガー回路の応答	71
4.5	用意したシングルミュオンモンテカルロデータの概要	75
5.1	PS board から送信されるモニターデータのデータタイプ	95
5.2	各試験にかかる所要時間	103
A.1	Strip Segment Reconstruction 出力のソフトウェアシミュレーターと Vivado シミュレーター比較結果	115
A.2	Wire Strip Coincidence 出力のソフトウェアシミュレーターと Vivado シミュレーター比較結果	115
D.3	Inner Coincidence を統合した後のデバイスのリソース使用状況	121
D.4	Inner Coincidence 統合後のタイミング解析の結果	122
D.5	Track Selector の最適化前後のタイミング解析の結果	124

第 1 章

序論

1.1 素粒子標準模型

素粒子標準模型は物質を構成する基本粒子である素粒子とそれらの間に働く相互作用を記述した理論体系である。標準模型を構成する素粒子は、物質を構成するフェルミオン、相互作用を媒介するゲージボゾン、質量の起源となるヒッグス粒子に分けられる。表 1.1 にフェルミオンの性質をまとめる。フェルミオンは強い相互作用をしないレプトンと、強い相互作用をするクォークに分けられ、それぞれ 3 世代 6 種類の素粒子で構成される。表 1.2 にボゾンの性質をまとめる。ボゾンは強い相互作用を媒介するグルーオン、弱い相互作用を媒介する W ボゾンと Z ボゾン、電磁相互作用を媒介する光子などのゲージボゾンと質量の起源であるヒッグス粒子で構成されている。

素粒子標準模型はこれまで多くの実験により高い精度で検証されているものの、天文観測から明らかになった暗黒物質の存在や、ヒッグス粒子の質量に対するファインチューニング問題など、多くの問題点を内包している。これらの問題を体系的に解決する理論として、ボゾンとフェルミオンの変換に対する対称性を導入する、超対称性理論が有力視されており、その直接的な検証は素粒子物理学の最重要課題である。一般に、超対称性粒子は大きな質量を持つことが予想されており、加速器を用いたエネルギーフロンティア実験はその検証において重要な役割を果たす。

表 1.1 標準模型におけるフェルミオンとその性質 [1]

		名称	表記	質量	電荷	スピン
クォーク	第 1 世代	アップ	u	2.2 MeV	+2/3	1/2
		ダウン	d	4.7 MeV	-1/3	1/2
	第 2 世代	チャーム	c	1.3 GeV	+2/3	1/2
		ストレンジ	s	93 MeV	-1/3	1/2
	第 3 世代	トップ	t	173 GeV	+2/3	1/2
		ボトム	b	4.2 GeV	-1/3	1/2
レプトン	第 1 世代	電子ニュートリノ	ν_e	< 2 eV	0	1/2
		電子	e	511 keV	-1	1/2
	第 2 世代	ミューニュートリノ	ν_μ	< 0.19 MeV	0	1/2
		ミューオン	μ	106 MeV	-1	1/2
	第 3 世代	タウニュートリノ	ν_τ	< 18.2 MeV	0	1/2
		タウ	τ	1.78 GeV	-1	1/2

表 1.2 標準模型のボゾン

	名称	表記	質量	電荷	スピン
ベクターボゾン	グルーオン	g	0	0	1
	W ボゾン	W^+	80.4 GeV	± 1	1
	Z ボゾン	Z^0	91.2 GeV	0	1
	光子	γ	0	0	1
スカラーボゾン	ヒッグス	h	125 MeV	0	0

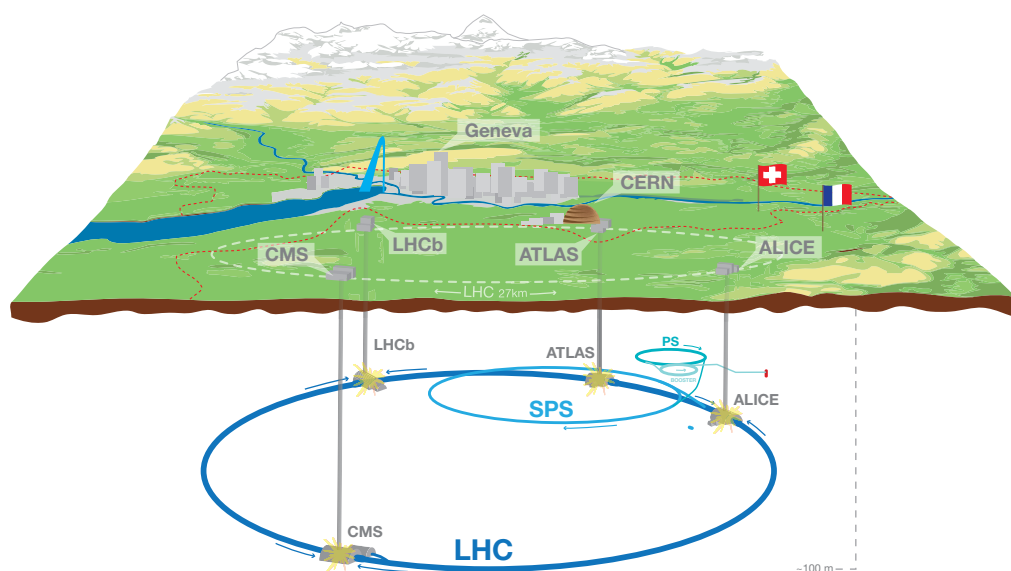


図 1.1 LHC 加速器の概観 [2]。スイスとフランスの国境をまたがって、周長 26.7 km の円形加速器が設置されている。

1.2 LHC-ATLAS 実験

スイスのジュネーブにある欧州原子核研究機構 (CERN) では、エネルギーフロンティア加速器である Large Hadron Collider (LHC) が稼働中である。LHC はスイスとフランスの国境付近に設置された陽子陽子衝突型加速器で、その周長は 26.7 km に及ぶ (図 1.1)。LHC には陽子陽子衝突が行われるポイントが 4 つあり、各衝突点ではそれぞれ特徴の異なる検出器を用いた独立した実験が行われている。本研究はその一つである A Toroidal LHC ApparatuS (ATLAS) 実験に関するものである。ATLAS 実験は LHC 加速器の高いミノシティと重心系エネルギーを活用して、素粒子標準模型の精密測定や超対称性粒子をはじめとする新粒子探索を行っている。

図 1.2 に ATLAS 検出器の全体像を示す。ATLAS 検出器は複数の検出器で構成された大型汎用検出器で、内部

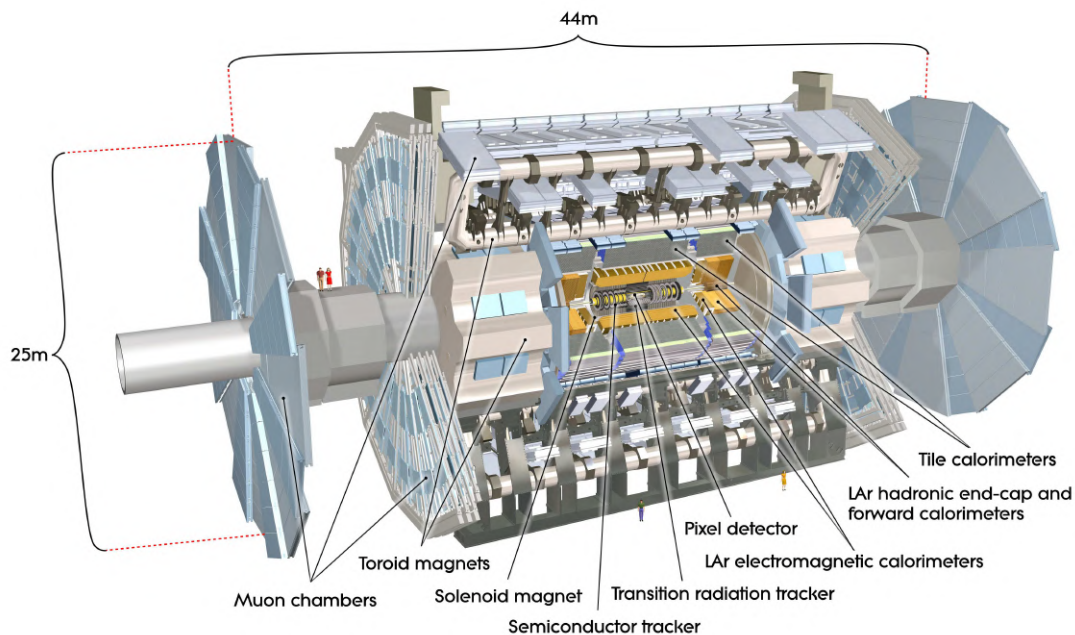


図 1.2 ATLAS 検出器の概要図 [3]。複数の検出機で構成された大型汎用検出器で、その長さは 44 m、直径は 25 m に及ぶ。内側から内部飛跡検出器、カロリメーター、ミュオンスペクトロメーターという 3 つの検出器群で構成される。

飛跡検出器、カロリメーター、ミュオンスペクトロメーターという 3 つの検出器群で構成される。最内層に設置された内部飛跡検出器は衝突点で生じた荷電粒子の飛跡を再構成し、運動量を測定する。その外側に設置されたカロリメーターは電子、光子、ハドロンを検出しそのエネルギーを測定する。最外層に設置されたミュオンスペクトロメーターは内側の検出器を透過してきたミュオンを捉え、その運動量を測定する。これらの検出器で測定された信号をイベントごとに集約することで、陽子陽子衝突で生じた事象を再構成し、物理解析を行なっている。

LHC 加速器内部では、陽子はおよそ 10^{11} 個ずつの束にまとめられ、陽子バンチを形成する。陽子バンチは LHC を周回しながら、最大エネルギー 7.0 TeV まで加速され、それぞれの衝突点において 40 MHz の頻度で交差する。1 回のバンチ交差で発生した大量の崩壊粒子は、それぞれの検出器にヒット信号を残す。それにより生じる情報量は 1 回の陽子衝突あたり約 2 MByte 程度である。これは 1 秒間に約 80 TBps のデータ量に相当し、その全てをストレージに転送し、保存することは技術的に不可能である。一方、陽子陽子衝突で生じるほとんどの事象は移行運動量がハドロンスケール程度の非弾性散乱で、エネルギーフロンティア物理の観点からは興味の薄いものである⁴¹。

そこで ATLAS 実験では莫大な背景事象の中から、興味のある事象のみを高速で選別するトリガーシステムを採用している。ATLAS のトリガーは 2 段階で構成されており、本研究で取り扱う 2029 年以降のシステムでは、ハードウェアで構成される初段トリガーで事象レートを 40 MHz から 1 MHz に、ソフトウェアで構成される後段トリガーで 1 MHz から 10 kHz まで選別する。トリガーシステムで選択されなかったデータは 2 度と取り戻すことができない。そのため、物理的に興味の深い事象を確実に補足し、そうでないものを効率的に排除する精度の高いトリガーシステムを実現することは、物理プログラム成功のための必要条件といえる。



図 1.3 LHC 実験のスケジュール [4]。LHC 実験は 2010 年に本格的に稼働を開始し、2024 年現在では Run 3 が進行中である。2026 年から高輝度 LHC 実験に向けた装置の設置が始まり、2029 年から運転が開始される。

1.3 高輝度 LHC-ATLAS 実験に向けた Phase-II アップグレード

図 1.3 に LHC の運転スケジュールを示す。LHC 実験は 2010 年から本格的に稼働を開始し、2024 年現在では第三期運転 (Run 3) が進行中である。Run 3 では 2025 年までに積分ルミノシティにして累計 450 fb^{-1} の統計量が蓄積されることが予定されている。その後、3 年間の Long Shutdown (LS3) を経て、LHC 加速器は高輝度化のため大幅にアップグレードされる。瞬間最高ルミノシティは現在の $2 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ から $5 \sim 7.5 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ に増強され、2040 年の実験終了までに蓄積される統計量は $3000 \sim 4000 \text{ fb}^{-1}$ に達する。これにより、新物理探索や標準模型の精密測定に対する感度は大きく向上することが期待される。

一方で、瞬間ルミノシティの増加に伴い、一回のバンチ交差ごとに発生する陽子陽子衝突数 (パイルアップ) が増え、背景事象が大幅に増加する。これに対応するために、高輝度 LHC-ATLAS 実験では検出器、データ収集システム、トリガーシステムが大幅にアップグレードされる。この一連のアップグレードを Phase-II アップグレードと呼ぶ。Phase-II アップグレードでは、内部飛跡検出器が全てシリコン検出器に置き換えられるほか、初段トリガーレートは 100 kHz から 1 MHz に、初段トリガーレイテンシーは $2.5 \mu\text{s}$ から $10 \mu\text{s}$ に拡張される。これに伴って、各検出器のエレクトロニクスシステムの多くがアップグレードされる。

エンドキャップ部^{*2}初段ミューオントリガーを担当する Thin Gap Chamber (TGC) 検出器でも一部を除いたすべてのエレクトロニクスが刷新される。TGC 検出器信号のデジタル化および陽子バンチ交差とのタイミング同期を担当する前段回路は、FPGA を搭載したものに置き換えられる。前段回路からのヒットデータをもとにトリガー判定を行う後段回路は、大規模 FPGA と System on Chip (SoC)、双方向で 2 Tbps の大規模光 I/O、を搭載したものに置き換えられる。

*1 超対称性で予言される新粒子やヒッグス粒子の断面積は、陽子陽子衝突の全断面積より $\mathcal{O}(10^{10})$ 程度小さい。

*2 ATLAS 検出器では円筒の側面部分をパレル、底面部分をエンドキャップと呼ぶ。詳細は 2.1.1 節で述べる。

1.4 本論文の目的・内容と構成

本研究は、高輝度 LHC-ATLAS 実験に向けて、TGC 検出器の後段回路に実装されるトリガー論理回路を完成させることを目的とする。そのために、まずトリガー回路を後段回路全体ファームウェアへと統合し、実機上で動作させるための準備を整える。次に、実装したトリガー回路の動作検証および性能評価に向けて、実機を用いたシングルボード試験システムを開発し、ミューオン飛跡のシミュレーションデータに対するトリガー効率を測定することで、トリガー回路が期待される性能を有しているか検証する。また、本研究では多くの要素技術を共有する課題として、前段回路に対する品質保証試験の開発にも取り組む。これにより、2024 年から開始される 1500 枚に及ぶ前段回路の量産に向けて必要不可欠なインフラを構築する。

これらの開発研究を通じて、本研究は将来の高エネルギー実験に幅広い応用が期待される、高速 FPGA や SoC を活用した次世代型の検証モデルを確立した。

本論文の構成は以下の通りである。第 2 章では高輝度 LHC-ATLAS 実験に向けた TGC 検出器システムのアップグレードについて説明し、各エレクトロニクスの役割を紹介する。第 3 章では高輝度 LHC-ATLAS 実験でのエンドキャップ部ミューオントリガーロジックの詳細を説明し、トリガー回路の統合について述べる。第 4 章では、トリガー論理回路の性能評価のために開発したシングルボード試験システムと、それを用いて行った試験の結果を述べる。第 5 章では前段回路の量産に向けた、品質保証試験システムの設計・開発・実装について説明する。最後に 6 章で本研究のまとめと今後の展望について述べる。

第 2 章

高輝度 LHC-ATLAS 実験に向けた TGC 検出器システムのアップグレード

2.1 LHC-ATLAS 実験における TGC 検出器

本節では本研究のテーマである、Thin Gap Chamber (TGC) 検出器について説明する。それに先んじて、ATLAS 検出器で使われる座標系と、ATLAS 検出器における磁場構造について説明する。

2.1.1 ATLAS 実験における座標系

ATLAS 実験では主に 2 種類の座標系が用いられる。図 2.1 にその概要を示す。1 つ目の座標系である直交座標系は、原点を検出器中心に取り LHC の中心方向を x 、地上方向を y に取った右手系で定義する。この時、 z 軸はビーム軸方向に沿って定義される。 $z > 0$ の領域を A side、 $z < 0$ の領域を C side と呼ぶ。2 つ目の座標系である円筒座標系は、ビーム軸を中心とした方位角方向を ϕ 、ビーム軸からの天頂角方向を θ 、同型方向を R と定義する。また、 θ 方向を表す変数として、擬ラピディティ (Pseudorapidity)

$$\eta = -\ln\left(\tan\frac{\theta}{2}\right) \quad (2.1)$$

を定義し、粒子の四元運動量を記述する際^{*1}やビーム衝突点から見た時の各検出器の立体角を議論する際に利用する。本研究で扱うミューオンシステムの議論でも η を利用し、 $|\eta| < 1.05$ の円筒側面領域をバレル領域 (Barrel)、 $|\eta| > 1.05$ の円筒底面領域をエンドキャップ (Endcap) と区別する。

ATLAS 実験では粒子の状態を表す物理量として z 軸垂直方向の運動量 (横方向運動量、 p_T) やエネルギー (横方向エネルギー、 E_T) を利用する。高エネルギーの陽子陽子衝突は、陽子の内部構造であるパートン同士の衝突であり、その担う運動量の差により衝突重心系は実験室系では z 軸方向にローレンツブーストされている。一方で横方向の運動量は (交差角の効果や、フェルミ運動量を除けば) 実験室系で厳密にゼロとなる。終状態粒子の運動量の z 軸方向の成分はパートン同士の運動量の非対称による寄与が大きく、パートン同士の衝突重心系の全エネルギーの良い指標にはならない。一方、で横方向の運動量はパートン同士の衝突重心系の全エネルギーの良い指標となるため、横方向運動量や横方向エネルギーが事象選別の重要な指標となる。オンライン事象選別であるトリガーでも横方向運動量、横方向エネルギーを指標にして興味のある事象を選択する。

^{*1} η はハドロンコライダーで生じる物理現象を説明する際によく利用される。2 粒子間の η の差は z 軸方向のブーストに対してローレンツ不変であり、始状態の粒子の z 方向の運動量に関わらず崩壊事象を統一的に記述することができる。

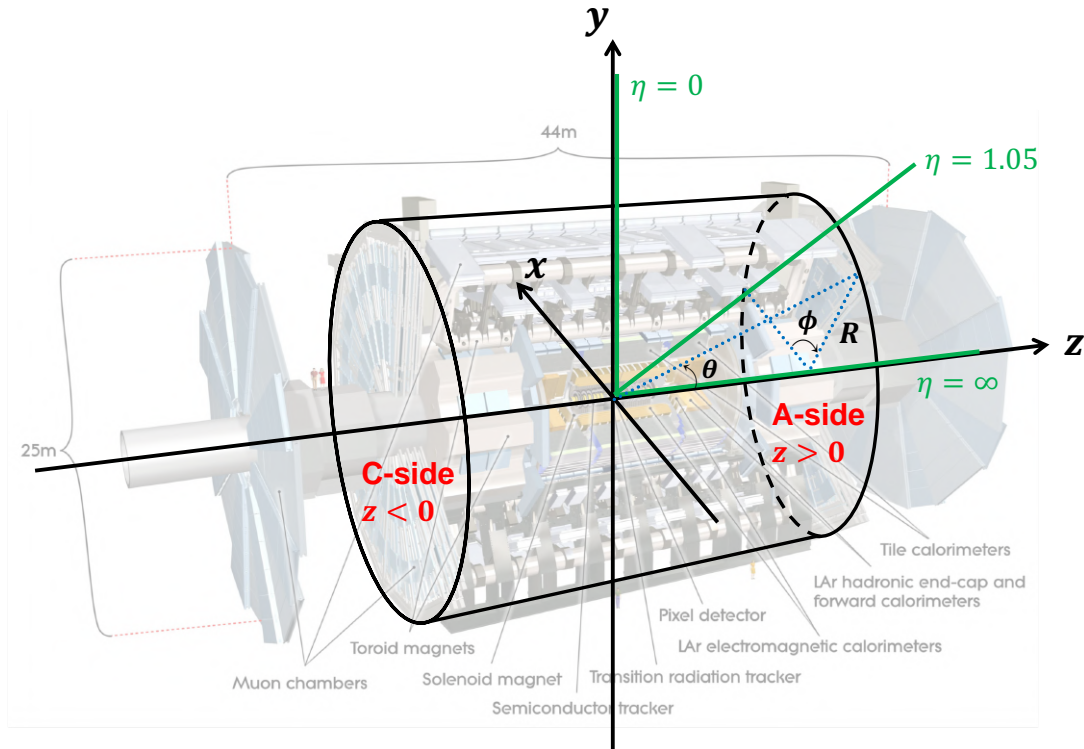


図 2.1 ATLAS 検出器で用いられる座標系。直交座標系は、原点を検出器中心に取り LHC の中心方向を x 、地上方向を y に取った右手系で定義する。 $z > 0$ を A side、 $z < 0$ を C side と呼ぶ。円筒座標系は、ビーム軸を中心とした方位角方向を ϕ 、ビーム軸からの天頂角方向を θ 、同型方向を R と定義する。最外層のミュオンシステムは $|\eta| = 1.05$ でバレル領域とエンドキャップ領域に分かれる。

2.1.2 ATLAS 検出器における超伝導磁石

ATLAS 検出器では荷電粒子の運動量を測定するため、2種類の超伝導磁石が設置されている。図 2.2 に超伝導磁石の配置を示す。1つ目は、内部飛跡検出器内で荷電粒子を曲げるのに利用されるソレノイド磁石である。ソレノイド磁石は内部飛跡検出器とカロリメーターの間の領域に設置され、 z 軸方向の磁場を生成する。2つ目は、内部検出器を透過してきたミュオンを曲げるのに利用されるトロイド磁石である。トロイド磁石はバレル部用とエンドキャップ部用で2種類用意される。バレル部とエンドキャップ部のトロイド磁石はそれぞれ ϕ 方向に8回対称の構造をとっており、互いの干渉を避けるため 22.5 度ずらして設置される。トロイド磁石は主に ϕ 方向の磁場を生成し、 η 方向に荷電粒子を曲げるが、磁石の構造によって磁場は場所ごとに不均一になっている。

2.1.3 ミュオンスペクトロメーター

ミュオンスペクトロメーターは ATLAS 検出器最外層に設置された検出器で、カロリメーターを透過したミュオンの横方向運動量を測定する。ミュオンスペクトロメーターはトロイド磁場の構造に合わせて ϕ 方向に8回対称になっており、そのうちバレル部のトロイド磁石が位置する領域を Small sector、トロイド磁石間に位置する領域を Large sector と呼ぶ。ミュオンスペクトロメーターの全体像を図 2.3 に示す。

高輝度 LHC-ATLAS 実験でエンドキャップ部ミュオントリガーに関わる検出器には、TGC 検出器、New Small Wheel (NSW)、Resistive Plate Chamber (RPC)、Monitored Drift Tube (MDT)、Tile カロリメーターがあ

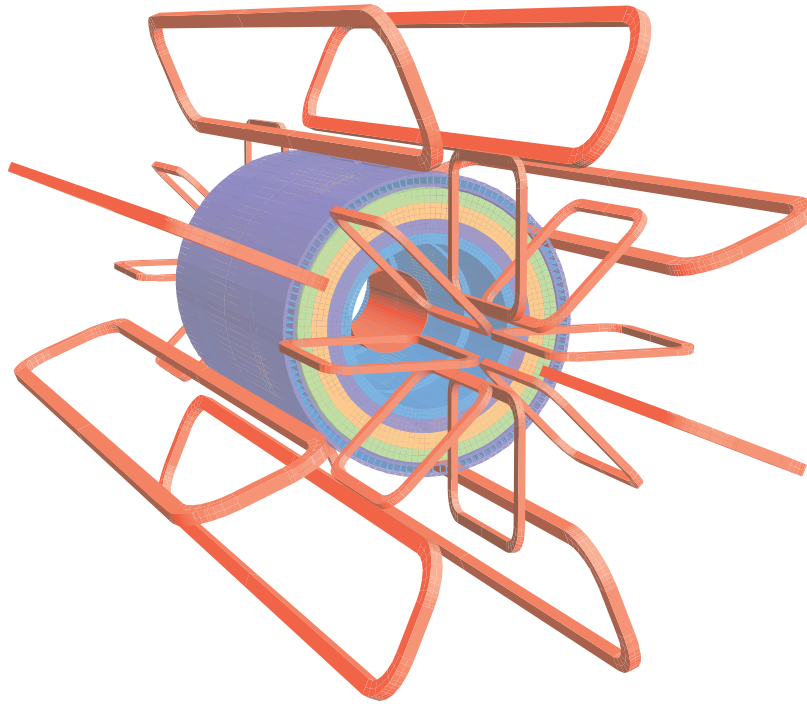
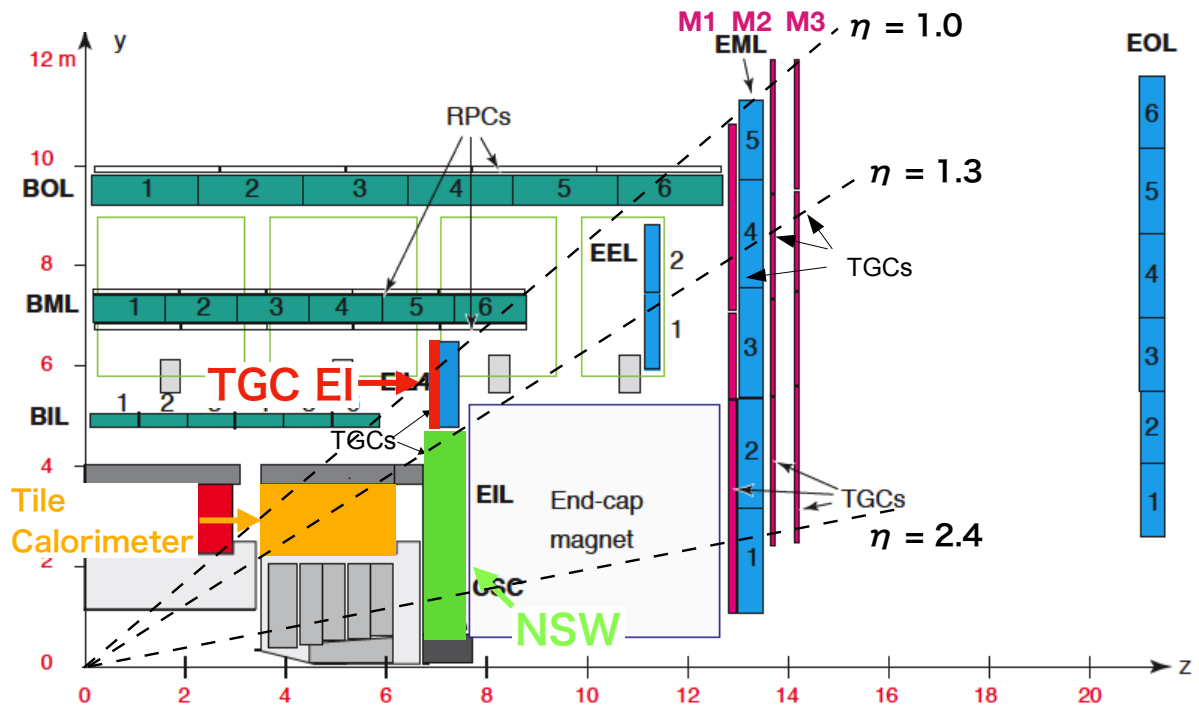


図 2.2 超伝導磁石の配置 [3]。内部飛跡検出器を囲うようにソレノイド磁石が、カロリメーターの外側にトロイド磁石が設置されている。バレル部とエンドキャップ部のトロイド磁石はそれぞれ ϕ 方向に 8 回対称の構造をとっており、互いの干渉を避けるため 22.5 度ずらして設置される。

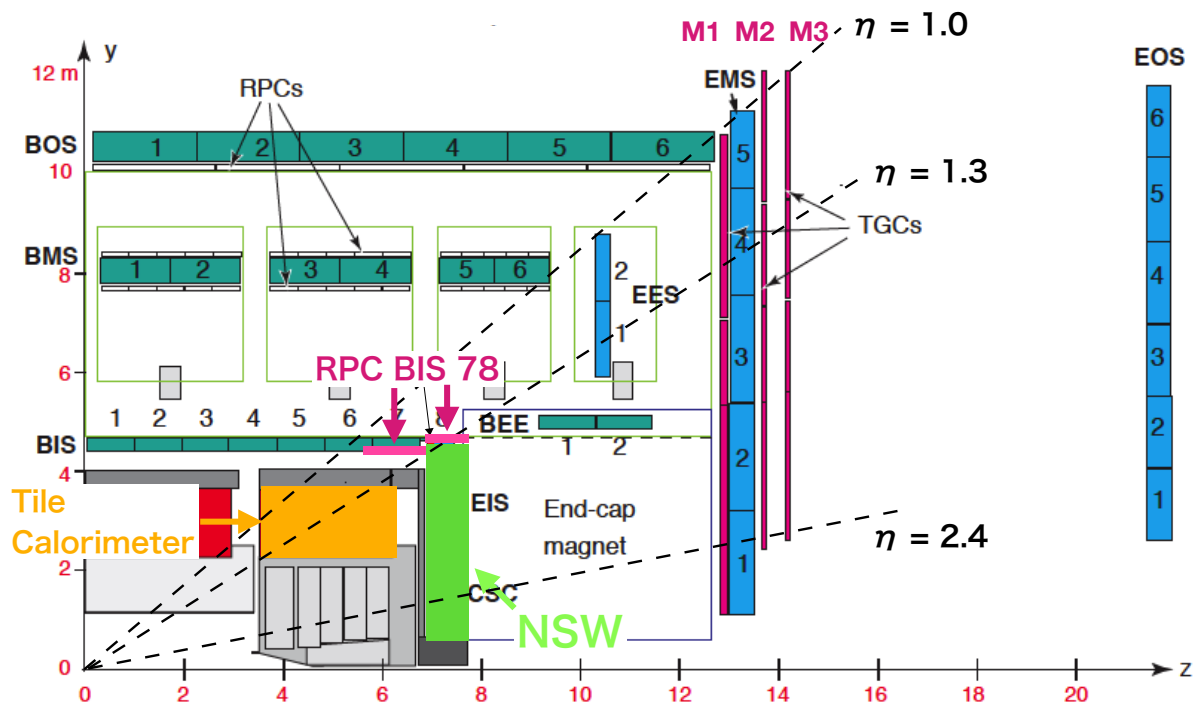
る。TGC 検出器については後ほど詳述する。NSW はエンドキャップトロイド磁場の内側に設置されたトリガーおよび精密測定用のミュオン検出器であり、 $1.3 < |\eta| < 2.7$ の全 ϕ 方向をカバーする。small-strip TGC (sTGC) と Micromegas (MM) の 2 種類の検出器を 4 層ずつ組み合わせた構造をしており、荷電粒子が通過した位置および飛跡の角度情報を取得する。RPC は $|\eta| < 1.05$ のバレル領域に設置されたトリガー用のミュオン検出器である。直交するストリップで η と ϕ の位置情報を読み出す。Run 3 以降では Small sector に RPC BIS78 というエンドキャップ領域をカバーする検出器が設置され、エンドキャップミュオントリガーで利用される。MDT はバレル領域およびエンドキャップ領域に設置された精密測定用のミュオン検出器である。ドリフトチューブを 6 層または 8 層並べた構造をとっており、荷電粒子が通過した位置および飛跡の角度情報を再構成する。MDT はドリフト時間が長く、Run 3 までは初段トリガーで用いることができなかったが、高輝度 LHC-ATLAS 実験からは拡張されるレイテンシーを活用して、初段トリガーにも用いられる。Tile カロリメーターは電磁カロリメーターの外側に設置されたハドロンカロリメーターであり、複数の層が重なった構造をとっている。最外層に到達する粒子のほとんどがミュオンであることを利用して、エンドキャップ領域のトリガー判定に用いられる。

TGC 検出器

TGC 検出器は $1.05 < |\eta| < 2.4$ のエンドキャップ領域に設置されたミュオントリガー用の検出器である。エンドキャップトロイド磁石より内側に位置する Endcap Inner (EI) と外側に位置する Big Wheel (BW) に大別される。TGC BW は z 方向に 3 つのステーションが連なって構成され、衝突点に近い方から M1、M2、M3 ステーションと呼ぶ(図 2.3 参照)。各ステーションは 2 層または 3 層のガス層で構成されており、2 層のものを Doublet、3 層のものを Triplet と呼ぶ。BW では M1 が Triplet、M2、M3 が Doublet になっている。



(a) Large sector でのミュオン検出器の配置



(b) Small sector でのミュオン検出器の配置

図 2.3 高輝度 LHC-ATLAS 実験でのミュオンスペクトロメーターの R - z 平面断面図 [5]。TGC BW 以外のミュオンスペクトロメーターはトロイドマグネットの磁場構造に合わせて 8 回対称になっており、 ϕ 方向に Large Sector、Small sector という 2 種類の sector に分かれている。エンドキャップ部のミュオントリガーには $1.05 < |\eta| < 1.3$ の領域では TGC、RPC BIS78、TGC EIL4、Tile カロリメーターが、 $1.3 < |\eta| < 2.4$ の領域では TGC と NSW が用いられる。

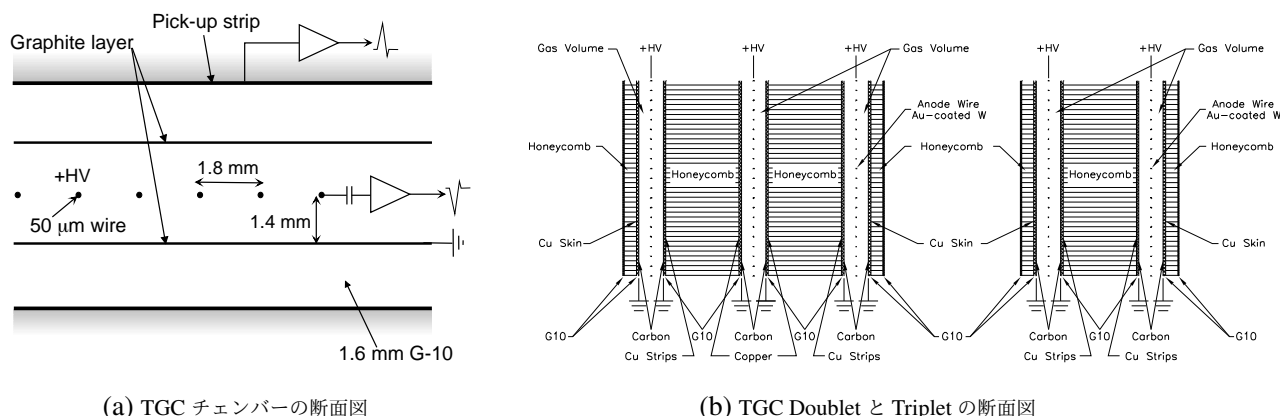


図 2.4 (a) は TGC チェンバーの断面図を表す [3]。ワイヤーとストリップが直交して張られていることがわかる。(b) は TGC Doublet と Triplet の断面図である。それぞれ 2 層、3 層のガスギャップで構成されており、各ガスギャップの間はペーパーハニカムにより満たされている。

TGC チェンバーの構造を図 2.4 に示す。TGC はアノードワイヤー間隔が 1.8 mm、アノードと GND の間隔が 1.4 mm である MWPC である。ワイヤーは R 方向、ストリップは ϕ 方向に直交して張られており、それぞれのヒット信号からミューオンが通過した 2 次元位置を検出することができる。ガス層には $\text{CO}_2/n\text{-C}_5\text{H}_{12}$ が 55:45 で混合されたガスが充填されている。荷電粒子が TGC に入射すると、電磁相互作用によりガス分子が電離される。電離した電子はワイヤーに印加された 2.8 kV の電圧によりワイヤー方向に集められ、ワイヤー付近に到達すると強い電場により電子雪崩を生じる。ワイヤーでは電子雪崩で生じた正イオンのドリフトが、ストリップではそれらの鏡像電荷が電流信号として検出される。

TGC 検出器はトリガー用の検出器であり、ワイヤー間隔が小さいため時間応答がよい。これにより陽子交差頻度である 25 ns より細かな時間分解能でミューオンを検出し、そのミューオンがどの陽子バンチ交差に由来するのかを識別 (Bunch Crossing Identification, BCID) することができる。一方、TGC 検出器はそれほど高い位置分解能が求められていないため、ワイヤー電極を 4 ~ 20 本まとめてから読み出しを行う。結果としてワイヤー、ストリップは合計 32 万チャンネルをもつ。

図 2.5 に TGC BW の写真を示す。BW の外側の領域 ($1.05 < |\eta| < 1.92$) はエンドキャップ領域、内側の領域 ($1.92 < |\eta| < 2.4$) はフォワード領域と呼ばれ、異なる構造を取っている。エンドキャップ領域では ϕ 方向に 48 回対称になるように、フォワード領域では ϕ 方向に 24 回対称になるようにチェンバーが設置されている。エンドキャップ領域の 1/48、フォワード領域の 1/24 はトリガー回路的に独立しており、それぞれが "トリガーセクター" と呼ばれる。また、電源供給、ガス供給、電気回路制御、読み出しの観点から BW は ϕ 方向に 12 個のセクターに分割されており、これを 1/12 セクターと呼ぶ。

2.2 TDAQ システムと PhaseII アップグレード

LHC では 25 ns の間隔で陽子バンチが衝突するため、衝突で生じたすべてのデータを保存することはできない。限られた読み出し帯域とオフラインの計算リソースを最大限有効活用するためには興味のある衝突事象のみを記録するトリガーが重要となる。またトリガー判定がなされたイベントに対して正しくデータを取得するには、トリガーシステムとデータ取得 (data acquisition, DAQ) システムが連動して機能する必要がある。ATLAS 実験では、トリガーとデータ取得をまとめて Trigger and Data Acquisition (TDAQ) システムと呼ぶ。本節では Run 3 での TDAQ システムと PhaseII での TDAQ システムについて説明し、PhaseII アップグレードにおける変更点を述べる。

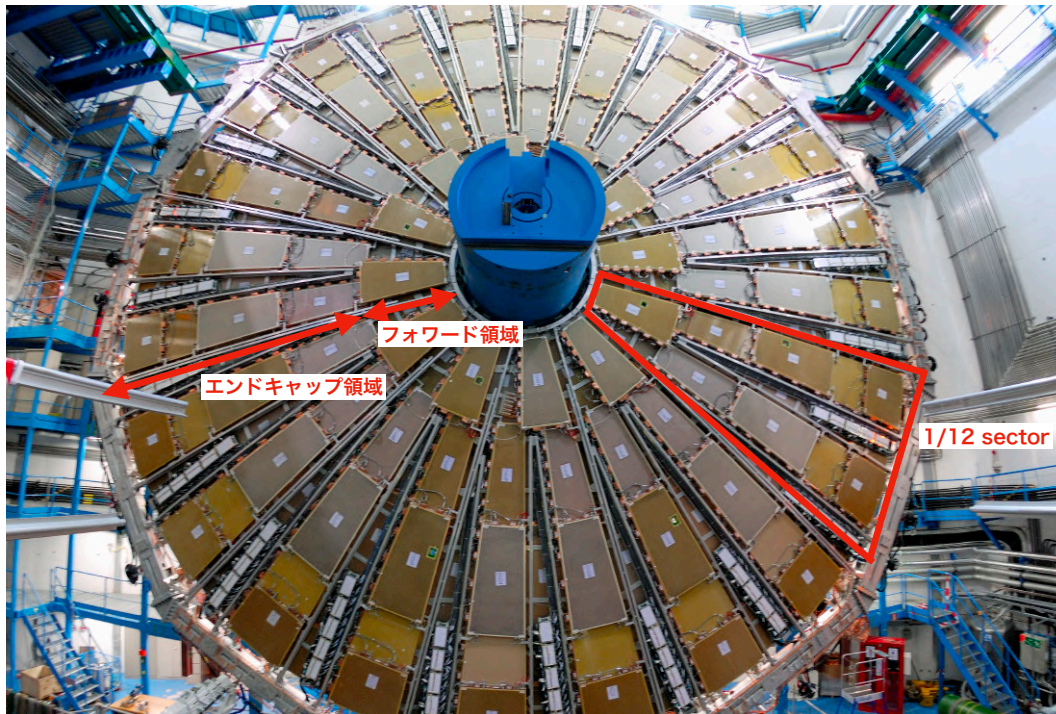


図 2.5 TGC 検出器の正面写真 (M1)[6]。TGC 検出器は電源供給、ガス供給、電気回路制御、読み出しの観点から ϕ 方向に 12 回対称になっており、赤枠で囲った範囲を 1/12 セクターと呼ぶ。また、TGC 検出器は η 方向に 2 種類の構造をとっており、円盤の外側の領域 ($1.05 < |\eta| < 1.92$) をエンドキャップ領域、円盤の内側の領域 ($1.92 < |\eta| < 2.4$) をフォワード領域と呼ぶ。エンドキャップ領域では ϕ 方向に 48 回対称になるよう、フォワード領域では ϕ 方向に 24 回対称になるようにチェンバーが設置されている。

2.2.1 Run 3 での TDAQ システム

図 2.6 に Run 3 における TDAQ システムの概要を示す。ATLAS のトリガーシステムは Level-1 という初段のハードウェアトリガーと、それに続く High Level Trigger (HLT) という後段のソフトウェアトリガーから構成される。

Level-1 Trigger

Level-1 Trigger は 25 ns 間隔で行われるすべての陽子バンチ交差の中から、物理的に興味のある衝突事象を高速で選別する初段トリガーである。L1 Trigger 判定に従い、40 MHz の陽子バンチ交差に対して 100 kHz までのレートでデータが読み出される。L1 Trigger は全ての衝突のデータを処理し、トリガー判定を行う必要があるため、ASIC や FPGA などのハードウェアを利用し、高速処理を実現する。

ATLAS での L1 Trigger システムは主にカロリメータトリガー (Level-1 Calo) と ミューオントリガー (Level-1 Muon) で構成される。Level-1 Calo はカロリメーターからのエネルギー情報をもとに、高いエネルギーを持つ電子、光子、ジェットを検出する。Level-1 Muon は RPC と TGC からの情報をもとに、横方向運動量の大きなミューオンを検出する。RPC と TGC からのトリガー情報は MUon-to Central Trigger Processor Interface (MUCTPI) で統合される。Level-1 Calo と Level-1 Muon からの信号は Central Trigger Processor (CTP) に渡され、総合的に Level-1 トリガー判定が行われる。L1 Trigger によって選別されたイベントについては、各検出器のフロントエンド回路に Level-1 Accept (L1A) が送られ、そのバンチ交差に由来する検出器のヒット信号が Readout Driver

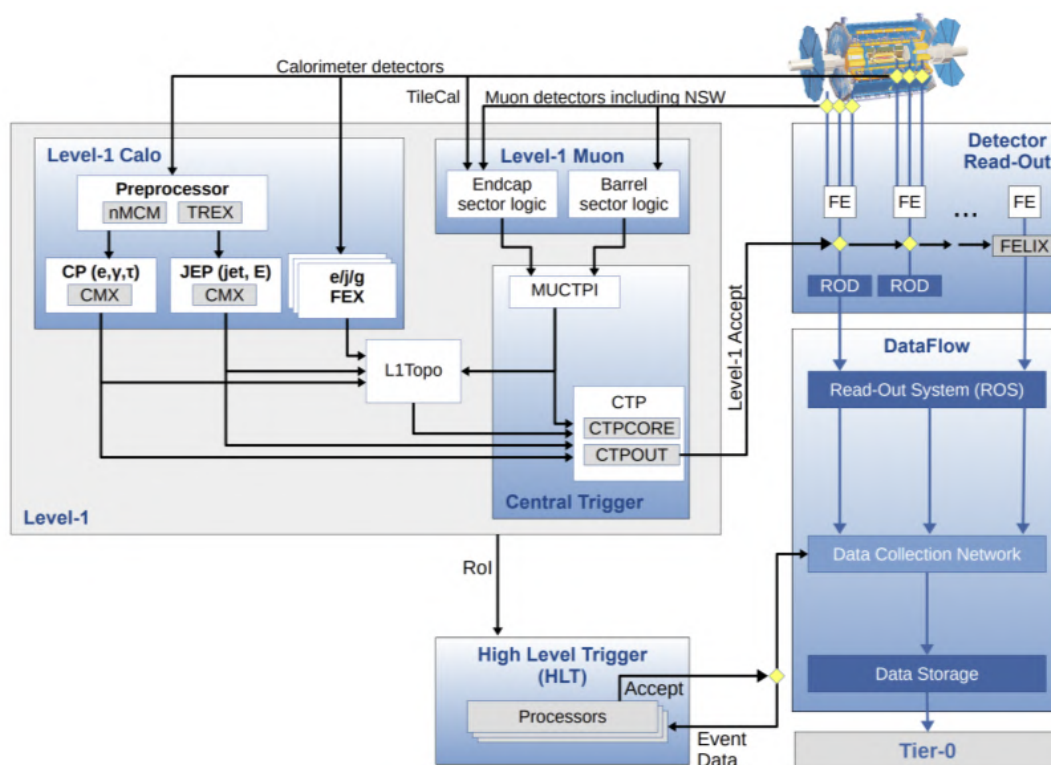


図 2.6 Run 3 における TDAQ システムの概要 [7]。トリガーシステムは Level-1 Trigger という初段のハードウェアトリガーと High Level Trigger (HLT) という後段のソフトウェアトリガーから構成される。L1 Trigger は Level-1 Calo と Level-1 Muon に大別され CTP で総合的なトリガー判定がなされる。L1 Trigger をパスしたイベントは HLT でより精度の高いトリガー判定が行われ、CERN Permanent Storage に保存される。

(ROD) へと送られる。

Level-1 Trigger ではバンチ交差が生じてから L1A が届けられるまでの時間 (Level-1 レイテンシー) が一定である Fixed Latency Scheme を採用している。L1A が出されるまでの間、陽子バンチ交差で生じるデータは各フロントエンドエレクトロニクス上のバッファに保管される。現行システムでは L1 レイテンシーは $2.5 \mu\text{s}$ に設定されており、それを満たすようフロントエンドエレクトロニクスの設計が行われている。

High Level Trigger (HLT)

HLT は初段トリガーをパスしたイベントから最終的にストレージに保存する事象を選ぶ役割を担う、ソフトウェアベースのトリガーである。初段トリガーでは使われなかった内部飛跡検出器や MDT などの精密測定用検出器からの情報も利用して、より高い精度でイベント再構成を行う。HLT によりトリガーレートは 3.3 kHz まで削減され、トリガーをパスしたイベントは、データセンターのストレージに記録される。その後 CERN のコンピューティングセンターである Tier-0 において処理され、イベントが再構成される。

トリガーマニュー

ATLAS 実験は陽子陽子衝突で生じるさまざまな事象を取得することで、幅広い終状態を持つ多様な物理解析を展開する。広範な物理事象を取得するため、複数のトリガー選別条件が用意されており、それぞれに適切なトリガーレートが分配されている。図 2.7 にトリガーマニューと呼ばれる、各トリガー選別条件とそれに割り振られた

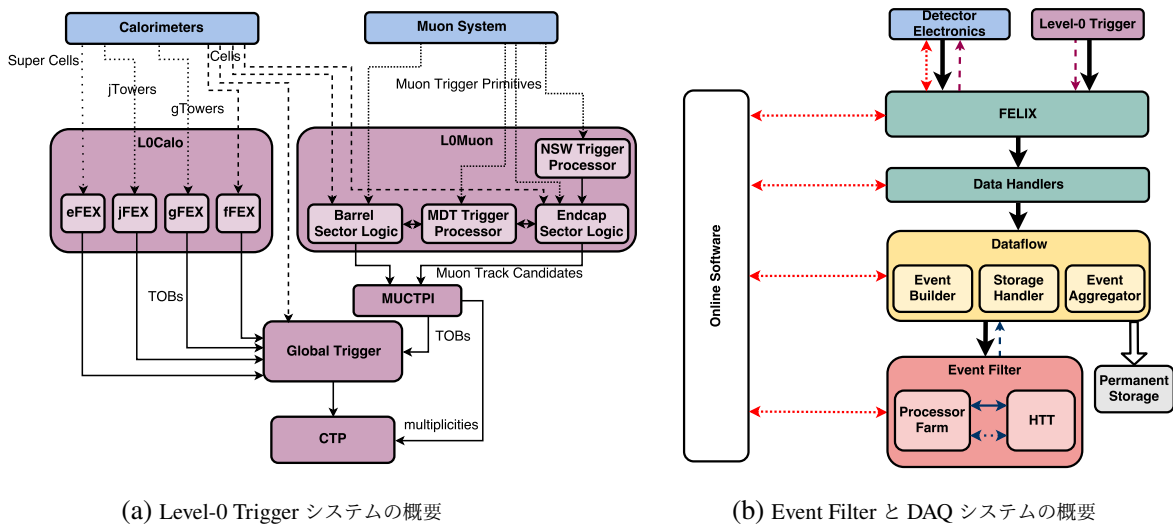
Trigger	Typical offline selection	Trigger Selection		L1 Peak Rate [kHz]	HLT Peak Rate [Hz]
		L1 [GeV]	HLT [GeV]	$L=2.0 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$	
Single leptons	Single isolated μ , $p_T > 27 \text{ GeV}$	20	26 (i)	16	218
	Single isolated tight e , $p_T > 27 \text{ GeV}$	22 (i)	26 (i)	31	195
	Single μ , $p_T > 52 \text{ GeV}$	20	50	16	70
	Single e , $p_T > 61 \text{ GeV}$	22 (i)	60	28	20
	Single τ , $p_T > 170 \text{ GeV}$	100	160	1.4	42
Two leptons	Two μ , each $p_T > 15 \text{ GeV}$	2×10	2×14	2.2	30
	Two μ , $p_T > 23, 9 \text{ GeV}$	20	22, 8	16	47
	Two very loose e , each $p_T > 18 \text{ GeV}$	2×15 (i)	2×17	2.0	13
	One e & one μ , $p_T > 8, 25 \text{ GeV}$	$20 (\mu)$	7, 24	16	6
	One loose e & one μ , $p_T > 18, 15 \text{ GeV}$	15, 10	17, 14	2.6	5
	One e & one μ , $p_T > 27, 9 \text{ GeV}$	22 (e, i)	26, 8	21	4
	Two τ , $p_T > 40, 30 \text{ GeV}$	20 (i), 12 (i) (+jets, topo)	35, 25	5.7	93
	One τ & one isolated μ , $p_T > 30, 15 \text{ GeV}$	12 (i), 10 (+jets)	25, 14 (i)	2.4	17
One τ & one isolated e , $p_T > 30, 18 \text{ GeV}$	12 (i), 15 (i) (+jets)	25, 17 (i)	4.6	19	
Three leptons	Three very loose e , $p_T > 25, 13, 13 \text{ GeV}$	$20, 2 \times 10$	$24, 2 \times 12$	1.6	0.1
	Three μ , each $p_T > 7 \text{ GeV}$	3×6	3×6	0.2	7
	Three μ , $p_T > 21, 2 \times 5 \text{ GeV}$	20	$20, 2 \times 4$	16	9
	Two μ & one loose e , $p_T > 2 \times 11, 13 \text{ GeV}$	$2 \times 10 (\mu)$	$2 \times 10, 12$	2.2	0.5
	Two loose e & one μ , $p_T > 2 \times 13, 11 \text{ GeV}$	$2 \times 8, 10$	$2 \times 12, 10$	2.3	0.1
Single photon	One loose γ , $p_T > 145 \text{ GeV}$	24 (i)	140	24	47
Two photons	Two loose γ , each $p_T > 55 \text{ GeV}$	2×20	2×50	3.0	7
	Two γ , $p_T > 40, 30 \text{ GeV}$	2×20	35, 25	3.0	21
	Two isolated tight γ , each $p_T > 25 \text{ GeV}$	2×15 (i)	2×20 (i)	2.0	15
Single jet	Jet ($R = 0.4$), $p_T > 435 \text{ GeV}$	100	420	3.7	35
	Jet ($R = 1.0$), $p_T > 480 \text{ GeV}$	111 (topo: $R = 1.0$)	460	2.6	42
	Jet ($R = 1.0$), $p_T > 450 \text{ GeV}$, $m_{\text{jet}} > 45 \text{ GeV}$	111 (topo: $R = 1.0$)	$420, m_{\text{jet}} > 35$	2.6	36
b -jets	One b ($\epsilon = 60\%$), $p_T > 285 \text{ GeV}$	100	275	3.6	15
	Two b ($\epsilon = 60\%$), $p_T > 185, 70 \text{ GeV}$	100	175, 60	3.6	11
	One b ($\epsilon = 40\%$) & three jets, each $p_T > 85 \text{ GeV}$	4×15	4×75	1.5	14
	Two b ($\epsilon = 70\%$) & one jet, $p_T > 65, 65, 160 \text{ GeV}$	$2 \times 30, 85$	$2 \times 55, 150$	1.3	17
	Two b ($\epsilon = 60\%$) & two jets, each $p_T > 65 \text{ GeV}$	$4 \times 15, \eta < 2.5$	4×55	3.2	15
Multijets	Four jets, each $p_T > 125 \text{ GeV}$	3×50	4×115	0.5	16
	Five jets, each $p_T > 95 \text{ GeV}$	4×15	5×85	4.8	10
	Six jets, each $p_T > 80 \text{ GeV}$	4×15	6×70	4.8	4
	Six jets, each $p_T > 60 \text{ GeV}$, $ \eta < 2.0$	4×15	$6 \times 55, \eta < 2.4$	4.8	15
E_T^{miss}	$E_T^{\text{miss}} > 200 \text{ GeV}$	50	110	5.1	94
B -physics	Two μ , $p_T > 11, 6 \text{ GeV}$, $0.1 < m(\mu, \mu) < 14 \text{ GeV}$	11, 6	11, 6 (di- μ)	2.9	55
	Two μ , $p_T > 6, 6 \text{ GeV}$, $2.5 < m(\mu, \mu) < 4.0 \text{ GeV}$	$2 \times 6 (J/\psi, \text{topo})$	$2 \times 6 (J/\psi)$	1.4	55
	Two μ , $p_T > 6, 6 \text{ GeV}$, $4.7 < m(\mu, \mu) < 5.9 \text{ GeV}$	$2 \times 6 (B, \text{topo})$	$2 \times 6 (B)$	1.4	6
	Two μ , $p_T > 6, 6 \text{ GeV}$, $7 < m(\mu, \mu) < 12 \text{ GeV}$	$2 \times 6 (Y, \text{topo})$	$2 \times 6 (Y)$	1.2	12
Main Rate				86	1750
B-physics and Light States Rate					200

図 2.7 Run2 でのトリガメニューの一例 [8]。解析で利用される典型的なオブジェクトを取得するための Level-1 および HLT でのトリガー閾値が定められる。全体を通してトリガーレートの制約を守るよう設計される。

トリガーレートをまとめたリストを示す。トリガメニューでは、解析で利用される lepton、jet、消失横方向エネルギー (E_T^{miss}) などの典型的なオブジェクトを取得するための、Level-1 および HLT トリガー閾値が定められている。

2.2.2 高輝度 LHC-ATLAS 実験での TDAQ システム

高輝度 LHC-ATLAS 実験ではパイルアップによる背景事象が大幅に増加し、トリガーレートが増加する。現行の TDAQ システムのままでは、読み出し能力の限界からトリガー制約を厳しくせざるを得ず、その結果、興味のある物理事象へのアクセプタンスを落としてしまう。そこで高輝度 LHC-ATLAS 実験に向けて、大規模な TDAQ システムのアップグレードが行われる。初段トリガーレートは 100 kHz から 1 MHz へ、後段トリガーレートは 3.3 kHz から 10 kHz へと拡張される。さらに、初段トリガーレイテンシーも $2.5 \mu\text{s}$ から $10 \mu\text{s}$ へと拡張される。図 2.8 に高輝度 LHC-ATLAS 実験での TDAQ システムの概要を示す。高輝度 LHC-ATLAS 実験では初段トリガー



(a) Level-0 Trigger システムの概要

(b) Event Filter と DAQ システムの概要

図 2.8 高輝度 LHC-ATLAS 実験における TDAQ システムの概要 [9]。(a) に Level-0 Trigger システムの概要を示す。Level-0 Trigger は Level-0 Calo と Level-0 Muon に大別され、CTP で総合的なトリガー判定がなされる。CTP で後段に送られるべきと判断された場合、FELIX を経由して各フロントエンドエレクトロニクスに LOA 信号が分配される。(b) に Event Filter と DAQ システムの概要を示す。LOA を受けた各システムは検出器からのヒットデータを FELIX に送る。FELIX は受け取ったデータを Event Filter に渡す。EF ではソフトウェアベースのトリガー判定が行われ、最後まで残ったデータが CERN の Permanent Storage に保存される。

を Level-0 Trigger、後段トリガーを Event Filter (EF) と呼ぶ。

L0 Trigger は L0 Calo、L0 Muon、Global Trigger、CTP で構成される。L0 Muon では、新たに精密測定用の MDT もトリガーに用いられるようになる。TGC や RPC の情報と組み合わせることで、より精度の高いトリガー判定を実現する。Global Trigger は L0 Calo と MUCTPI からの位置や p_T 、 E_T などの情報を基に、その事象が特徴的なトポロジーを持つものか判定し、その結果を CTP に送る。CTP は L0 Calo、L0 Muon、L0 Global からの入力に基づき、トリガーメニュー (図 2.9) に従い、各トリガー条件に指定されたプリスケリングファクターを適用してトリガー判定を行う。各フロントエンドエレクトロニクスには Front-End Link eXchange (FELIX) を経由して Level-0 Accept (LOA) 信号が分配される。LOA を受けた各エレクトロニクスは、該当する陽子バンチ交差に由来するデータを FELIX に送信する。FELIX はこれらのデータを Event Filter に転送する。Event Filter ではソフトウェアベースのトリガー判定が行われ、トリガーレートは 10 kHz まで削減される。最終的に残った生データは、CERN のデータセンターのストレージに保存される。

2.3 TGC 検出器の読み出し、トリガー、電気回路システムと PhaseII アップグレード

高輝度 LHC-ATLAS 実験に向けた TDAQ システムのアップグレードに伴い、TGC 検出器の読み出し、トリガー、電気回路システムも大幅にアップグレードされる。本節ではまず、TGC 検出器におけるトリガーのコンセプトを説明する。次に Run 3 での TGC 検出器システムと PhaseII での TGC 検出器システムについて説明し、PhaseII アップグレードにおける変更点を述べる。

Trigger Selection	Run 1 Offline p_T Threshold [GeV]	Run 2 (2017) Offline p_T Threshold [GeV]	Planned HL-LHC Offline p_T Threshold [GeV]	L0 Rate [kHz]	After regional tracking cuts [kHz]	Event Filter Rate [kHz]
isolated single e	25	27	22	200	40	1.5
isolated single μ	25	27	20	45	45	1.5
single γ	120	145	120	5	5	0.3
forward e			35	40	8	0.2
di- γ	25	25	25,25		20	0.2
di- e	15	18	10,10	60	10	0.2
di- μ	15	15	10,10	10	2	0.2
$e - \mu$	17,6	8,25 / 18,15	10,10	45	10	0.2
single τ	100	170	150	3	3	0.35
di- τ	40,30	40,30	40,30	200	40	0.5 ⁺⁺⁺
single b -jet	200	235	180	25	25	0.35 ⁺⁺⁺
single jet	370	460	400			0.25
large- R jet	470	500	300	40	40	0.5
four-jet (w/ b -tags)		45 [†] (1-tag)	65(2-tags)		20	0.1
four-jet	85	125	100	100		0.2
H_T	700	700	375	50	10	0.2 ⁺⁺⁺
E_T^{miss}	150	200	210	60	5	0.4
VBF inclusive			2x75 w/ ($\Delta\eta > 2.5$ & $\Delta\phi < 2.5$)	33	5	0.5 ⁺⁺⁺
B -physics ⁺⁺				50	10	0.5
Supporting Trigs				100	40	2
Total				1066	338	10.4

図 2.9 高輝度 LHC におけるトリガーメニューの例 [9]。解析に利用される典型的なオブジェクトに対して L0 Trigger および Event Filter でのトリガーレートが分配されている。L0 Trigger レートは Run 3 の約 10 倍、Event Filter でのレートは約 6 倍に増強される。

2.3.1 TGC トリガーのコンセプト

衝突点からエンドキャップ方向 ($1.05 < |\eta| < 2.4$) に飛来するミューオンは、エンドキャップトロイド磁石で曲げられ TGC 検出器に入射する。TGC 検出器の各層ではワイヤー、ストリップによる 2 次元読み出しで、TGC を通過したミューオンの (R , ϕ) 座標を検出する。TGC 検出器は z 方向に 3 つのステーションを持っており、ステーション間のコインシデンスをとることでミューオンの 3 次元飛跡を再構成し、それをもとに運動量を概算する。より具体的な運動量概算手法の概要を図 2.10 に示す。

TGC 検出器はエンドキャップトロイド磁場の外側に位置するため、ミューオンは検出器を直線的に通過する。TGC 検出器では、3 つのステーションのヒットから再構成したミューオンの飛跡 (図中の実際の飛跡) と、M3 のヒット点 (ピボット) と衝突点を直線的結んだ無限運動量飛跡のなす角を分別変数に利用して、 p_T を概算する。具体的に Run 3 のロジックでは、ミューオンが実際に残したヒット点と無限運動量飛跡の M1 および M2 の交点との位置の差分 (dR , $d\phi$) を利用する^{*2}。特に、ミューオンはエンドキャップトロイド磁場で主に R 方向に曲げられるため、 dR は p_T と強い相関を持った値となる。一方、 ϕ 方向にはあまり曲げられないため、衝突点から飛来するミューオンの $d\phi$ は p_T によらず小さい値を取る。そのため $d\phi$ は再構成したミューオン飛跡が、衝突点に由来するものであることを担保する目的で利用される。

^{*2} 高輝度 LHC-ATLAS 実験では実際の飛跡と無限運動量飛跡のなす角 ($\Delta\theta$, $\Delta\phi$) が利用される。本質的に等価なロジックである。

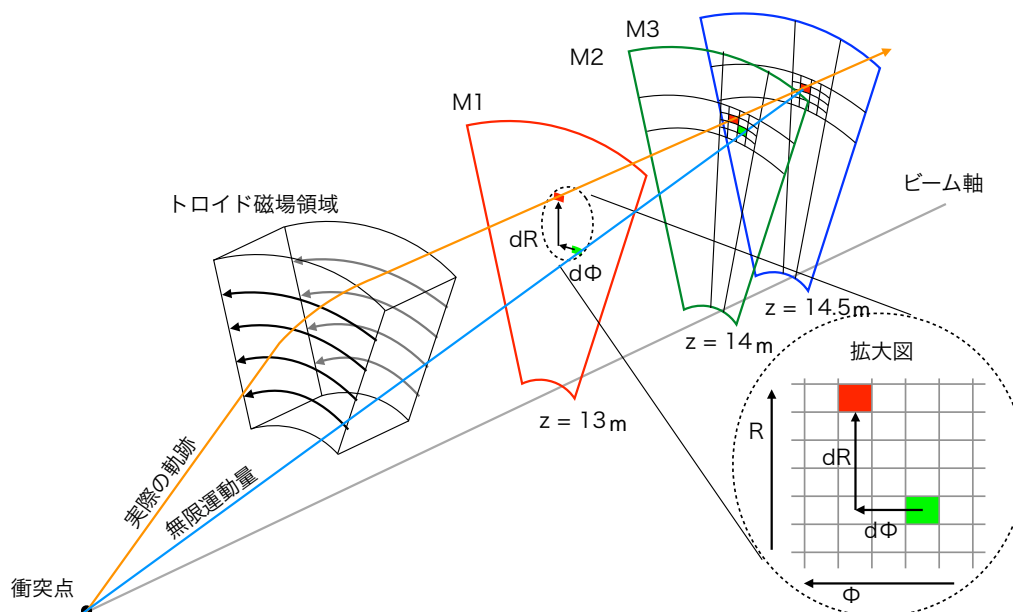


図 2.10 TGC におけるトリガーのコンセプト [10]。ミュオンが TGC に残したヒット点から再構成した実際の飛跡と無限運動量飛跡の位置の差分 ($dR, d\phi$) を利用して p_T を概算する。

2.1.2 節で述べたように、エンドキャップ領域に生成されるトロイド磁場は、 ϕ 方向にも R 方向にも均一ではない上、その大きさも一様でない。そのため、 $(dR, d\phi)$ と p_T の関係はミュオンの飛来する場所に依存する複雑な関数となり、代数的にも電気回路的にも求めるのは難しい。そこでシミュレーションを用いて、あらかじめ $(dR, d\phi)$ と p_T の関係性をまとめたテーブル (Look Up Table, LUT) を領域ごとに用意することで、複雑な計算をすることなく、高速で p_T を計算する。この手法をパターンマッチングと呼ぶ。

上述した TGC BW のヒットデータのみを用いたトリガー判定を行っていた Run 1 では、フェイクトリガーと呼ばれる、衝突点に由来しない荷電粒子に対して誤ってトリガーを発行してしまう事象が多く発生していた。図 2.11 に Run 1 でのミュオントリガーをパスしたイベント数の η 分布を示す。TGC がトリガーを担当する $1.05 < |\eta|$ の領域では、オフラインで再構成されたミュオンイベントよりはるかに多くのイベントがトリガーをパスしていることがわかる。この差分の多くがフェイクトリガーによるものであると考えられる。フェイクトリガーの主な原因として、陽子陽子衝突や、ハドロンカロリメーター内で生じた中性ハドロンが、エンドキャップトロイド磁石の支持構造体と相互作用し、荷電粒子を放出するケースが挙げられる。

この問題に対処するため、Run 2 以降のトリガーでは Inner Coincidence という新しいロジックが追加された。Inner Coincidence の概要を図 2.12 に示す。Inner Coincidence は、TGC 検出器で再構成したミュオン飛跡と、エンドキャップトロイド磁石内部に設置された検出器で再構成したミュオン飛跡のマッチングをとるもので、これによって衝突点に由来する粒子とそうでないものを見分けることが可能となる。さらに、NSW などの精密測定用の検出器で再構成された飛跡情報を組み合わせることで、TGC BW 単体での p_T 計算より精度を向上させることができる。

2.3.2 Run 3 での TGC 検出器システム

上記のトリガーコンセプトを実現する現行 (Run 3) の TGC TDAQ システムのブロック図を図 2.13 に示す。図の赤色の線はトリガー判定のための信号の流れを示しており、LHC の陽子バンチ交差に同期して Fixed latency で動作する (トリガーパスと呼ぶ)。青色の線は L1A 後のデータ読み出しのための信号の流れを示しており、LHC と

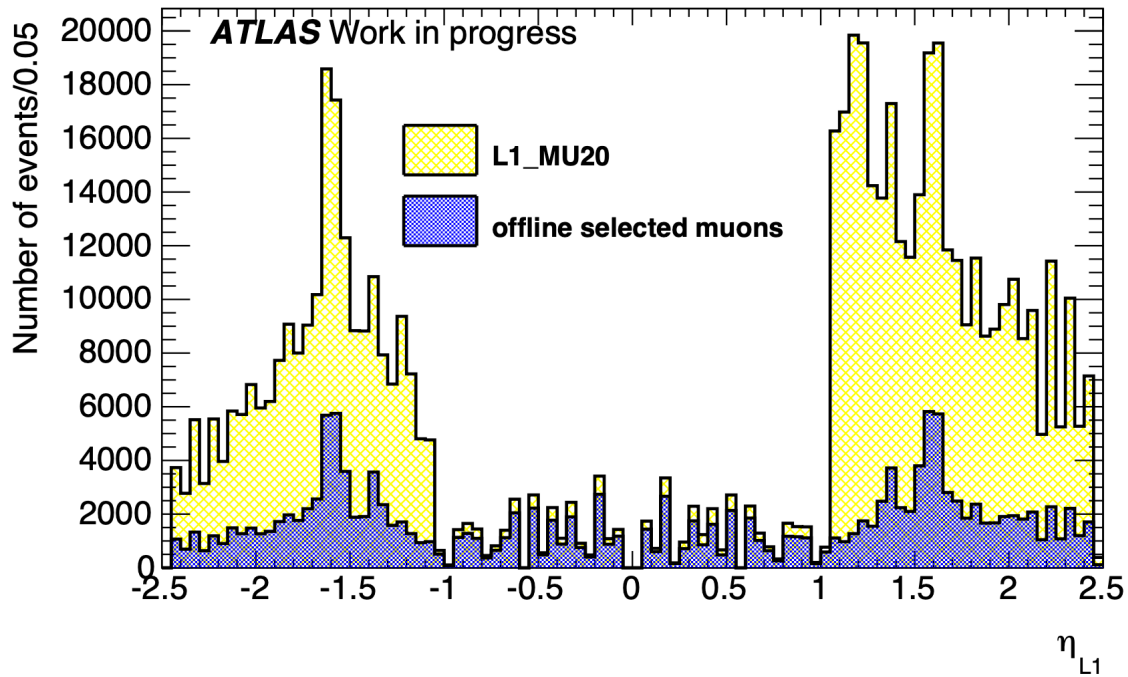


図 2.11 Run 1 における、L1 Muon をパスしたイベント数とオフラインで再構成されたミュオンイベント数の比較。TGC がトリガーを担当する $1.05 < |\eta|$ の領域で、オフラインで再構成されたミュオンイベント数より多くのイベントがミュオントリガーをパスしていることがわかる。この差分がフェイクトリガーによるものと考えられる。

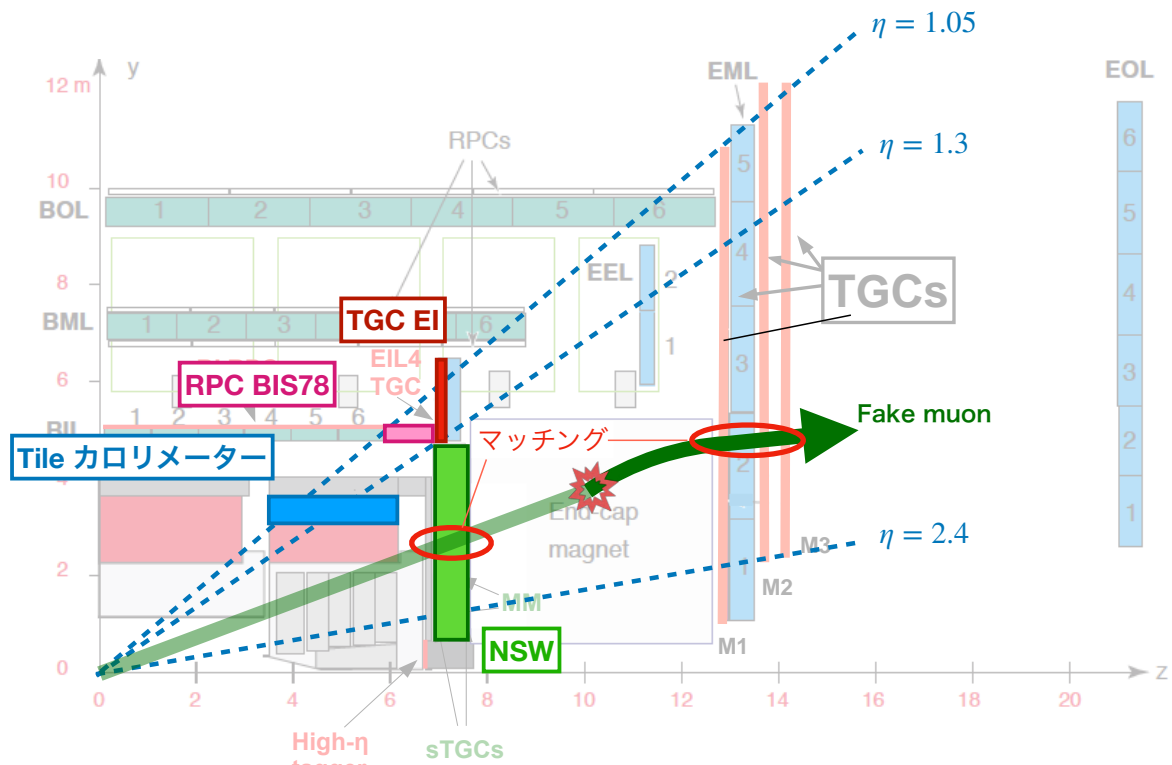


図 2.12 Inner Coincidence の概念図 [11]。エンドキャップトロイド磁場内部の検出器にヒットを要求することで、衝突点由来でない荷電粒子 (Fake muon) に対してトリガーを発行してしまう事象を削減する。

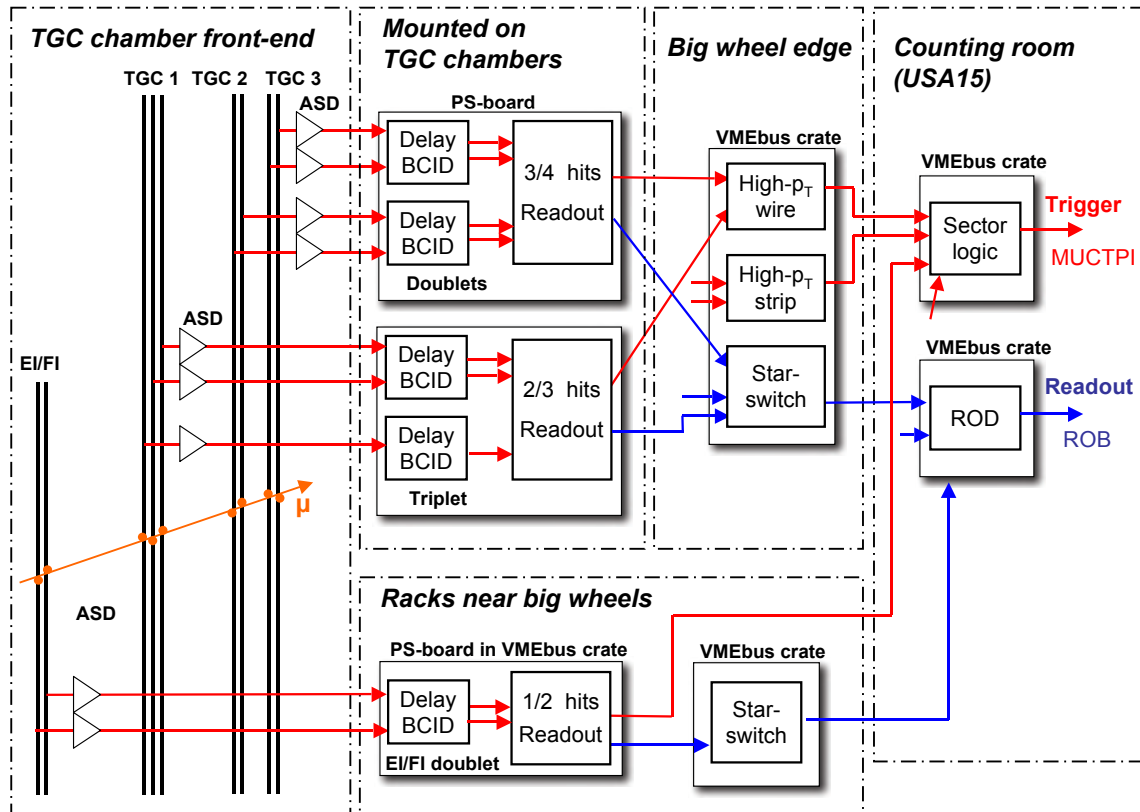


図 2.13 Run 3 における TGC TDAQ システム。赤線がトリガーパス、青線が読み出しパスを表す。TGC 検出器からの電流信号は ASD でデジタル信号に変換され、PS board に集められる。PS board では BCID が行われ、ワイヤーとストリップそれぞれでステーション内コインシデンスおよび M2-M3 コインシデンスが取られる。その後 HPT で M1-M3 コインシデンスがとられたのち、SL でワイヤー・ストリップ間コインシデンス及び磁場内部に位置する検出器とのコインシデンスがとられる。SL で得られた p_T 情報は MUCTPI を介して CTP に送られ、L1 トリガー判定が行われる。L1 トリガー判定を待っている間、ヒットデータは PS board でバッファーされ、L1A 信号が発行された場合には SSW、ROD を通して後段に読み出される。

は非同期に動作する (読み出しパスと呼ぶ)。TGC 検出器にミューオンが入射すると、ガスレイヤーに張られたワイヤーとストリップでアナログの電流信号が発生する。アナログ信号は TGC 検出器に直接取り付けられた Amplifier Shaper-Discriminator (ASD) に集められる。ASD は信号を電圧信号に変換し増幅した後、閾値電圧との比較によりデジタル信号を生成する。デジタル信号は後段の PS board 上に搭載された Patch-Panel ASIC (PP ASIC) において、どの陽子バンチ交差に由来するものか識別される (BCID)。この後から信号はトリガーパスと読み出しパスに分けられる。

トリガーパスは、Slave Board (SLB) ASIC、High-pt (HPT)、Sector Logic (SL) で構成される。SLB ではまずワイヤーとストリップ、それぞれでステーション内コインシデンスがとられる。また M2、M3 の間のコインシデンスがとられ、 $(dR_{23}, d\phi_{23})$ が計算される。その後 HPT で M1、M3 間のコインシデンスがとられ、 $(dR_{13}, d\phi_{13})$ が計算される。SL では M3 におけるヒット位置および $(dR_{13}, d\phi_{13})$ をもとにワイヤー・ストリップ間のコインシデンスがとられ、 p_T が概算される。また Inner Coincidence により、背景事象の除去およびより精度の高い p_T 計算が行われる。SL で再構成された入射位置と p_T 情報は MUCTPI に送られ、CTP へ渡される。CTP でトリガー条件を満たしたバンチ交差には L1A が発行される。

読み出しパスは SLB、Star SWitch (SSW)、ReadOut Driver (ROD) というパスを経由して後段へ渡される。PP

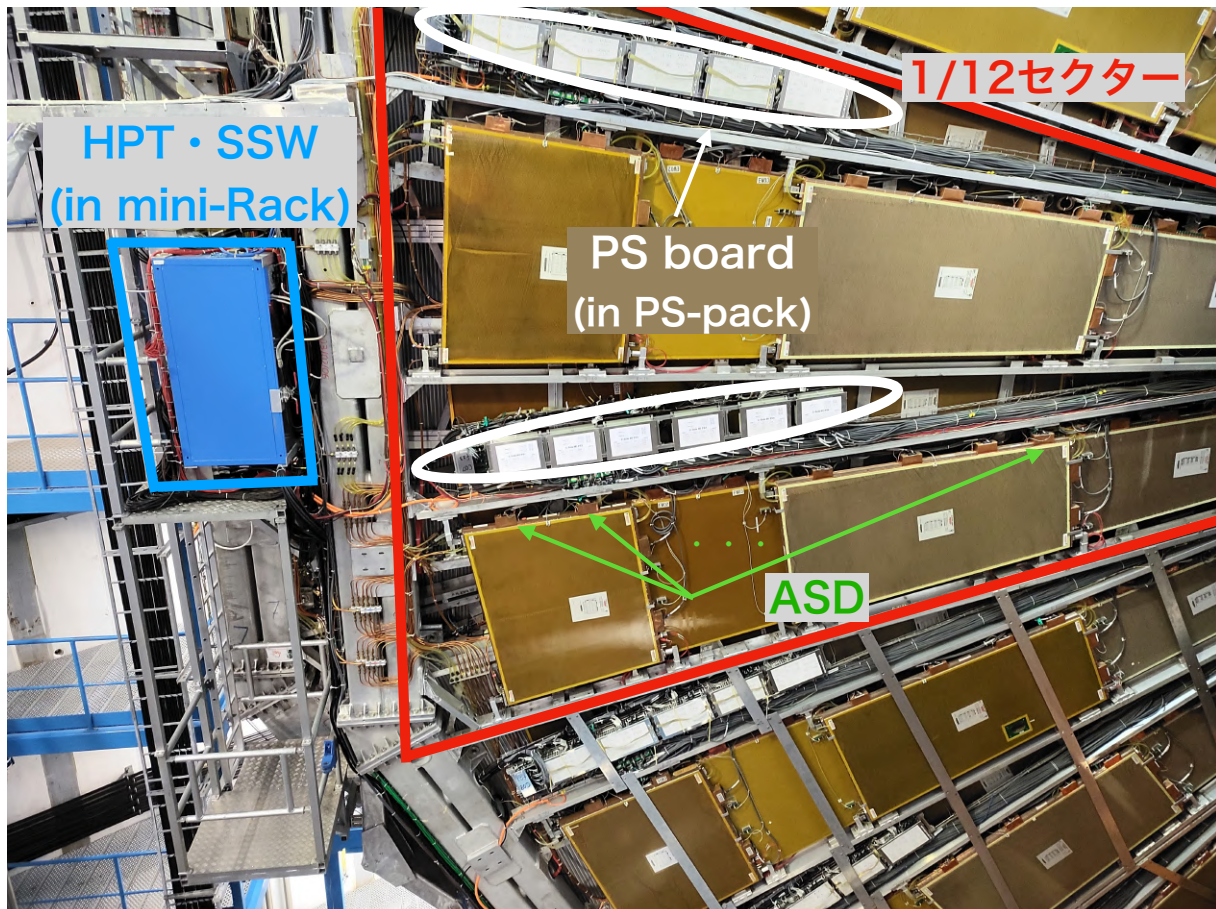


図 2.14 TGC エレクトロニクスの設置場所。ASD は TGC チェンバーに直接取り付けられたアダプタボードに設置される。PS board は検出器付近の PS-pack に収納される。HPT および SSW は BW の外側に設置された Mini-Rack 内の VME クレイトに収められる。

ASIC で同期された信号は、トリガー判定が完了するまで SLB 内の L1 Buffer に保存される。SLB では最大 128 イベント分の信号をバッファリングすることが可能である。前節で述べた通り、L1 Trigger は Fixed latency Scheme を採用しており、L1 Buffer にデータが入ってからそのイベントに L1A が出されるまでの時間 (L1 latency) は固定されている。そのため SLB は L1A を受信した後、L1 latency だけ前のデータを読み出すことで、正しくデータを読み出すことができる。SLB から読み出されたデータは SSW でゼロサプレスと呼ばれるデータ圧縮が行われ、イベント毎にパッケージされる (Event Building)。ROD は 1 つのセクター内の全ての SLB-SSW から送られたデータ集約し、さらに後段の ReadOut System (ROS) へとデータを送信する。

これらの TGC エレクトロニクスのうち、ASD、PS board、HPT、SSW は ATLAS 実験室内に設置され、フロントエンドエレクトロニクスと呼ばれる。図 2.14 にフロントエンドエレクトロニクスの設置場所を示す。ASD は TGC チェンバーに直接取り付けられたアダプタボードにマウントされる。PS board は 2 枚ごとにアルミニウムで作られたケース (PS-pack) に収納され、TGC チェンバー付近に設置される。HPT および SSW は BW の外側にある Mini-Rack 内の VME クレイトに収められる。

SL、ROD 以降のエレクトロニクスは USA15 という ATLAS 回路室内に設置され、バックエンドエレクトロニクスと呼ばれる。HPT と SL および SSW と ROD は Run 3 では G-link と呼ばれる旧式の光シリアル通信規格で接続される。G-link におけるシリアルデータ転送レートはおよそ 800 Mbps である。

次に TTC 系について述べる。Level1 Buffer より前の読み出し回路およびトリガー回路は、Fixed latency scheme

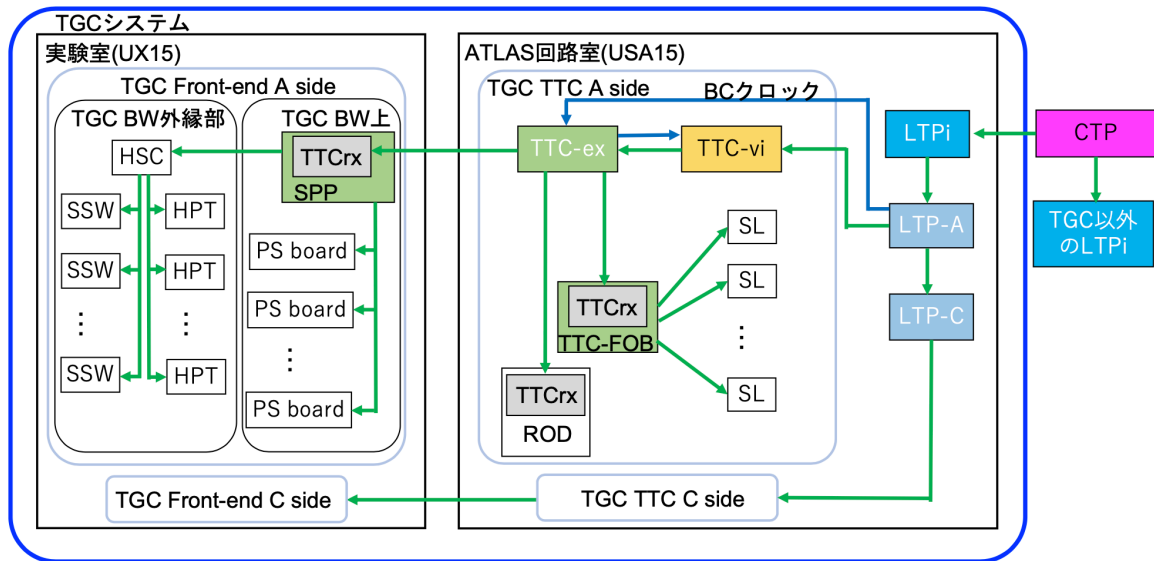


図 2.15 Run 3 TGC システムにおける TTC システムの概要 [3]。TTC 信号は CTP から各検出器のサブシステムの LTP に分配され、TTCvi、TTCex、TTCrx を介して PS board、HPT、SSW、SL へと分配される。

を実現するため LHC の陽子バンチ交差と同期して動作する必要がある。この同期のために、各検出器と LHC を同期させるシステムを Timing, Trigger and Control (TTC) システムとよび、そのために配布される信号を TTC 信号と呼ぶ。TTC 信号には陽子バンチ交差と同期した 40.079 MHz の LHC バンチ交差クロック (LHC クロック) や L1A 信号などが含まれる。Run 3 における TTC 系の概要を図 2.15 に示す。TTC 信号は CTP から各検出器サブシステムの Local Trigger Processor (LTP) に配られる。その後 TTC 信号は TTCvi、TTCex、TTCrx と呼ばれる TTC 専用モジュールおよび専用線を通じて、PS board、HPT、SSW、SL へ分配される。

以上に述べた Run 3 の TDAQ システムでは、高輝度 LHC-ATLAS 実験のトリガーおよび読み出し性能の要求を満たすことができない。この限界は、主に SLB に設置される L1 Buffer の容量と、ATLAS 実験室と回路室間の帯域幅に由来している。Run 3 では、L1 Buffer は最大 128 イベント分のデータしか保存できないため、高輝度 LHC-ATLAS 実験の L0 latency の要求である $10 \mu\text{s}$ (400 バンチ) の間データを保持することができない。さらに、1 MHz の初段トリガーレートに対応するデータ量を読み出すためには、現行システムの ATLAS 実験室と回路室間の帯域幅では不十分である。

2.3.3 高輝度 LHC-ATLAS 実験での TGC 検出器システム

前節で述べた問題を解決するため、高輝度 LHC-ATLAS 実験では TGC 検出器の電子学を大幅にアップグレードする。主な変更点は PS board で BCID したすべてのヒット信号を、ヒットの有無に関わらず、すべての BC について SL に送る点である。Run 3 のトリガーパスでは、検出器からの信号は SLB や HPT でのコインシデンスを通じて、段階的にデータ量を削減しながら回路室へ送られた。一方、高輝度 LHC-ATLAS 実験では、実験室と回路室の間で新しく高速シリアル通信技術を導入し、帯域幅を大幅に拡張することで、PS board が処理した全チャンネル分のヒットビットマップをすべて、回路室の SL へ送信することができる。回路室では、ボードサイズや放射線耐性に対する制約が少なく、SL には大規模な FPGA を搭載することができる。その結果、CTP から L0A が発行されるまでデータを保管しておく L0 Buffer も余裕を持って実装することができ、 $10 \mu\text{s}$ の L0 latency に対応することが可能となる。また SL に 1 つのトリガーセクター内の全ての PS board からのヒット信号を集約させることで、TGC BW 7 層分のヒットデータを利用した、より包括的なトリガーアルゴリズムを実現する。高輝

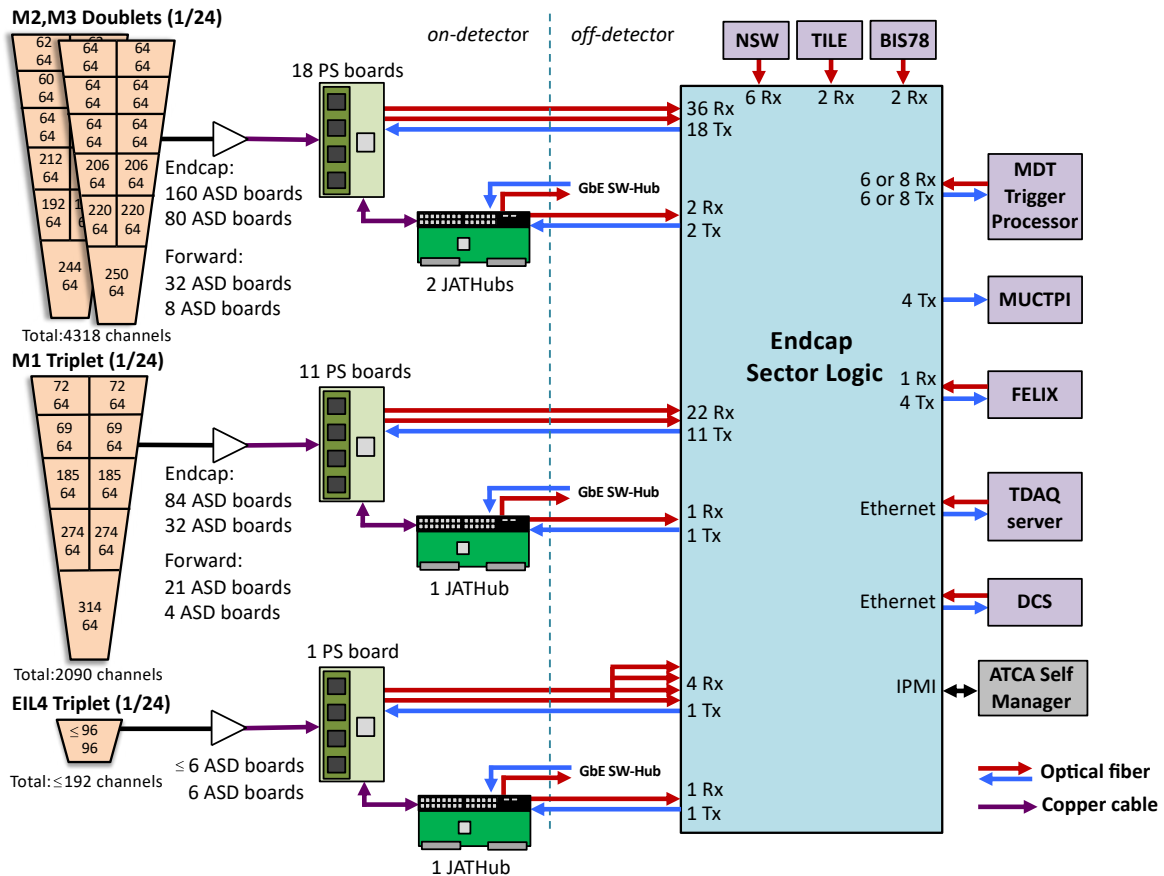


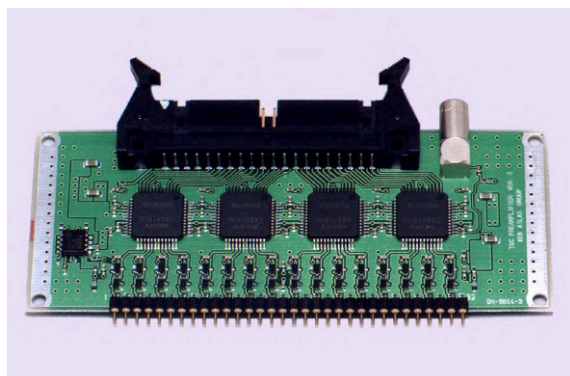
図 2.16 高輝度 LHC-ATLAS 実験における TGC 検出器システムの概要 [5]。TGC からの電流信号は ASD でデジタル信号に変換され、PS board 上の PP ASIC で BCID される。PP ASIC からのヒット信号はヒットの有無に関わらず、PS board FPGA から光ファイバーを通じて SL へ送られる。1 枚の SL は 1 つの 1/24 セクターを担当し、PS board からのヒット信号に加えて TGC EI、NSW、RPC BIS78、Tile カロリメーターの情報も利用して p_T を判定する。その後 SL は再構成した飛跡情報を MDT Trigger Processor に送信し、より精度の高い p_T を計算する。SL からのトリガー出力は、MUCTPI を介して CTP へ送信され、L0 トリガー判定が行われる。CTP で L0 トリガー判定を待っている間、衝突データは SL でバッファされ、L0A が出されたものは FELIX を通して後段へ読み出される。TTC 信号は CTP から FELIX を介して各 SL に分配され、PS board へのコントロール信号に乗せられ PS board に分配される。JATHub はデータパスとは独立したモジュールで、PS board の制御を担当する。

度 LHC-ATLAS 実験でのトリガーロジックの詳細は 3.1 節で説明する。

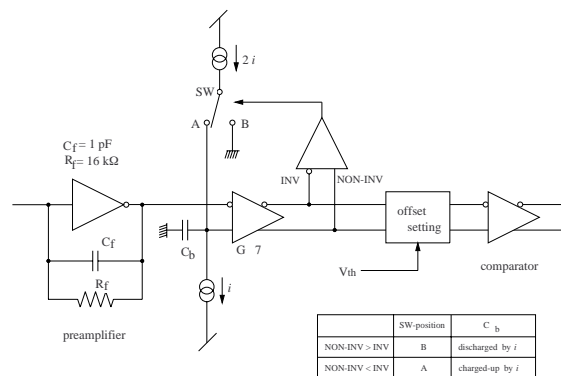
全体像

図 2.16 に高輝度 LHC-ATLAS 実験での TGC 検出器エレクトロニクスの概要を示す。ASD は現行システムのものを引き続き使用する。一方で、Run 3 で使われていた PS board、SLB、HPT、SL はすべて撤去され、新しく Primary Processor board (PS board)、JTAG Assistance Hub (JATHub)、Endcap Sector Logic (SL) が設置される。PS board は Run 3 と同様に PS-pack に格納され、TGC 検出器付近に設置される。JATHub は Mini-pack 内の VME クレーンに設置される。SL は USA15 の ATCA クレーンに設置される。1 つの SL は TGC の 1/24 セクターを担当し、このセクターを担当する最大 31 枚の PS board からヒットデータを受け取る。

トリガー回路および読み出し回路は主に SL 上の FPGA に集約される。TGC で生じる電流信号は ASD でデジタル信号に変換された後、PS board 上の PP ASIC で BCID される。PP ASIC からのヒット信号は、PS board 上



(a) ASD ボードの写真



(b) ASD チップの回路ブロック図

図 2.17 ASD の概要 [12]。(a) に ASD ボードの写真を示す。1 枚の ASD ボードには 4 枚の ASD チップが搭載されており、1 つにつき 4 チャンネル分のヒット信号を処理する。(b) に ASD チップの回路図を示す。ASD チップは前段増幅回路、差動電圧増幅回路、コンパレータから構成される。

の FPGA でまとめられ、光ファイバーを通じて SL に送信される。SL は、PS board から送られる TGC BW 7 層の情報に加え、TGC EI、NSW、RPC BIS78、Tile カロリメーターの情報も用いてミューオンの p_T を概算する。その後 SL は MDT Trigger Processor (MDTTP) にミューオン飛跡候補を送信し、よりよい分解能で p_T を再構成する。SL のトリガー出力は MUCTPI を通じて CTP へ送信され、L0 トリガー判定が行われる。CTP でのトリガー判定を待っている間、ヒットデータは SL 内でバッファされ、LOA が発行されたバンチ交差事象のデータは FELIX を通じて後段へ読み出される。

TTC 信号は CTP から FELIX を通じて各 SL に分配され、PS board へのコントロール信号に乗せて各 PS board に分配される。JATHub はデータバスと独立しており、PS board の制御を担当する。このシステムではバックエンドの SL とフロントエンドの PS board が光リンクのみで接続されており、トリガー、データ読み出し、コントロール、TTC の分配をコンパクトに実現している。

以下に TGC 検出器における読み出し、トリガー、制御に関わるそれぞれのエレクトロニクスとその役割を説明する。

Amplifier-Shaper-Discriminator (ASD)

ワイヤー、ストリップからの電流信号は TGC のチェンバーに直接取り付けられた、Amplifier-Shaper-Discriminator (ASD) ボードで電圧信号に変換された後に増幅され、閾値電圧との比較による信号識別を経て、最終的に LVDS 規格のデジタル信号へ変換される。図 2.17 に ASD の概要を示す。ASD はチャージアンプである前段増幅器 (Preamplifier)、差動電圧増幅回路、コンパレータから構成されている。前段増幅回路では、0.8 V/pC のゲインを用いて、16 ns の時定数で電流信号を電圧信号に変換する。その信号は差動電圧増幅回路で 7 倍に増幅され、コンパレータで閾値電圧を超えている時間に対応するパルス長の LVDS 信号に変換される。この閾値電圧は PS board から設定できるようになっている。また ASD には TGC のチャージ出力をエミュレートする Test Pulse 源が実装されており、ASD 以降のトリガー・データバスの試験に利用される。1 枚の ASD ボードには 4 枚の ASD チップが搭載されており、1 枚あたり 4 チャンネル、全部で 16 チャンネルの信号を処理する。TGC の読み出しチャンネルはおよそ 32 万チャンネルであるため、システム全体ではおよそ 2 万枚の ASD ボードが設置される。

Primary Processor board (PS board)

ASD ボードからのヒット信号は、次に PS board に送られる。PS board は PP ASIC と Xilinx 社製の Kintex-7 FPGA という 2 種類の集積回路を搭載している。PP ASIC は可変遅延機能により各 ASD からの入力信号の到着時間を揃え、その信号がどの陽子バンチ交差に由来するかを識別する (BCID)。PS board FPGA は BCID されたヒット信号を高速光シリアル通信に乗せて SL へ転送する。図 2.18 に PS board の最終試作機の写真とブロック図を示す。PS board には 8 つの PP ASI が搭載され、そのうち 4 つはメザニカードに乗せられる。1 つの PP ASIC は 2 台の ASD と接続され、合計で 32 チャンネルの信号を処理する。PS board FPGA は 8 つの PP ASIC からのヒット信号をまとめて、合計 256 チャンネル分のヒットビットマップを SL に送る。

PS board はボード上に DAC (Digital Analog Converter)、ADC (Analog Digital Converter)、クロックジッタークリーナー、QSPI フラッシュメモリーなどの電子素子を搭載している。DAC は ASD のコンパレーターにアナログの閾値電圧を供給する。PS board FPGA から閾値電圧の大きさを設定することができる。ADC は DAC から供給される閾値電圧をモニターする。クロックジッタークリーナーは、PS board FPGA がシリアルデータから再構成した LHC バンチ交差クロックのジッターを低減する。QSPI フラッシュメモリーは不揮発性のメモリーで、ボードの電源が落とされた場合でも書き込まれた値を保持することができる。これを利用して、PS board FPGA のファームウェアや、ボードごとに最適化された PP ASIC の遅延値などのパラメーターを保存する役割を果たす。また、インターフェイスとして、電気信号を光信号に変換する SFP+ モジュール、Cat-6 ケーブルのコネクターである RJ45 を有しており、それぞれ SL と JATHub に接続される。

PS board は TGC システム全体で 1434 枚設置され、それぞれの個体の量産が 2024 年から始まる。本研究では、量産される各個体にハードウェアの不具合がないことを検証する品質保証試験の設計、およびそれに向けた機能開発を行なった。これに関する詳細は 5 章で議論する。以下に PP ASIC と PS board FPGA の役割について詳しく説明する。

Patch-Panel ASIC (PP ASIC)

図 2.19 に PP ASIC のブロック図を示す。PP ASIC は主に可変遅延回路と陽子バンチ識別回路で構成される。PP ASIC が ASD から受け取る LVDS 信号の入力時間には、チャンネルごとに $O(10\text{ ns})$ のばらつきが存在する。これは、衝突点から検出器までの飛行時間 (Time-of-Flight) や、ASD から PP ASIC までの LVDS ケーブルの長さがチャンネルごとに異なるためである。また、同一チャンネル内でもイベントごとに信号の到着時間が 20 ~ 30 ns 程度変動する (図 2.20)。これは、ミューオンの入射位置に応じて、電荷が検出される位置から ASD までの距離が異なり、結果として信号の伝播遅延や電荷のドリフト時間にばらつきが生じるためである。

このばらつきを揃えるために、各 ASD ごとに固有の遅延を加えるのが可変遅延回路である。可変遅延回路は 1 ns 以下の刻み幅で、最大 45 ns までの遅延をかけることができる。PP ASIC に一番遅く到着する ASD からの信号の立ち上がりに、他の ASD からの信号の立ち上がりを合わせるように遅延パラメーターが設定される。この遅延パラメーターは PS board の FGPA から設定可能である。

可変遅延回路でタイミングが揃えられた後、信号は陽子バンチ識別回路に送られる。陽子バンチ識別回路は、PP ASIC に入射するデジタル信号の立ち上がりを検出し、その信号がどの陽子バンチ交差に由来するのかが識別する。陽子バンチ回路の概念図を図 2.21 に示す。前述の通り、同じチャンネル内であってもヒット信号の到着時間はイベント毎に 20 ~ 30 ns 程度の幅を持ち、この時間幅のうちに来るヒット信号には同じ BCID を付与する必要がある。そのため、同じ BCID を付与する時間幅 (有効ゲート幅) を ASD ごとに設定できるようになっており、信号到着時間幅が 25 ns を超える場合に対応するため、有効ゲート幅は 25 ns 以上に設定できるようになっている。その場合、2 つの有効ゲートが重なる時間が存在するが、このタイミングに入射した信号には 2 バンチ分の信号が出力される。

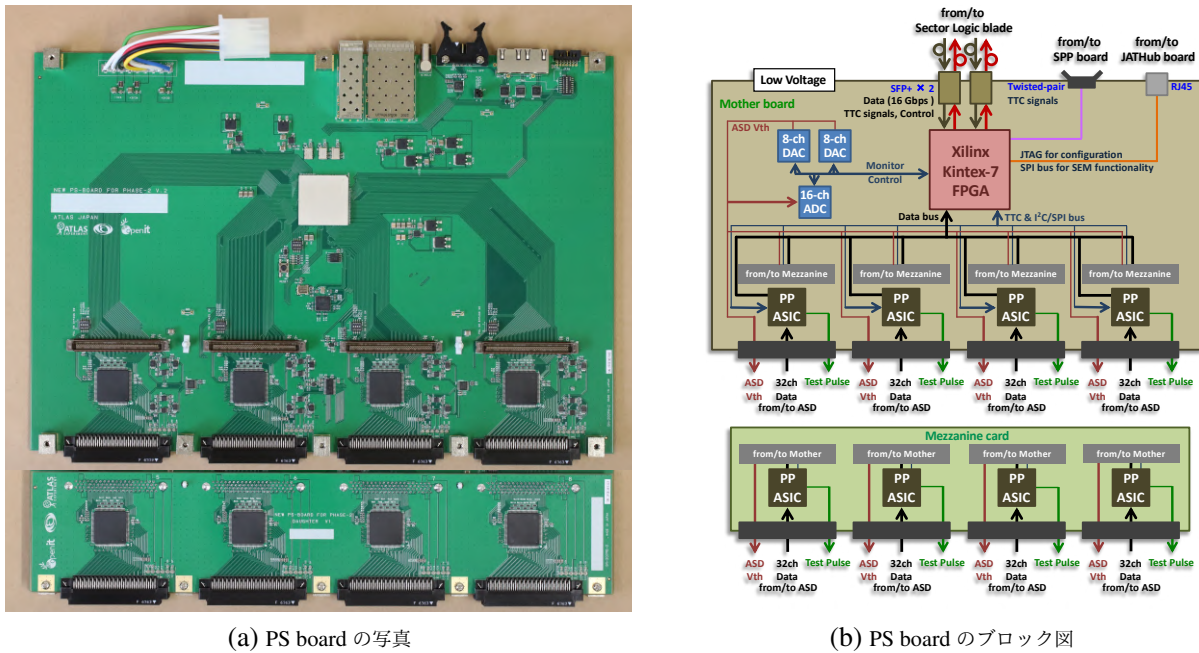


図 2.18 PS board の概要。1 枚の PS board は、1 つの FPGA と 8 つの PP ASIC (メインボードに 4 個、メザニンボードに 4 個) を搭載している。1 つの PP ASIC は 2 台の ASD と接続され、合計で 32 チャンネルの信号を処理する。また、PS board は DAC、ADC、クロックジッタークリーナー、QSPI フラッシュメモリーなどの素子を搭載しており、これらは PS board FPGA からコンフィギュレーションすることができる。

有効ゲート幅も PS board の FPGA から設定することができる。

PS board FPGA

PS board FPGA は SL にヒットデータを送信することに加えて、PS board 上の各素子の制御/監視および LHC バンチ交差クロックの再構成を担当する。PP ASIC により LHC バンチ交差クロックと同期された 256 チャンネルのヒット信号は、PS board FPGA でまとめられ、光ファイバーを通じて SL に転送される。PS board から SL に送信されるデータフォーマットを図 2.22 に示す。256 チャンネルのヒットデータに加え、64 bit のヘッダーが付与され、合計 320 bit のデータが 25 ns おきに送信される。320 bit のデータは 2 本の光ファイバーに分けられ、32 bit ごとにワードという単位でまとめられる。ワード 0 ではヘッダーが、ワード 1~4 にはヒットデータが詰められる。データ転送には、高速シリアル通信対応のトランシーバーの種である、GTX トランシーバーが用いられ、FPGA 内のパラレルデータをシリアルデータに変換する際には 8b10b コーディングが用いられる。そのため、1 本の光ファイバーのラインレートは $160 \text{ bit} \times 10/8 \times 40 \text{ MHz} = 8 \text{ Gbps}$ となる。

PS board FGPA は SL からのコントロール信号を 1 本の光ファイバーを通じて受け取る。コントロール信号のデータフォーマットを図 2.23 に示す。SL はワード 2 と 3 で定義された Address、Data、Command を利用して PS board FPGA 内のレジスタを操作する。PS board FPGA と QSPI フラッシュメモリーは SPI バスで接続されており、SL が FPGA を介して SPI バスを制御することで、QSPI フラッシュメモリーにデータを書き込むことができる。この通信パスを利用して、PP ASIC の遅延パラメーターや有効ゲート幅、ASD に供給する閾値電圧などの制御用パラメーターを QSPI フラッシュメモリー上に保存することができる。

PS board FPGA は QSPI フラッシュメモリーに保存された制御用パラメーターを読み取り、PP ASIC、ASD に自動で分配する。これを自立型制御機構と呼ぶ。この機構により、1434 枚の PS board には同じファームウェア^{*3}

^{*3} 正確には PS board から SL に送るデータフォーマットに 3 種類のバラエティが必要であるため、3 種類のファームウェアを使い分けるこ

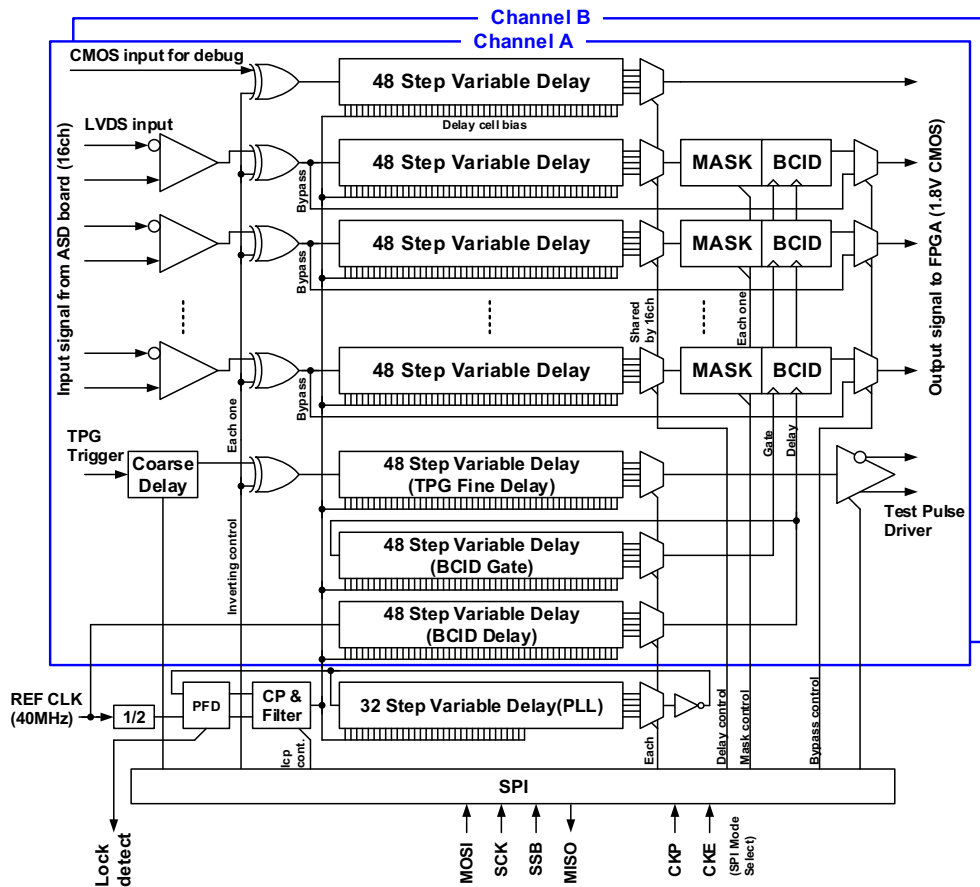


図 2.19 PP ASIC 回路のブロック図 [13]。Channel A と Channel B がそれぞれ 1 つの ASD ボードに対応し、1 つの PP ASIC は 32 チャンネルの LVDS 信号を受け取る。PP ASIC は主に可変遅延回路と陽子パンチ識別回路で構成され、この 2 つを用いて BCID を行う。

を使用し、ボード毎に異なるパラメーターの設定は各個体が自動で行う、という次世代的な制御モデルを実現している。PS board FPGA には PS board 上の素子の状態を監視するための機構も備わっており、DAC に分配した設定値、ADC、xADC*4 のモニター値、光トランシーバーのロック信号を定期的に読み出し、SL に送信する。これにより SL はフロントエンドの PS board の状態を常に把握し、異常が生じた際には瞬時に対応することができる。

TTC 信号もこのコントロール信号に乗せられて PS board に分配される。PS board FPGA は、ワード 0 で定義される Comma ワードを使用して、シリアルデータから LHC バンチ交差クロックを再構成する。この時、適切な BCID を実現するために、SL や PS board のリセットや再コンフィギュレーションによってクロックの位相関係が変化しない、固定位相でのクロック再構成を行う。PS board で再構成された LHC バンチ交差クロックは、PS board 間で位相を揃えるための適切な遅延が加えられた後、ジッタークリーナーを経由して FPGA、GTX トランシーバー、PP ASIC へ分配される。

さらに、PS board は放射線損傷に対する堅牢性を実現するシステムを搭載している。PS board は実験室に置かれるため、FPGA は電子回路上のメモリのビットが反転してしまう Single Event Upset (SEU) などの放射線損傷を受ける可能性がある。先行研究 [17] によると、1 つの PS board FPGA では 3 時間に一回程度 SEU が発生す

とになる。

*4 Xilinx Analog-Digital Converter、FPGA 内部の温度、電源電圧、外部からのアナログ信号を監視するためのモジュール。PS board に供給される 3.3 VD (デジタル)、+3 VA (アナログ)、-3 VA をモニターする

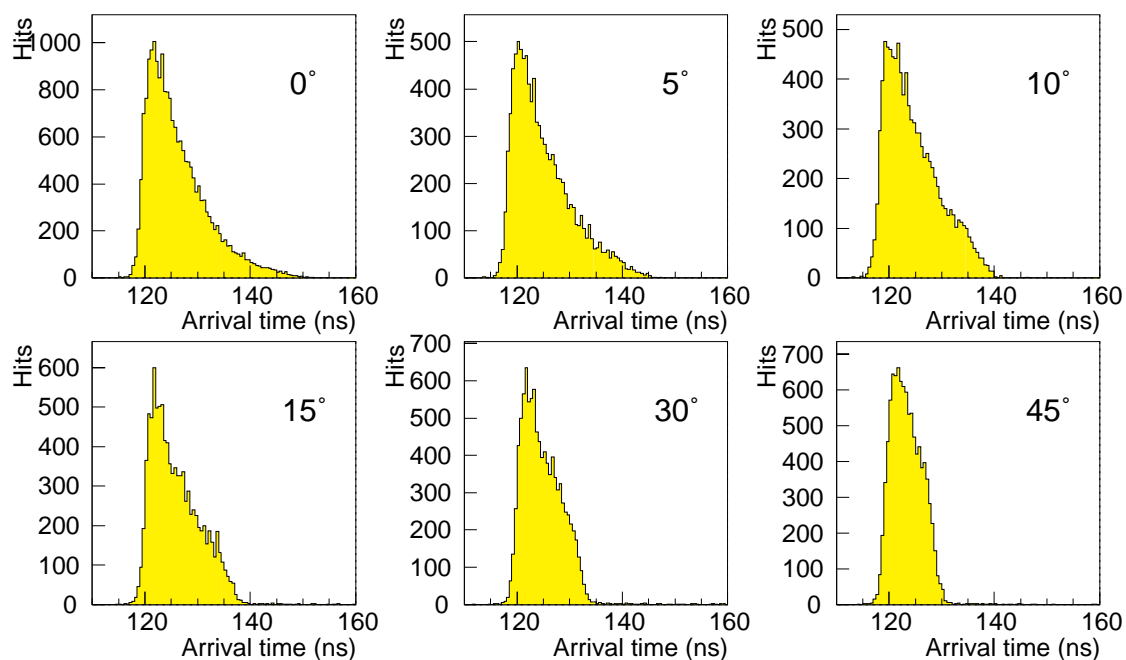


図 2.20 ミューオンが TGC 検出器に入射してから、ASD に信号が到着するまでの時間分布 [14]。時間分布はミューオンが入射する角度に依存して、20 ~ 30 ns の幅を持つ。

る。FPGA には修復可能な SEU (1 ビットエラーおよび隣接する 2 ビットエラー) を自動的に修復する Soft Error Mitigation Controller (SEM) が実装されている。また修復不可能な SEU (隣接しない 2 ビットエラーおよび 3 ビット以上のエラー) が生じた際には、PS board の制御を担当する JATHub に対して救難信号を送り、JATHub に FPGA の再コンフィギュレーション用の信号を送信させることでこれに対処する。このように、放射線損傷に対して堅牢なシステムを実現することで、データ取得時のデッドタイムを最小限に抑えることができる。

JTAG AssisTance Hub (JATHub)

JATHub はデータパスとは独立した、PS board 制御用の回路である。概要を図 2.24 に示す。JATHub は Xilinx 社製の Zynq-7000 SoC をメインドライバーとして搭載する。Zynq はプロセッサ部分である Processing System (PS) と、FPGA 部分である Programmable Logic (PL) で構成されている。PS 部分では Linux などのカーネルを立ち上げることが可能で、C 言語などで記述されたアプリケーションを実行して FPGA を操作することができる。

JATHub はインターフェイスとして、PS board と LVDS 通信するための RJ45 コネクタと、イーサネット通信を行うための SFP+ モジュールを搭載している。1 枚の JATHub は最大で 11 枚の PS board と接続可能であり、それぞれ 2 本の Cat-6^{*5}ケーブルで接続される。一本の Cat-6 ケーブルは JTAG 線と呼ばれ、JATHub を起点に遠隔で PS board のファームウェアを書き込む際に利用される。もう 1 本の Cat-6 ケーブルは Recovery/Monitor 線と呼ばれ、PS board の放射線損傷に対する回復、および LHC バンチ交差クロックのモニターに利用される。

図 2.25 に JATHub の Recovery/Monitor 線の概要を示す。PS board FPGA で自己修復不可能な SEU が生じた場合、Recovery Request 線 (RcvB 線) を通じて救難信号が出される。JATHub は救難信号を受け取ると、PS board に FPGA 再コンフィギュレーション用の Program 信号 (PROGB 線) を送信する。この一連の手続きをリカバリー手続きと呼ぶ。Monitor 線 (MON) は PS board が再構成した LHC バンチ交差クロックを JATHub に送信するた

^{*5} Category-6 twisted pair cable。4 対 8 線の動線で構成されており、4 種類の差動信号線を束ねている。

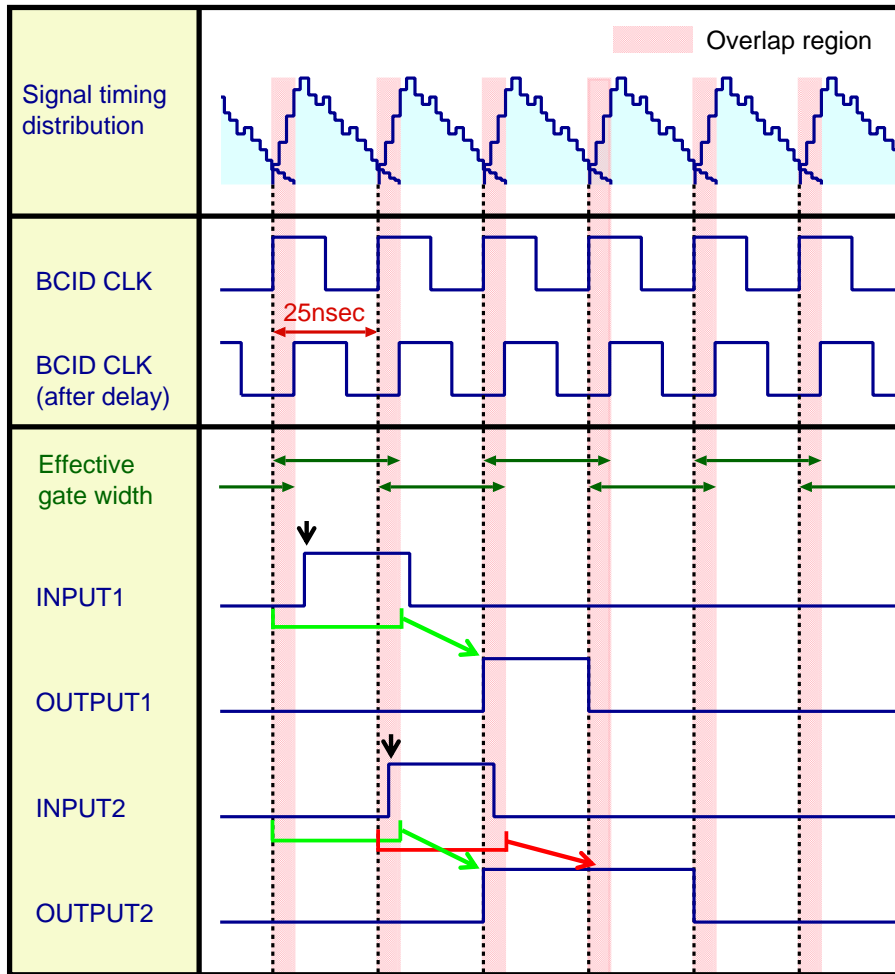


図 2.21 陽子バンチ識別回路のタイミングチャート [15]。BCID CLK の立ち上がりのタイミングで INPUT が high であった時、次の BCID CLK から 1 BC 分デジタル信号が出力される (図の INPUT1 の場合に対応)。BCID CLK の立ち上がりのタイミングが有効ゲートが重なる領域に来た場合には、2 BC 分のデジタル信号を出力する (図の INPUT2 の場合に対応)。

SFP+ 0

Word (32 bit)	Forth byte [31:24]						Third byte [23:16]	Second byte [15:8]	First byte [7:0]		
Word-0	Status bit [1:0] = 2'b11	xADC read	ADC read	DAC read	DAC write	QSPI status	Si5345 Lock	PP ASIC status	SEM status [2:0]	BCID [11:0]	Comma = 0xbc
Word-1	32 bit hit data										
Word-2	32 bit hit data										
Word-3	32 bit hit data										
Word-4	32 bit hit data										

SFP+ 1

Word (32 bit)	Forth byte [31:24]				Third byte [23:16]	Second byte [15:8]	First byte [7:0]				
Word-0	PS data [7:0]				Data type [3:0]	BCID [11:0]	Comma = 0xbc				
Word-1	32 bit hit data										
Word-2	32 bit hit data										
Word-3	32 bit hit data										
Word-4	32 bit hit data										

図 2.22 PS board から SL へ送るデータフォーマット [16]。256 チャンネルのヒットデータに 64 bit のヘッダーが付与された、合計 320 bit のデータが 2 リンクに分けられ、25 ns おきに送信される。ヘッダーワードには主に PS board FPGA やその他の素子の状態を表すモニター用のデータが含まれる。

SFP+ 0													
Words (32-bit)	Forth byte [31:24]					Third byte [23:16]				Second byte [15:8]			First byte [7:0]
Word-0	0x0	Soft Reset	TPT	BCR	1b0	4b0	BCID (11:0)			Comma = 0xbc			
Word-1	8b0					DAC Bitmap [1:0]	SCL	SDA	1b0	CS Bitmap [10:0]			SPIRST SIRSTB SCLK SDI
Word-2	Address					FPGA TP pattern ID				Command			
Word-3	8b0					Data				Data			
Word-4	Footer = 0x0e0d					SL Full board ID				SL Full board ID			

図 2.23 SL から PS board へ送るコントロール信号のデータフォーマット [16]。Word 0 には TTC 信号、Word 1 にはビットバンギング用のデータ、Word 2、Word 3 には PS board FPGA 内のレジスタ操作用のデータが含まれる。

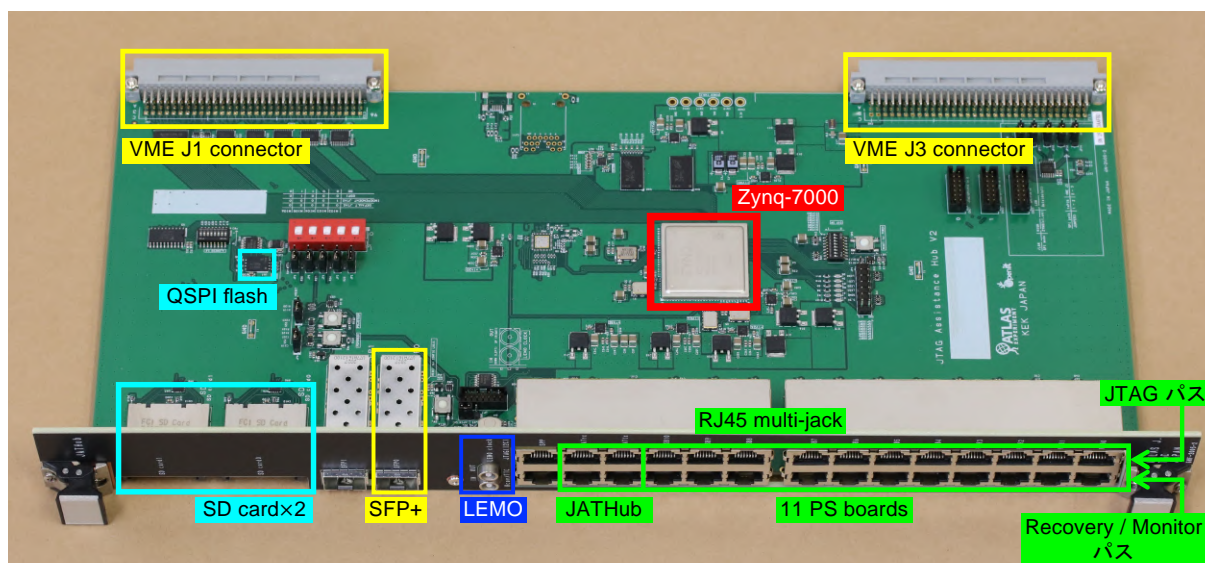


図 2.24 JATHub の写真 [16]。JATHub はメインドライバーとして CPU と FPGA が統合された SoC デバイスである Zynq-7000 を搭載している。PS board とのインターフェイスとして LVDS 通信用の RJ45 コネクタと、イーサネット通信用の SFP+ モジュールを持つ。Mini-Rack 内の VME クレイトに設置されるため、VME コネクタを有している。

めに利用される。JATHub は接続されるそれぞれの PS board で再構成されたクロックの位相をモニターし、その位相差を測定する。

PS board の量産試験について議論する 5 章では、JATHub の PS 領域と PL 領域には実験本番とは異なるシステムを実装し、JATHub をコンパクトな DAQ システムとして利用する。このシステムの詳細は 5.2 節で説明する。

Sector Logic (SL)

PS board から出力された TGC BW 全 7 層分のヒットデータは SL に集められる。SL は Virtex UltraScale+ FPGA という大規模 FPGA と Zynq UltraScale+ MPSoC の 2 種類の集積回路を搭載している。Virtex UltraScale+ FPGA のデバイスは SL FPGA と呼ばれ、PS board から受信したヒットデータを用いてトリガー計算を行う。また、SL FPGA は CTP から L0A が発行されるまでのデータのバッファリングおよび読み出しも担当する。Zynq UltraScale+ MPSoC は Virtex UltraScale+ FPGA や PS board のコントロールマスターとして機能する。

SL の第一試作機の写真を図 2.26 に示す。SL は Advanced Telecommunications Computing Architecture (ATCA) 規格のボードである。ユーザーは ATCA クレイトの Shelf manager から CERN で開発された Intelligent Platform Management Controller (IPMC) カードを介して、SL の電圧や温度のモニターや遠隔での電源操作を行うことができる。SL は外部とのインターフェイスとして電気信号を光信号に変換するための FireFly を送信用に

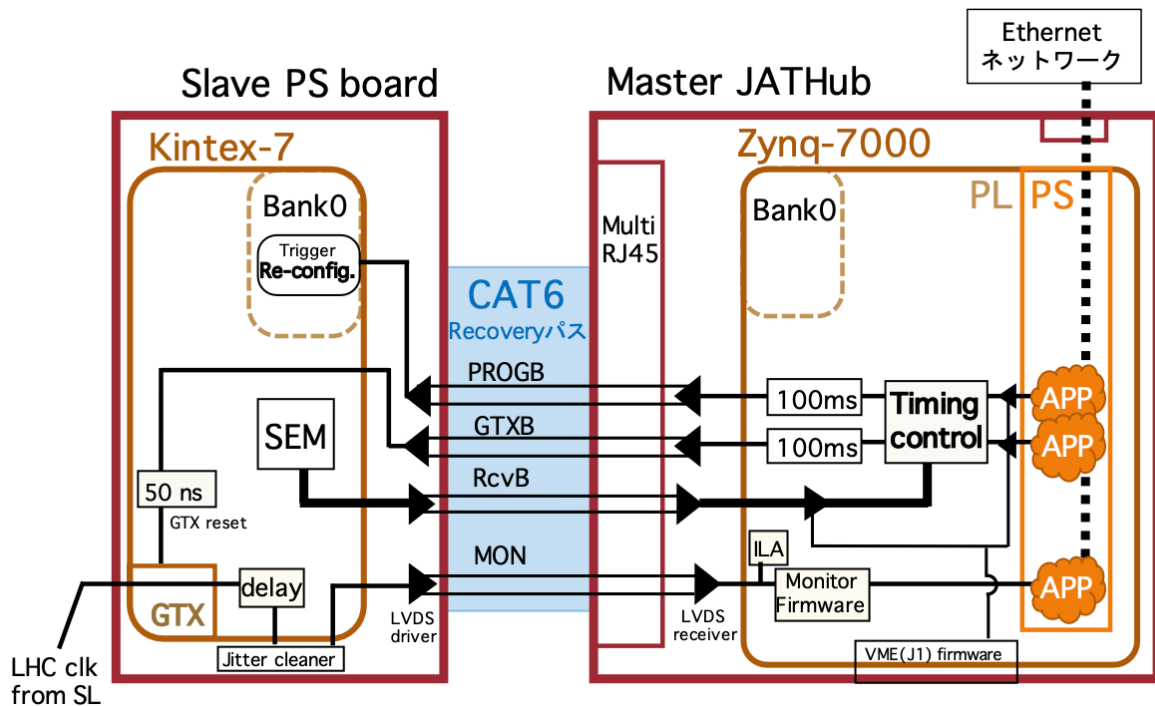


図 2.25 JATHub によるリカバリー手続きの概要 [18]。PS board に自己修復不可能な SEU 事象が発生した場合、Recovery Request 線 (RcvB 線) を通じて JATHub に救難信号が発出される。これを受信した JATHub は Program 線 (PROGB 線) を通じてリセット信号を送ることで PS board FPGA の再コンフィギュレーションを行う。

10 個、受信用に 10 個搭載している。それぞれが 12 レーンを束ねるため、送受信 120 リンクの光通信が可能である。1G イーサネットケーブルのインターフェイスである RJ45 コネクターも搭載しており、イーサネットを介したネットワーク通信も可能である。

1 枚の SL は TGC の 1/24 セクターからの信号処理を担当し、合計 31 台の PS board と接続する (BW 用に 29 枚、EI 用に 2 枚)。Aside と Cside を合わせて、合計で 48 枚の SL が設置される。以下にそれぞれの集積回路の機能の詳細を述べる。

SL FPGA

SL FPGA に実装するファームウェアの概要を図 2.27 に示す。ファームウェアは大別してトリガー回路、読み出し回路、コントロール回路に分けられる。PS board から光リンクを経由して送られるヒット情報は、トリガー回路および読み出し回路に入れられる。

トリガー回路は、PS board から送られる BW 7 層分のヒットデータを用いて p_T 判定を行なった後、エンドキャッププロトイド磁石より内側にある NSW、RPC BIS78、Tile カロリメーターからの情報も用いて、より精度の高い p_T の概算を行う。SL で再構成されたミューオン飛跡の一部は、より精度の高い p_T 計算のため MDTTP へ送信される。MDTTP で処理された飛跡情報は、SL に送り返され、MDTTP に送られなかったものと合わせて MUCTPI へ送信される。トリガー回路はこれまでに ATLAS TGC グループの共同研究として開発が進められてきた。本研究では開発されたトリガーマジュールの全体ファームウェアへの統合および試験システムの開発を行った。この詳細は 3 章、4 章で議論する。

読み出し回路は、PS board から受信したヒットビットマップとそのイベントに対応するトリガーデータをバッファリングしておき、LOA が発行されたイベントのデータを選択的に後段へ転送する役割を担う。読み出されるデータにはゼロサプレスという圧縮処理が行われ、すべての PS board からのデータはイベントごとにバッキング (Event

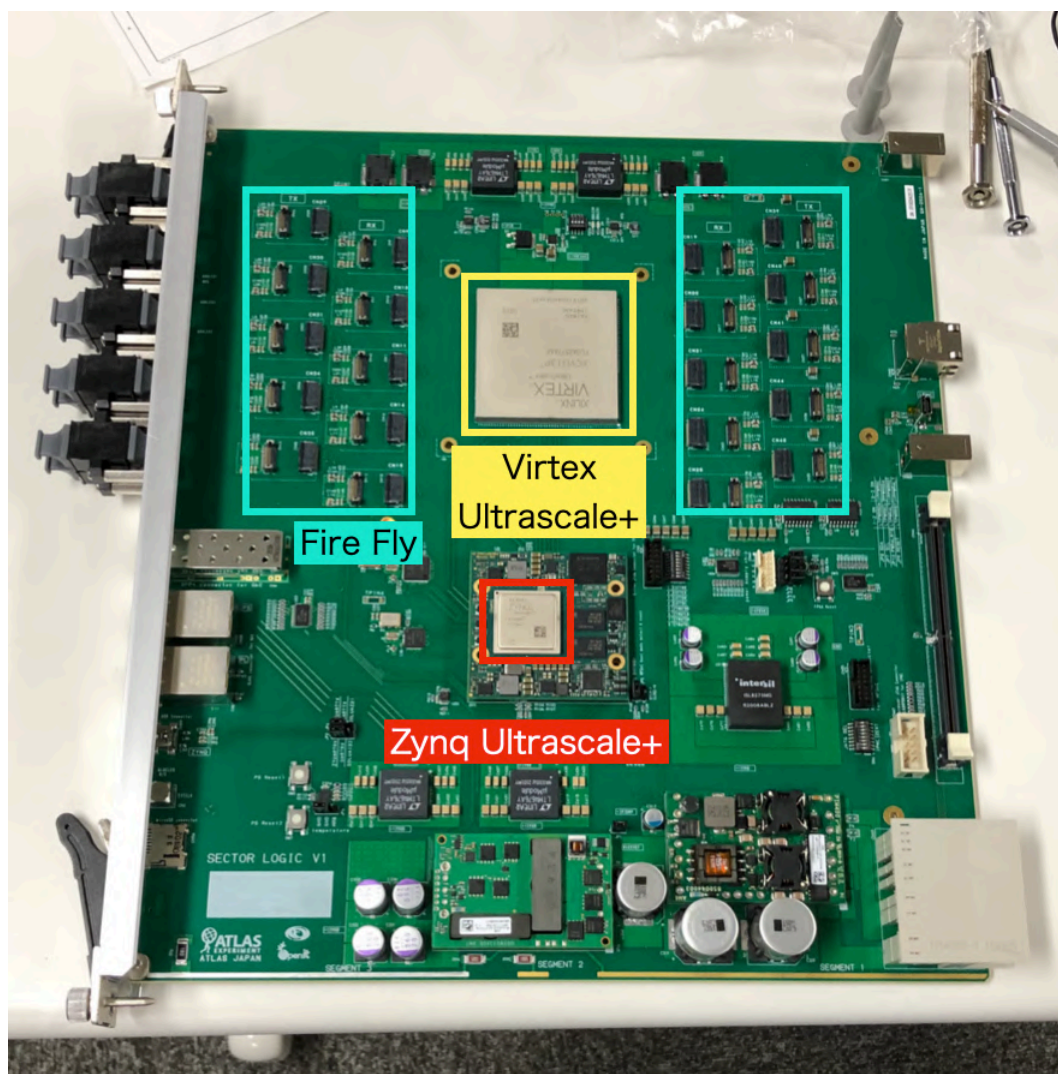


図 2.26 SL 第一試作機の写真。Virtex UltraScale+ FPGA と Zynq UltraScale+ MPSoC という 2 種類の集積回路を搭載している。他のモジュールとのインターフェイスとして、FireFly を送信用に 10 個、受信用に 10 個搭載している。1 つの FireFly は 12 リンクを束ねるため、合計 120 リンクの光通信が可能である。

Building) され、FELIX に送られる

コントロール回路は、LHC バンチ交差クロックと同期する必要のないスローな制御を担当する。MPSoC を起点に SL FPGA 内のレジスタを操作することで、SL のトリガー・読み出しに関連するパラメータの設定や、PS board の制御を行う。

SL FPGA は 4 つのシリコンダイ (Super Logic Region, SLR) で構成される大規模な FPGA である。図 2.28 に示すように、隣接する SLR は Super Long Line (SLL) と呼ばれる専用のワイヤーで接続されており、これを通じて信号の送受信が行われる。しかし、SLL を介した信号の伝搬は通常よりも大きなレイテンシーが生じる。さらに、SLL の位置は SLR 内で物理的に固定されているため、SLL を過剰に使用する設計ではタイミング制約を満たすことが難しくなる。ファームウェアを物理的な制約 (タイミング制約やリソース使用量など) を満たしつつ効果的に実装するためには、I/O や各種ロジックを FPGA 上の適切な場所へ配置することが重要となる。

図 2.29 に設計された SL FPGA のフロアマッピングを示す。PS board からのヒット信号は SLR 0, SLR 2, SLR 3 で受信され、磁場内部の検出器からの信号は SLR 1 で受信される。この配置に基づき、TGC のヒット信号のみ

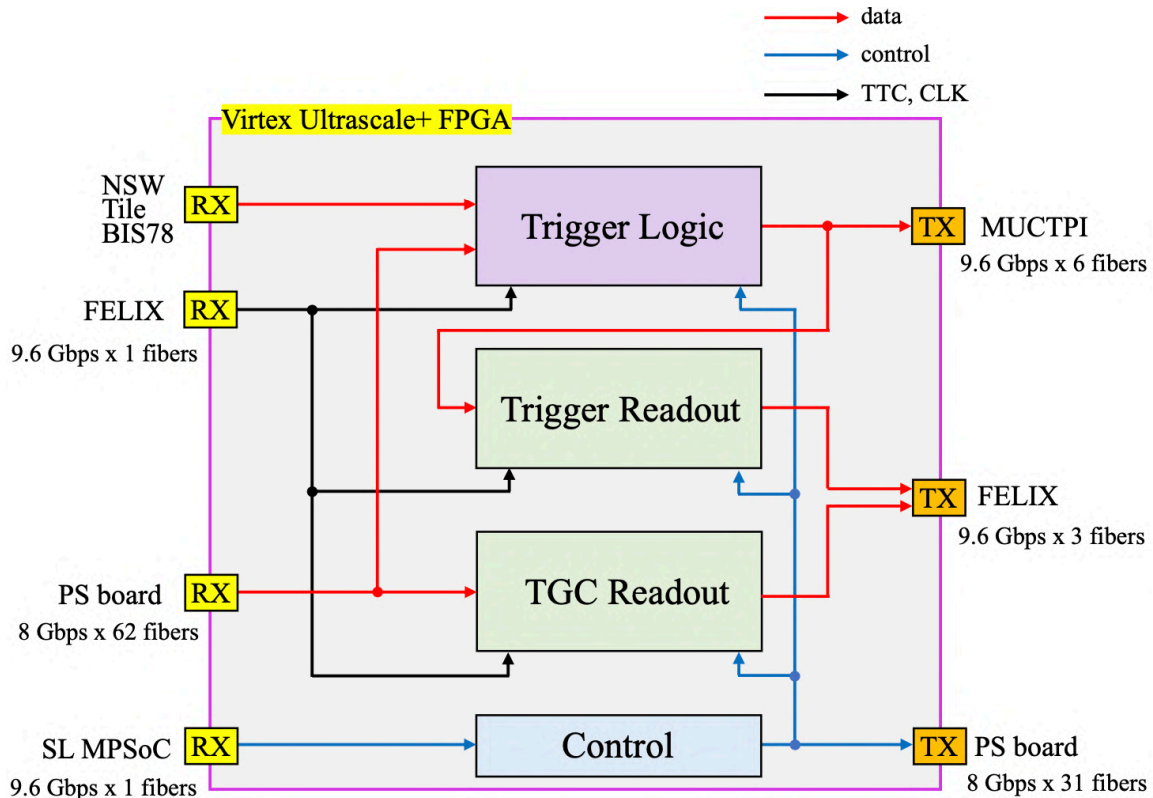


図 2.27 SL FPGA に実装されるファームウェアの概要 [19]。SL FPGA は大別してトリガー回路、読み出し回路、コントロール回路に分けられる。トリガー回路は PS board や磁場内部に位置する検出器からのヒットデータを受け取り、トリガー出力を MUCTPI に送信する。読み出し回路は PS board からのヒットデータをバッファし、FELIX から LOA が出されたイベントのデータを取り出し、FELIX に送信する。コントロール回路は SL MPSoC からコントロール用信号を受け取り、各ロジックの制御を行う。

を使用するトリガーロジック (TGC BW Coincidence) はそれぞれの SLR に配置され、Inner Coincidence は SLR 1 に配置される。これにより、8,000 bit に及ぶ PS board からのヒット信号はそれぞれの SLR で十分にデータ量が削減された後、SLR を跨いで送信される。TGC BW Coincidence はトリガーセクターごとに SLR を分けて配置され、SLR 0 にエンドキャップ $\phi 0$ 、SLR 2 にエンドキャップ $\phi 1$ 、SLR 3 にフォワード領域のコインシデンスロジックが配置される。

SL から FELIX にヒットデータを送信するリンクは SLR 3 に配置される。SLR を超える信号をできるだけ小さくするため、読み出し回路は Zero Suppress などのデータ圧縮処理をそれぞれの SLR 内で行い、SLR 3 で各 SLR からのヒット信号を集め、イベントごとにパッキングする。また、MPSoC と SL FPGA 間のチップ間通信リンクも SLR 3 に実装され、各 SLR のレジスタ操作は SLR 3 を介して行われる。読み出し回路やコントロール回路などの Fixed latency での実装が求められていないロジックでは、キューイングを適用しリソースを十分に節約している。

Zynq MPSoC

Zynq MPSoC は SL FPGA および PS board のコントロールマスターとして動作する。加えて ATLAS の TDAQ システムや Detector Control System (DCS) とのインターフェイスとしての役割も果たす。Zynq MPSoC も PS と PL から構成されるシリコンデバイスである。PS にはプロセッサやメモリが搭載されていて、SL では標準的な OS

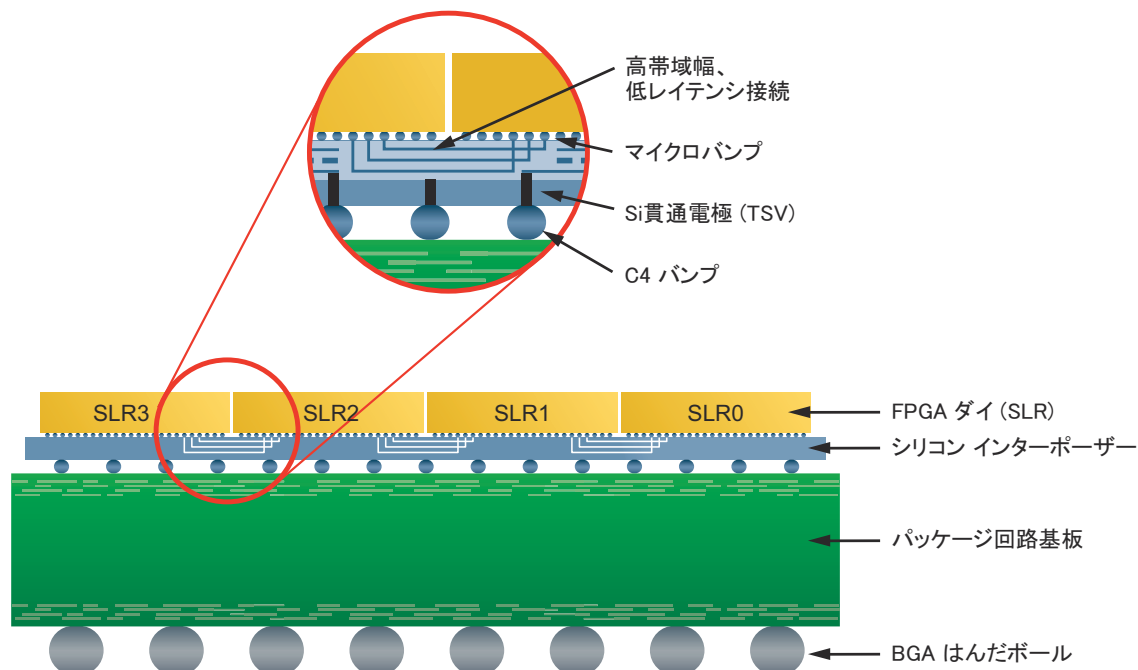


図 2.28 SLR の概略図。SL FPGA は 4 つの SLR で構成されており、隣接する SLR は SLL で接続されている。これにより SLR 間で信号を送受信することができる。

である CentOS 7 を起動する。^{*6}

SL の MPSoC は Enclustra 社が開発している Mercury XC5 メザニンカードに搭載されている。このメザニンには高速通信を行うための IO が搭載されており、SL FPGA と MPSoC は 2 レーンの高速シリアル通信を行うことができる。これを利用して、MPSoC から SL FPGA のコントロールや、SL FPGA から MPSoC へのデータ読み出しを行なっている。Mercury XU5 には他にも、DDR4 SDRAM や eMMC、Gigabit Ethernet PHY、USB PHY、QSPI フラッシュメモリなどが搭載されている。市販のメザニンを活用することで、SL ボードの開発コストを下げることに加え、メンテナンスを容易にすることができる。Mercury XU5 メザニンカードの構造と、SL ボード上における接続関係を図 2.30 に示す。

^{*6} Run 4 で使用する OS は今後選定される。

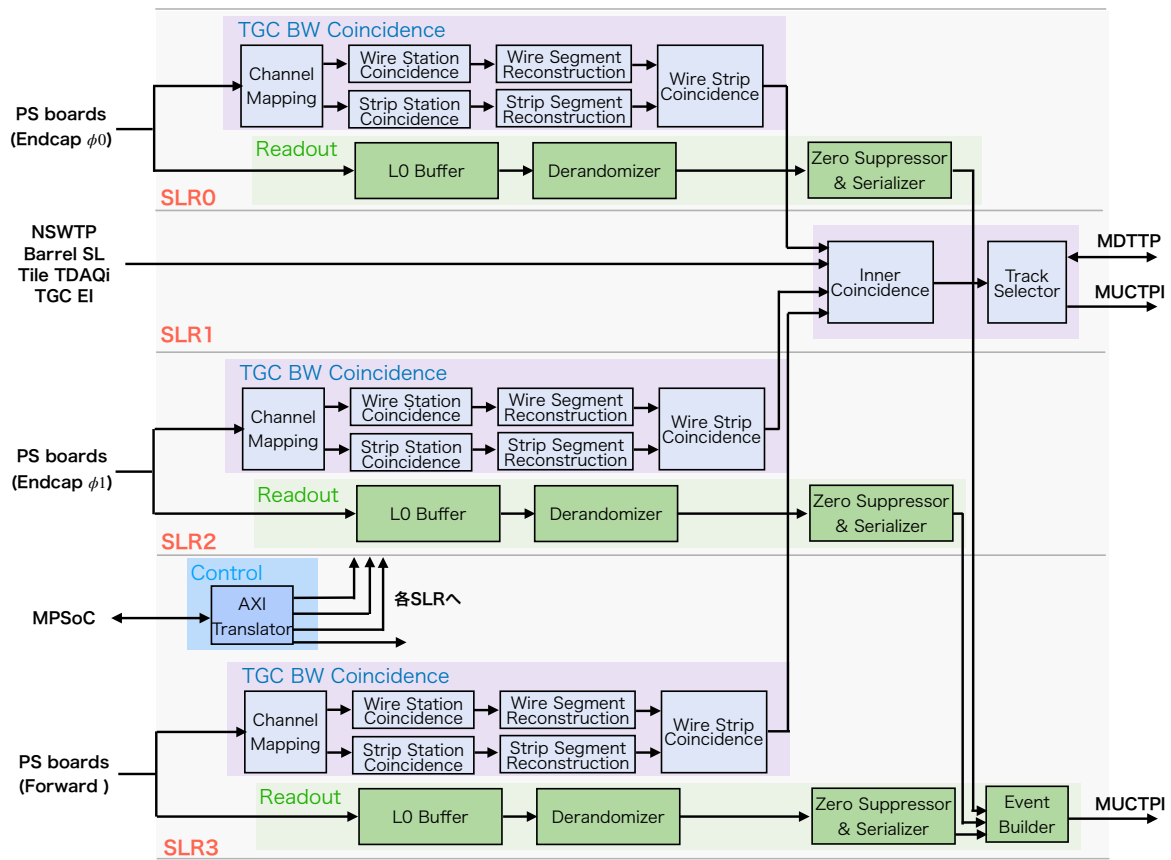


図 2.29 SLR FPGA のフロアマップ。PS board からのヒット信号は SLR 0、SLR 2、SLR 3 で受信し、磁場内部の検出器からの飛跡情報は SLR 1 で受信する。それに合わせて SLR 0、2、3 には TGC BW Coincidence および読み出し回路の一部を配置し、SLR 1 には Inner Coincidence を配置する。MPSoC とのインターフェイスは SLR 3 に配置し、これを介して各 SLR 内のレジスタ操作を行う。

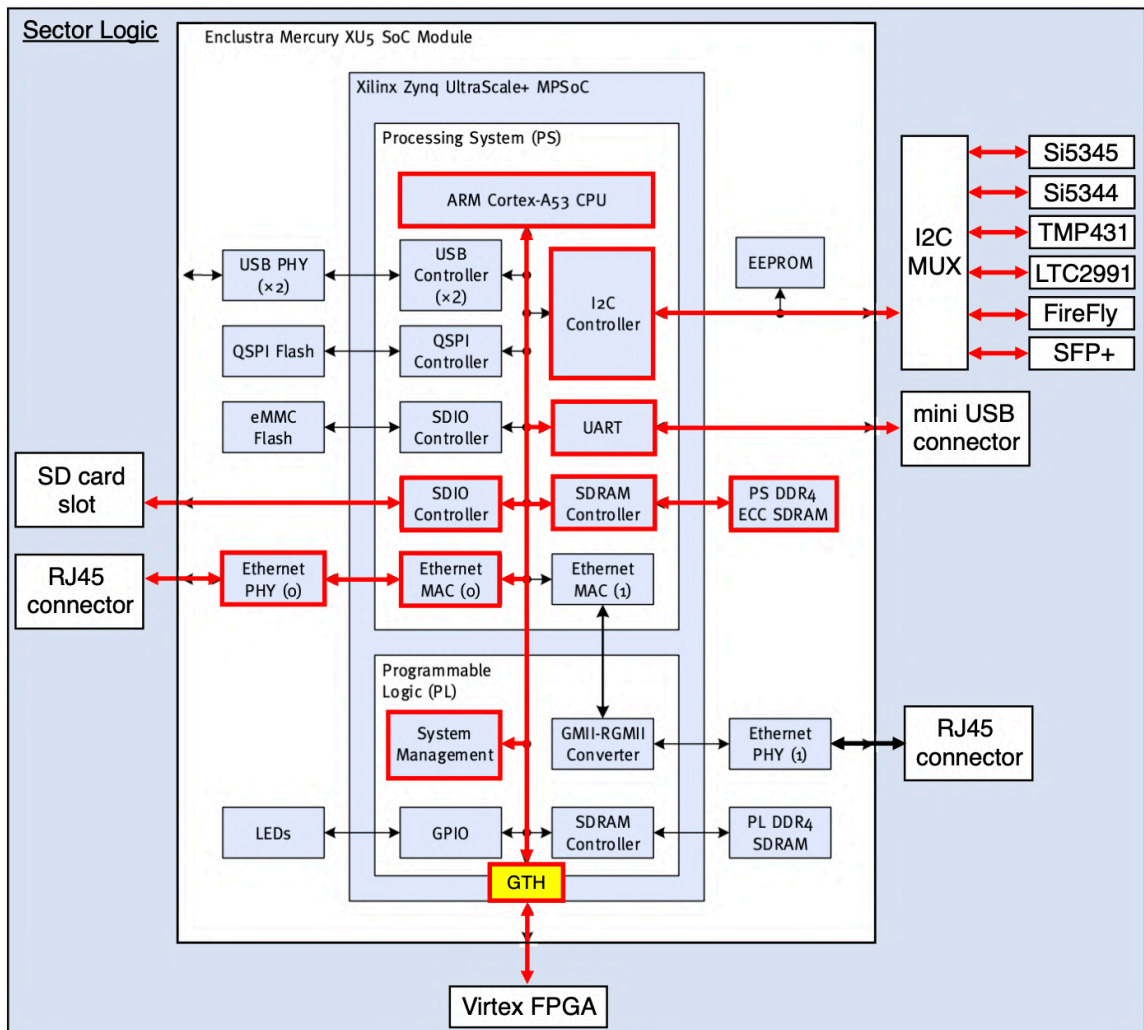


図 2.30 Mercury XU5 メザニンカードの構造と SL ボード上における接続関係 [19]

第 3 章

高輝度 LHC-ATLAS 実験に向けたエンド キャップ部ミュオントリガー回路の統合

高輝度 LHC-ATLAS 実験において、エンドキャップ部トリガーロジックは大規模論理回路として SL FPGA 上に実装される。本章では、まず、先行研究で開発されたトリガーロジックの概要および論理回路としての実装方法について説明する。次に、本研究で行った、トリガー論理回路の全体ファームウェアへの統合について議論する。

3.1 高輝度 LHC-ATLAS 実験におけるトリガーロジックの概要と論理回路実装

前章で述べたように、PhaseII アップグレードにより、SL は 1/24 セクター内の TGC BW 7 層からのヒットビットマップを、ヒットの有無に関わらず、すべての BC について受信するようになる。そのため、Run 3 で SLB、HPT、SL に分割されていたトリガーロジックは、SL に集約される。高輝度 LHC-ATLAS 実験でのミュオントリガー回路の全体像を図 3.1 に示す。SL FPGA に実装されるトリガーロジックは Channel Mapping、Station Coincidence、Segment Reconstruction、Wire Strip Coincidence、Inner Coincidence、Track Selector という 6 段階のトリガーモジュールで構成される。

Channel Mapping は、PS board から受信したヒットデータをコインシデンスロジックの入力に適した形へと並び替える。Station Coincidence はステーション内の 2 層または 3 層の間のコインシデンスをとり、そのステーションにおけるミュオンのヒット位置を表す"代表点"を出力する。Segment Reconstruction は各ステーションの代表点を組み合わせてミュオンの飛跡を再構成し、M3 の代表点をピボットとした無限運動量飛跡とのなす角 $\Delta\theta$ 、 $\Delta\phi$ を算出する。Wire Strip Coincidence は Wire で算出した $\Delta\theta$ と Strip で算出した $\Delta\phi$ からミュオンの横方向運動量閾値 p_T を概算する。TGC 検出器に飛来するミュオンはエンドキャップトロイド磁場により、主に η 方向に曲げられるため、 $\Delta\theta$ は p_T 再構成において有効な分別変数となる。一方、ミュオンは基本的には ϕ 方向に曲

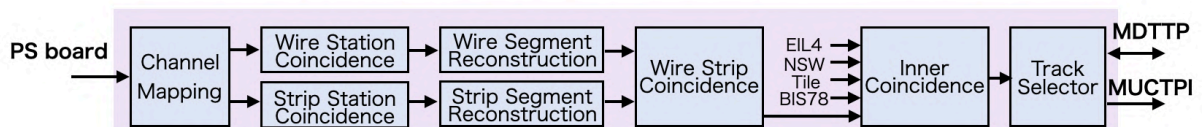


図 3.1 トリガー回路の全体像。高輝度 LHC-ATLAS 実験でのエンドキャップ部ミュオントリガー回路は、Channel Mapping、Station Coincidence、Segment Reconstruction、Wire Strip Coincidence、Inner Coincidence、Track Selector という 6 段階のトリガー回路で構成される。

げられず、 $\Delta\phi$ はミュオンが衝突点から飛来したものであることを担保する目的で利用される。ここまでのトリガーロジックは TGC 検出器からのヒットデータのみを利用するものであり、TGC BW Coincidence と呼ぶ。Inner Coincidence は TGC BW Coincidence で再構成されたミュオン飛跡候補と、磁場内部の検出器 (NSW、BIS78、RPC、EIL4 TGC、Tile カロリメーター) で得られたヒット情報の間でコインシデンスをとり、フェイクトリガーを削減するとともに p_T 精度の向上を図る。Inner Coincidence までのロジックは、1つの 1/24 セクターごとに最大 112 個のミュオン飛跡候補を出力する。Track Selector は、112 個のミュオン飛跡候補から p_T が大きい順に 6 つまで候補を絞り込む。そのうち 3 つは MDTTP に転送され、さらに高い精度で p_T が計算される。最終的に、MDTTP から SL に送り返された 3 つの飛跡候補と、MDTTP へ転送しなかった 3 つの飛跡候補を合わせて MUCTPI に送信する。

以下にそれぞれのロジックの概要と、論理回路への実装方法を説明する。

3.1.1 Channel Mapping

概要

Channel Mapping は PS board から受信する TGC BW 全チャンネルのヒット情報 (128 bit x 62 link) を、飛跡再構成に先んじてトリガー入力に適したフォーマットへとマッピングする。ここでは単にチャンネルを並び変えるだけでなく、TGC 検出器のジオメトリーに合わせて設計されたフロントエンドのチャンネル構造を、トリガーロジックとして取り扱いやすいものへと変換する。TGC BW のエンドキャップ領域は η 方向に M1 は 4 つ、M2、M3 は 5 つのチェンバーで構成されており、それぞれ不感領域がないようにオーバーラップを持って設置されている (図 2.14 および図 3.2 参照)。オーバーラップするチャンネル同士は OR 処理を施す。また、ストリップのコインシデンスではミュオンが η 方向に曲げられ、複数のチェンバーに跨ってヒットを残す場合がある。これに対応するため、図 3.2 に示すように、M3 のそれぞれのチェンバーに対してステーション間でコインシデンスを取る際には、M1、M2 ではより広い範囲のチェンバーの情報を参照できるように、OR を取った情報を後段に流す。

Channel Mapping の論理回路実装

Channel Mapping モジュールは単純なワイヤーと OR 回路で実装される。

3.1.2 Station Coincidence

概要

図 3.3 に Station Coincidence の概要を示す*¹。TGC 検出器はスタックリング構造を取っており、ステーション内のワイヤーは互いに η 方向にずらして、ストリップは互いに ϕ 方向にずらして設置されている。3 層の各チャンネルが重複してカバーする η 領域、 ϕ 領域を代表点 (スタガードチャンネル) と呼ぶ。Station Coincidence は M1 の 3 層、M2 の 2 層、M3 の 2 層のヒットチャンネルを入力として、コインシデンスが取れた代表点を出力する。これによりデータ量を落としながら、より位置分解能を上げてミュオンのヒット位置を特定することができる。Wire と Strip の Station Coincidence はそれぞれ独立に実装される。

Wire Station Coincidence の論理回路実装

このモジュールの駆動クロックは、LHC バンチ交差クロックに同期した 40 MHz クロックで、レイテンシーは 1 クロックチック (25 ns) である。M1 ステーションにおけるコインシデンスロジックの概要を図 3.4 に示す。ステーションコインシデンスは AND 回路と OR 回路の組み合わせ回路として実装される。3 層中 3 層にヒットがあった

*¹ Station Coincidence を Intra Station Coincidence、Segment Reconstruction を Inter Station Coincidence と呼ぶ場合もある。

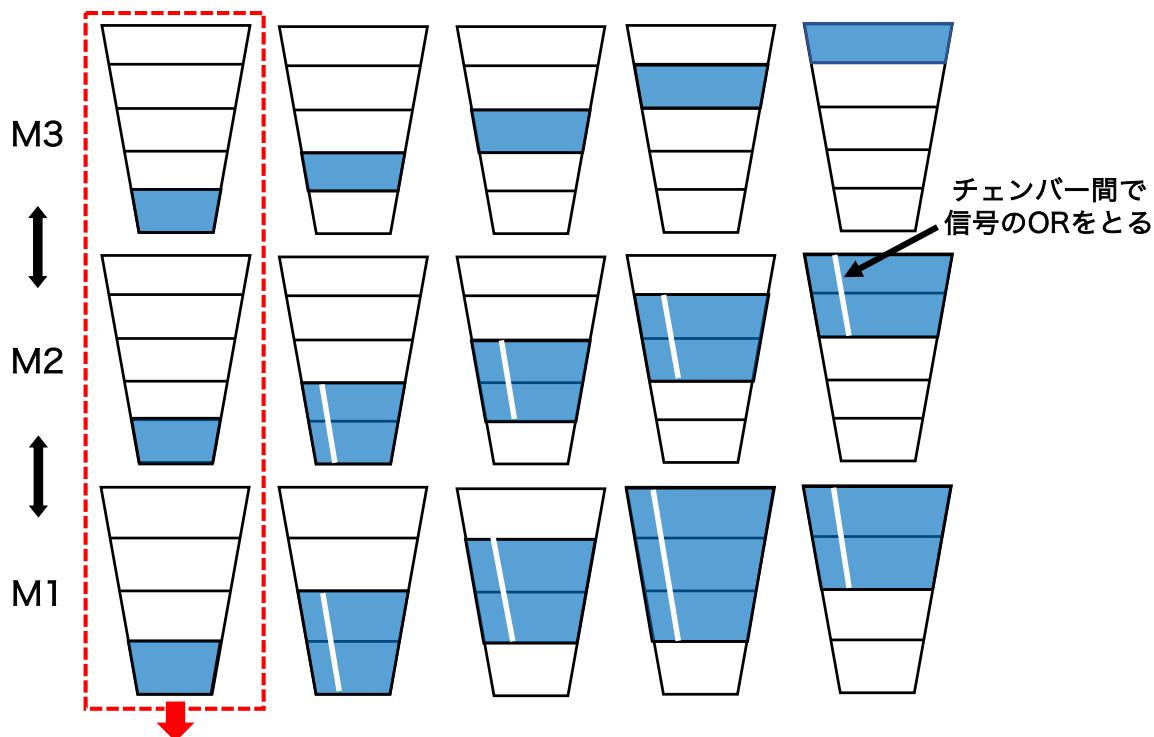


図 3.2 ストリップにおけるチェンバー間の OR の取り方 [11]。M3 のそれぞれのチェンバーに対してステーション間でコインシデンスを取る際、M1、M2 ではより広い範囲のチェンバーの情報を参照できるように OR を取った情報を後段に流す。

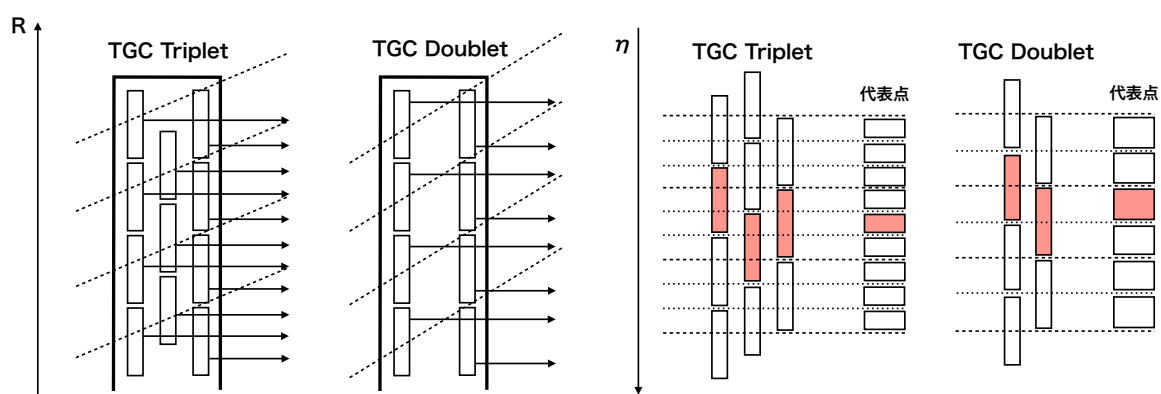
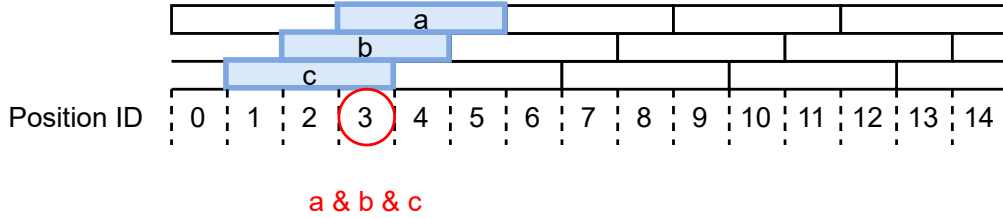
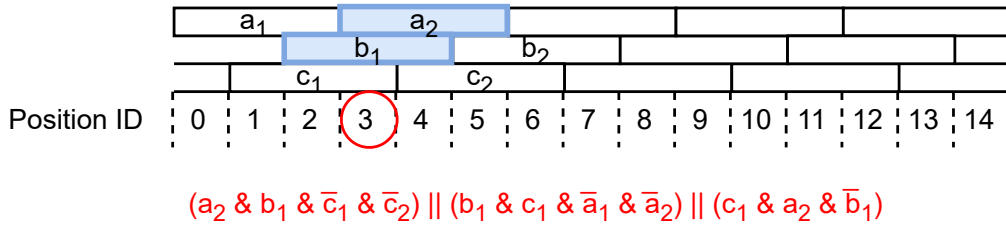


図 3.3 Station Coincidence の概要 [20]。TGC 検出器ではステーション内のワイヤーは η 方向に、ストリップは ϕ 方向にずらして設置されており、各層のチャンネルが重複してカバーする η 領域を代表点として定義する。Station Coincidence では 2 層または 3 層でコインシデンスのとれた代表点の ID を後段に送る。

(1) 3/3 coincidence module



(2) 2/3 coincidence module



(3) 1/3 coincidence module

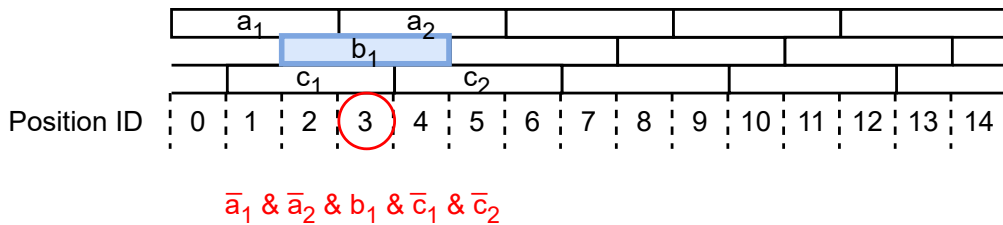


図 3.4 M1 Triplet におけるコインシデンスロジック [21]。3 層中 3 層にヒットがあった場合に代表点を出力する 3/3 コインシデンス、3 層中 2 層の場合に代表点を出力する 2/3 コインシデンス、3 層中 1 層の場合に代表点を出力する 1/3 コインシデンスが用意されている。

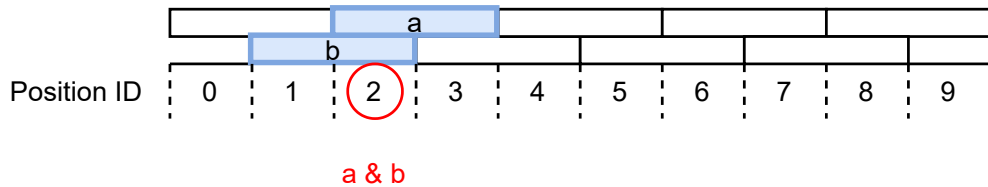
場合に代表点を出力する 3/3 コインシデンス、2 層にヒットがあった場合の 2/3 コインシデンス、1 層にヒットがある場合の 1/3 コインシデンスが独立に用意されており、それぞれが並列に動作する。

M2、M3 ステーションは 2 層で構成されているため、2/2 コインシデンスと 1/2 コインシデンスが用意されており、それぞれのロジックは図 3.5 のように実装される。

Strip Station Coincidence の論理回路実装

このモジュールの駆動クロックは、LHC バンチ交差クロックに同期した 40 MHz クロックで、レイテンシーは 1 クロックチック (25 ns) である。コインシデンスのロジックは基本的にワイヤーと同じで、M1、M2、M3 ともに 2 層構造になっているためそれぞれ 2/2、1/2 ロジックが並列に走っている。

(1) 2/2 coincidence module



(2) 1/2 coincidence module

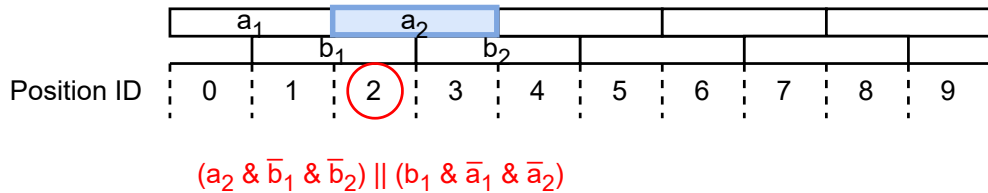


図 3.5 M2, M3 ステーションにおけるコインシデンスロジック [21]。2 層中 2 層にヒットがあった場合に代表点を出力する 2/2 コインシデンス、2 層中 1 層の場合に代表点を出力する 1/2 コインシデンスが用意されている。

3.1.3 Segment Reconstruction

概要

Segment Reconstruction では Station Coincidence で得られた、各ステーションの代表点の組み合わせから、無限運動量飛跡と実際の飛跡のなす角度 ($\Delta\theta$, $\Delta\phi$) を算出する。ミュオンはエンドキャプトロイド磁場により主に η 方向に曲げられるため、 $\Delta\theta$ は p_T を再構成する上で有効な分別変数となる。一方、 ϕ 方向にはほとんど曲げられず、有効な分別変数とはなり得ない。ミュオンが衝突点から飛来したものであることを担保するための、追加的な条件として利用される。Segment Reconstruction の概念図を図 3.6 に示す。角度情報の概算には、パターンマッチングと呼ばれる手法を用いる。この手法ではあらかじめ、代表点の組み合わせとそこから計算される角度情報の対応関係をまとめたテーブル (パターンリスト、LUT ともよぶ) を用意することで、複雑な計算をせずに高速で角度情報を再構成する。

Wire segment Reconstruction の論理回路実装

このモジュールの駆動クロックは、LHC クロックに同期した周波数 160 MHz のクロックで、レイテンシーは 12 クロックチック分 (75 ns) である。

Wire segment Reconstruction では Unit、Subunit と呼ばれる単位領域でトリガーセクターを分割して、Subunit ごとに並列にコインシデンスロジックを用意する。ワイヤーロジックはエンドキャップ領域を 37 分割、フォワード領域を 16 分割したものを Unit として定義する。図 3.7 に Unit の構造を示す。1 つの Unit は M1 ステーションの 96 代表点、M2 ステーションの 32 代表点、M3 ステーションの 16 代表点をカバーする。また、1 つの Unit を

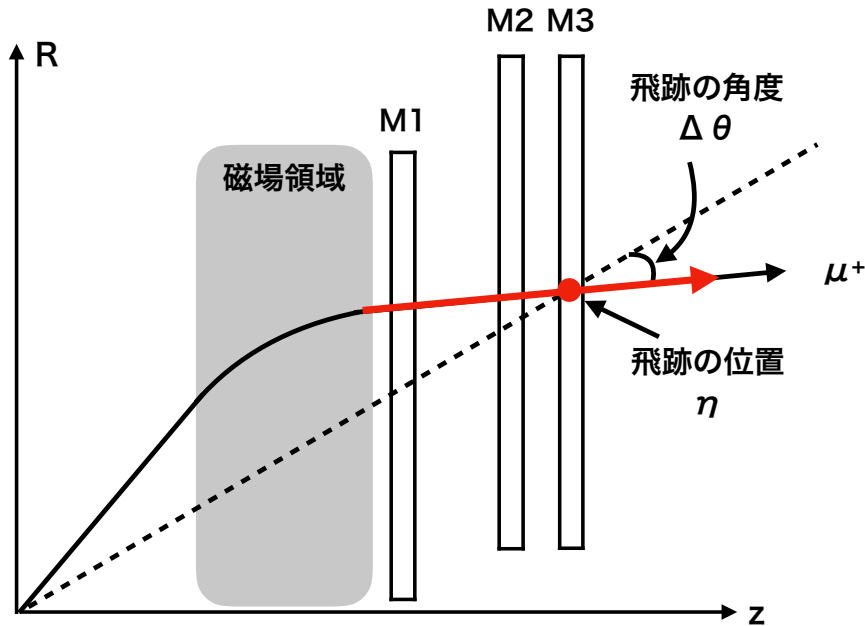


図 3.6 η 方向のパターンマッチングの概念図 [20]。黒い線が実際のミューオンの飛跡を表し、赤い線が TGC のヒットから再構成される飛跡を表す。黒い点線は M3 ステーションの代表点と衝突点を結んだ、無限運動量飛跡を表す。M1、M2、M3 の代表点の組み合わせから、実際の飛跡と無限運動量飛跡とがなす角度 $\Delta\theta$ を再構成する。

4 等分するように Subunit を定義し、Subunit に 1 つ、パターンマッチング用の LUT が用意する。Unit や Subunit の大きさは必要な LUT のサイズによる制約で決められ、FPGA の RAM リソースに合わせて最適化されている。Unit の範囲は、とある M3 チャンネルにヒットを残した p_T 5 GeV のミューオンが飛跡を残しうる M1、M2 の代表点を網羅するよう設計されている。

最終的にはそれぞれの Subunit から最大 1 つの飛跡情報を出力するため、SL 全体では最大 360 の飛跡候補が出力される。Wire segment Reconstruction の各 Subunit 内でのロジックの概要を図 3.8 に示す。各 Subunit は Address Specifier · Segment Extractor · Segment Selector で構成される。以下でそれぞれのモジュールについて述べる。

Address Specifier

Wire Station Coincidence で得られた各ステーションの代表点を組み合わせて、LUT にアクセスするためのアドレスを作成する。1 つの Subunit は Station Coincidence から M1 の代表点を最大 6 つ、M2 のものを最大 4 つ、M3 のものを最大 2 つ、それぞれ受け取るため、M1、M2、M3 の代表点の組み合わせは最大 $6 \times 4 \times 2 = 48$ パターン存在する。Address Specifier はこの中から 8 パターンを選抜して Segment Extractor へ送る。パターンを選択するにはマッチレイヤーが多いものが優先する。具体的に優先順位を定めたテーブルを表 3.1 に示す。より上位にリストされているコインシデンスパターンが優先して送られる。また同率の組み合わせが複数存在する場合には、 η がより小さいものが選ばれる。なお、同表中の Fraction の値はレイヤーあたりの検出効率を 94 % と仮定して算出したものである。

Segment Extractor

Wire segment Reconstruction で利用される LUT は FPGA の URAM 上に格納される。Segment Extractor では Address Specifier で作られたアドレスをもとに、URAM にアクセスし、対応するデータ (Wire segment) を出力する。Wire segment のデータフォーマットを表 3.2 に示す。URAM は Dual port で設計されており、1 つの Subunit

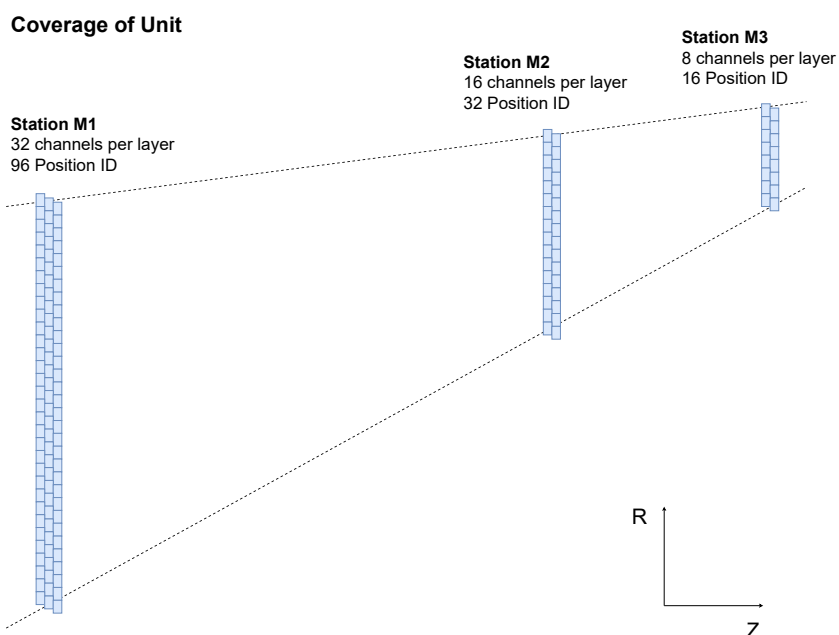


図 3.7 Wire segment Reconstruction における Unit[21]。1 つの Unit は M1 の 96 代表点、M2 の 32 代表点、M3 の 16 代表点をカバーする。

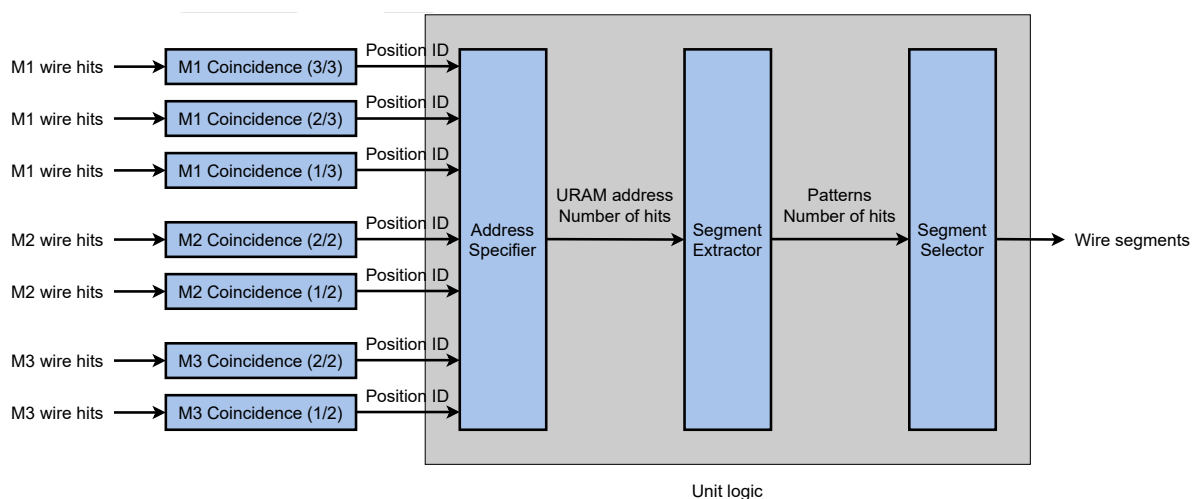


図 3.8 Wire segment Reconstruction のブロックダイアグラム [21]。Wire segment Reconstruction は、Station Coincidence で得られた代表点の組み合わせから、LUT のアドレスを作成する Address Specifier、LUT から対応するデータを取り出す Segment Extractor、Segment Extractor から得られる最大 8 つの飛跡候補から後段に送る 1 つを選別する Segment Selector で構成される。

表 3.1 Wire Station Coincidence におけるコインシデンスパターン

Coincidence Pattern	Hit Pattern			Faraction
	M1	M2	M3	
7/7	3/3	2/2	2/2	0.649
6/7A	2/3	2/2	2/2	0.124
6/7B	3/3	1/2	2/2	0.083
6/7C	3/3	2/2	1/2	0.083
5/7A	2/3	1/2	2/2	0.016
5/7B	1/3	2/2	1/2	0.016
5/7C	3/3	1/2	1/2	0.011
5/7D	1/3	2/2	2/2	0.008
Total				0.988

が 2 ポート利用する。そのため 1 つの Subunit は 40 MHz クロック 1 チックの間で 8 つのデータを処理する。

表 3.2 Wire segment のフォーマット

# of bits	Name
1	Flag of successful reconstruction
2	Number of the stations with hits used for coincidence
8	Angle difference $\Delta\theta$ between the segment and the vector from IP
12	Global η position of the segment

Segment Selector

Segment Extractor から送られる最大 8 つの Wire segment から マッチレイヤーの多さを基準に最大 1 つ選択して、Wire Strip Coincidence へと送信する。マッチレイヤーも同じものが複数ある場合には $\Delta\theta$ がより小さいものを選ぶ。

Strip segment Reconstruction

このモジュールの駆動クロックは、LHC クロックに同期した周波数 240 MHz のクロックで、レイテンシーは 21 クロックチック分 (87.5 ns) である。Strip segment Reconstruction も Unit、Subunit 呼ばれる単位領域でトリガーセクターを分割して、Subunit ごとに並列にコインシデンスロジックを用意する。Strip では 1 つのチェンバーを 4 分割するよう Unit が定義される。エンドキャップ領域は 5 枚のチェンバーで構成されるため 20 Unit、フォワード領域は 1 枚のチェンバーで構成されるため 5 Unit 存在する。図 3.9 に Unit の構造を示す。1 つの Unit は M1 ステーションの 40 代表点、M2 ステーションの 24 代表点、M3 ステーションの 16 代表点をカバーする。また、1 つの Unit を 2 等分するように Subunit が定義されており、Subunit に 1 つパターンマッチング用の LUT が用意される。

最終的にはそれぞれの Unit から最大 1 つの飛跡情報を出力するため、SL 全体では最大 45 の飛跡情報が出力される。Strip segment Reconstruction の各 Subunit 内でのロジックの概要を図 3.10 に示す。各 Subunit は Address Specifier · Segment Extractor · Segment Selector で構成される。以下でそれぞれのモジュールについて述べる。

Coverage of Unit

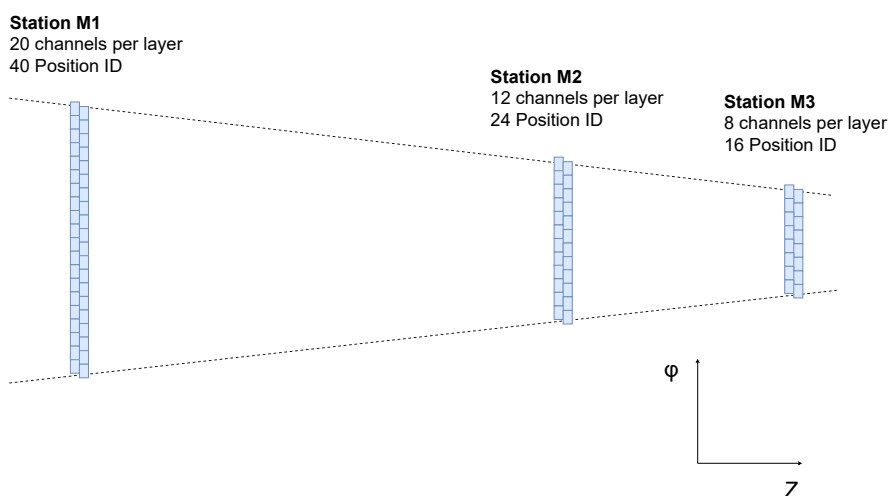


図 3.9 Strip segment Reconstruction における Unit[21]。1つの Unit は M1 の 40 代表点、M2 の 24 代表点、M3 の 16 代表点をカバーする。

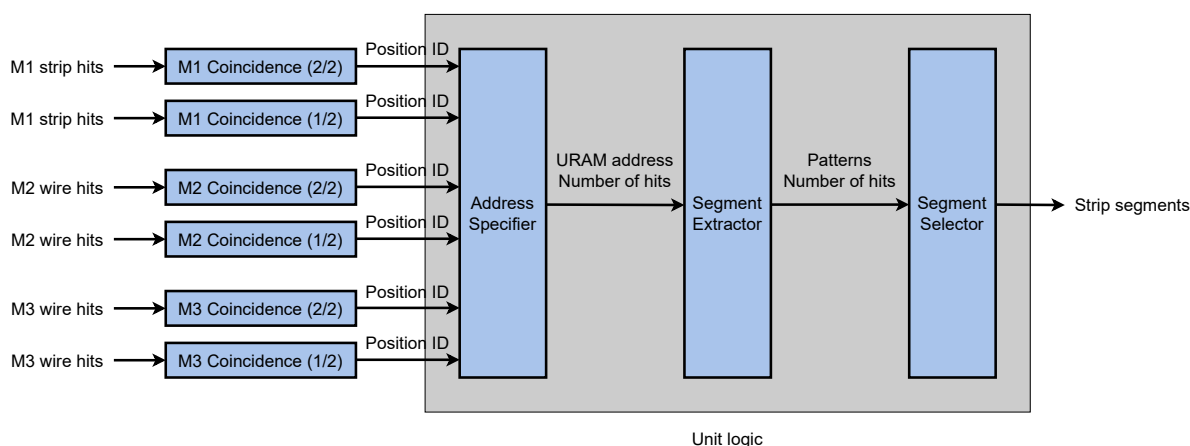


図 3.10 Strip segment Reconstruction のブロックダイアグラム [21]。Strip segment Reconstruction は、Station Coincidence で得られた代表点の組み合わせから、LUT のアドレスを作成する Address Specifier、LUT から対応するデータを取り出す Segment Extractor、Segment Extractor から得られる最大 6 つの飛跡候補から後段に送る 1 つを選別する Segment Selector で構成される。

Address Specifier

Strip Station Coincidence で得られた各ステーションの代表点を組み合わせて、LUT にアクセスするためのアドレスを作成する。1つの Subunit は Station Coincidence から M1 の代表点を最大 4 つ、M2 のものを最大 4 つ、M3 のものを最大 2 つ、それぞれ受け取るため、M1、M2、M3 の代表点の組み合わせは最大 $4 \times 4 \times 2 = 32$ パターン存在する。Address Specifier はこの中から 6 つのパターンを選抜して Segment Extractor へ送る。パターンを選択する際にはワイヤーのロジックと同様にマッチレイヤーが多いもの優先する。具体的に優先順位を定めたテーブルを表 3.3 に示す。

表 3.3 Strip segment Reconstruction における Wire segment のデータフォーマット

Coincidence Pattern	Hit Pattern		
	M1	M2	M3
6/6	2/2	2/2	2/2
5/6A	2/2	1/2	2/2
5/6B	1/2	2/2	2/2
5/6C	2/2	2/2	1/2
4/6A	1/2	1/2	2/2
4/6B	2/2	1/2	1/2
4/6C	1/2	2/2	1/2
Total			

表 3.4 Strip segment のフォーマット

# of bits	Name
2	Number of the stations with hits used for coincidence
6	Local ϕ position in the chamber
9	Angle difference $\Delta\phi$ between the segment and the vector from IP

Segment Extractor

Strip segment Reconstruction で利用される LUT は FPGA の URAM 上に格納される。Segment Extractor では Address Specifier で作られたアドレスをもとに、URAM にアクセスし、対応するデータ (Strip segment) を出力する。Strip segment のデータフォーマットを表 3.4 に示す。URAM は Dual port で設計されており、2 つの Subunit が 1 ポートずつ利用する。そのため 1 つの Subunit は、40 MHz クロック 1 チックの間に 6 つのデータを処理する。

Segment Selector

Segment Extractor から送られる最大 6 つの Strip segment から マッチレイヤーの多さを基準に、Unit ごとに最大 1 Segment 選択して、Wire Strip Coincidence へと送信する。マッチレイヤーも同じものが複数ある場合には $\Delta\phi$ がより小さいものを選ぶ。

3.1.4 Wire-Strip Coincidence

概要

Wire Strip Coincidence では Wire segment Reconstruction で算出した $\Delta\theta$ と Strip segment Reconstruction で算出した $\Delta\phi$ を組み合わせることで、横方向運動量 p_T を概算する。 p_T の計算も Coincidence Window と呼ばれる LUT を用いて行う。Coincidence Window の例を図 3.11 に示す。

Wire Strip Coincidence の論理回路実装

駆動クロックは LHC バンチ交差クロックに同期した 160 MHz クロックで、レイテンシーは 6 クロックチック分 (37.5 ns) である。Wire Strip Coincidence では Region と呼ばれる単位領域を新たに設定し、Region ごとに並列にコインシデンスロジックを走らせる。Region は後段の Inner Coincidence のコインシデンスをとる単位に合わ

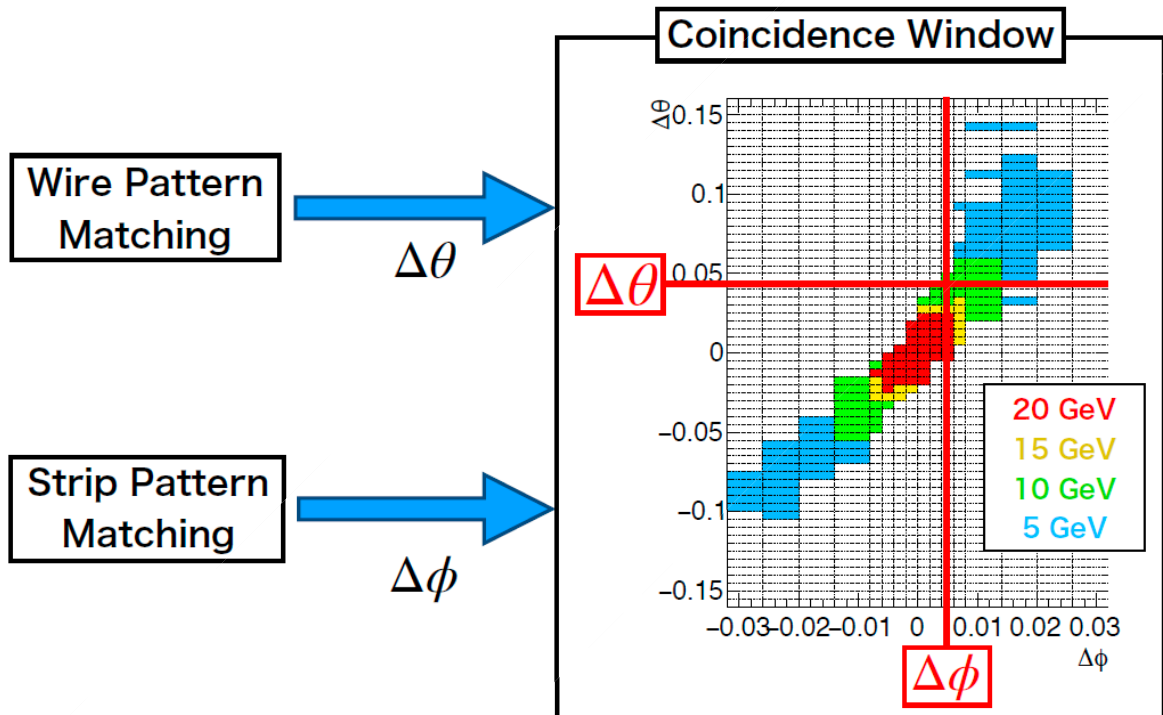


図 3.11 Wire Strip Coincidence における Coincidence Window の例 [21]。Wire および Strip segment Reconstruction で再構成された $\Delta\theta$ 、 $\Delta\phi$ から p_T 閾値を概算する。 $\Delta\phi$ で再構成したミューオンが衝突点に由来するものであることを担保し、 $\Delta\theta$ で p_T を決めている。 $\Delta\phi$ の値域が $\Delta\theta$ の関数になっているのは、この例で示した領域が荷電粒子を ϕ 方向にも曲げる磁場を持つ例外的な点であるためである。現状 p_T は 5 GeV、10 GeV、15 GeV、20 GeV の 4 段階で出力している。本番運用時には 4 bit、16 段階で出力する予定である。

せて定義されている。Wire Strip Coincidence における Region の定義を図 3.12 に示す。エンドキャップ領域は $|\eta| < 1.3$ では 8 Unit Region、 $|\eta| < 1.3$ では 32 Unit Region という異なる大きさの Region を定義し、それぞれの領域を 22 分割、13 分割する。フォワード領域は 32 Unit Region で 8 分割する。

図 3.13 に 1 つの Unit の構造を示す。8 Unit Region はトリガーセクターの全 ϕ 領域に当たる、ストリップ 4 Unit からの Strip segment と ワイヤー 2 Subunit からの Wire segment を組み合わせて、最大 1 つの飛跡候補を出力する。32 Unit Region はストリップ 4 Unit からの Strip segment と、ワイヤー 8 Subunit からの Wire segment を組み合わせて、最大 4 つの候補を出力する。SL 全体では最大 180 の飛跡候補が出力される。

Wire Strip Coincidence の各 Unit 内でのロジックの概要を図 3.14 に示す。各 Unit は p_T Calculator、Wire Position Corrector、Block Selector で構成される。以下でそれぞれのモジュールについて説明する。

p_T Calculator

p_T Calculator は Wire segment に含まれる 8 bit の $\Delta\theta$ 及び Strip segment に含まれる 9 bit の $\Delta\phi$ から、Coincidence Window にアクセスするためのアドレスを作成し、Coincidence Window に格納された 4 bit の p_T を出力する。1 つの p_T Calculator は 1 つの Wire segment と 4 つの Strip segment を組み合わせ、全ての組み合わせに対して p_T を出力する。Coincidence Window は FPGA の Dual port BRAM 上に格納されているため、この処理は 160 MHz の 2 クロックチックで完了する。

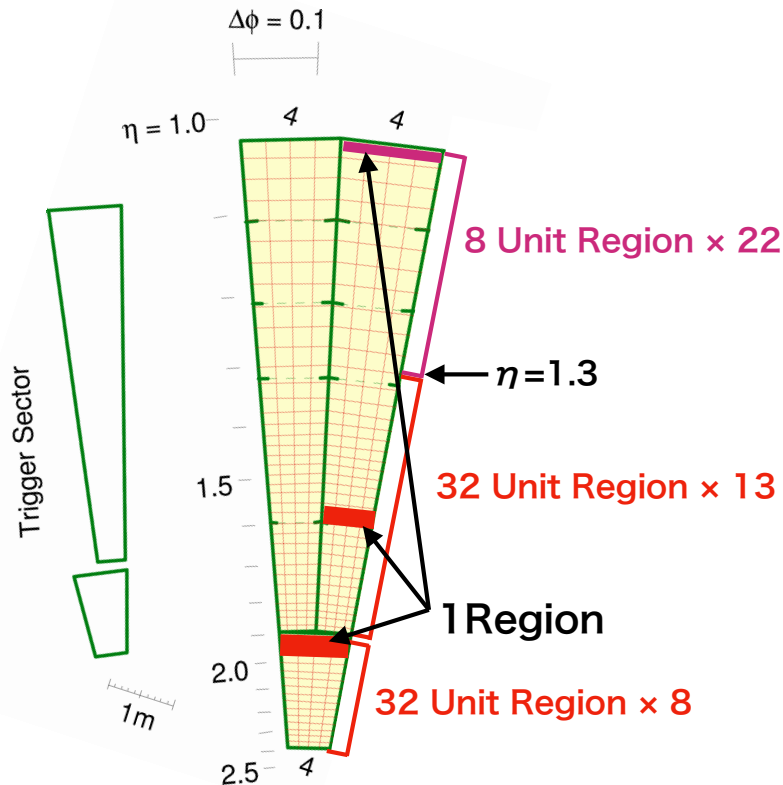


図 3.12 Wire Strip Coincidence の Region の定義 [11]。エンドキャップ領域は $|\eta| < 1.3$ 領域では 8 Unit、 $1.3 < |\eta|$ 領域では 32 Unit という異なる大きさの Region を定義し、それぞれの領域を 22 分割、13 分割する。フォワード領域は 32 Unit Region で 8 分割する。

Wire Position Corrector

Wire segment や Strip segment ではミュオンのヒットがあった位置情報を代表点番号で表しているが、磁場内部の検出器とのコインシデンをとるにあたって、位置情報を η 座標位置に変換する必要がある。しかし、TGC 検出器における代表点番号と η の関係は Unit Region ごとにばらつきがあることが知られており*2、全ての Unit で共通に変換できるわけではない。そこで wire と strip の代表点と η 座標の対応関係をパターンリストとして保存し、Unit Region ごとに η の再構成を行う。これを担当するモジュールを Wire Position Corrector と呼ぶ。

Block Selector

Block Selector はワイヤーとストリップの組み合わせで生じる複数のミュオン飛跡候補から、後段に送る飛跡候補を選択する。飛跡の選抜はマッチした層数の多さを基準に行い、マッチした層数も同じ場合は角度がより小さいものを選択する。8 Unit Region では 4 つのストリップ飛跡と 2 つのワイヤー飛跡から再構成される計 8 個の飛跡候補から、最大 1 つの候補を絞り込む。32 Unit Region では 4 つのストリップ飛跡と 8 つのワイヤー飛跡の組み合わせからなる 32 個の飛跡候補から最大 4 つの候補を選ぶ。この時、2 つは $\Delta\theta$ が正の方向に曲げられたもの、もう 2 つは負に曲げられたものが選ばれるように設計する。これは J/ψ 粒子などから生じる、2 つの異符号ミュオンを優先的に取れるようにするための実装である。Wire Strip Coincidence から Inner Coincidence に送られる飛跡情報のフォーマットを図 3.5 に示す。

*2 wire の代表点が η に対して均一に並んでいないこと、 ϕ 位置ごとに代表点と η の対応関係が異なること、が原因。

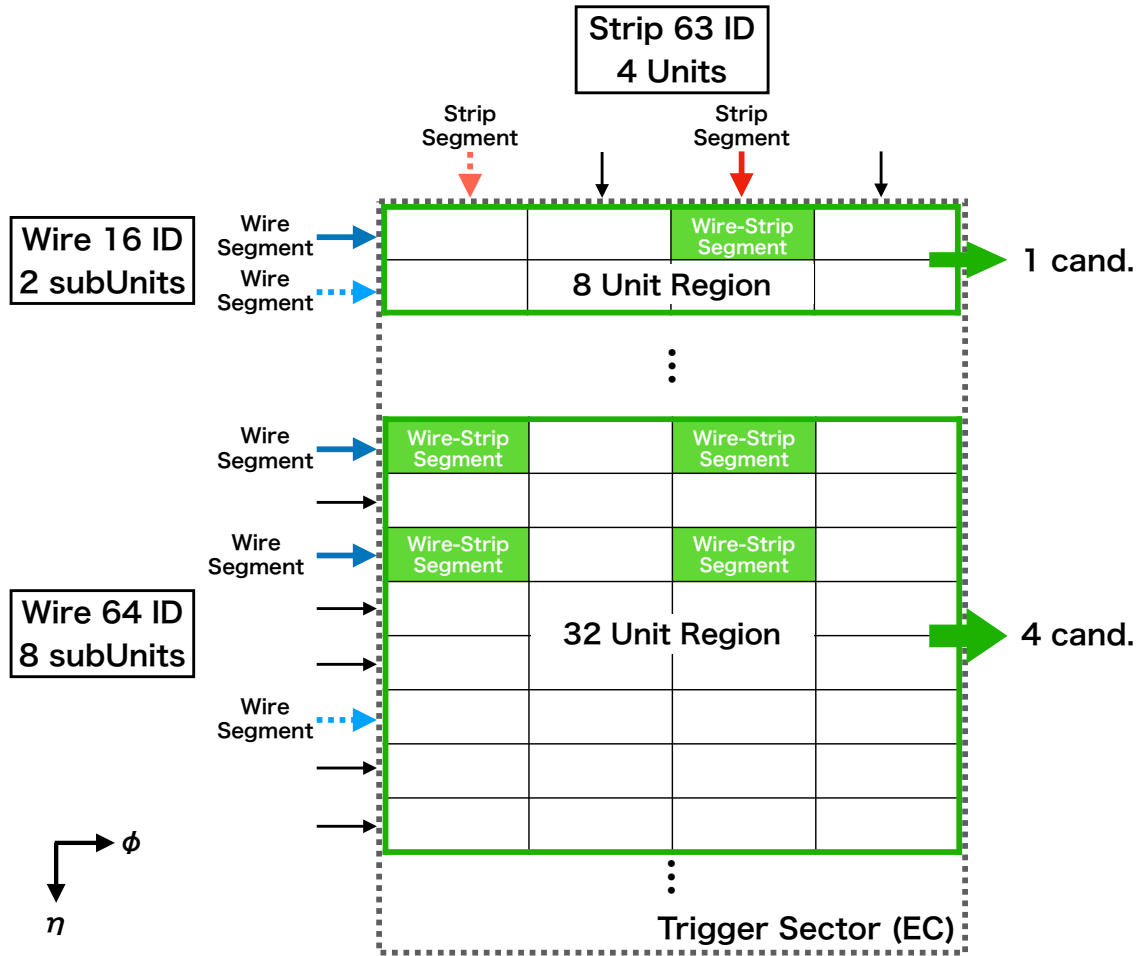


図 3.13 8 Unit Region、32 Unit Region におけるコインシデンスの概要 [11]。8 Unit Region ではストリップ 4 Unit からの Strip segment とワイヤー 2 Subunit からの Wire segment の組み合わせから、最大 1 つの飛跡候補を出力する。32 Unit Region はストリップ 4 Unit からの Strip segment と、ワイヤー 8 Subunit からの Wire segment の組み合わせから、最大 4 つの候補を出力する。

表 3.5 Wire Strip Coincidence における飛跡候補のデータフォーマット

# of bits	Name
1	Valid flag which means this region receive the more than one Strip segment and Wire segment
4	p_T threshold reconstructed using Coincidence Window
8	Local η position in the unit
3	Unit ID
7	Angle difference $\Delta\theta$ between the segment and the vector from IP
2	Number of stations with the hits used for Wire segment reconstruction
4	Angle difference $\Delta\phi$ between the segment and the vector from IP
2	Number of stations with the hits used for Strip segment reconstruction

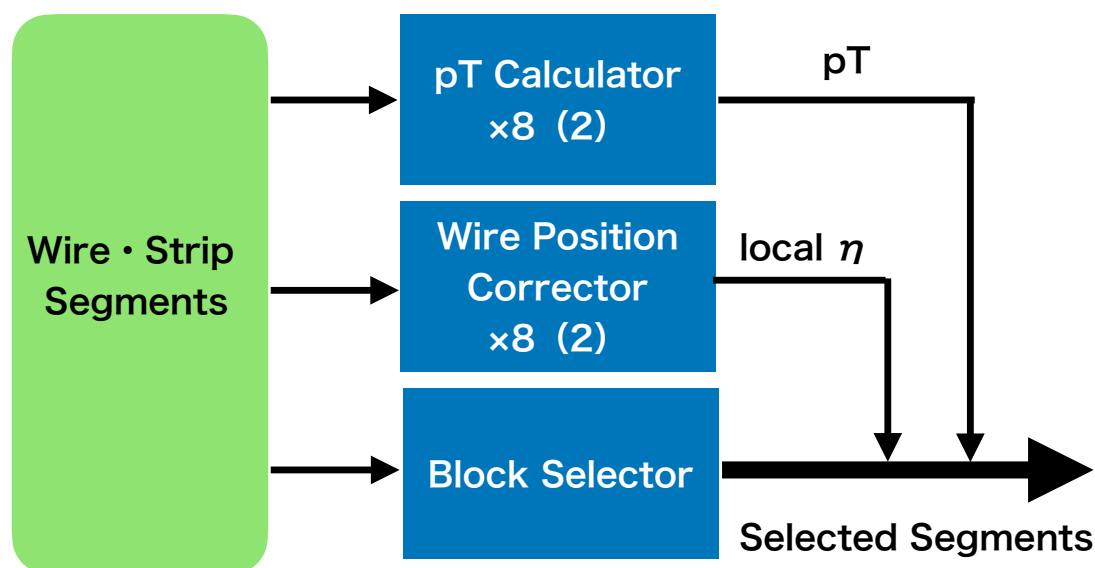


図 3.14 Wire Strip Coincidence ファームウェアの概要 [11]。Wire segment および Strip segment の情報をもとに、 p_T を計算する p_T Calculator、local な η 情報を計算する Wire Position Corrector、複数のミュオン飛跡候補から後段に送信するものを選択する Block Selector で構成される。

3.1.5 Inner Coincidence

概要

Inner Coincidence では TGC BW コインシデンスで再構成された飛跡候補と磁場領域内部の検出器 (NSW、RPC BIS78、TGC EI、Tile カロリメーター) でコインシデンスをとることで、衝突点に由来しないフェイクトリガーの削減と p_T 分解能の向上を図る。図 3.15 に各検出器がカバーする η 、 ϕ 領域を示す。 $1.3 < \eta < 2.4$ の領域は NSW が網羅的にカバーしており、 $1.05 < \eta < 1.3$ 領域では TGC EI、RPC BIS78、Tile カロリメーターがそれぞれ相補的にカバーしている。

磁場内部に位置する検出器はそれぞれ異なる特徴を持っているため、コインシデンスをとる検出器ごとに独立したロジックが用意されている。ここでは具体例として NSW とのコインシデンスロジックについて説明する。

図 3.16 に NSW とのコインシデンスアルゴリズムの概要を示す。衝突点から飛来するミュオンは NSW にヒットを残した後、エンドキャップトロイド磁場により主に η 方向に曲げられ、TGC にヒットを残す。そのため TGC BW Coincidence で再構成された η 位置 (η_{TGC}) と NSW で再構成された η 位置 (η_{NSW}) の差は p_T と相関をもつ。この差を $d\eta$ と定義する。

$$d\eta = \eta_{TGC} - \eta_{NSW} \quad (3.1)$$

一方、衝突点から飛来するミュオンは検出器内部の物質 (主に物質の大きいカロリメーター) と多重散乱を起こすことがあり、この場合では $d\eta$ のみでは p_T が高いものと低いものを見分けることができない (図 3.16 中参照)。そこで、ミュオンが NSW に入射した角度 ($\Delta\theta_{NSW}$) を分別変数として追加することで、この問題を克服する。この計算には $d\eta$ 、 $\Delta\theta_{NSW}$ を入力、 p_T を出力とした Coincidence Window を利用する。

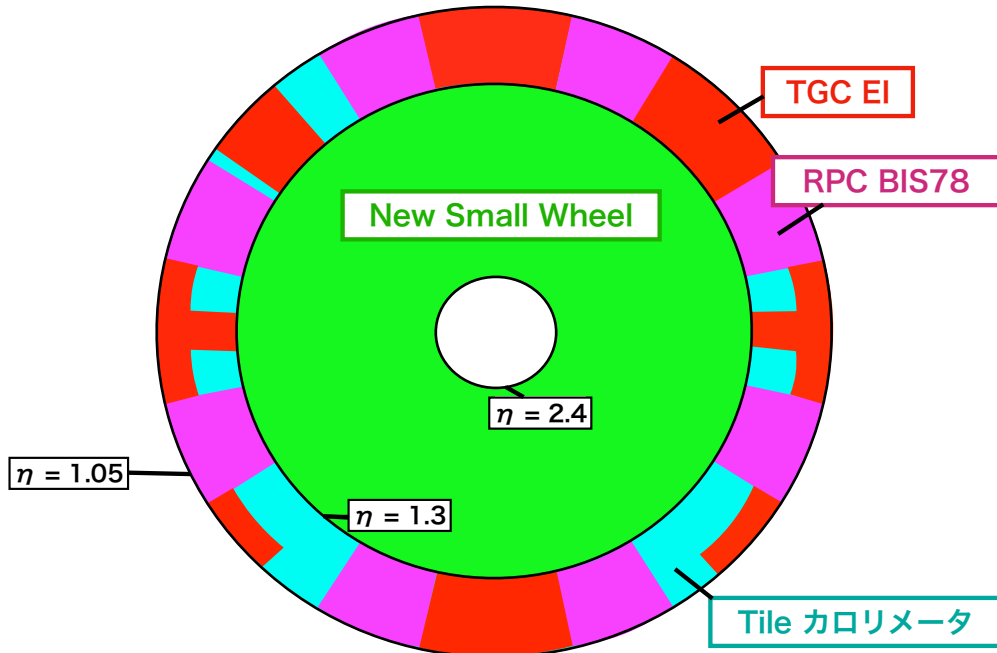


図 3.15 各磁場内部の検出器がカバーする η , ϕ 領域 [20]。 $|\eta| < 1.3$ 領域は NSW がカバーする。 $1.3 < |\eta|$ 領域は TGC EI、RPC BIS 78、Tile カロリメータがそれぞれ相補的にカバーしている。

Inner Coincidence の論理回路実装

このモジュールの駆動クロックは、LHC クロックに同期した周波数 160 MHz のクロックで、レイテンシーは 7 クロックチック分 (43.75 ns) である。Inner Coincidence も Region を 1 単位として並列に処理が行われる。 η 位置ごとにどの検出器とコインシデンスをとるかが決められており、 $|\eta| < 1.3$ の領域では主に RPC、EI、Tile カロリメータと、 $|\eta| > 1.3$ の領域では主に NSW とコインシデンスをとる。8 Unit Region は Wire Strip Coincidence から 1 つの飛跡候補を受け取り、磁場内部の検出器とコインシデンスをとった後、1 候補を出力する。32 Unit Region は Wire Strip Coincidence から 4 つの飛跡候補を受け取り、2 つまで候補を絞って出力する。その結果 SL 全体では最大 112 候補を出力する。

Inner Coincidence の各 Unit 内でのロジックの概要を図 3.17 に示す。各 Unit は Decoder、Coincidence、Which Inner で構成される。以下に例として NSW とのコインシデンスをとる Region について説明する。

- Decoder

1 つの SL は NSW で再構成された飛跡を最大 16 個受け取る。この中から TGC BW コインシデンスで再構成された飛跡候補と組み合わせる 4 候補を選ぶのが Decoder である。ここでは p_T が大きいミューオンの検出効率を高く保つため、 $|d\eta|$ が小さいものを優先的に選別する。

- NSW Coincidence

TGC BW の 1 つの飛跡候補と NSW の 4 つの飛跡候補でコインシデンスをとる。Coincidence Window は FPGA の URAM に格納されており、8 bit の $d\eta$ 情報と 4 bit の $\Delta\theta$ を入力、4 bit の p_T を出力とする。計算された飛跡のうち、 p_T が最大の候補が保存され 40 MHz ごとに出力される。

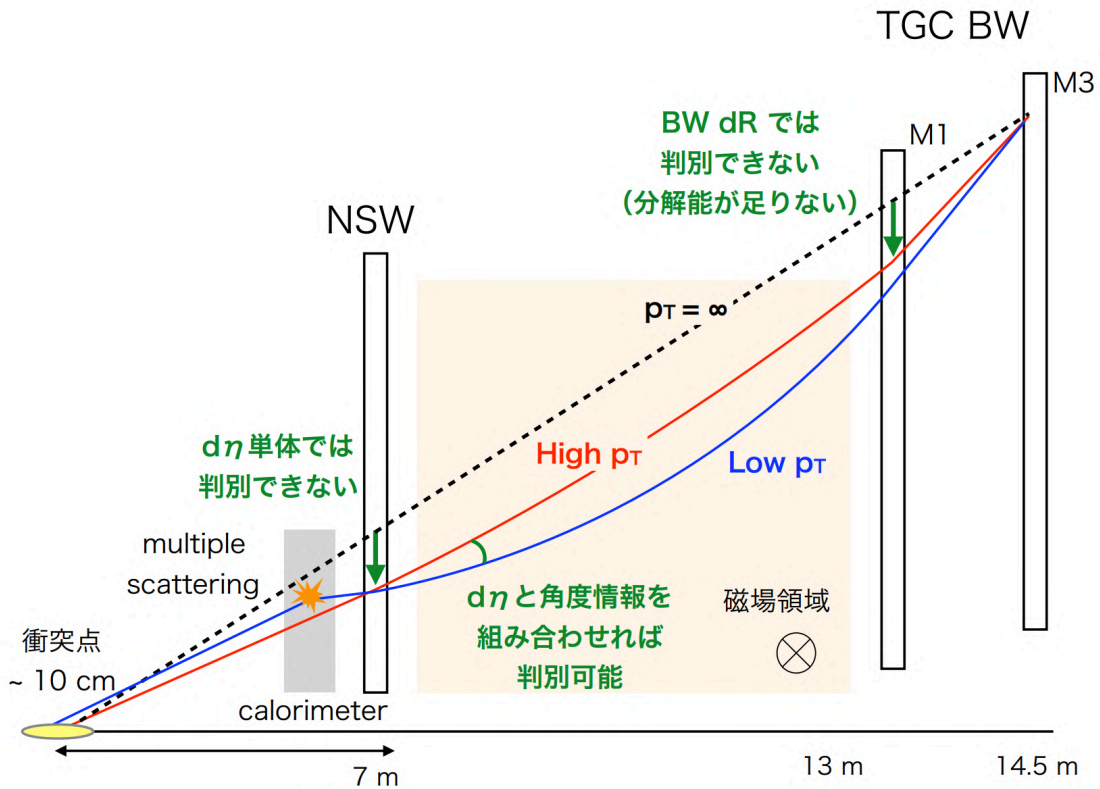


図 3.16 NSW を用いたコインシデンスロジックの概要 [10]。 p_T が小さいミュオンほど磁場領域で曲げられるため、基本的には $d\eta$ を用いることで p_T を判定することができる。しかし、青線で示すように NSW に入射する前に多重散乱を受けた場合には、 $d\eta$ だけでは High p_T のミュオンと Low p_T のミュオンを判別することができない。そこで NSW に入射した角度 ($\Delta\theta_{NSW}$) を分別変数として追加することで、より正確に p_T を判定することができる。

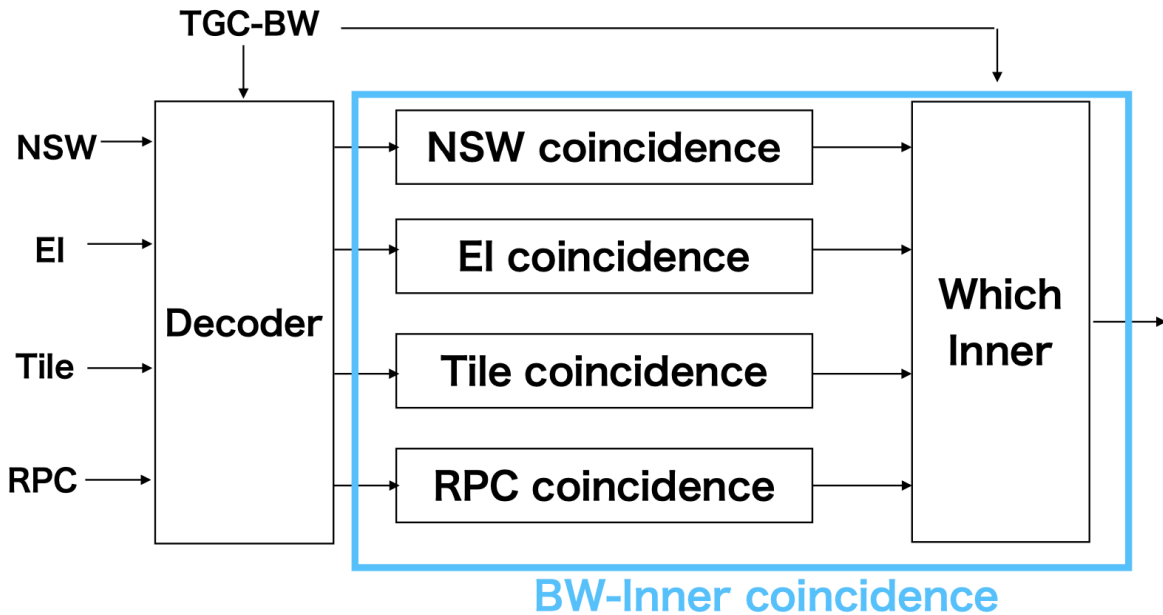


図 3.17 Inner Coincidence の概要 [22]。 Inner Coincidence は NSW から受け取った飛跡候補のうち、コインシデンスに利用する候補を選ぶ Decoder、各検出器とのコインシデンスロジック、どの検出器とのコインシデンス結果を採用するか選択する Which Inner で構成される。

表 3.6 Inner Coincidence における飛跡候補のデータフォーマット

# of bits	Name	Comment
14	TGC η	η in global candidate at the pivot plane of TGC
9	TGC ϕ	ϕ in global candidate at the pivot plane of TGC
8	TGC p_T	TGC transverse momentum
4	TGC p_T threshold	TGChighest p_T threshold satisfied
1	TGC charge	TGC TC charge
3	Coincidence type	Identifier of coincidence type
14	MDT η	η coordinate of the MDT segment position of the innermost station
8	MDT p_T	MDT transverse momentum
4	MDT p_T threshold	MDT highest transverse momentum threshold satisfied
1	MDT charge	MDT charge
4	MDT Processing Flag	Type of reconstructed muon
2	# of segments	Number of MDT segments associated to the muon
3	Segment quality flag	Quality of each segment
49	Reserved	

- Which Inner

Which Inner モジュールは複数の検出器とのコインシデンスロジックが並列に走る $|\eta| < 1.3$ の領域において、どの検出器とのコインシデンス結果を最終出力とするかを決定する。

Inner Coincidence から Track Selector に送られる飛跡候補のデータフォーマットを表 3.6 に示す。

3.1.6 Track Selector

概要

Track Selector は Inner Coincidence から出力される最大 112 個のミューオン飛跡候補から MUCTPI に送る 6 候補を選ぶ。このとき Inner Coincidence で判定された p_T が高いものを優先的に選択する。

Track Selector の論理回路実装

モジュールの駆動クロックは LHC クロックに同期した周波数 160 MHz のクロックで、レイテンシーは 5 クロックチック (31.25 ns) である。概要を図 3.18 に示す。Track Selector は 112 個のミューオン飛跡候補を p_T が大きい順に並び替えるソーティングロジックとして実装される。特に Batcher の奇遇マージソート法 [23] という、高速かつ並列にソーティングを行うメソッドをデジタル回路として実現している。具体的には 16 個の 8-key sorting network と 15 個の 16-key merging network で構成されており、インプットが 128、アウトプットが 8 のソーティングロジックを構成している。8-key sorting network および 16-key merging network の概要を図 3.19、図 3.20 に示す。横線はワイヤーを示し、縦線がコンパレーターを表す。左から右に 1 つの飛跡候補同士の比較が行われ、その集合体としてロジックが実装される。

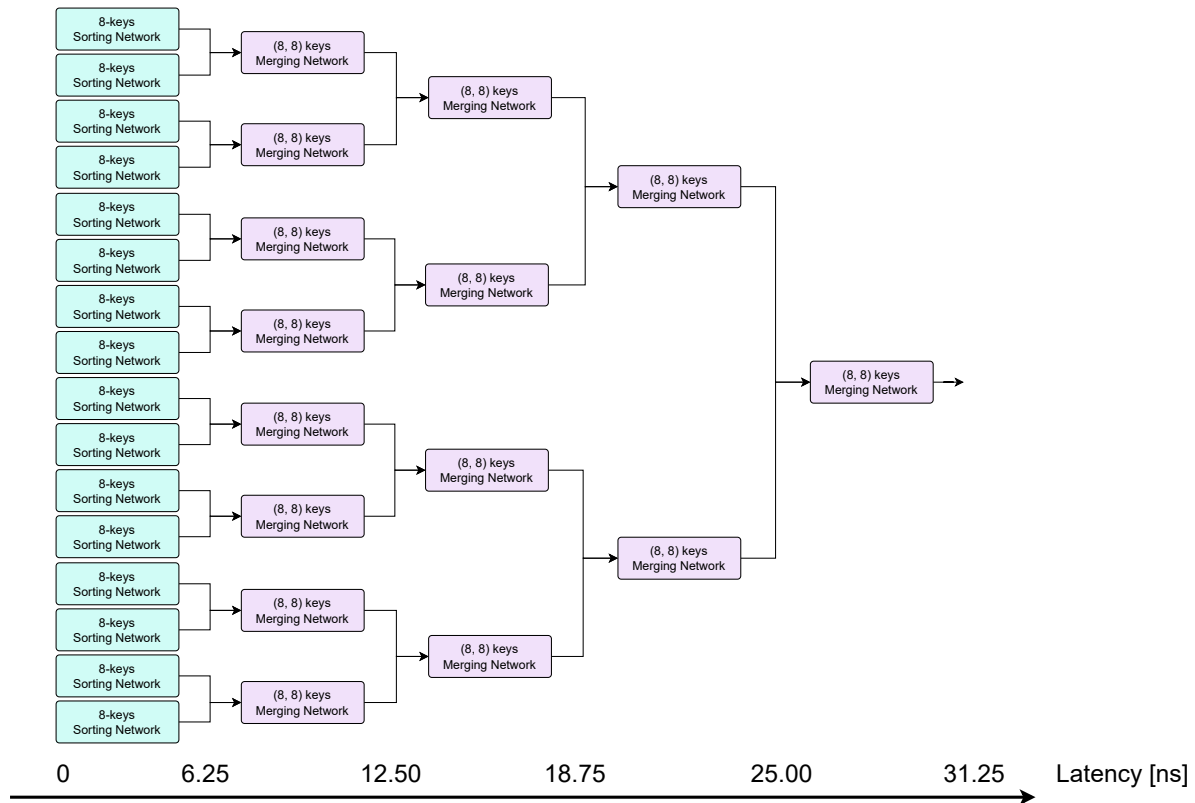


図 3.18 Track Selector の概要。16 個の 8-key sorting network、と 15 個の 16-key merging network を用いて、128 個のミュオン候補から p_T が大きい順に 8 候補を選択する。このうち 6 個を MUCTPI に送信する。

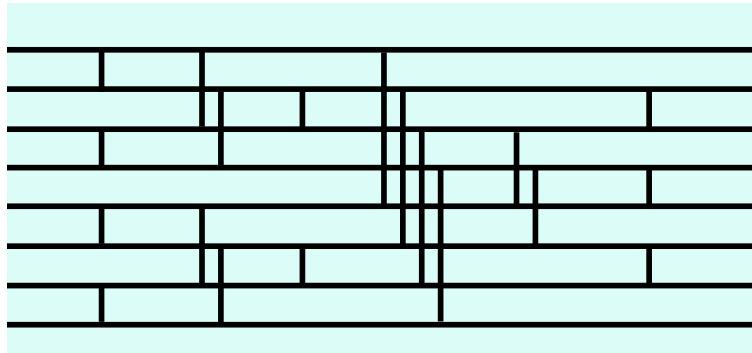


図 3.19 8-key sorting network の概要。横線はワイヤーを表し、縦線はコンパレーターを表す。

3.2 トリガー論理回路の統合

2章で述べたように、SL FPGA は 4 つのシリコンダイで構成されており、トリガーロジックはそれぞれの SLR 上に並列に配置される (図 3.21)。PS board とのインターフェイスが実装される SLR 0・2・3 には、TGC BW Coincidence が実装される。トリガーセクターごとに SLR が分けられ、SLR 0 に Endcap $\phi 0$ 、SLR 2 に Endcap $\phi 1$ 、SLR 3 に Forward のロジックが配置される。磁場内部の検出器や MUCTPI、MDTTP とのインターフェイスは SLR 1 に実装され、Inner Coincidence、Track Selector が配置される。

高輝度 LHC-ATLAS 実験に向けたトリガーロジックの開発は、これまでに各トリガーモジュールの論理回路実

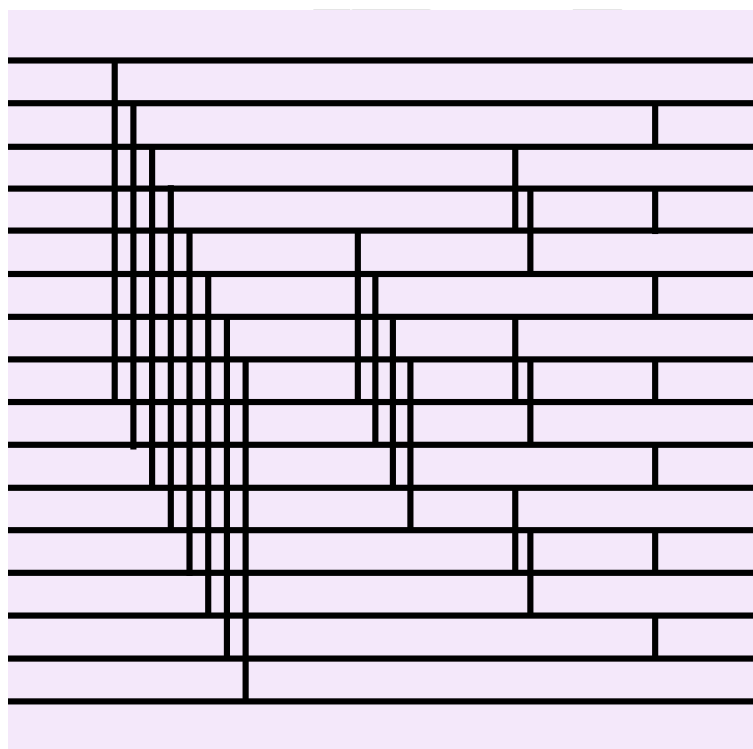


図 3.20 16-key merging network の概要。横線はワイヤーを表し、縦線はコンパレーターを表す。

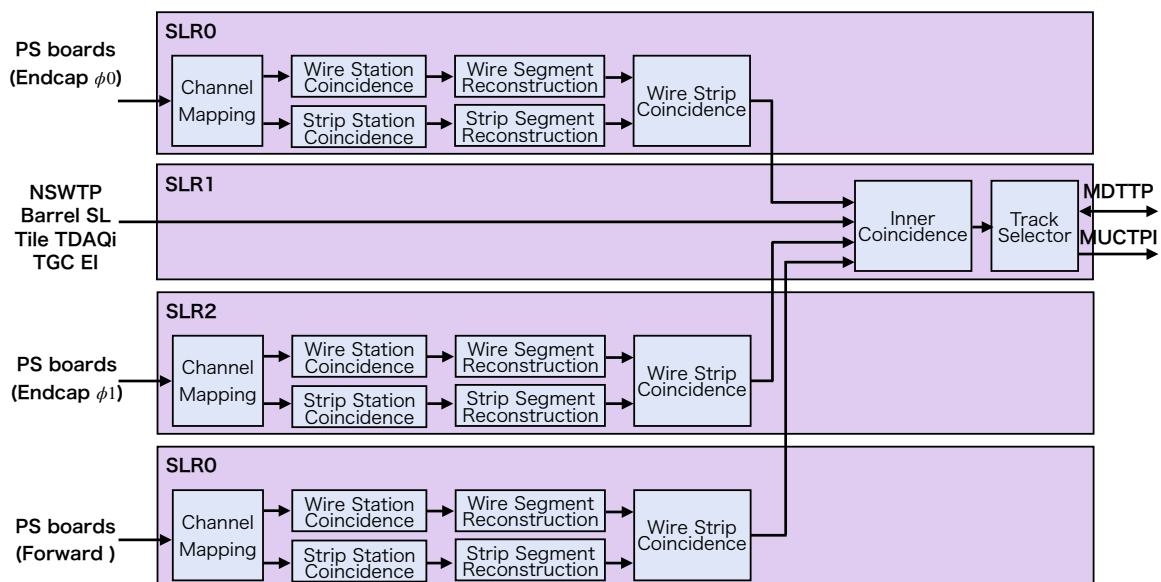


図 3.21 SL FPGA 内におけるトリガーロジックの配置。SLR 0・2・3 には、PS board とのインターフェイスと TGC BW Coincidence が実装される。トリガーセクターごとに SLR が分けられ、SLR 0 に Endcap ϕ_0 、SLR 2 に Endcap ϕ_1 、SLR 3 に Forward のロジックが配置される。SLR1 には磁場内部の検出器や MUCTPI、MDTTP とのインターフェイスが実装され、Inner Coincidence、Track Selector が配置される。

表 3.7 Wire Strip Coincidence における信号線

接続するモジュール	信号線 (bit 幅)
Wire segment Reconstruction	Wire segment (EC : 148 subunit \times 20 bit, FW : 64 subunit \times 20 bit)
Strip segment Reconstruction	Strip segment (EC : 20 unit \times 17 bit, FW : 5 unit \times 17 bit)
LUT Manager for p_T LUT	Address + Data (EC : 35 region \times 18 bit, FW : 8 unit \times 18 bit)
LUT Manager for η LUT	Address + Data (EC : 35 region \times 17 bit, FW : 8 unit \times 17 bit)

装が完了し、Segment Reconstruction まで全体ファームウェアへの統合が進められている。本研究では引き続き Wire Strip Coincidence、Inner Coincidence、Track Selector の統合を進め、一連のトリガーロジックの統合を完了させる。以下にそれぞれの実装について述べる。

3.2.1 Wire Strip Coincidence の統合

Wire Strip Coincidence を統合ファームウェアに組み込むため、前段の Wire segment Reconstruction、Strip segment Reconstruction の出力を配線した。また、モジュール内に存在する p_T 閾値概算用の LUT と η 位置算出用の LUT をコンフィギュレーションするパスをそれぞれ接続した。Wire Strip Coincidence モジュールに接続した各信号線とその bit 幅について表 3.7 にまとめる。

Wire Strip Coincidence は Wire segment Reconstruction の 1 つの Subunit から 1 つの飛跡情報を受け取る。1 つの飛跡情報は 20 bit で構成され、8 bit の $\Delta\theta$ 、10 bit の M3 代表点 ID、2 bit のマッチレイヤーの情報が含まれる。Endcap 領域には合計 148 の Subunit が存在するため、合計 2960 bit、Foward 領域では 64 の Subunit が存在するため、合計 1280 bit の信号線が接続される。Strip segment Reconstruction からは 1 つの Unit から 1 つの飛跡情報を受け取る。1 つの飛跡情報は 17 bit で構成され、9 bit の $\Delta\phi$ 、6 bit の M3 代表点 ID、2 bit のマッチレイヤー情報が含まれる。Endcap 領域には計 20 の Unit が存在するため、合計 340 bit、Foward 領域では 5 つのユニットが存在するため、合計 85 bit の信号線が接続される。

Wire Strip Coincidence モジュール内に配置されている p_T 出力用の BRAM と η 出力用の BRAM のコンフィギュレーションは MPSoC から行う。MPSoC からの書き込みを仲介するためのモジュールとして Wire Strip Coincidence 用の LUT Manager を実装した。MPSoC から各モジュールの LUT Manager までは RAM ID、Data、address、Write enable という共通の信号線を接続する。LUT Manager は RAM ID をモニターし、担当するモジュールに該当する RAM ID が指定されたときのみ、address と Data をラッチして RAM に書き込み動作を行う。これにより MPSoC から各 RAM に個別に信号線を用意する必要がなく、MPSoC から統一的に LUT をコンフィギュレーションすることができる。また、モジュール内の BRAM は LHC バンチ交差クロックを逡倍した 160 MHz クロックで動作するのに対し、LUT の書き込みは安定供給が保証されているオンボード発振器を基とする 50 MHz クロックで行われる。この 2 つは異なるクロックソースを使用しているため、クロックドメインを適切にまたぐための仕掛けも実装した。

3.2.2 Inner Coincidence の統合

Inner Coincidence を実機上で稼働させるために、Wire Strip Coincidence の出力、磁場内部の検出器からの飛跡情報、LUT のコンフィギュレーションパスを実装した。Inner Coincidence モジュールに接続した各信号線とその bit 幅を表 3.8 にまとめる。

Inner Coincidence は Wire Strip Coincidence の 8 Region から 1 つ、32 Region から 4 つの飛跡情報を受け取

表 3.8 Inner Coincidence における信号線

接続するモジュール	信号線 (bit 幅)
Wire Strip Coincidence SLR 0	8 region (31 bit x 1 cand x 22 regions) + 32 region (31 bit x 4 cand x 13 regions)
Wire Strip Coincidence SLR 2	8 region (31 bit x 1 cand x 22 regions) + 32 region (31 bit x 4 cand x 13 regions)
Wire Strip Coincidence SLR 3	32 region (31 bit x 4 cand x 8 regions)
LUT Manager for NSW coincidence	Address + Data (86 bit x 78 region)
LUT Manager for RPC coincidence	Address + Data (21 bit x 46 region)
LUT Manager for EI coincidence	Address + Data (15 bit x 46 region)
LUT Manager for Tile coincidence	Address + Data (18 bit x 46 region)
Test Register for NSW	19 bit x 16 cand
Test Register for RPC	24 bit x 4 cand
Test Register for EI	22 bit x 4 cand
Test Register for Tile	16 bit x 4 cand

る。1つの飛跡情報は31bitで構成され、1bitのvalid信号、4bitの p_T 閾値、8bitの η positionに加え、計17にリダクションしたWire segmentおよびStrip segmentが含まれる。そのため、エンドキャップ領域を処理するSLR 0、SLR 2からそれぞれ4,650bit、フォワード領域を処理するSLR 3から992bitの信号線がSLRを跨いで接続される。

Inner CoincidenceはNSW、EI、Tile、RPCからの位置情報や角度情報を入力として利用する。しかし、今のところ磁場内部の検出器から送られるデータの詳細が決まっておらず、それを受け取るためのインターフェイスも実装されていない。そこで、今回はMPSoCから書き込むことができる試験用のレジスタをこれらの代わりに接続した。各種LUTへのコンフィギュレーションは、Inner Coincidence用のLUT Managerを実装し、Wire Strip Coincidenceと同様に配線した。

Inner Coincidenceは112個のミューオン飛跡候補を選別し、1飛跡候補につき128bitの情報を出力する。これらの信号線はTrack Selectorと接続した。Track Selectorから出力される6候補×128bitの信号は、現状MDTTPやMUCTPIへのインターフェイスは実装中であるため、MPSoCから読み出せる試験用のレジスタと配線した。

3.2.3 トリガーロジック統合後のリソース使用量

本研究により、トリガーロジックのすべてのモジュールがSLFPGAへ統合され、トリガーロジック全体を含むリソース使用量を初めて見積もることができるようになった。リソース使用状況を表3.9に示す。表中の値は、1つのSLR中のリソースに対する使用量の割合を百分率で表したものである。Totalにはトリガーロジックだけでなく、コントロール、リードアウトなどのロジックも含めた統合ファームウェア全体のリソース使用状況を示している。また、現状、トリガーロジックはタイミング違反を起こすことなく実装を完了しているが、そのために幾つかの最適化研究を行なった。この詳細はAppendix Dで説明する。

表 3.9 最適化後のデバイスのリソース使用状況

Name	Block	LUT (17280000)	REG (34560000)	CLB (2160000)	LUT as Memory (791040)	BRAM (2688)	URAM (1280)
SLR 0 EC ϕ 1	Wire Station Coincidence	7.4	1.48	22.2	0	0	0
	Strip Station Coincidence	0	0.2	0.84	0	0	0
	Wire segment Reconstruction	16.28	4.44	26.64	0	0	45.88
	Strip segment Reconstruction	6.24	3.08	13.44	0.08	0	2.43
	Wire Strip Coincidence	2.4	2.92	10.68	0	37.52	0
	Total	57.2	25.68	91.76	3.6	74.4	51.56
SLR 1	Inner Coincidence	66.92	18.88	87	3	28.88	50
	Track Selector	7.56	2.8	17.56	0	0	0
	Total	79.16	26.52	99.24	4.88	28.88	50
SLR 2 EC ϕ 1	Wire Station Coincidence	8.88	1.48	22.2	0	0	0
	Strip Station Coincidence	0	0.2	0.92	0	0	0
	Wire segment Reconstruction	14.8	4.44	29.6	0	0	45.88
	Strip segment Reconstruction	6.24	3.08	13.6	0.08	0	2.43
	Wire Strip Coincidence	2.44	2.92	11.28	0	37.52	0
	Total	58.64	27.04	93.25	3.96	80.2	51.56
SLR 3 FW	Wire Station Coincidence	1.44	0.64	3.36	0	0	0
	Strip Station Coincidence	0	0.04	0.12	0	0	0
	Wire segment Reconstruction	6.4	1.28	10.88	0	0	19.84
	Strip segment Reconstruction	1.24	0.6	2.64	0.04	0	1.24
	Wire Strip Coincidence	1.48	1.4	4.4	0	14.28	0
	Total	25.36	13.36	50.12	1.84	33.24	20.32

第 4 章

高輝度 LHC-ATLAS 実験に向けたエンド キャップ部ミュオントリガー回路の性能 評価

4.1 トリガー論理回路試験システムの開発

4.1.1 シングルボード試験システムのコンセプト

高輝度 LHC-ATLAS 実験に向けたエンドキャップ部ミュオントリガー開発は、これまでにモジュールごとの論理回路実装が完了し、前章までに全体ファームウェアへの統合が完了した。精度の高いミュオントリガーを実現するためには、実装したトリガー論理回路が正常に動作していること、また期待したトリガー性能を実現できていることを SL 実機を用いて検証することが重要である。しかし、実験開始前の段階でハードウェア上で動作する大規模トリガー回路をどのように試験するかというのは、一般的な課題であり、工夫が必要となる。SL においても、ヒットビットマップの送信を担う PS board やトリガーデータの読み出しを行う FELIX とのシステム統合が進むまでは、実機を用いてトリガー性能を検証することは困難である。本研究では、SoC を活用したシングルボード試験システムの開発により、この課題を克服した。

シングルボード試験システムの設計において、以下の 4 つの要求を満たすシステムが有用であると考えた。

1. 前段回路 (PS board) や読み出し回路 (FEILX) を用いず SL 単体で試験できること
2. シミュレーションデータや過去の Run で取得した実データなど、現実的なデータセットをトリガー回路に入力し、その応答を調べることができること
3. 大統計量での試験ができること
4. トリガー回路が意図しない挙動を示した場合、系統的に問題の切り分けができること

トリガー回路は大規模な論理回路であるため、その開発・修正には時間を要する。PS board や FELIX の開発状況に左右されずに、SL 単体で試験を進めることは、精度の高いトリガーの実現に向けて重要である。また、トリガーの運用に際して、ファームウェアのアップグレードが必要な場合でも、更新したファームウェアに対して迅速に試験することができるシステムを有していることの価値は大きい。2つ目、3つ目の観点はトリガー論理回路を詳細に調査するために重要である。大統計量のデータを用いることにより、トリガー回路の包括的かつ詳細な検証が行え、論理回路実装の局所的な不具合も発見しやすくなる。テストベクターだけでなく現実的な飛跡データも使用することで、トリガー効率、運動量分解能、トリガーレートなどを実機を用いて直接評価することができるようになる。また、試験においてトリガー回路から予期しない応答が得られた場合に、その問題を切り分けて議論でき

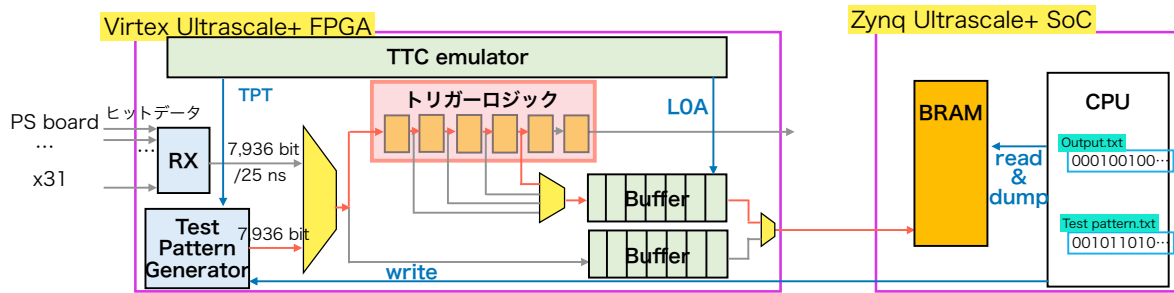


図 4.1 トリガー試験ファームウェアの概要。MPSoC 上の CPU 領域に起動した Linux を起点に、試験システムのコントロール、データ読み出しを行う。システムは SL 上の水晶発振器で生成した 40.079 MHz クロックを LHC 陽子バンチ交差クロックの代わりに利用する。FELIX から分配される TPT、LOA、BCR などの TTC 信号は SL 内部の TTC emulator で生成する。トリガー回路へのヒットビットマップの入力は Test Pattern Generator から行う。トリガー回路の出力は読み出し回路を通じて MPSoC 上の BRAM にダンプし、それを Linux から読み出すことでトリガー演算結果を取得する。

ることは、効率的なデバッグに重要である。

本研究では SL が FPGA と CPU が一体となった拡張性に富んだデバイスである SoC を搭載していることに着目し、上述の要請を満たす実機試験システムを開発した。以下にその詳細を述べる。

4.1.2 シングルボード試験システムの実装

ファームウェアの全体像

図 4.1 に実装したシングルボード試験システムの全体像を示す。トリガー回路のデータ入力及びデータ読み出しは MPSoC 上の Linux から制御する。SL 上の水晶発振器で生成した 40.079 MHz クロックを LHC 陽子バンチ交差クロックの代わりとし、基準クロックに利用する。FELIX から配られる Test Pulse Trigger (TPT) や LOA などの TTC 信号も FPGA 上の TTC emulator で擬似的に生成する。これにより PS board や FELIX を用いない SL 単体での試験を実現する。

トリガー回路へのデータ入力は Test Pattern Generator が担う。Test Pattern Generator は PS board から受信するヒットビットマップをエミュレートした試験用データ (テストパターン、合計 256 bit x 31 PS board = 7936 bit / BC) を BRAM に格納し、TPT ごとに 1 BC 分のデータをトリガー回路に入力する。このように、PS board からの出力自体をエミュレートするシステムにすることで、Channel Mapping から Track Selector までトリガー回路全体を通して試験することができる。またこの設計により、元のデータセットの種類に関わらず、任意のデータセットをテストパターンとして統一的に扱うことができ、汎用的な利用が可能となっている (テストパターンの作成方法は後述する)。さらに、Test Pattern Generator 内の BRAM は MPSoC 上の Linux から何度でも書き換えることができるため、任意のイベント数で試験を行うことができる。

トリガー回路にはその中間データをプローブするための読み出し回路を実装する。トリガーアルゴリズムの適した箇所、適した情報を選択的に読み出す機構を実装することにより、トリガー回路内部で行われた処理を系統的に調査することが可能である。中間データは読み出し回路内部で効率的に圧縮され、データ量を削減した上で、高速チップ間通信を利用して MPSoC へ送られる。

表 4.1 Wire Strip Coincidence の読み出しフォーマット。

# of bits	Name
1	The flag indicates that the region has received at least one wire segment and strip segment
4	p_T threshold reconstructed using Coincidence
3	Reserved
7	$\Delta\theta$ from Wire Segment Reconstruction
2	Number of stations with the hits used for wire segment
4	$\Delta\phi$ from Strip Segment Reconstruction
2	Number of stations with the hits used for strip segment
1	Reserved

Test Pattern Generator

Test Pattern Generator は幅 128 bit、深さ 64 列の BRAM にテストパターンを保存し、TTC emulator から TPT が供給されたタイミングで 1BC 分のデータをトリガー回路に投入する。1 つの BRAM が PS board 1 リンク分の信号を保存する。全 31 PS board \times 2 リンクの信号をエミュレートするため、Test Pattern Generator には合計 62 個の BRAM が並列に配置される。

BRAM には coe 形式のファイルを利用して初期値を設定できるほか、MPSoC から何度でもテストパターンを上書きすることができる。そのため、BRAM の深さに制限されることなく任意のイベント数を用いた試験を行うことができる。以下にそれぞれの機能の実装について述べる。

トリガー回路中間状態の読み出し回路

6 段階のトリガーモジュールで構成されるトリガー回路には、各モジュールの出力をプローブするための読み出し回路を実装する。トリガー読み出しでは、トリガーモジュール間で渡されるデータの中から、アルゴリズムを調査する上で必要なデータのみを選択的に読み出すことで、効率の良い論理回路の検証ができるようになっている。

例として Wire Strip Coincidence の出力として読み出すデータを説明する。Wire Strip Coincidence は 8 Unit region から 1 つのミューオン候補を、32 Unit region から 4 つのミューオン候補をそれぞれ出力し、最大 180 個のミューオン飛跡候補を Inner Coincidence に送る。読み出し回路はこの最大 180 個のミューオン候補を並列に読み出せるよう設計しており、各領域ごとに表 4.1 のフォーマットに飛跡情報を成形する。各モジュールの出力のうち、最上位 1 bit を valid 信号として定義する。後段の Candidate Selector は valid 信号が立てられた領域の情報のみを選択的に読み出すため、効率的にデータを削減することができる。

それぞれのモジュールから読み出すデータの出力ビット幅を表 4.2 に示す。読み出し回路において、数千ビットの信号を一度に処理しようとする、タイミング制約を満たしてロジックを配置することが難しくなる。そこで bit 幅が大きな信号線については、いくつかの並列な処理レーンに分割して読み出しを進める。また、全てのトリガーモジュールの読み出し回路を同時に実装するのはリソースの観点で不可能である。そのため、目的に応じてどのトリガーモジュールの出力を読み出すか選択し、ファームウェアを分けて試験を行う。読み出し回路の概要を図 4.2 に示す。以下に各モジュールの役割を説明する。

表 4.2 各モジュールの出力ビット幅。

トリガー回路	SLR	出力ビット幅 (bits)	レーン数
Channel Mapping	0	2732	4
	2	2732	4
	3	1000	2
Wire Station Coincidence	0	1768	3
	2	1768	3
	3	804	3
Strip Station Coincidence	0	1890	3
	2	1890	3
	3	378	3
Wire Segment Reconstruction	0	3404	5
	2	3404	5
	3	1472	2
Strip Segment Reconstruction	0	360	1
	2	360	1
	3	72	1
Wire Strip Coincidence	0	1776	3
	2	1776	3
	3	768	1

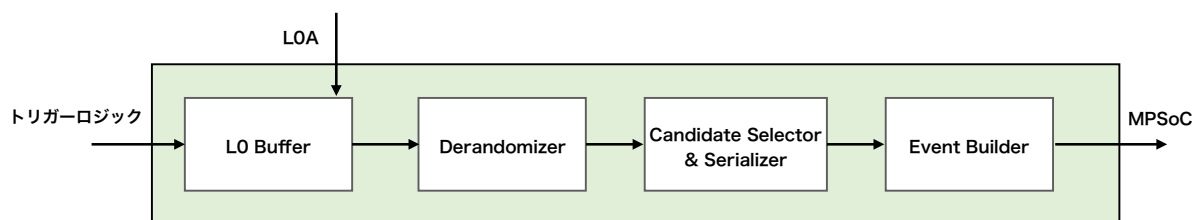


図 4.2 トリガー読み出し回路の概要。トリガー回路からの出力は L0 Buffer に格納され、LOA が出されたイベントのみ Derandomizer に送信される。トリガーモジュールからの出力は、すべてのユニットからの出力を並列に並べてあるが、Candidate Selector はこの中から飛跡を再構成できたユニットからの出力のみを抜き出して、Event Builder に送信する。Event Builder では複数のモジュールからの出力を 1 つのパケットにまとめて、MPSoC へと送る。

L0 Buffer

トリガー読み出しに際してフォーマットされたデータは L0 Buffer にダンプされる。L0 Buffer は入力ビットマップと同じ幅をもつ深さ 512 列の BRAM あり、TTC emulator から LOA が出されるまでのバッファリングを行う。トリガーモジュールには 40 MHz、160 MHz、240 MHz で駆動するものが存在するが、いずれのモジュールでも 1 つの陽子バンチ交差由来の出力は 25 ns ごとに横並びに揃えられ、40 MHz の LHC クロックで動作する書き込みポインタに従って、L0 Buffer に格納される。一方、読み出しは 240 MHz で行われ、読み出されたデータは Derandomizer に送られる。

表 4.3 Serializer から Event Builder に送られるパケットのフォーマット

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE	rsvd				data loss flag				0xB0E												LOID											
EOE	rsvd				data loss flag				0xB0E												LOID											
data word	Unit address				BC tag				bitmap																							
buffer	0xBFF								rsvd												LOID											

Derandomizer

Derandomizer は入力ビットマップと同じ bit 幅をもつ深さ 512 列の FIFO で、後段で行われる圧縮およびシリアル化の処理待ち用バッファとして動作する。後段の Candidate Selector から読み出し命令を受け取るたびに、FIFO の先頭のイベントを出力する。

Candidate Selector & Serializer

Candidate Selector は各トリガーモジュールのすべてのユニットから出力される飛跡候補の中から、ミューオン候補を再構成できたユニットの出力のみを選択して後段に送る。これにより大幅なデータ削減が達成される。選択された飛跡情報には、どのサブユニットからの出力かを示す 6 bit の識別 ID と 2bit の bunch tag が付与され、32 bit 幅のワードとして後段の Serializer に送信される。Serializer は 32 bit 幅のワードをクロックチックごとに 1 ワードずつ処理し、ある LOA に対応する 1 イベント分のパケットをまとめて、後段の Event Builder に渡す。図 4.3 にパケットのフォーマットを示す。パケットにはイベント境界を示すための特別なワード Start of Event (SOE)、End Of Event (EOE) が挿入される。

Event Builder

上述の Candidate Selector までは読み出しレーンごとに処理が行われる。Event Builder はこれらの並列なレーンの出力を直列化し、MPSoC に送信するためのフォーマットに成形する。Serializer から受信する 32 bit のワードは Event Builder 内の処理待ちバッファに格納される。Event Builder は処理待ちバッファのデータを 240 MHz のクロックチックごとに 1 ワードずつ図 4.3 のフォーマットに成形する。meta switch は読み出しを行うトリガーモジュールの選択状態を表す。meta data は各トリガーモジュールごとに送られ、そのトリガーモジュールのラベル (data flavor)、読み出しを行う bunch tag (previous, current, next, next to next) の選択状態を表す。実機試験ではこのように FPGA 内でエンコードされたデータを、ソフトウェア上で逐次的にデコードすることで、トリガー回路からの出力を一意に再構成して、テキストファイルにダンプする。

SL FPGA から MPSoC へのデータ転送

SL FPGA で成形されたデータは MPSoC PL 領域内の BRAM に格納されたのち、PS から読み出される。SL FPGA から MPSoC へのデータ転送は、高速シリアル通信を利用する。図 4.4 にチップ間通信の概要を示す。

Event Builder から出力される 32 bit 幅のワードは AXI sender に送られる。AXI sender は受信したデータをデータ幅 32 bit、アドレス幅 32 bit のバス通信である AXI プロトコルで、AXI chip2chip へ送る。AXI chip2chip は AXI 形式のデータを AXI Stream 形式に変換し、Aurora 64B/66B Master と通信する。AXI chip2chip 以降のデータ転送では AXI バーストと呼ばれる、連続したデータブロックを高速転送するための手法が採られている。AXI バーストは Master から Slave に送信するワード数を事前に定義 (本システムでは 256 ワード) することで、通常のハンドシェイキングで生じるバスのアイドルタイムを最小限にする。Aurora 64B/66B Master は AXI Stream 形式のデータをシリアル通信のデータにエンコードし、ギガビットトランシーバーを用いてチップ間通信を行う。

シリアルデータは、MPSoC の PL に実装された Aurora 64B/66B Slave で受信され、再び AXI Stream 形式にデ

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Start of Packet	Start								D5.6								D5.6								K28.1											
Header 0	Trigger Type																Format version																			
Header 1	Run Type																Run Number								Rsvd				Fiber ID				Sector Logic ID			
Header 2	Rsvd																LOID [31:0]																			
Header 3	Rsvd																BCID								Rsvd				LOID [37:32]							
Header 4	Enable Data Type bitmap [31:0]																																			
Header 5	Enable Data Type bitmap [63:32]																																			
Header 6																																				
Trigger Data Header (not zero-suppressed)	Data type (e.g.type=0 for data for MUCTPI, max 64 types)								Words per cand				BC readout eable bitmap (NN, N, C, P)				Number of candidates (Next-to-Next BC)				Number of candidates (Next-to-Next BC)				Number of candidates (Current BC)				Number of candidates (Previous BC)							
Data	Data																																			
Trailer 0	Data loss flag bitmap [31:0]																																			
Trailer 1	Data loss flag bitmap [63:32]																																			
Trailer 2	Rsvd																Number of PP ASIC hit words (max 992)																			
Trailer 3	Error bitmap																																			
End of Packet	BUSY status								CRC20								K28.6																			
Idle word	D5.6								D5.6								D5.6								K28.5 (comma)											
Start of busy	D5.6								D5.6								D5.6								K28.2											
End of busy	D5.6								D5.6								D5.6								K28.3											

図 4.3 Event Builder で成形されるトリガー出力のフォーマット [21]

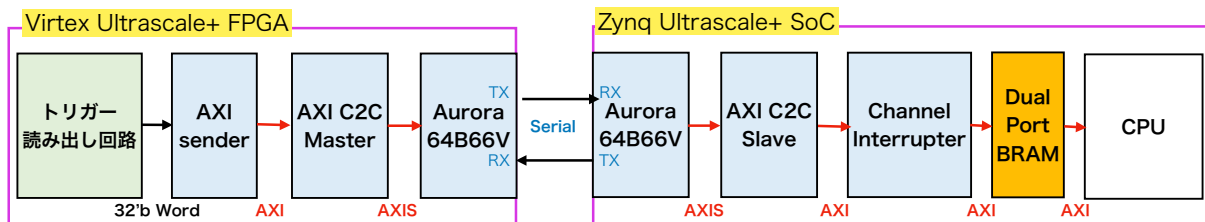


図 4.4 SL FPGA から MPSoC へのチップ間通信。トリガー回路からの出力は AXI sender から AXI プロトコルに乗せられ、Aurora64B66V でシリアル通信のデータにエンコードされ、MPSoC へ送られる。そのデータは MPSoC で再び AXI に載せ替えられ、BRAM にダンプされる。

コードされた後、AXI chip2chip で AXI 形式へと変換される。MPSoC で AXI 形式に戻されたデータは Channel Interrupter を中継して、幅 32bit、深さ 8192 列の Dual port BRAM にダンプされる。BRAM のもう片方のポートは MPSoC の PS に AXI 形式で接続されており、BRAM を介して PL から PS へのデータ送信が可能になっている。AXI バーストを利用していることで 1 イベント当たりのワード数は 256 に固定されている。そのため、BRAM に格納できるイベント数も固定であり、最大 $8192/256 = 32$ イベントである。一度 BRAM が full になった場合でも、MPSoC からマニュアルでリセットをかけることで、データ取得を再開できる。

アプリケーションの概要

トリガー試験を行うために開発した、MPSoC 上で走るアプリケーションの概要を説明する。このアプリケーションは SL の SD カード上にテキストファイルとして保存された LUT やテストパターンをハードウェアに書き込み、トリガー演算の結果を SD カード上のテキストファイルにダンプする。図 4.5 にアプリケーションのフローチャートを示す。

試験の最初には LUT の書き込みを行う。LUT に利用されている URAM はファームウェア上で初期値を設定することができないため、ファームウェアのリセットのたびに書き込みを行う必要がある。Wire Strip Coincidence までのすべての LUT を書き込むのに、概ね 20 分程度要する。

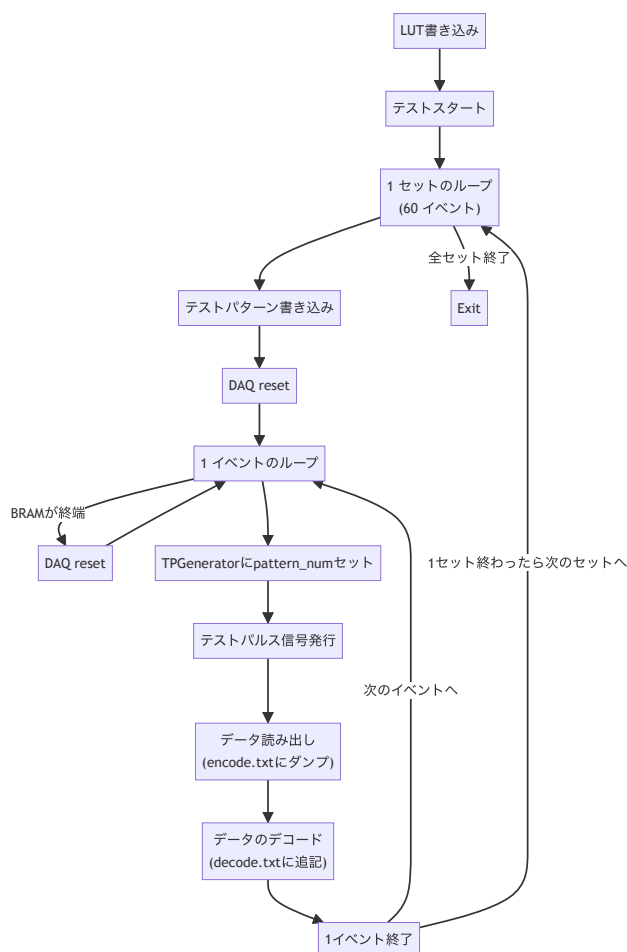


図 4.5 アプリケーションのフローチャート。Test Pattern Generator の BRAM の容量に合わせて、60 イベントを 1 セットに試験を行う。1 イベントのループでは、テストパルス信号の発行、データ読み出し、データのデコード、テキストファイルへのダンプを行う。

試験は Test Pattern Generator に一度に書き込めるイベント数である 60 イベントを 1 セットとして、テストパターンの書き込みと 60 イベント分のテストを繰り返す。1 イベントのループでは、最初に Test Pulse Generator のパターン番号を指定し、60 イベントの中からトリガー回路に投入するイベント番号を指定する。次にマニュアルでテストパルス信号を駆動し、トリガー回路にデータを入力する。トリガー回路の出力は読み出し回路を経てエンコードされたのち、MPSoC 上の BRAM に格納される。格納されたデータを MPSoC の PS から取り出し、アプリケーション上でデコードする。得られたビット配列は、最終出力として SD カード上のテキストファイルにダンプされる。イベントのループを回している過程で、MPSoC の BRAM が full になった場合はその都度リセットをかける。

試験に要する時間は、Linux から MPSoC 上の FPGA にアクセスするのにかかる時間が律速しており、10,000 イベントの試験におよそ 1 分程度かかる。

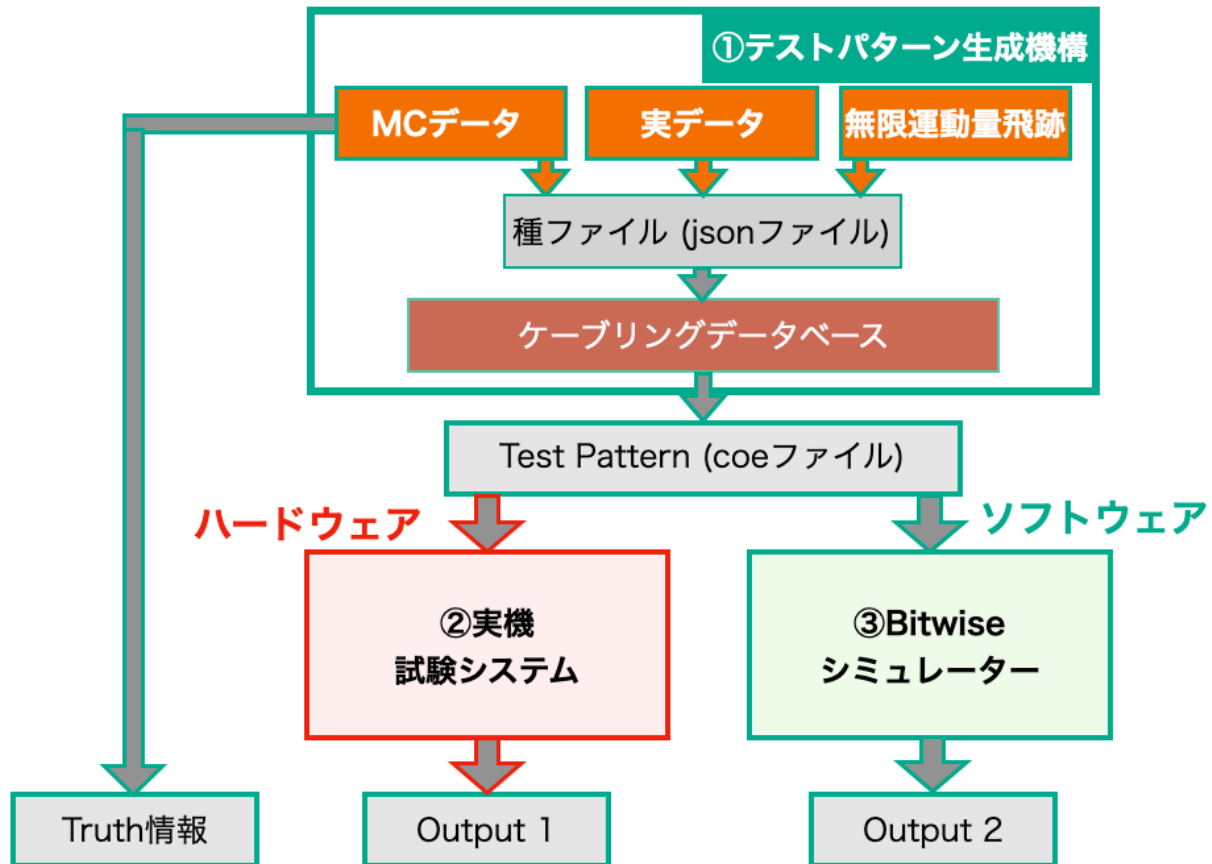


図 4.6 トリガー論理回路検証機構。テストパターン生成機構、SL シングルボード試験システム、Bitwise シミュレーターで構成される。テストパターン生成機構は MC データ、実データ、無限運動量飛跡などをもとに、シングルボード試験システムと Bitwise シミュレーターに共通するテストパターンを生成する。シングルボード試験システムはこれを入力として、実機上で動作するトリガー回路の演算結果を出力する。Bitwise シミュレーターもこれを入力として、トリガー演算を Bit レベルで再現したソフトウェアエミュレーターの演算結果を出力とする。この 2 つの出力を相互に比較・検証することで、トリガー論理回路を系統的に調査する。

4.1.3 トリガー論理回路検証システムの全体像

本研究で開発したシングルボード試験システムに加え、共同研究 [24] で開発が進められたテストパターン生成機構、Bitwise シミュレーターを組み合わせることで、トリガー論理回路を系統的に調査していくための検証システムを構築した。図 4.6 に検証システムの全体像を示す。

テストパターン生成機構はシミュレーションデータや実データから、SL が受信するヒットビットマップを再現したテストパターンを作成するツールである。ここではまず、元のデータセットから、TGC 検出器のヒットチャンネル情報を抽出し、種ファイルとして JSON ファイルを作成する。次に、種ファイルのデータを TGC 検出器から SL までの複雑な配線情報 (ASD、PS board、SL と続く物理的なケーブルの配線に加え、各デジタル回路内の信号の並び替え情報等も含む) を一元的に管理するケーブリングデータベースと組み合わせ、SL のインプットを再現したテストパターンを生成する。テストパターンは SL 実機、Vivado シミュレーション^{*1}、Bitwise シミュレーターに共

*1 実装した論理回路の信号遷移を逐次的にエミュレートするソフトウェア。Vivado の標準的な機能として実装されている。

通で用いることができるよう coe ファイル形式で出力される^{*2}

Bitwise シミュレーターは SL トリガー論理回路をビットレベルで再現した C++ ベースのシミュレーターである。このシミュレーターはテストパターンを入力とし、LUT も実機と同じものを使用する。さらに、各トリガーモジュールの出力フォーマットも実機とシミュレーションで統一的に定義しているため、完全に同一のコンフィギュレーションでトリガー演算を行い、その出力をビット単位で比較することができる。Bitwise シミュレーターはソフトウェアで実装してあるため、実機では出力していないモジュール内部の信号線の情報も簡単に出力することができる。これによりトリガー論理回路のより詳細なデバッグが可能である。

これらの検証システムにより、シミュレーションデータに対するトリガー応答をハードウェアとソフトウェアで計算し、その出力をモジュールごとに系統的に比較・検証する盤石な開発基盤を構築した。以下では、この検証システムを用いて行った、トリガー論理回路の動作検証および性能評価について議論する。

4.2 無限運動量飛跡を用いたトリガー回路の動作検証

本章では前節で述べたトリガー回路検証システムと無限運動量飛跡と呼ばれるトイデータセットを用いて行った、トリガー回路の初頭的な動作検証試験について述べる。無限運動量飛跡とは、衝突点から直線的に TGC 検出器に入射するミューオンをエミュレートした試験用のイベントであり、トリガーロジックとしては 100% 再構成されることが期待されるイベントセットである。このイベントセットに対する TGC BW コインシデンス (Channel Mapping から Wire Strip Coincidence まで) の応答を調べることで、実装したトリガー回路が期待通り動作していることを確かめる。

以下ではまず、テストパターン生成機構に含まれる無限運動量飛跡生成機構について述べた後、試験結果およびその考察について述べる。

無限運動量飛跡生成機構

前章で述べたように TGC 検出器は位置分解能向上およびデータ量の削減を実現するため、スタッガリング構造を取っている。ステーション内の 2 層または 3 層のガスチェンバーは、各チャンネルのカバーする η 領域がずれるよう設置されており、ステーション内コインシデンスにより、位置分解能を 2、3 倍に高めた代表点を算出する。ここで、M3 の各代表点と同じ η に位置する M1、M2 代表点に同じ番号が割り振られるよう定義したものを η ID と呼ぶ。 η ID は各ステーション内の代表点に通し番号的に振られている訳ではなく、M3 の代表点を起点に、その η に一番近い代表点を選ぶようにして値を割り振っている。^{*3}

η ID はケーブリングデータベースにより、各検出器のチャンネル番号と紐づけられている。テストパターン生成機構にはそれらを活用して、任意の 2 次元座標点に入射する無限運動量飛跡を作成する機構が備わっている。とある (η, ϕ) 位置に入射するミューオンをエミュレートしたテストパターンが作りたい場合、該当する η ID と ϕ 方向の代表点番号 (スタッガード ID) を指定するだけで、それに対応する 7 層分のヒット情報が含まれた、テストパターンが生成される。

^{*2} テストパターンを格納する BRAM は、coe ファイルによって初期値を設定することができる。そのため Vivado シミュレーションではテストパターンは coe ファイルに格納されていることが好ましい。これに合わせるように、シングルボード試験システム、Bitwise シミュレーターも coe ファイルを入力形式とするよう設計している。

^{*3} TGC 検出器が設計された当初は、各ステーションで η の位置分解能が均一になるようにワイヤーがバンドルされていたため、ステーション内で通し番号的に η ID を割り振ることができるはずだった。しかし、設置の段階で TGC 検出器の設置位置が η 方向にずれたため、これはできなくなった。

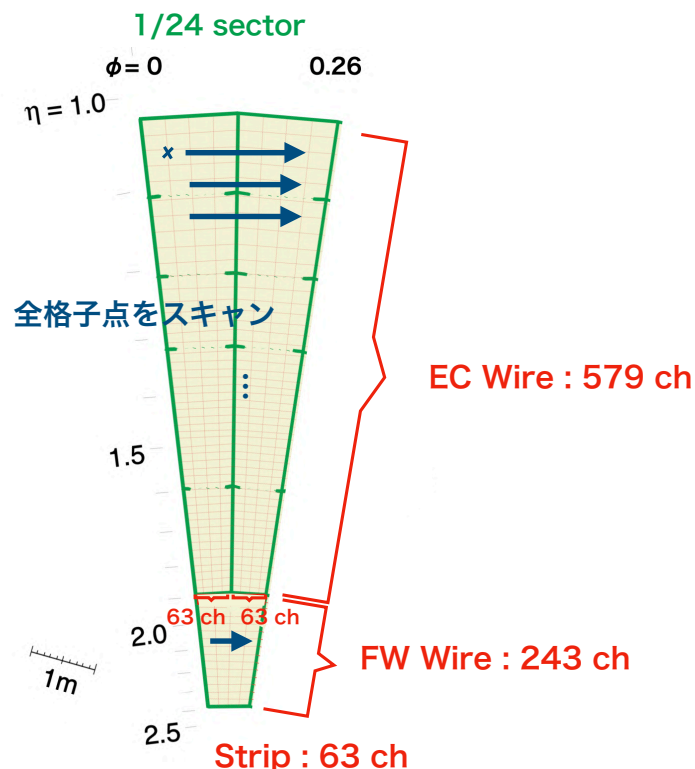


図 4.7 用意した無限運動量飛跡のデータセット。FW、EC 領域に存在する全 2 次元格子点に対して網羅的にテストパターンを用意。EC $\phi 0$ 、 $\phi 1$ 領域は Wire が 579 個、Strip が 63 個あるため合計 36,477 の格子点が存在する。FW 領域は Wire が 243 個、Strip が 63 個あるため合計 15,309 の格子点が存在する。

無限運動量飛跡試験の概要

トリガー論理回路および作成された LUT に対する最初の試験として、無限運動量飛跡に対するトリガー応答を調査した。ここでは論理回路実装や LUT 作成の際に生じた不具合を網羅的に洗い出すため、M3 の Wire、Strip で張られる全ての 2 次元格子点に対して無限運動量飛跡を用意した。図 4.7 にイベントセットの概念図を示す。1 つの SL が担当する TGC 1/24 セクターでは、フォワード領域には M3 の Wire 代表点が 243 個、Strip 代表点が 63 個存在し、合計 15,309 の 2 次元格子点が存在する。エンドキャップ領域には $\phi 0$ 、 $\phi 1$ それぞれで M3 の Wire 代表点が 579 個、Strip 代表点が 63 個存在し、合計 36,477 の格子点が存在する。このイベントセットに対する、Wire Segment Reconstruction、Strip Segment Reconstruction、Wire Strip Coincidence の応答を調べることで、トリガー回路の動作を調査した。

図 4.8～図 4.10 に各モジュールごとの結果を示す。横軸に M3 における Strip のスタッガード ID、縦軸に M3 における Wire のスタッガード ID をとる。各格子点をピボットとする無限運動量飛跡をシングルボード試験システムに投入し、飛跡再構成に成功した場合はその格子点を白色、失敗した場合は黒色で塗り潰す。飛跡再構成に成功したイベントの割合を表 4.4 にまとめる。同様の試験を Bitwise シミュレーターで行ったところ、全てのトリガーモジュールで飛跡再構成の成功率は 100 % であった。

表 4.4 無限運動量飛跡に対する飛跡再構成の成功率

	Strip Segment Reconstruction	Wire Segment Reconstruction	Wire Strip Coincidence
FW	100 %	99.9 %	99.2 %
EC0	100 %	99.7 %	97.0 %
EC1	100 %	99.9 %	96.5 %

Strip Segment Reconstruction の結果

Strip Segment Reconstruction ではフォワード領域とエンドキャップ領域の全格子点に対して飛跡再構成に成功した。この結果は Channel Mapping、Strip Station Coincidence、Strip Segment Reconstruction、の論理回路実装が適切に行われたこと、作成された LUT が抜けなく実装されていること、そして LUT の書き込みやタイミング制御などトリガーを稼働させるのに必要なコントロール機能が正確に動作していることを示している。さらに、この結果はシングルボード試験システム自体が正確に動作していることも示唆している。MPSoC からのテストパターンを書き込むバスとトリガー回路の読み出しバスは安定して動作しており、TTC emulator、トリガー回路、読み出しバスが固定レイテンシーで制御されていることを示している。これらのコントロールおよび読み出しバスは実験本番でも使われるシステムであり、SL 統合ファームウェア全体が精度良く動作していることを示している。この結果を得られるまでの過程で、本研究によって Strip LUT のミスを発見し、修正を行った。デバッグの過程を Appendix B に詳述する。

Wire Segment Reconstruction の結果

Wire Segment Reconstruction では、フォワード領域で 99.9 % (15,287 / 15,309)、エンドキャップ $\phi 0$ 領域で 99.7 % (35,370 / 36,477)、エンドキャップ $\phi 1$ 領域で 99.9% (36.181 / 36.477) の飛跡再構成に成功した。一方で、全領域において特定の構造を持たない $O(0.1\%)$ の Inefficiency が観測された。^{*4}

この Inefficiency に関しては、Bitwise シミュレーターでは確認されていないため、シングルボード試験システム固有の問題であると考えられる。これまでの調査の結果、飛跡再構成に失敗する格子点の位置は、データ取得のたびに変わることが判明している。失敗するイベントの割合は試験ごとに概ね一定で、Wire Segment Reconstruction では約 0.1 % である。現時点では、この問題がハードウェア上のトリガー回路自体の問題に起因するのか、それとも読み出し回路の問題に起因するのかの区別がついておらず、問題の解決には至っていない。今後、詳細な調査を進め、原因の解明と修正に努める。一方で、後述するトリガー回路の性能評価においては $O(10\%)$ 程度の Inefficiency について議論するため、この $O(0.1\%)$ 程度の Inefficiency は今後の議論には影響しないと考える。

この結果を得られるまでの過程で、本研究によって無限運動量飛跡生成機構に問題を発見し、修正を行った。デバッグの過程を Appendix B に詳述する。

Wire Strip Coincidence の結果

Wire Strip Coincidence では、フォワード領域で 99.6 % (15,179 / 15,309)、エンドキャップ $\phi 0$ 領域で 97.0 % (35,390 / 36,477)、エンドキャップ $\phi 1$ 領域で 96.5% (35.201 / 36.477) の飛跡再構成に成功した。一方で、フォワード領域では Wire スタッガード ID 190 番に該当する飛跡が全て再構成されないことが確認された。エンド

^{*4} エンドキャップ $\phi 1$ 領域の Wire スタッガード ID 0 ~ 2 の範囲でも Inefficiency が見られるが、これは M3 と M1 の η 方向のカバレッジの違いによるものであると理解されている。 $\phi 1$ 領域の $\eta \sim 2.0$ 付近の領域では、M3 が M1 よりも広い η 範囲をカバーしている。そのため、M3 の代表点をピボットにして代表点を割り振ると同じ代表点をもつチャンネルでも η 位置にずれが生じ、直線的な飛跡にならない。

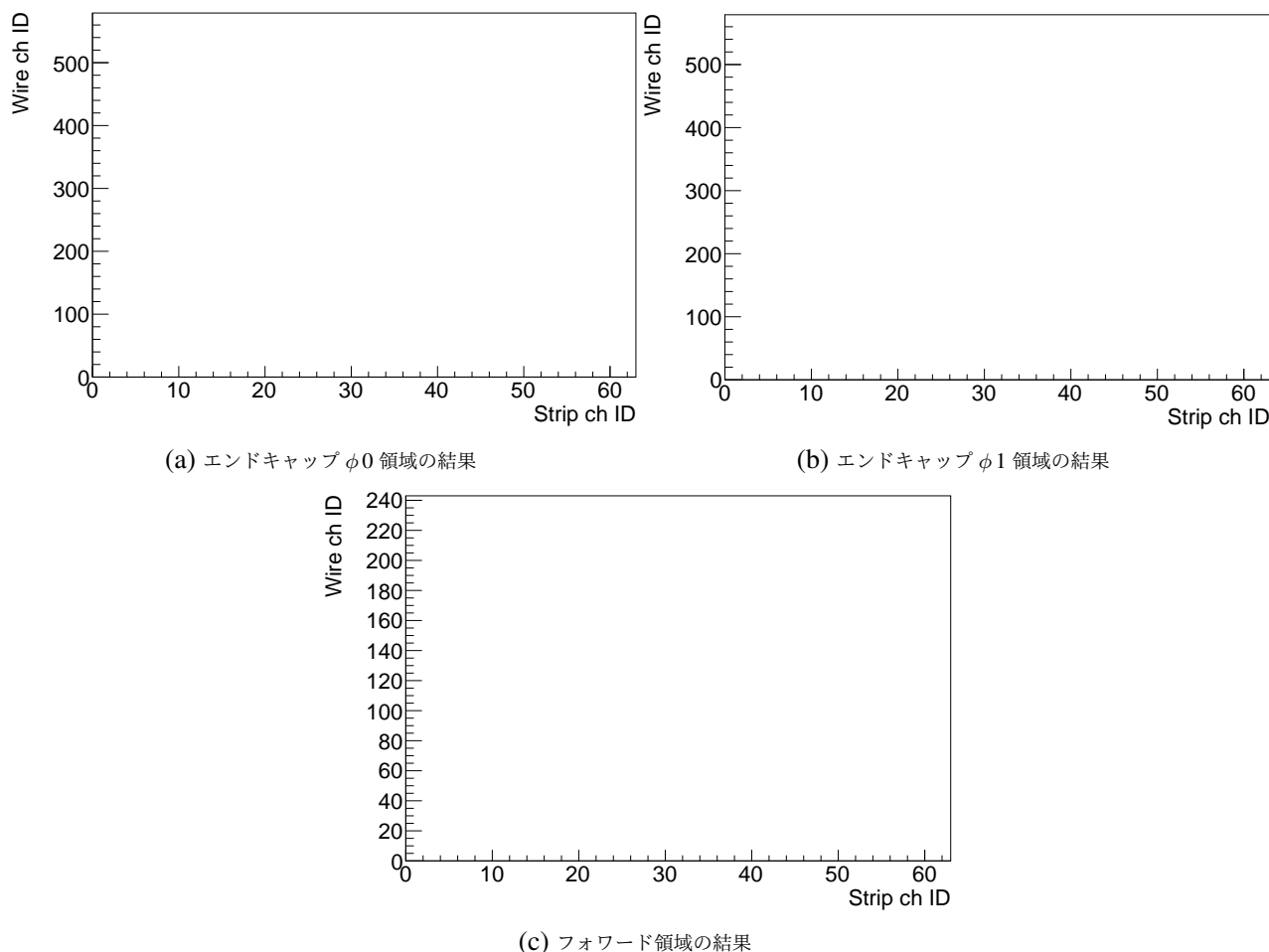


図 4.8 無限運動量飛跡に対する、Strip Segment Reconstruction の応答。横軸に M3 における Strip のスタagger ID、縦軸に M3 における Wire のスタagger ID をとる。各 2 次元格子点をピボットとする無限運動量飛跡をシングルボード試験システムに投入し、 $0 \leq \Delta\phi$ の Segment を再構成できた場合にはその格子点を白色、できなかった場合は黒色で塗り潰す。この結果は、全ての 2 次元格子点で飛跡再構成に成功したことを示す。

キャップ領域では、Wire スタagger ID 410 番以降の領域で、 $\phi 0$ と $\phi 1$ のどちらにも規則的な構造を持った Inefficiency が観測された。この Inefficiency は、同様の LUT を利用している Bitwise シミュレーターでは再現されないため、LUT の原因ではなく論理回路の問題であると考えられる。Wire Strip Coincidence では Wire スタagger ID が 419 より小さい領域は 32 Unit region で処理され、大きい領域は 8 Unit region で処理される。そのため、この構造的な Inefficiency は 8 Unit region のファームウェアの問題である可能性が高い。今後 Vivado シミュレーターと Bitwise シミュレーターの途中出力を比較することで、問題が発生している箇所を特定し、論理回路の修正を進める。

4.2.1 まとめ

無限運動量飛跡を用いた試験によって、TGC BW Coincidence の 95 % 以上の領域でトリガー回路が正常に動作していることを確認することができた。一方で、Wire Strip Coincidence では、トリガーロジックの論理回路実装において発生したと思われる不具合を発見することができた。このような数 % の局所的な Inefficiency は、網羅的かつ緻密に検証を進めたからこそ見つけられたもので、これまで発見されてこなかったものである。任意の位置に

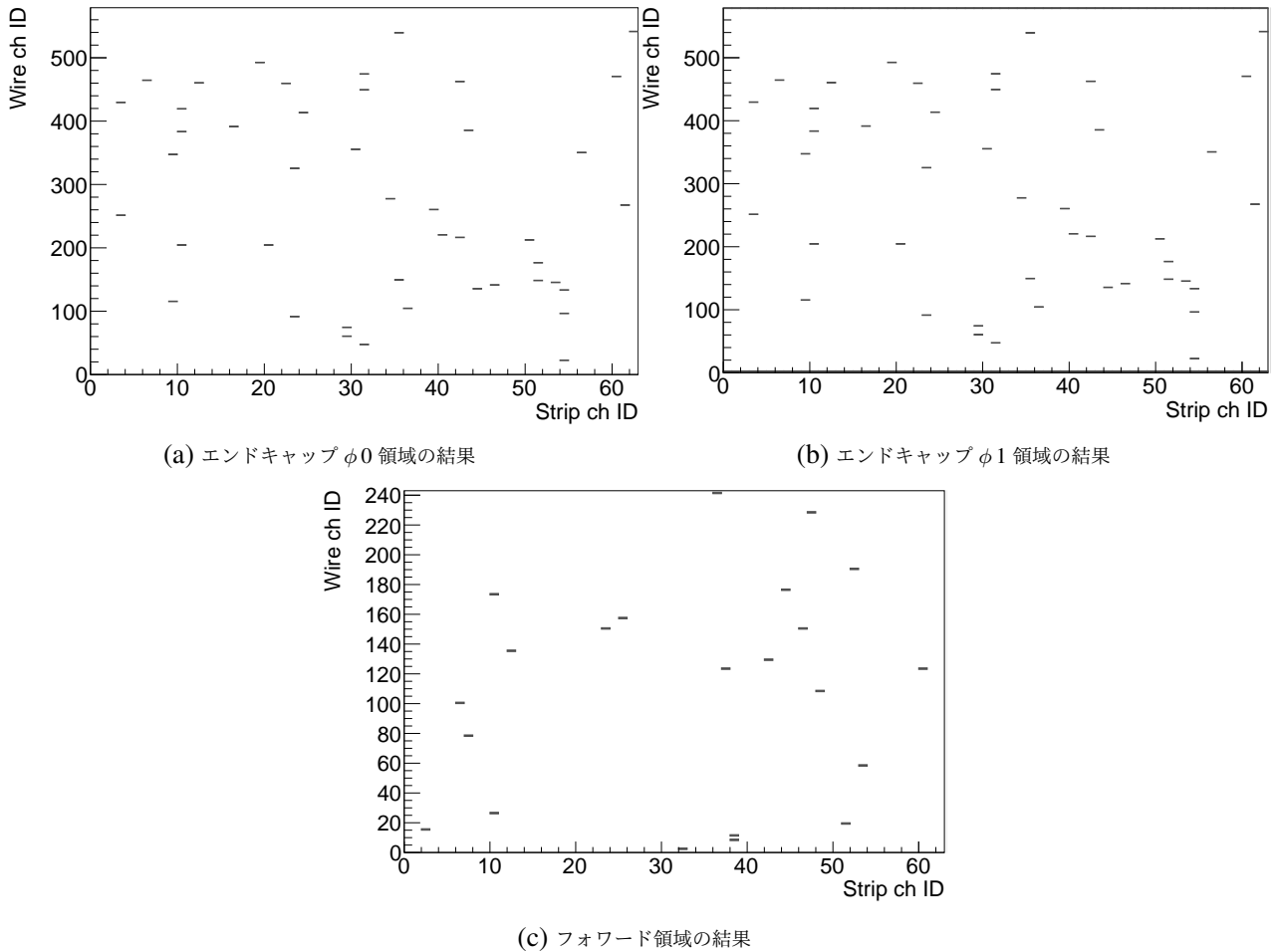


図 4.9 無限運動量飛跡に対する、Wire Segment Reconstruction の応答。横軸に M3 における Strip のスタッガード ID、縦軸に M3 における Wire のスタッガード ID をとる。各 2 次元格子点をピボットとする無限運動量飛跡をシングルボード試験システムに投入し、 $0 \leq \Delta\theta$ の Segment を再構成できた場合にはその格子点を白色、できなかった場合は黒色で塗り潰す。全体の $\mathcal{O}(0.1\%)$ 程度の格子点で飛跡再構成に失敗していることがわかる。

入射するミュオン飛跡を生成できる無限運動量飛跡生成機構と、ハードウェア上で実際に動作するトリガー回路に対して、大統計量の試験を行うことができるシングルボード試験システムの真価が発揮された例である。これらの不具合を実験開始前に発見し、修正することは、高輝度 LHC-ATLAS 実験の運転初日から、最大パフォーマンスでのミュオントリガーを実現する上で、重要な役割を果たす。

4.3 シミュレーションデータを用いたトリガー回路の性能評価

本節では、シングルミュオンモンテカルロデータを用いたトリガー性能評価試験について述べる。シングルミュオンイベントとは、1本のミュオンが衝突点から検出器に入射する過程をエミュレートしたもので、ミュオンが TGC 検出器を素通りする事象や、多重散乱により飛跡が大きく曲げられる事象など、実際に起こりうる物理過程を考慮したものになっている。そのためこのイベントセットを利用することで、現実的なトリガー性能を評価することができる。また、シミュレーションデータでは Truth 情報が含まれているため、有限の運動量をもつミュオンイベントに対して適切な p_T 判定を下すことができるのか、などより詳細に論理回路の検証を行うことができる。

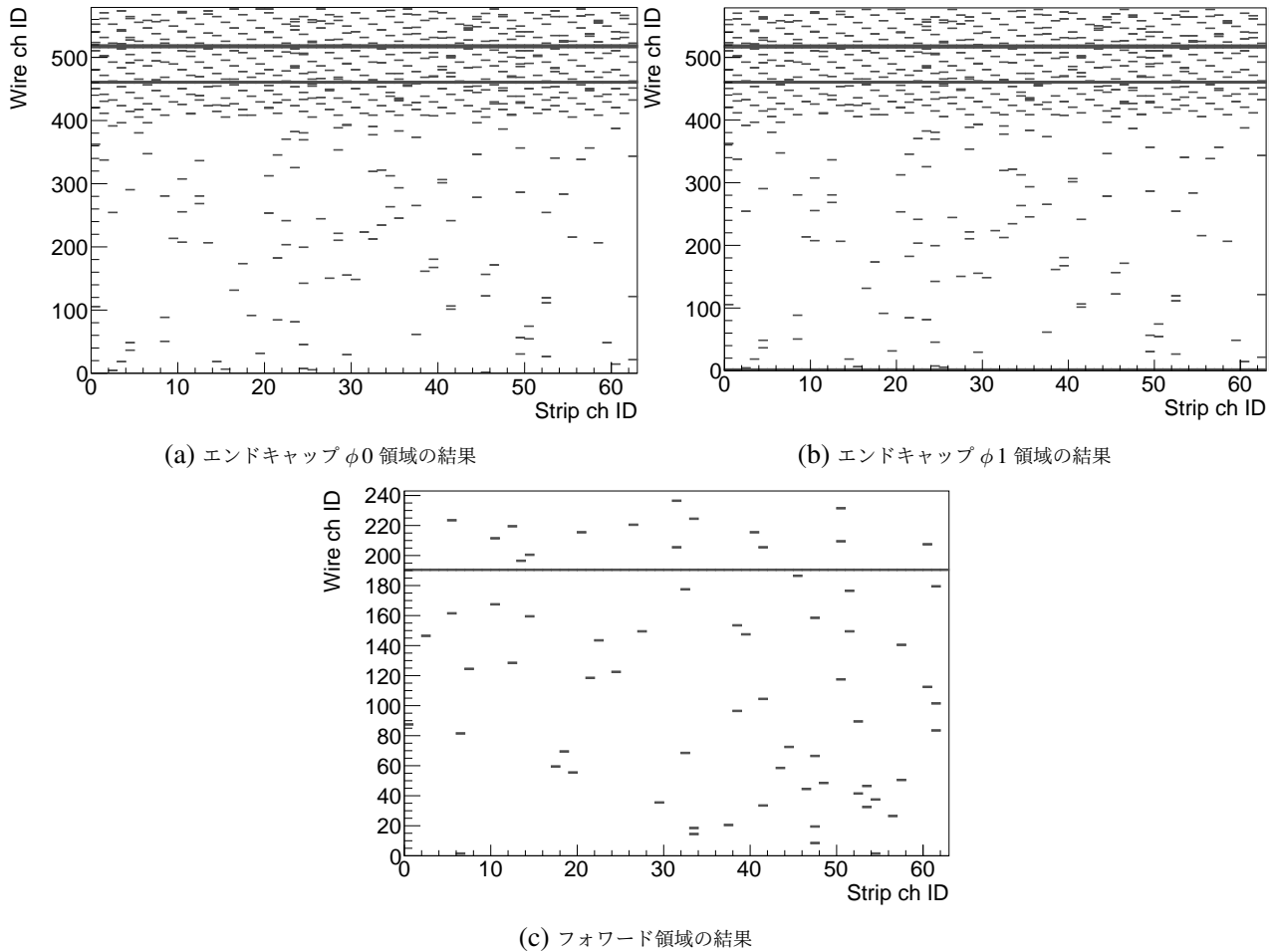


図 4.10 無限運動量飛跡に対する、Wire Strip Coincidence の応答。横軸に M3 における Strip のスタッガード ID、縦軸に M3 における Wire のスタッガード ID をとる。各 2 次元格子点をピボットとする無限運動量飛跡をシングルボード試験システムに投入し、 $5 \text{ GeV} \leq p_T$ の Segment を再構成できた場合にはその格子点を白色、できなかった場合は黒色で塗り潰す。Forward 領域では Wire スタッガード ID 190 番に該当するイベントが全て再構成に失敗している。Endcap 領域では Wire スタッガード ID 410 番以降の領域で $\phi 0$ と $\phi 1$ のどちらにも規則的な構造を持った Inefficiency が見られる。

用意したデータセットの概要を表 4.5 にまとめる。パイルアップは含まれておらず、 p_T は 0 GeV から 50 GeV 、 η 、 ϕ は TGC 検出器がカバーする全領域に対して満遍なく用意した。また、本試験ではトリガー回路自体の性能に焦点を当てた検証を行うため、M1、M2、M3 の各ステーションに少なくとも 1 つのヒットがあることを要求している。これにより、多重散乱などで飛跡が大きく曲げられ、TGC 検出器に入射しなかったイベントなどを除外している。

Strip Segment Reconstruction のトリガー効率

この試験では p_T が十分に大きい事象に対するトリガー応答を確認することを目的とし、Truth $p_T 20 \text{ GeV}$ 以上のミューオンに対する検出効率を評価する。Efficiency は式 4.1 のように定義する。

$$\text{Efficiency} = \frac{\text{Strip Segment Reconstruction で } 0 \leq \Delta\phi \text{ を再構成できたイベント数}}{\text{Truth } p_T 20 \text{ GeV 以上のイベント数}} \quad (4.1)$$

Strip Segment Reconstruction の検出効率の η および ϕ 依存性を図 4.11 に示す。黒色の点は SL 実機の出力結

表 4.5 用意したシングルミュオンモンテカルロデータの概要

Parameter	
p_T	$0 < p_T < 50 \text{ GeV flat}$
η	$1.06 < \eta < 2.4 \text{ flat}$
ϕ	$0 < \phi < 2\pi \text{ flat}$
イベント数	500,000
イベントカット	1つのトリガーセクター内の M1、M2、M3 各ステーションにそれぞれ 1つ以上のヒットがあることを要求

果、赤色の点は Bitwise シミュレーターの出力結果を表している。全領域での平均トリガー検出効率は 97.4 % であり、先行研究 [20] の結果と矛盾のない結果が得られた。また、Strip Segment Reconstruction では実機と Bitwise シミュレーターの結果が多くの領域で一致している。これは Bitwise シミュレーターがトリガー論理回路を極めて正確に再現できていることを示している。しかし、 $2 < \eta$ のフォワード領域の、特に 1/24 セクターの境界領域 (トリガー回路は TGC BW の 1/24 領域ごとに独立した回路になっている。 ϕ 方向に $2\pi / 24 = 0.26$ おきに 1/24 チェンバーの境界が存在する。) で、両者の Efficiency に 5 % 程度の違いが生じている。原理的にはこの 2 つの出力は完全に一致するべきものであり、この差異は Bitwise シミュレーターとファームウェアで実装されているロジックに、わずかな違いが存在することを示している。今後、両者のロジックを精密に比較し、出力の不一致を解消する。

また、この領域では 15 % 程度の Inefficiency が生じている。これに関してはファームウェア自体の論理回路実装のミス、もしくは LUT の不具合である可能性が高い。今後これに関する調査も進め、原因の特定および修正を行う。

Wire Segment Reconstruction のトリガー効率

Strip の場合と同様に、Efficiency は式 4.2 のように定義する。

$$\text{Efficiency} = \frac{\text{Wire Segment Reconstruction で } 0 \leq \Delta\theta \text{ を再構成できたイベント数}}{\text{Truth } p_T \geq 20 \text{ GeV 以上のイベント数}} \quad (4.2)$$

Wire Segment Reconstruction の検出効率の η および ϕ 依存性を図 4.11 に示す。全領域での平均トリガー効率は 96.3 % であり、先行研究 (A.2) と矛盾しない結果が得られた。

トリガー効率の η 依存性に着目すると、 $\eta \sim 1.4$ 付近で約 5 % の Inefficiency が見られる。この Inefficiency に対して Bitwise シミュレーターを用いた調査したところ、Channel Mapping、Wire Station Coincidence、および Wire Segment Reconstruction の Address Specifier までは期待通り動作していることが確認され、LUT に該当するイベント情報が含まれていないことがわかっている。今後は LUT のような局所的な Inefficiency の解消に向けた LUT の最適化にも取り組み、トリガーパフォーマンスの最大化を目指す。

Wire Strip Coincidence のトリガー効率

Wire Strip Coincidence は無限運動量飛跡を用いた試験で既にエンドキャップ領域に不具合が確認されているため、フォワード領域に限定して議論を進める。また Strip の試験で明らかになった 1/24 セクター境界領域 (1/24 セクターの両端から ϕ 方向に 1 割の領域) も除外する。ここでは Efficiency を式 4.3 のように定義する。

$$\text{Efficiency} = \frac{\text{Wire Strip Coincidence で } p_T \geq 20 \text{ GeV 以上と判定されたイベント数}}{\text{Truth } p_T \geq 20 \text{ GeV 以上のイベント数}} \quad (4.3)$$

図 4.13 に Wire Strip Coincidence のフォワード領域での検出効率の η 依存性及び ϕ 依存性を示す。平均のトリ

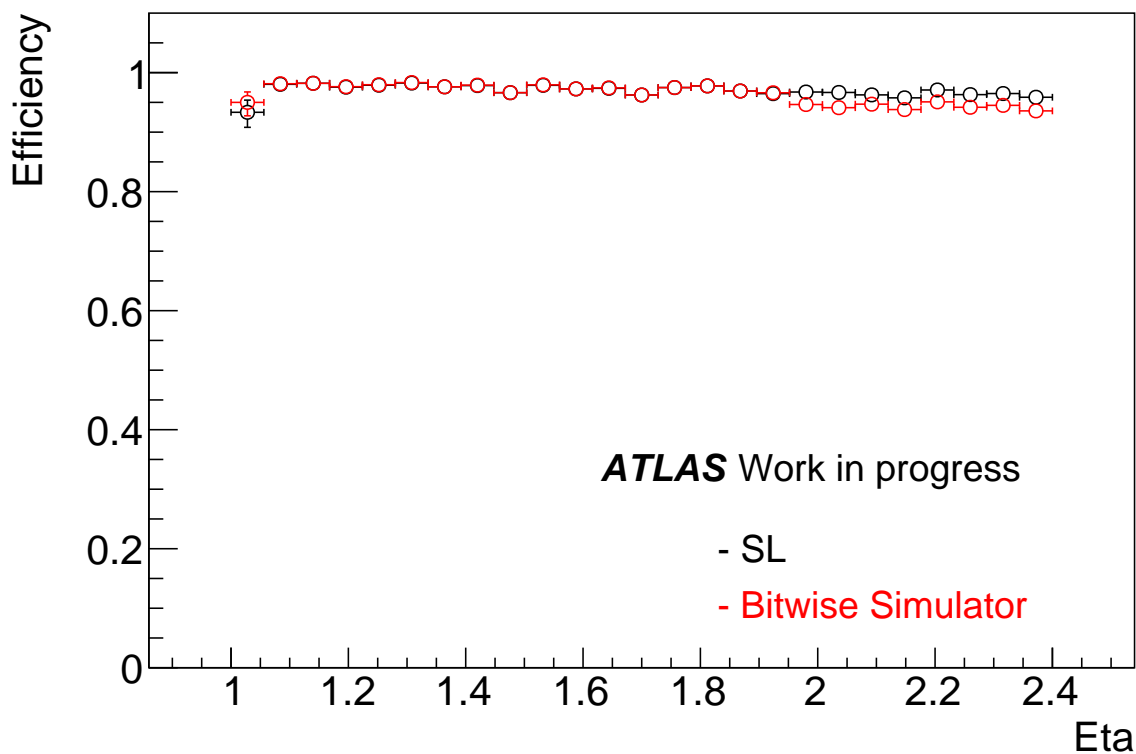
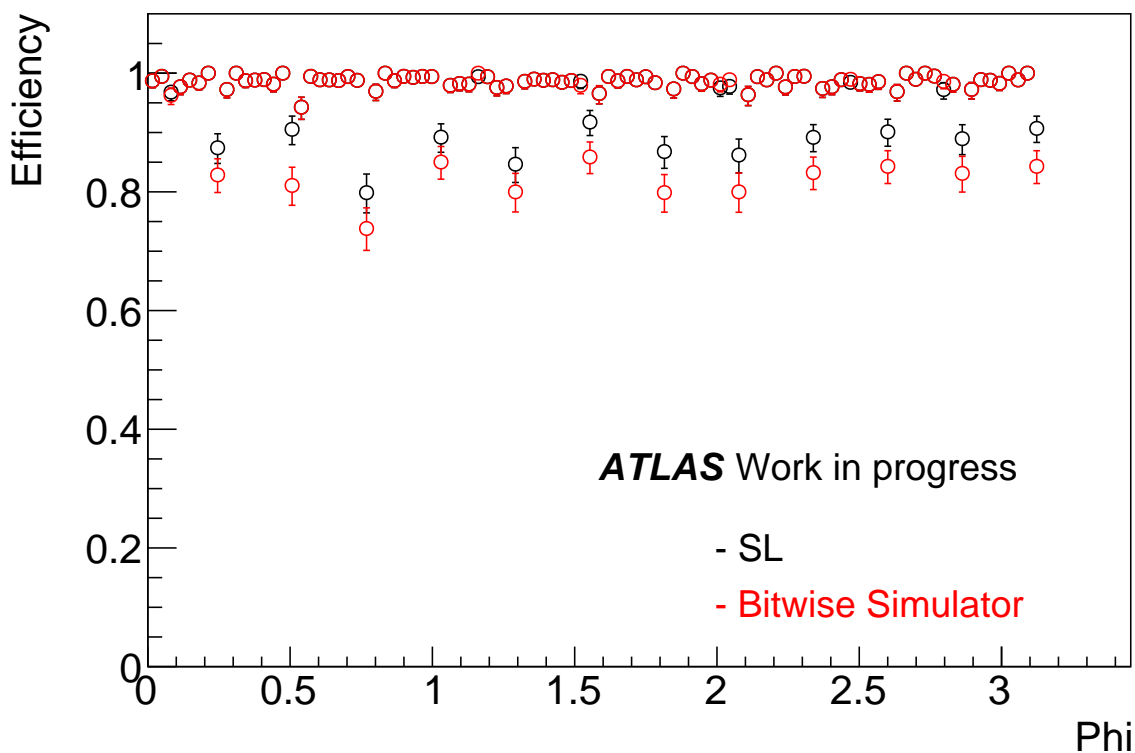
(a) Strip Segment Reconstructino 検出効率の η 依存性(b) Strip Segment Reconstructino 検出効率の ϕ 依存性

図 4.11 Strip Segment Reconstruction の検出効率。黒色のプロットが実機出力、赤色のプロットが Bitwise シミュレータの出力を表す。全領域のトリガー検出率は 97.4 % であり、先行研究と矛盾のない結果が得られた。トリガー効率の ϕ 依存性に着目すると、シングルボード試験システムでも Bitwise シミュレーターでも、チェンバー境界領域で 10 % 程度の Inefficiency が見られる。また、フォワード領域の、特にチェンバー境界領域で実機と Bitwise シミュレーターで出力が一致していないイベントが数 % 程度存在している。

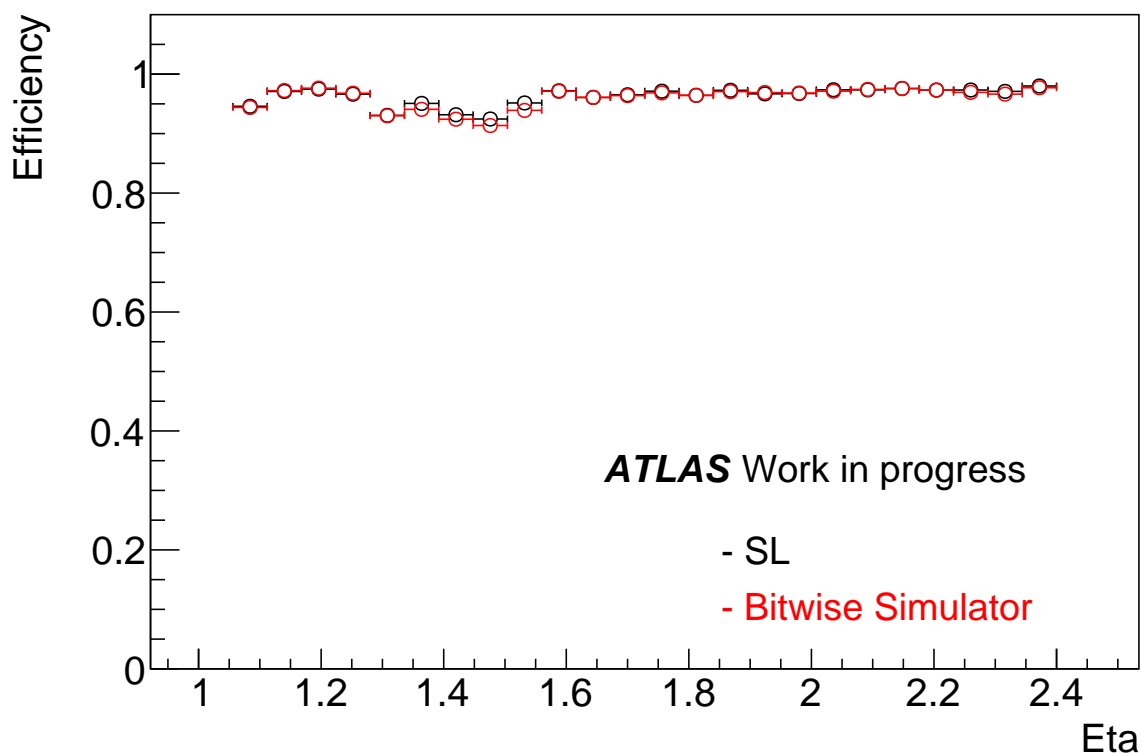
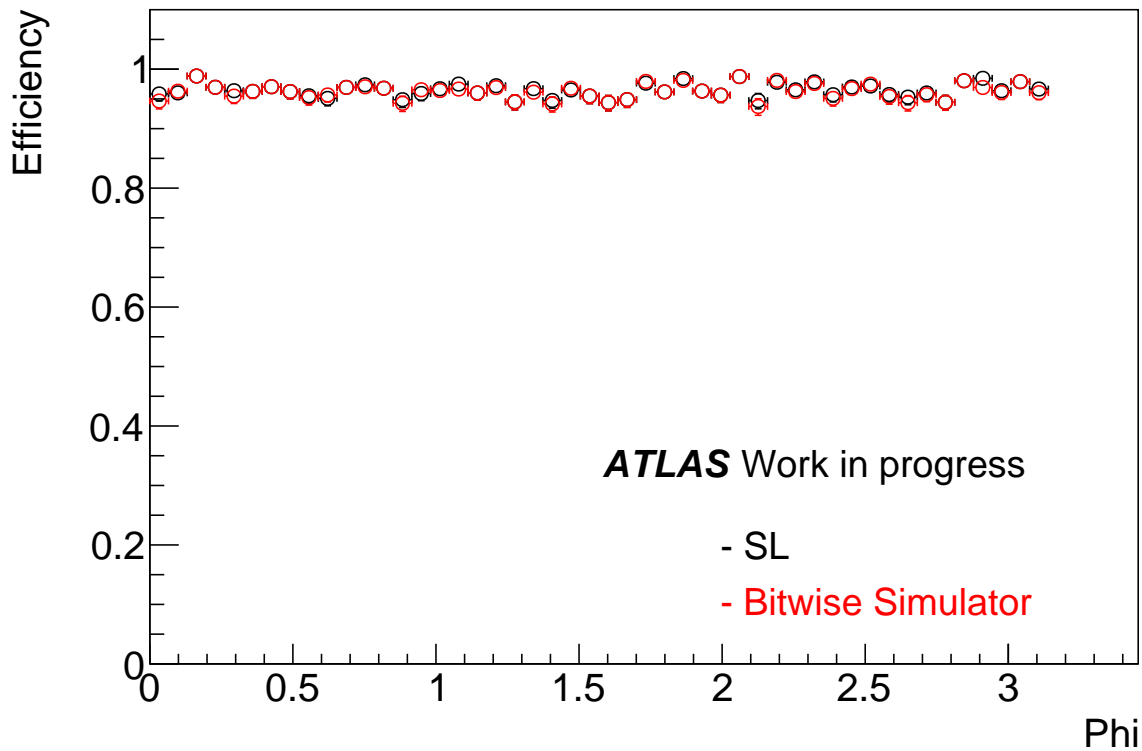
(a) Wire Segment Reconstruction 検出効率の η 依存性(b) Wire Segment Reconstruction 検出効率の ϕ 依存性

図 4.12 Wire Segment Reconstruction の検出効率。黒色のプロットが実機出力、赤色のプロットが Bitwise シミュレータの出力を表す。全領域のトリガー検出効率は 96.3 % であり、先行研究と矛盾のない結果が得られた。トリガー効率の ϕ 依存性に注目すると、 $\eta \sim 1.4$ 付近で 5 % の Inefficiency が見られる。

ガー効率は 93.5 % であり、先行研究 [11] の結果と矛盾のない結果が得られた。トリガー効率の η 依存性に着目すると、 $\eta \sim 1.95, 2.2$ 付近で 5 % の Inefficiency が見られる。この領域は実機の出力と Bitwise シミュレーターの出力で違いが大きく見られる領域でもあるため、Bitwise シミュレーターと実機の出力を詳細に比較することで、まずはこの差の解消に努める。

図 4.14 に Wire Strip Coincidence の p_T ごとの検出効率を示す。赤色、青色、緑色、ピンク色の各カーブは、それぞれがトリガー閾値 20 GeV、15 GeV、10 GeV、5 GeV と判断されたイベントの割合を示している。いずれの p_T 閾値に対してもプラトー領域の Efficiency は 93.5 % 程度であり、先行研究 [11] の結果と一致している (図 A.1)。実機の結果と Bitwise シミュレーターの結果も概ね一致している。

一方、先行研究の結果と比べて、Efficiency の立ち上がりが緩やかで、 p_T 20 GeV 以下のミューオンに対する除去効率が期待されるより低いことが明らかになった。^{*5}

例えば、Wire Segment Reconstruction で不当に $\Delta\theta$ を小さく見積もってしまうと、このように p_T を大きく見積もる不具合が生じる。今後は、Segment Reconstruction から出力される角度情報の大きさに対しても調査を進め、原因の解明を行う。

この結果を得るまでに、本研究によって Wire Segment Reconstruction の問題点を発見し、修正を行なった。詳細を Appendix C に記述する。

まとめ

本研究では、任意のデータセットに対するトリガー応答を高速で計算するシングルボード試験システムを開発し、無限運動量飛跡やモンテカルロデータを用いてトリガーロジックの性能評価試験を行った。その結果、Wire Strip Coincidence までの大部分の領域が期待される性能を有していることを初めて確認することができた。また、シングルボード試験システムと Bitwise シミュレーションの比較により、問題を系統的に特定する検証基盤も構築したことで、これまで発見が困難だった論理回路や LUT の局所的な不具合を発見し、多くの問題点を解決することができた (Appendix B ~ C 参照)。これからも、4.2 節と 4.3 で議論した、残された課題に対して詳細な調査とデバッグを進めていくことで、ミューオントリガーの精度を高めていく。

^{*5} 先行研究では、Bitwise ではなくフロートレベルでのソフトウェアシミュレーションを行なっている。また先行研究では Efficiency の分母として Offline で再構成されたミューオンのイベント数を採用しているのに対し、本件研究では Truth のイベント数を採用している。

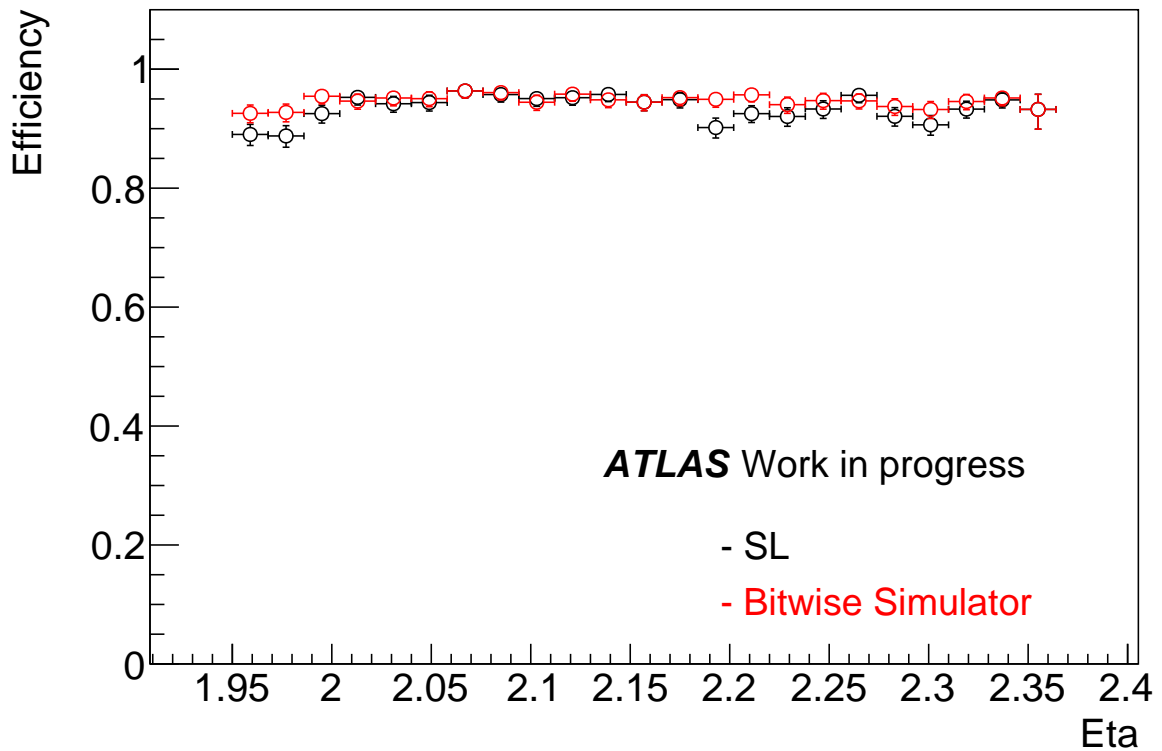
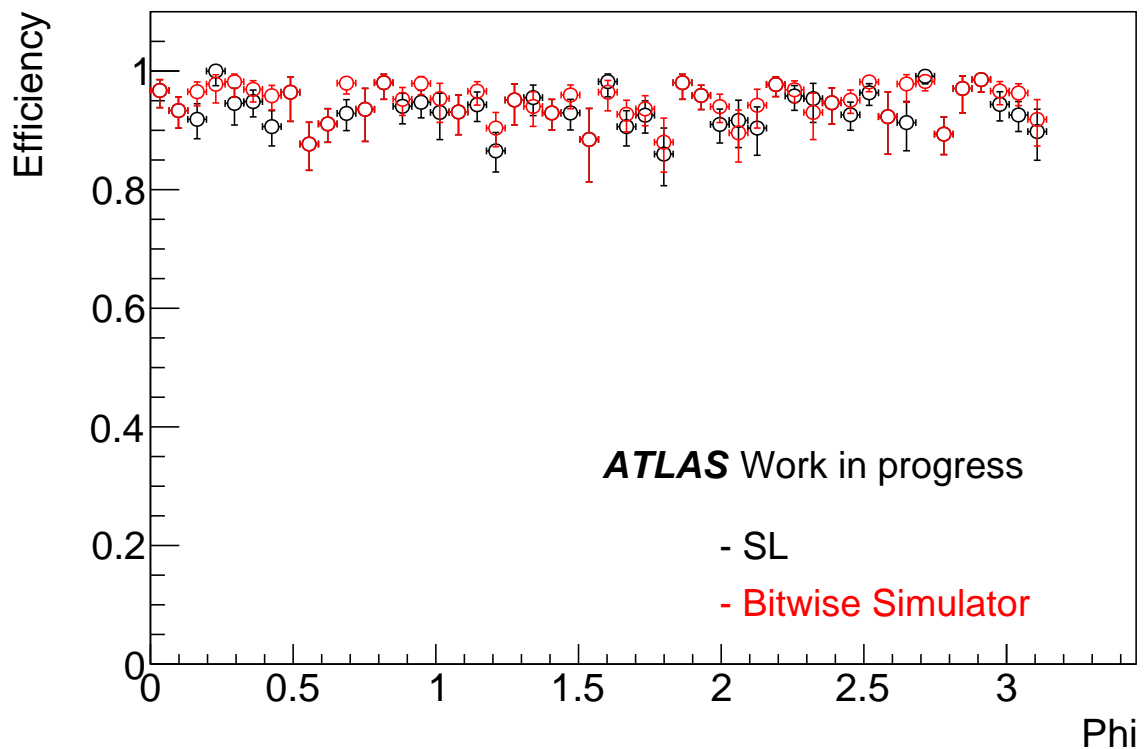
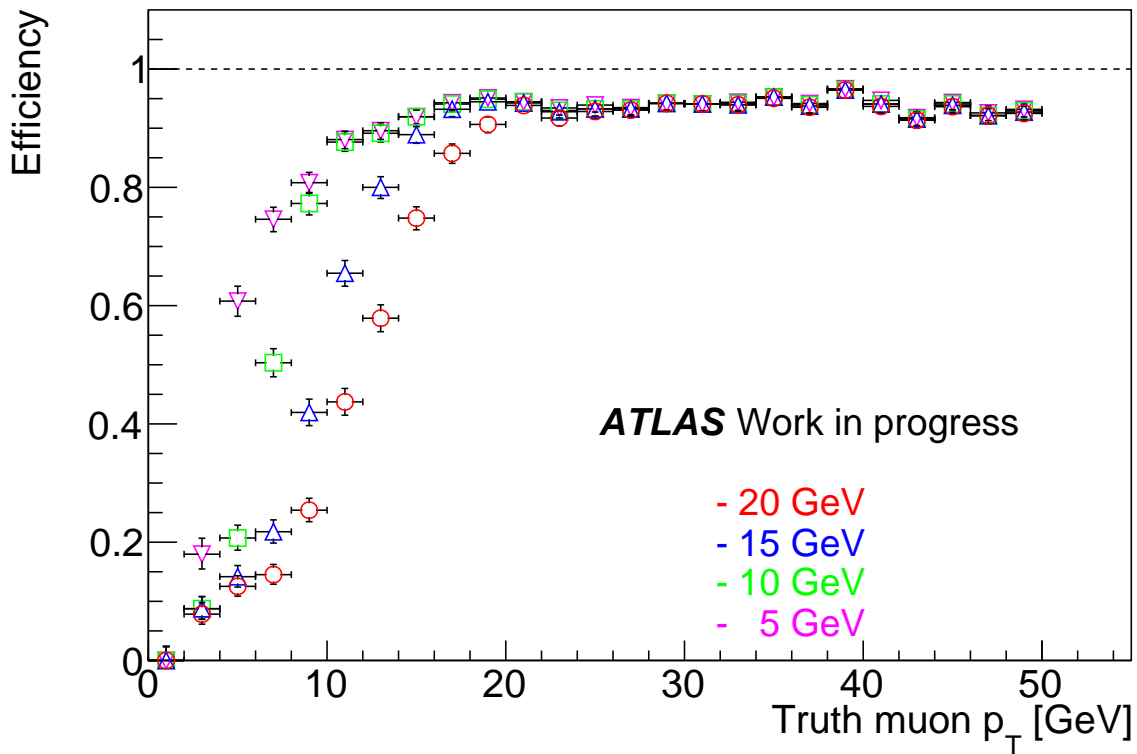
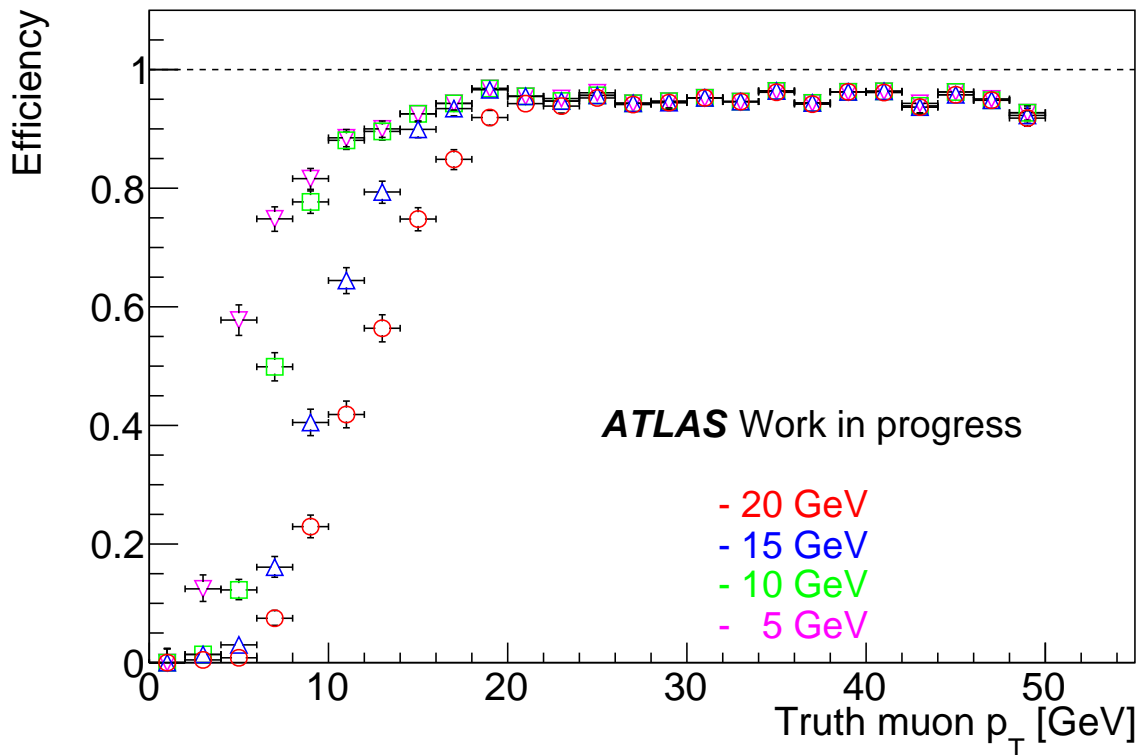
(a) Wire Strip Coincidence 検出効率の η 依存性(b) Wire Strip Coincidence 検出効率の ϕ 依存性

図 4.13 フォワード領域における Wire Strip Coincidence の検出効率。黒色のプロットが実機出力、赤色のプロットが Bitwise シミュレータの出力を表す。フォワード領域のトリガー検出効率は 93.5 % であり、先行研究と矛盾のない結果が得られた。トリガー効率の η 依存性に注目すると、 $\eta \sim 1.95, 2.2$ 付近で 5% の Inefficiency が見られる。



(a) 実機の結果



(b) Bitwise シミュレーターの結果

図 4.14 Wire Strip Coincidence フォワード領域における p_T ごとの検出効率。赤色、青色、緑色、ピンクの各プロットは、それぞれが p_T 閾値 20 GeV、15 GeV、10 GeV、5 GeV と判断されたイベントの割合を示している。プラトー領域の Efficiency はいずれの p_T 閾値に対しても 93.5% 程度であり、先行研究の結果と一致している。

第 5 章

PS board 品質保証試験に向けたコンパクト DAQ システムの開発

5.1 PS board 品質保証試験の設計

5.1.1 PS board 品質保証試験の概要

2029 年から始まる高輝度 LHC-ATLAS 実験に向けて、TGC 検出器エレクトロニクスは刷新される。フロントエンドエレクトロニクスの 1 つである PS board は、Run 3 までのエレクトロニクスに代わり、FPGA を搭載した新しいハードウェアデバイスへと置き換えられる。図 5.1 に PS board の量産スケジュールを示す。PS board はこれまでに第一試作機、第二試作機を用いた機能開発が進められ、2022 年にプレ量産が完了している。2024 年から 1400 枚以上に及ぶ本量産が始まり、2026 年から ATLAS 実験室への設置作業が開始される。

高輝度 LHC-ATLAS 実験の運転初日から安定したデータ収集を実現するためには、量産された各個体にハードウェアとしての欠陥がないことを詳細に検査することが必要である。そのために行う品質保証試験のことを Quality Assurance and Quality Control (QAQC) 試験と呼ぶ。

本章ではまず、PS board のハードウェアを網羅的に検査することができる試験内容を考案し、それを実現するセットアップを設計した。次に、4 章で述べたシングルボード試験システムの応用として、QAQC 試験に向けた試験システムを開発した。最後に、開発したシステムの動作確認とデモンストレーションを行い、PS board QAQC 試験に向けたシステム構築を完了させた。

PS board 量産のスケジュール

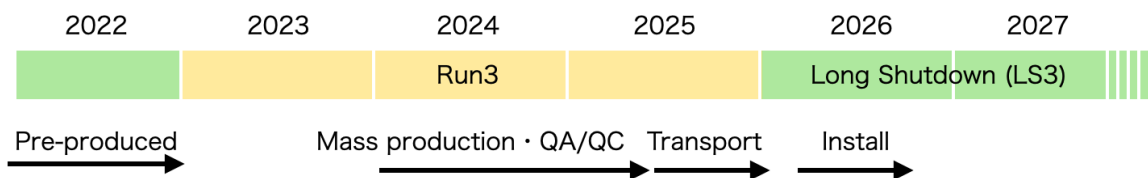


図 5.1 PS board 量産のスケジュール。これまでに第一試作機、第二試作機を用いた試験が行われ、2022 年にプレ量産が完了している。2024 年から 1400 枚以上に及ぶ本量産が始まり、2026 年から ATLAS 実験室への設置作業が開始される。

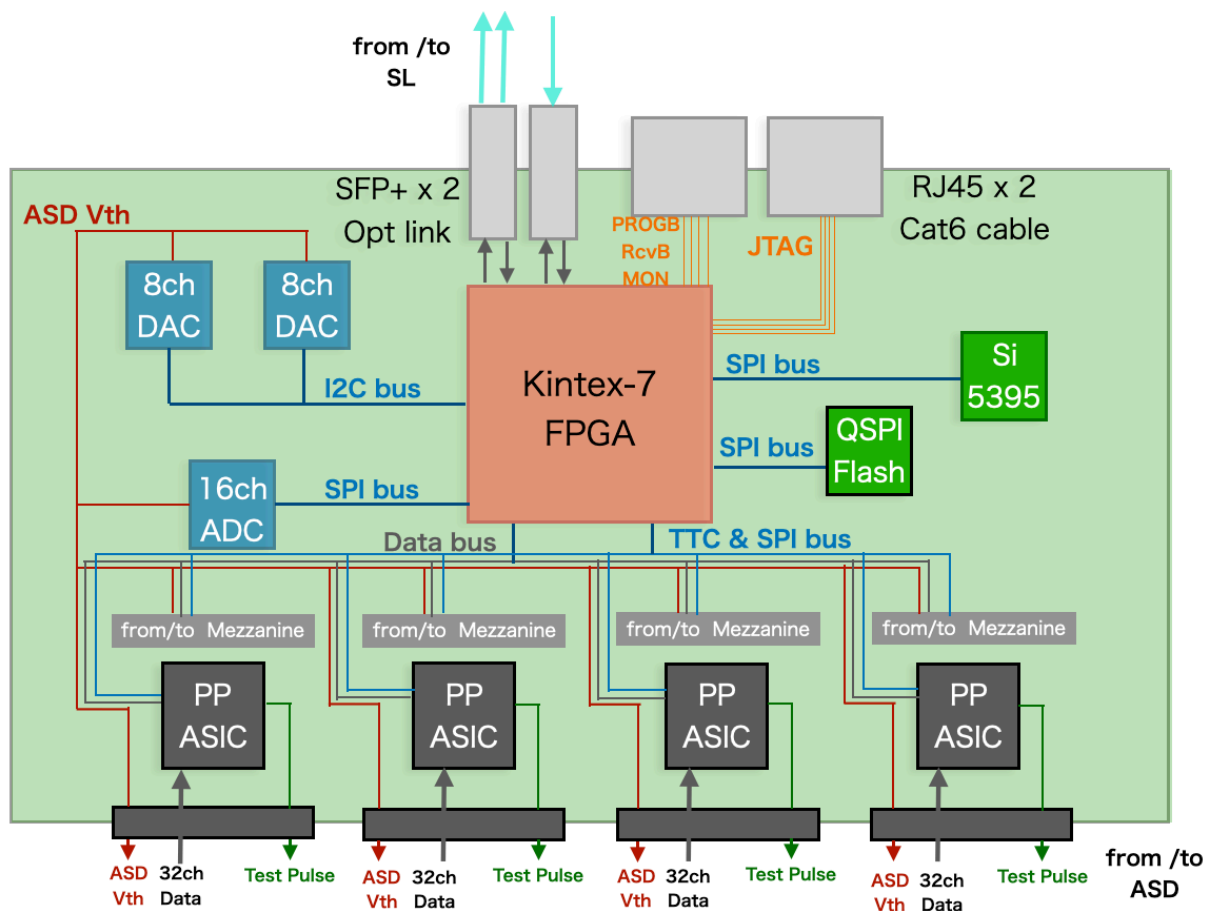


図 5.2 PS board の全体像。PS board に搭載されている素子とその間の配線を示す。PS board は SL と 3 本の光ファイバーで接続され、高速シリアル通信を行う。PS board FPGA から送られる電気信号は SFP+ モジュールで光信号へと変換される。また、PS board は JATHub と 2 本の Cat-6 ケーブルで接続される。1 本は JTAG 線と呼ばれ、JATHub から QSPI フラッシュメモリにファームウェアを書き込む際に利用される。もう 1 本は Recovery/Monitor 線と呼ばれ、PS board に自己修復不可能な SEU が発生した際のリカバリー手続およびクロックの位相測定に利用される。PS board FPGA と DAC は I²C バスで接続され、ADC、Si5395、QSPI、PPASIC とは SPI バスで接続される。DAC から ASD へはアナログの閾値電圧が供給され、ADC はそれをモニターする。PS board FPGA は PP ASIC に TTC 信号を送信し、ヒット信号を受信する。1 つの PP ASIC は 2 台の ASD と接続され、それぞれから 8 チャンネル分のヒットデータを受信する。

5.1.2 PS board に搭載された素子

QAQC 試験ではエレクトロニクス上のすべての素子を網羅的に検査し、素子の不良や実装上の欠陥を確実に検出することが求められる。そのため、PS board に搭載されている素子やそれらの間をつなぐ配線を精査し、試験に適したセットアップおよび試験内容を考案した。

図 5.2 に PS board に搭載されている素子、各素子間の配線を示す。また、以下に PS board 上に搭載されている各素子の役割と各素子間をつなぐ配線をまとめる。

1. SFP+ : エレクトロニクス上の電気信号と光ファイバー上の光信号を相互に変換する光トランシーバーモジュール。PS board FPGA は 3 本の光ファイバーを介して SL と通信する。2 本は送信用に定義されてお

- り、1枚のPS boardが担当する256チャンネルのヒット信号を送信する。1本は受信用に定義されており、コントロール信号およびTTC信号を受け取る。
2. RJ45 コネクタ : Cat-6 ケーブルを接続するためのコネクタ。JATHub と PS board は2本の Cat-6 ケーブルで接続され、LVDS 規格で通信する。1本は JTAG 線と呼ばれ、JATHub から QSPI フラッシュメモリにファームウェアを書き込む際に利用される。もう1本は Recovery/Monitor 線と呼ばれ、PS board に自己修復不可能な SEU が発生した際のリカバリー手続きおよび LHC パンチ交差クロックの位相測定に利用される。
 3. QSPI フラッシュメモリ : SPI バスによる高速データ転送が可能な不揮発性のメモリ。PS board ではファームウェアおよび制御用パラメータを保存するのに利用される。ファームウェアは JATHub から JTAG 線を操作することで、PS board FPGA を経由して書き込まれる。制御用パラメータは SL がコントロール信号に乗せて SPI プロトコルをビットバンギングし、PS board FPGA がそれを中継することで書き込まれる。PS board FPGA は自動でこれらのパラメータを読み出し、PP ASIC や DAC へ分配する(自律型制御機構)。
 4. PP ASIC : ASD からのヒット信号の処理用に開発された ASIC。可変遅延回路における信号遅延の大きさや、陽子パンチ識別回路の有効ゲート幅は ASD ごとに異なる値を設定する必要がある。これらのパラメータは自律型制御機構により、PS board FPGA から SPI バスを通じて設定される。その他に、PS board FPGA から TTC 信号や TPT 信号を受信し、16チャンネル分のヒット信号を送信する。PP ASIC は ASD にテストパルスを供給する。
 5. DAC : ASD のコンパレータにアナログの閾値電圧を供給する。PS board FPGA とは I²C バスで接続される。電圧の大きさや極性は ASD ごとに異なる値を設定する必要があり、自律型制御機構により設定される。設定されたパラメータは、自律型制御機構により定期的に読み出され、モニター用に SL に送信される。
 6. ADC : DAC から ASD に供給される閾値電圧をモニターする。PS board FPGA とは SPI バスで接続される。ADC から読み出された電流のモニター値は自律型制御機構により定期的に読み出され、SL に送信される。
 7. Si5395 : PS board FPGA がシリアルデータから再構成した LHC パンチ交差クロックのジッターを低減し、FPGA、PP ASIC、GTX トランシーバへ分配する。PS board FPGA とは SPI バスで接続される。Si5395 ではクロックの入出力ポートの設定や周波数の設定を行う必要があるが、これらのパラメータは1434枚のPS boardで共通である。そこで、このパラメータは QSPI フラッシュメモリではなく、ファームウェア内の BRAM に固定値として保存される。

5.1.3 セットアップおよび試験内容の設計

5.1.2 節で述べたすべてのインターフェイスと素子を網羅的に検証可能なセットアップとして、JATHub を用いた試験システムを考案した。JATHub は PS board を試験するための十分なインターフェイスを有していることに加え、拡張性に富んだ Zynq SoC デバイスをメインドライバーとして搭載している。この特性から、JATHub の PS を起点にすべての試験を完結させる、コンパクトな試験システムを実現できると考えた。図 5.3 にその概念図を示す。

このシステムでは PS board 1 台の試験に、JATHub 1 台を用い、JATHub と PS board を2本の Cat-6 ケーブルと3本の光ファイバーで接続する。このシステムのコンセプトは、SL が担う PS board 制御機能を JATHub に

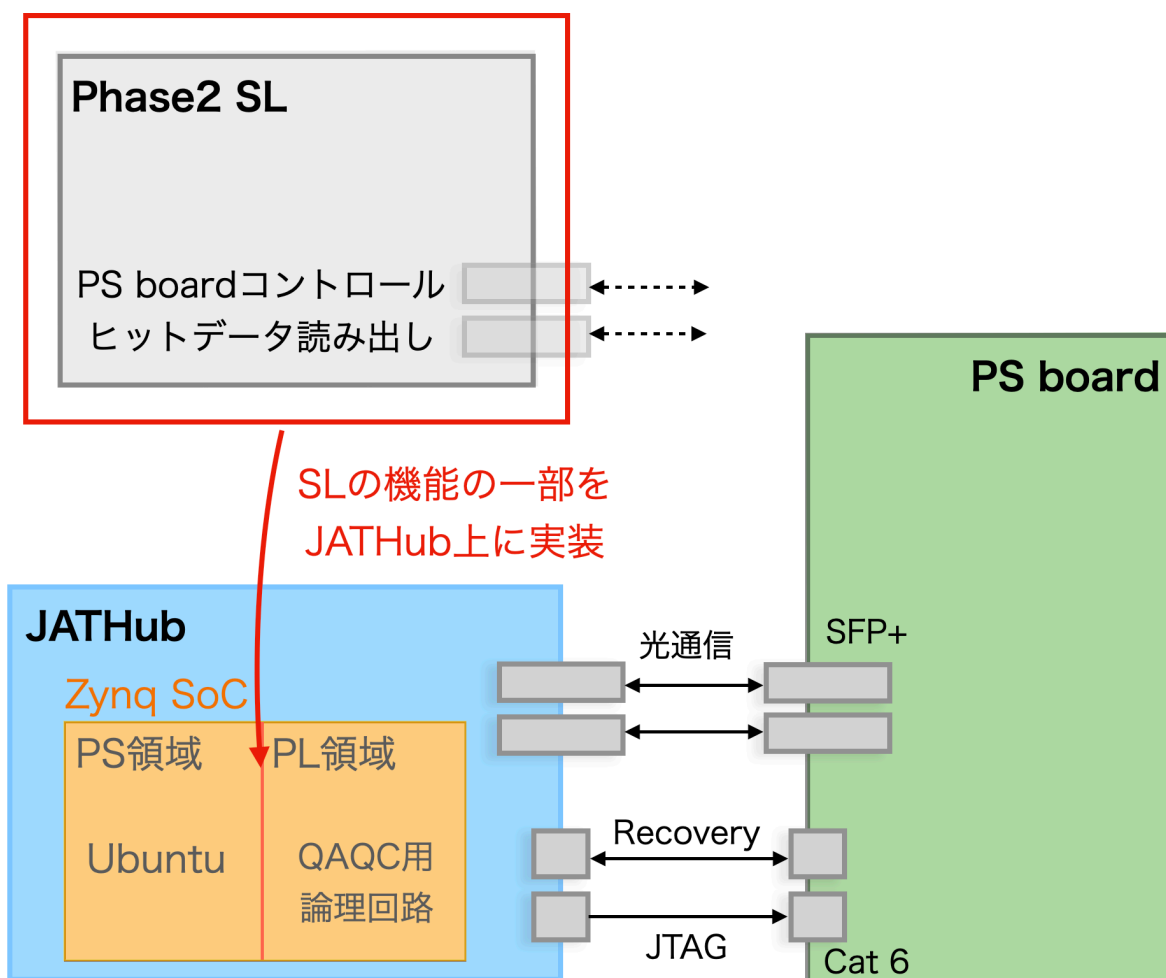


図 5.3 PS board QAQC 試験のセットアップ。試験では PS board 1 台に対して JATHub 1 台を利用する。JATHub と PS board は 2 本の Cat-6 ケーブルと 3 本の光ファイバーで接続する。JATHub に PS board を制御するを実装することで、SL を用いずに試験を完結させる。

実装することで、JATHub 1 台で PS board の試験を完結させることである。SL の駆動には ATCA クレートが必要で、大掛かりなセットアップが必要となる。一方、JATHub はデスクトップで給電することも可能であり、場所を選ばない汎用的な試験システムを実現する。以降、QAQC 試験のマスター用に開発する JATHub を、QAQC 用 JATHub と呼ぶ。

QAQC 用 JATHub の Zynq PS 領域には Ubuntu を起動する。試験では、ローカル PC からイーサネット経由で Ubuntu にアクセスし、Ubuntu 上で実行したアプリケーションを起点に試験用信号の送信や、試験用データの読み出しを行う。Ubuntu は汎用的な OS であり、既存のソフトウェアを用いてネットワークや web サーバーを構築することができる。また、コンパイラをインストールすることで Ubuntu 上で直接アプリケーションを作成することができるため、開発を簡単に進めることができる。

FPGA 部分である PL 領域には試験用に新しくファームウェアを開発する。JATHub が担う Cat-6 ケーブルを介して PS board を制御する機能 (JTAG 線をドライブする機能、リカバリー手続き、クロックの位相測定) に加えて、SL が担う光ファイバーを介して PS board を制御する機能 (高速光シリアル通信、固定位相でのクロック分配、コントロール信号の送信、固定時間でのヒットデータ読み出し) も実装することで、JATHub1 台で PS board の有する機能を網羅的に試験する。

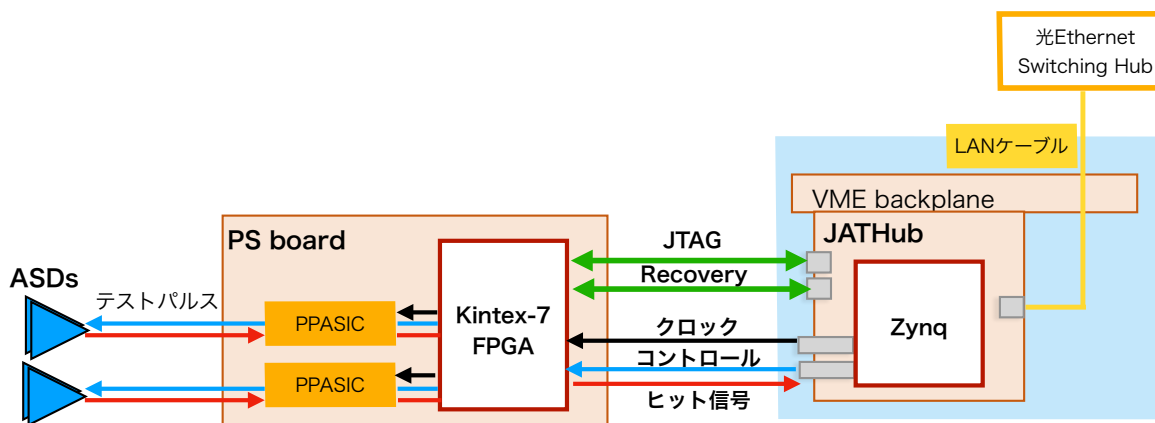


図 5.4 QAQC 用 JATHub を用いた ASD テストパルス試験の概要。QAQC 用 JATHub が試験のマスターとして、PS board の制御、TPT の発行、ヒットデータの読み出しを行う。Zynq SoC の PL 領域に PS board 制御と読み出しのための回路を実装し、Ubuntu 上のアプリケーションを起点に TPT の発行およびヒットデータの読み出しを行う。Ubuntu から読み出されたヒットデータは、SD カード上にテキストファイルとして保存される。

PS board ハードウェアの動作を網羅的に検証することができる試験として、ASD テストパルス試験と JTAG/Recovery/Clock monitor 試験を考案した。以下にそれぞれの試験の概要を示す。

ASD テストパルス試験

ASD テストパルスは、PP ASIC から ASD に送られる試験用の電荷であり、ASD から SL までのデータパスの試験に使用される。高輝度 LHC-ATLAS 実験の TGC システムでは、TPT 信号を含む TTC 信号は、CTP から SL を中継して各フロントエンドエレクトロニクスに分配される。テストパルス生成回路は TPT 信号を受信すると、参照クロックである 40 MHz クロックの立ち上がりと同期した差動の矩形波を ASD に送信する。

PS board QAQC 試験では QAQC 用 JATHub が試験のマスターとして、PS board を制御する。図 5.4 に概要を示す。Zynq SoC の PL 領域に PS board 制御とヒットデータ読み出し回路を実装し、Ubuntu 上のアプリケーションを起点にそれらを制御する。Ubuntu から読み出したヒットデータは、SD カード上にテキストファイルとして保存される。

この試験では TPT 信号を発行してから、固定時間後のヒットビットマップを読み出し、PS board が担当する 256 チャンネル全てにヒットデータが含まれていることを確認する。これにより、PS board の制御パスと読み出しパス、およびそれに関係する全ての素子、素子間の導通を検証することができる。具体的には光インターフェイス (SFP+) が正常に動作し光通信が問題なく行われていること。QSPI フラッシュメモリへのパラメータ書き込みおよび PP ASIC、DAC、Si5395 へのパラメータ分配が正常に行われていること。PP ASIC によるヒットデータの処理、DAC による閾値電圧の供給、クロックジッタークリーナーによるクロックの分配が正常に動作していること、などを網羅的に検証することができる。以下に具体的な手順について述べる。

1. QAQC 用 JATHub をマスターとして、Ubuntu 上のアプリケーションを起点に PS board に制御パラメータを設定する。コントロール信号を利用して SPI バスを制御し、PS board 上の QSPI フラッシュメモリにパラメータを書き込む。その後、PS board の自律型制御を駆動し、パラメータを DAC、PP ASIC に分配する。PP ASIC の制御パラメータには、ヒット信号遅延、有効ゲート幅、テストパルスの極性、テストパルスの波高、テストパルスの時間幅などが含まれる。

2. QAQC 用 JATHub からコントロール信号に乗せて TPT を発行する。また、固定レイテンシー後に L1A 信号を発行し、3BC 分 (Previous BC、Current BC、Next BC) のヒットデータを読み出す。TPT を発行してからヒット信号が返ってくるまでのレイテンシーは、試験セットアップ (シグナルケーブルや光ファイバーの長さなど) に依存する。試験開始前にレイテンシーを測定し、Current BC にヒットが返ってくるよう試験パラメーターを設定する。
3. TPT の発行とデータ読み出しを複数回繰り返す。読み出したデータにヒットが入っていた割合を Efficiency として評価する。PS board が期待通り動作している場合、Current BC の Efficiency が 1、Previous BC および Next BC の Efficiency が 0 になる。

JTAG/Recovery/Clock 試験

JATHub とのインターフェースの動作検証のため、JATHub からのファームウェアの書き込み、リカバリー手続き、クロック位相測定が正常に動作することを確認する。以下に各試験の概要を示す。

■JTAG 試験

JTAG 試験は Zynq に起動した OS から JTAG 線をドライブして、QSPI フラッシュメモリーにファームウェアを書き込む試験である。これには Serial Vector Format (SVF) player と呼ばれるアプリケーションを用いる。SVF player は SVF ファイルと呼ばれる、JTAG4 線をドライブするパターンを記述した ACSII (テキスト) ファイルを読み込み、PS-PL 間チップ通信を利用して JTAG 線を操作する。SVF ファイルは、QAQC 用 JATHub 内の SD カード上に置かれる。

■リカバリー試験

リカバリー試験は PS board から擬似的に救難信号を出力させ、一連のリカバリー手続きが正常に完了することを確認する。これにより Recovery Request 線および Program 線の導通とファームウェアのリセット機構の動作を検証する。

■クロック位相測定試験

クロック位相測定試験では、Monitor 線を通じて PS board が再構成した LHC バンチ交差クロックの位相を測定する。これによりクロック分配線の導通および Si5395 の動作を確認する。先行研究で開発された、JATHub 内部でのクロック位相測定の方法を図 5.5 に示す。JATHub 内の水晶発振器から生成した 40 MHz クロックを参照クロックとして、その立ち上がりのタイミングで LHC バンチ交差クロックをラッチする。参照クロックを 1/56 ns 刻みでスキャンしながら、ラッチを繰り返すことで LHC バンチクロックの位相を測定する。

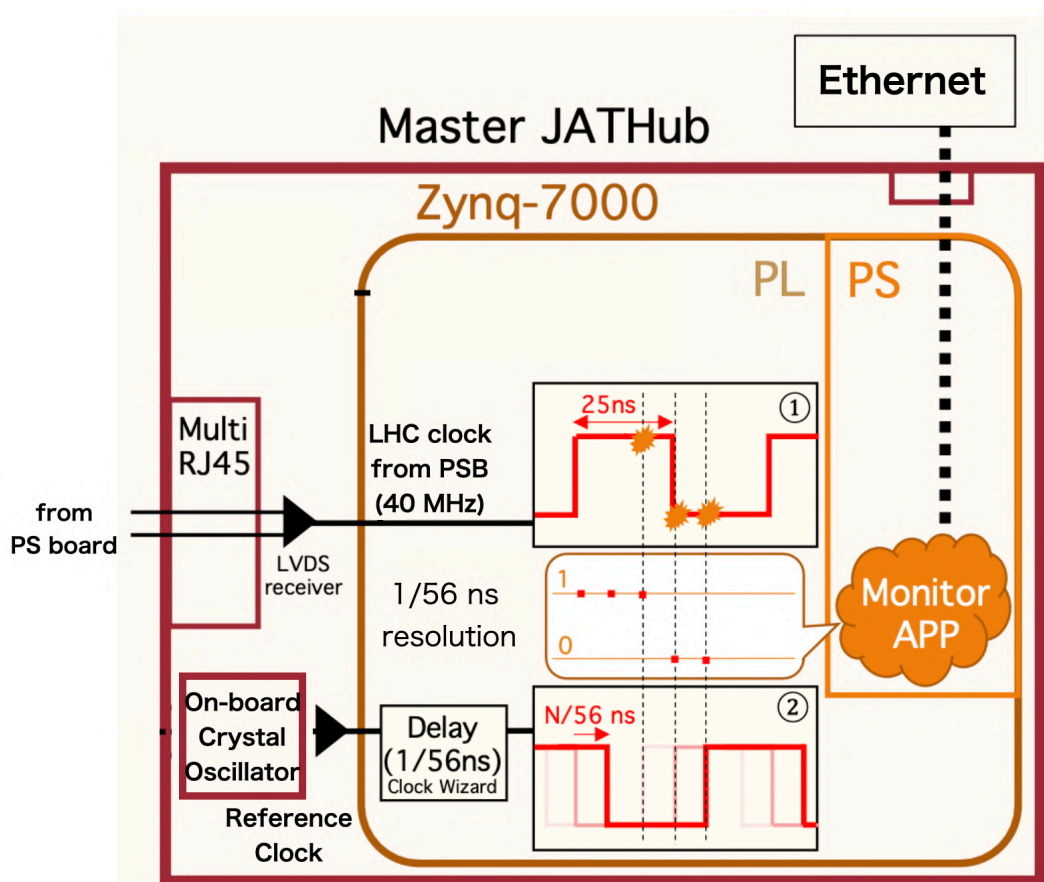


図 5.5 JATHub によるクロック位相測定概念図 [18]。PS board から Monitor 線 (MON 線) を通じて受信した 40 MHz クロックの位相を、JATHub 内部の水晶発振器で生成した 40 MHz クロックを用いて測定する。参照クロックを 1/56 ns 刻みでスキャンしながら、立ち上がりのタイミングで 40 MHz クロックをラッチすることで位相を測定する。

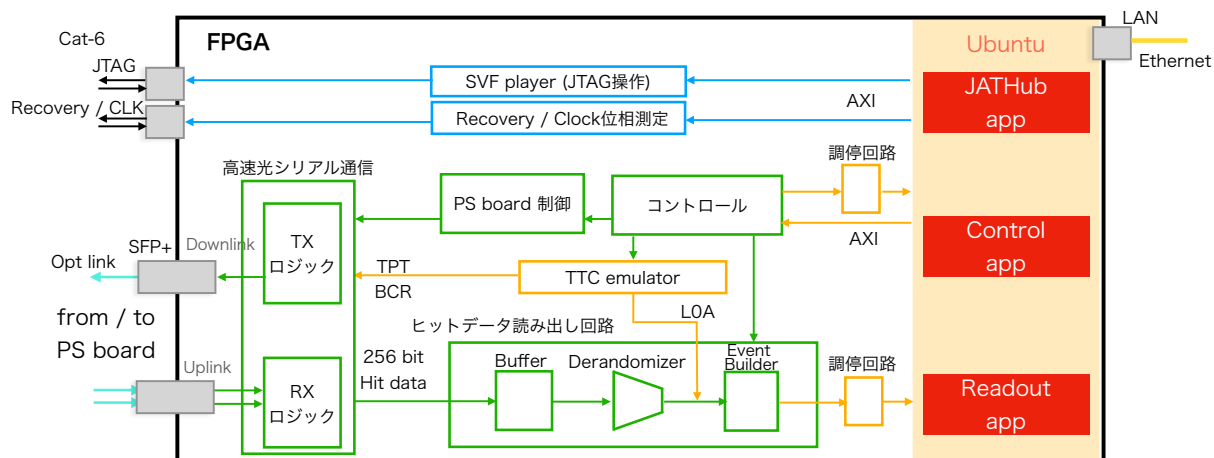


図 5.6 コンパクト DAQ システムの全体像。本システムは大別して、インフラ機能 (図中オレンジ色)、ASD テストパルス試験のための機能 (図中緑色)、JTAG/Recovery/Clock 試験機能 (図中青色) の 3 つの機能に分けられる。PS 領域には Ubuntu を起動し、アプリケーションを起点に FPGA を操作する。Ubuntu から FPGA の操作は AXI GPIO を利用し、読み出しには自作調停回路を用いる。TTC 信号も JATHub 内部で生成し PS board へ分配する。ASD テストパルス試験のための機能は、高速光シリアル通信、PS board への制御信号の送信、固定時間でのヒットデータ読み出しで構成される。JTAG/Recovery/Clock 試験に向けて、SVF player、リカバリー手続、クロック位相測定を実装する。

5.2 コンパクト DAQ システムの機能開発

本節では 5.1.3 節で考案した試験を実現するために開発した、QAQC 用 JATHub の機能実装について述べる。システムの全体像を図 5.6 に示す。本システムは大きく分けて 3 つの機能に分けられる。

1 つ目は QAQC 用 JATHub 単体で PS board 制御および DAQ を完結させるためのインフラの実装である (図中オレンジ色)。Zynq の PS 領域には Ubuntu を起動し、LAN ケーブルを介した Ethernet 通信を行う。ASD テストパルス試験と JTAG/Recovery/Clock 試験はいずれも Ubuntu 上のアプリケーションを起点に実行する。そのために Ubuntu から FPGA を操作する機能、FPGA から Ubuntu へデータを読み出す機能をそれぞれ実装する。QAQC 試験では JATHub をマスターとして、PS board、ASD のタイミング制御も行うため、JATHub 上の水晶発振器で生成した 160 MHz クロックを分周した 40 MHz クロックを LHC バンチ交差クロックの代わりに、基準クロックとして利用する。またそれぞれのシステムのタイミングを制御する TTC 信号もこれを元に JATHub 上で生成する。

2 つ目は ASD テストパルス試験のための実装で、基本的には SL が担う機能を模したものである (図中緑色)。これには PS board との高速光シリアル通信、固定位相でのクロック分配、PS board への制御信号の送信、固定時間でのヒットデータ読み出しが含まれる。

3 つ目は JTAG/Recovery/Clock 試験のための実装で、本番運用での JATHub の担う機能である (図中青色)。これには SVF player、リカバリー手続、クロック位相測定が含まれる。本研究に取り組んだ 2022 年当時、高輝度 LHC-ATLAS 実験用の SL は開発段階にあったため、ASD テストパルス試験のための機能実装は Run3 の SL を参考に SoC デバイス用に応用した。JTAG/Recovery/Clock 試験のための機能実装は、先行研究 [18] により機能開発が完了していたため、それを用いた。以下にそれぞれの機能実装について述べる。

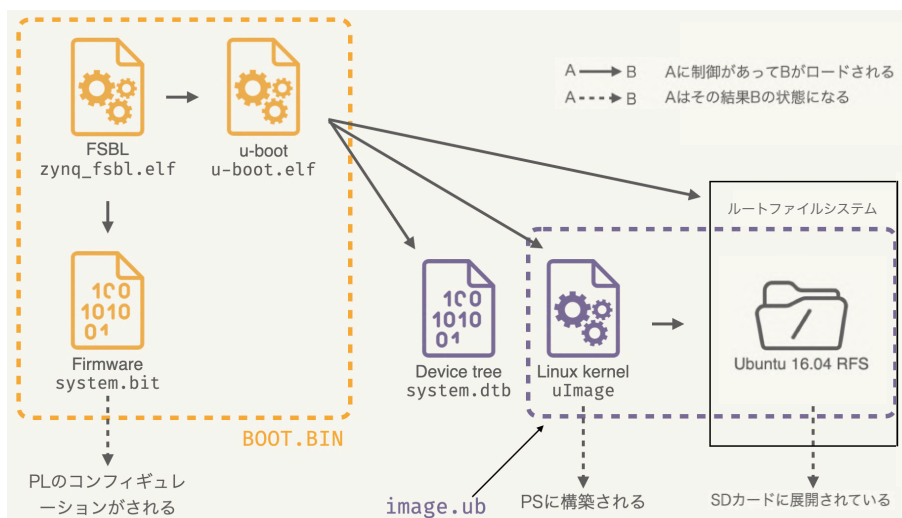


図 5.7 Ubuntu の起動シーケンス [25]。JATHub に電源を投入すると、まず FSBL がロードされ、ファームウェアのビットストリームが PL 領域に書き込まれる。次に PS 領域に Linux を起動するため、u-boot が起動し、デバイスツリー、カーネルイメージ、Ubuntu のルートファイルシステムがロードされる。

5.2.1 インフラの実装

Zynq PS 領域における Ubuntu の起動

Zynq SoC の PS 領域には標準的な Linux OS である Ubuntu を起動する。Ubuntu は汎用性と拡張性に富んだ OS で、ネットワークの設定や QAQC 用 JATHub 内部でのアプリケーション開発を容易に行うことができる。

Zynq 組み込みデザインの開発には 64 bit Ubuntu 18.04.6 を用いた。Zynq PL 領域に構築する自作論理回路の開発や PS 領域の IO 設計は、Xilinx 社が提供する "Vivado 2020.2" を利用した。Zynq PS 領域で走る Linux の設定は Xilinx 社が提供するクロスコンパイラ "Petalinux 2020.2" を利用した。Petalinux では Vivado で生成したハードウェア記述ファイルを元にデバイスツリーや Root File System (rootfs) を設定することで、Zynq の起動に必要なブートファイルを作成することができる。

QAQC 用 JATHub では Ubuntu の起動に SD カードを利用する。SD カードには 2 つのパーティション^{*1}を用意し、Zynq の起動に必要なブートファイルと Ubuntu のルートファイルシステムをそれぞれ展開する。図 5.7 に Zynq 上での Ubuntu 起動の流れを示す。JATHub に電源を投入すると以下のシーケンスで Ubuntu が起動する。

1. First Stage Boot Loader(FSBL) がロードされる
2. ファームウェアのビットストリームが SoC の PL 領域に書き込まれる
3. Linux カーネルや OS を起動するためのブートローダーである u-boot がロードされ、制御が移行される。
4. u-boot 制御下でデバイスのハードウェア情報を記述したデバイスツリーがロードされる。
5. u-boot 制御下で Linux kernel がロードされ PS 領域に構築される。
6. 制御が Linux カーネルに移行され Linux が起動する。

^{*1} ブートファイル用のパーティションは fat32、Ubuntu の rootfs 用のパーティションは ext4 で展開する。

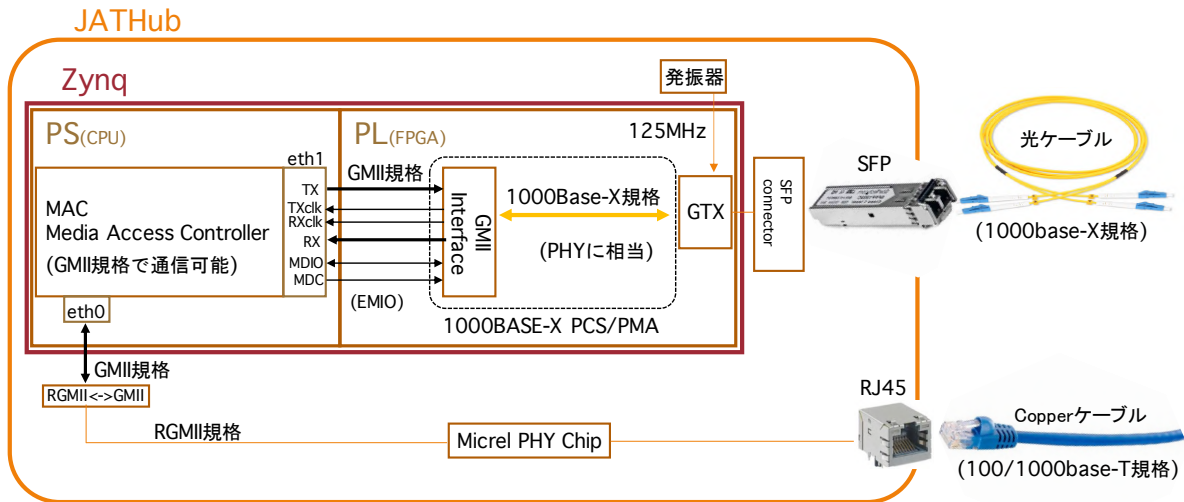


図 5.8 光 Ethernet 通信の仕組み [18]。QAQC 試験で用いる JATHub 試作 1 号機は光を介した方法と LAN ケーブルを介した方法の 2 種類で Ethernet 通信を行うことができる。QAQC 用 JATHub では光ファイバーを PS board との通信用に使用するため、LAN ケーブルを用いた方法を採用する。

LAN ケーブル経由のネットワーク通信

QAQC 試験で用いる JATHub 試作 1 号機は図 5.8 に示す、2 通りの方法で Ethernet 通信を行うことができる。1 つ目は LAN ケーブルを使用するもので、回路上に搭載された PHY chip (Micrel PHY Chip) を利用して、Ethernet 信号を CPU が扱える信号に変換する。2 つ目は光ファイバーを用いる方法で、GTX トランシーバーで受けた光信号を 1000BASE X PCS/PMA と呼ばれる IP ブロックを利用して処理する。QAQC 用 JATHub では 3 本の光ファイバーは PS board との通信用に利用するため、LAN ケーブルを用いる方法を採用する。

AXI GPIO を用いた PS-PL 間通信

PS 領域から PL 領域への通信は、汎用入出力インターフェースである AXI General Purpose Input Output (GPIO) を介して行う。AXI GPIO によって接続された PL のレジスタには固有の物理アドレスが割り当てられる。このアドレスは Vivado の Address Map で確認することができ、Address Editor にてユーザーが自由に変更することができる。

PS から AXI GPIO レジスタへは少なくとも 2 通りの方法でアクセスすることができる。1 つ目は Ubuntu ルートファイルシステム内の /dev/mem が提供するキャラクターデバイスをアプリケーションから直接開く方法である。/dev/mem を介したアクセスでは Ubuntu が扱うすべての物理アドレスに制限なくアクセスすることができるため、簡単に使用できる。一方、カーネル動作に必要なレジスタにも意図せずアクセスする危険があるため、カーネルを壊す危険性がある。2 つ目の方法は特定の AXI GPIO レジスタを User space I/O (UIO) としてデバイスツリーに登録し、アプリケーションから UIO ドライバーを介してアクセスする方法である。この方法では UIO に登録したアドレス以外へのアクセスは禁止されるため、カーネルを壊す危険なく安全に制御できる。本システムのコントロールパスにおいては、より実装が簡単な /dev/mem を用いる方法を採用した。

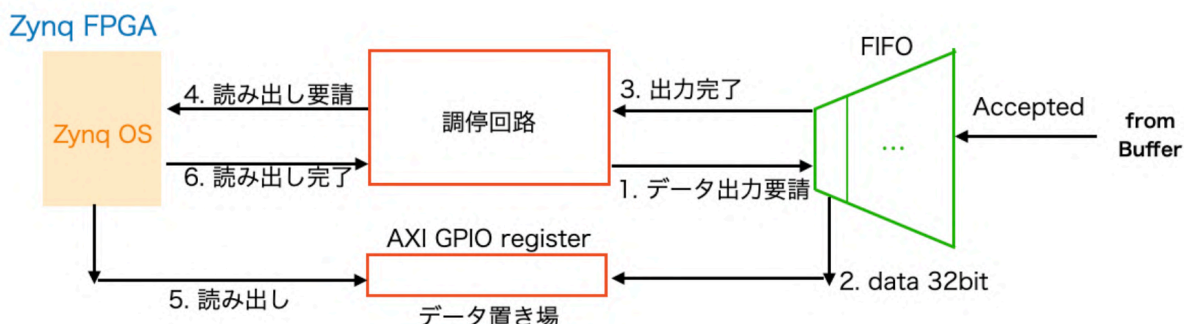


図 5.9 PL から PS へのデータ読み出しシステム概要。Ubuntu から FPGA 内のデータを読み出すために、AXI GPIO レジスタを利用する。FIFO がレジスタのデータを更新するタイミングと Ubuntu がレジスタのデータを読み出すタイミングは制御されている必要があり、制御のために調停回路を実装した。調停回路により、1~6 の動作順序が保証され、安定した読み出しが保証される。

PL から PS へのデータ読み出し

高速かつ安定的に Zynq PL 領域から PS 領域へとデータを読み出すため、本研究では自作調停回路を用いた読み出しシステムを開発した。図 5.9 に実装した読み出しシステムの概要を示す。

データの読み出しシステムは、FIFO (First In First Out) メモリー、AXI GPIO レジスタ、自作調停回路で構成される。FIFO メモリーは、データを一定の順番で保持するメモリーで、最初に格納されたデータが最初に取り出される。この特性を利用して、FIFO を PL 領域から PS 領域へデータを読み出す際のバッファとして動作させる。FPGA から読み出すデータは、FPGA 内の動作クロックに同期して FIFO メモリーに書き込まれる。書き込まれたデータは、PS 領域に起動した Ubuntu の動作するタイミングに合わせて読み出される。しかし、PS 領域から FIFO メモリーに直接アクセスする方法はないため、データの受け渡しには、AXI GPIO レジスタを利用する。

AXI GPIO レジスタは FPGA 内の FIFO からも Ubuntu からも任意のタイミングでアクセスすることができるため、両者の動作タイミングを調整する仕掛けが必要である。例えば、Ubuntu がデータを読み出す前に FIFO が AXI GPIO レジスタのデータを書き換えると、書き換える前のデータは Ubuntu から読み出されることなく失われることになる。また、FIFO がデータを書き換える前に Ubuntu が 2 回読み出し動作を行うと、同じデータを重複して読み出すことになる。

すべてのデータを漏れや重複なく読み出すために、FIFO と Ubuntu の動作順序を制御するのが自作調停回路である。この回路は Ubuntu からも FPGA からも操作することができる 1 bit のフラグと、ステートマシンで構成される。フラグは Ubuntu と FIFO 間の情報伝達に利用し、0 を FIFO からの書き込み待ち、1 を Ubuntu からの読み出し待ちと定義する。調停回路で実現される読み出しシーケンスを図 5.10 に示す。Ubuntu 側はフラグをモニターし、1 であることを確認すると AXI GPIO レジスタの値を読み出し、その後、フラグを 1 から 0 に下げる。FPGA 上に実装したステートマシンも同時にフラグをモニターし、0 であることを確認すると FIFO にデータ更新用の read enable 信号を送り、フラグを 0 から 1 に上げる。これにより FIFO が AXI GPIO レジスタのデータを更新する動作と Ubuntu がレジスタのデータを読み出す動作の順序が保たれ、漏れや重複のない読み出しが実現される。

作成した読み出しシステムの動作検証や性能評価は 5.3 節で行う。また、この読み出しシステムは PL 領域から PS 領域へのデータ読み出しに複数箇所使われており、汎用読み出しシステムと呼ぶ。

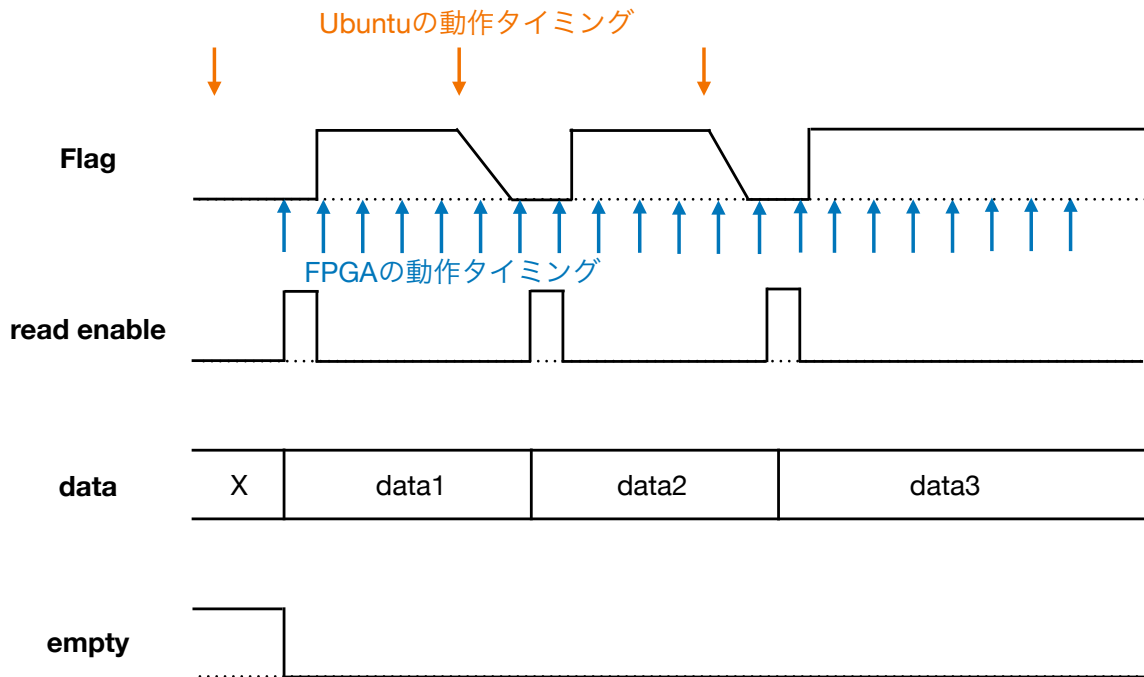


図 5.10 調停回路のシーケンス。Ubuntu 側と FPGA 上のステートマシン側で共通のフラグを操作・モニターすることで両者がコミュニケーションをとる。フラグは 0 を FIFO からの書き込み待ち、1 を Ubuntu からの読み出し待ちと定義する。Ubuntu はフラグをモニターし、1 になったら AXI GPIO からデータを読み出し、その後フラグを 0 に下げる。FPGA の調停回路もフラグをモニターし、0 になったら FIFO に read enable 信号を送信し、その後フラグを 1 に上げる。これにより、FIFO が AXI GPIO レジスタのデータを更新する動作と Ubuntu がレジスタのデータを読み出す動作の順序が保たれ、漏れや重複のない読み出しが実現される。

TTC emulator

ASD テストパルス試験では JATHub、PS board、ASD の間で固定位相で LHC バンチ交差クロックを分配し、それぞれのエレクトロニクスが同期して動作する必要がある。TGC システムでは LHC バンチ交差クロックは CTP から分配されるが、本試験システムでは JATHub 1 台で試験を完結させるため、JATHub 上の水晶発振器で生成されたクロックを基準クロックとして扱う。Bunch Counter Reset (BCR) 信号や Event Counter Reset (ECR) などの TTC 信号も、JATHub 内で生成し PS board へと分配する。この役割を果たすのが TTC emulator である。ここで発行された TTC 信号はコントロール信号のワード 0 に埋め込まれ、光ファイバーを介して PS board へと送られる。L1A 信号も TTC emulator から出力され、前述したように TPT と L1A を同期して制御することで Fixed latency の DAQ を実現している。以下に TTC emulator のサブモジュールを説明する。

- **TTC generator:** 40MHz LHC バンチ交差クロックで動作するカウンター。reset 信号でカウンターをリセットし、1 クロックチック毎にカウンターの値を 1 つずつインクリメントする。カウンターの値が 3564 に達したタイミングで BCR 信号を発行する。デフォルトの設定では TPT、L1A も BCR に合わせて 3564 BC に一回発行しているが、TPT、L1A の発行頻度は任意の値に設定することができる。TPT length を変更することで、TPT を複数 BC に渡って出力することも可能になっている。
- **TTC Delay:** 1 bit 幅、深さ 4096 列の BRAM で実装したディレイ回路。L1A、BCR、TPT に任意の遅延をかけることができる。L1A Delay を調整することで TPT から L1A を発行するまでのレイテンシーを変更す

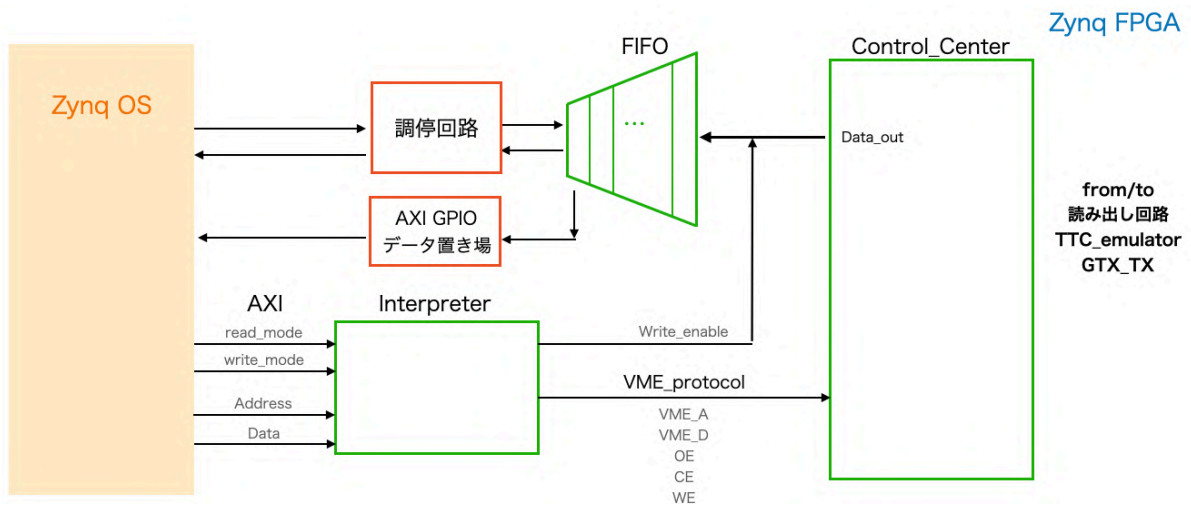


図 5.11 JATHub コントロール回路。Zynq の OS から PL 領域の Control Center 内のレジスタを操作することにより FPGA 内の各機能の制御や PS board へのコントロール信号を制御する。Control Center 内のレジスタ操作は Interpreter が仲介する。Ubuntu から AXI GPIO を介してアドレスとデータを指定すると、Interpreter は VME プロトコルに従ってレジスタの書き込みを行う。データの読み出しには自作調停回路を利用する。

ることができる。

- ID counter : BCR、ECR、L1A を受けて BCID、ECID、L1ID を数え上げるカウンター。ここで発行された BCID や L1ID は Ubuntu からの読み出しフォーマットに組み込まれて出力される。読み出したデータにおける L1ID の連続性や BCID を確認することでデータの欠損や重複をチェックすることができる。
- FPGA テストパルス発行機能 : PS board の持つ FPGA テストパルスを発行するためのモジュール。FPGA テストパルスは PS board 内の BRAM に保存される。BRAM の address を指定した状態で、TPT を発行すると BRAM から 256 bit のヒットビットマップが取り出され、ASD からのヒット信号の代わりに JATHub に送信される。

5.2.2 ASD テストパルス試験のための機能実装

コントロール

QAQC 用 JATHub 上の各機能の制御は Control Center から行う。Control Center では複数のレジスタがインスタンス化されており、このレジスタの値を書き換えることで FPGA 内の各機能を制御する。Control Center の操作は Ubuntu 上で実行したアプリケーションを起点に行う。図 5.11 に Ubuntu と Control Center の接続を示す。

Control Center 内のレジスタ操作は、VME プロトコルを模倣した独自のプロトコルに従って行い、Interpreter がその操作を仲介する。Zynq PS 領域と Interpreter は 16 bit の Data、12 bit の Address、読み書きモード選択用のそれぞれ 1 bit の信号線で接続される。Ubuntu は Control Center 内のレジスタアドレスを指定し、read mode または write mode を 1 に引き上げることで、レジスタの読み書きを行う。この設計により、PS から Control Center 内の各レジスタへ直接 AXI バスを接続する必要がなく、PS から PL に伸びる AXI バスの本数を必要最小限にとどめることができる。Control Center から Ubuntu へのデータ読み出しは、前述の汎用読み出しシステムを利用する。

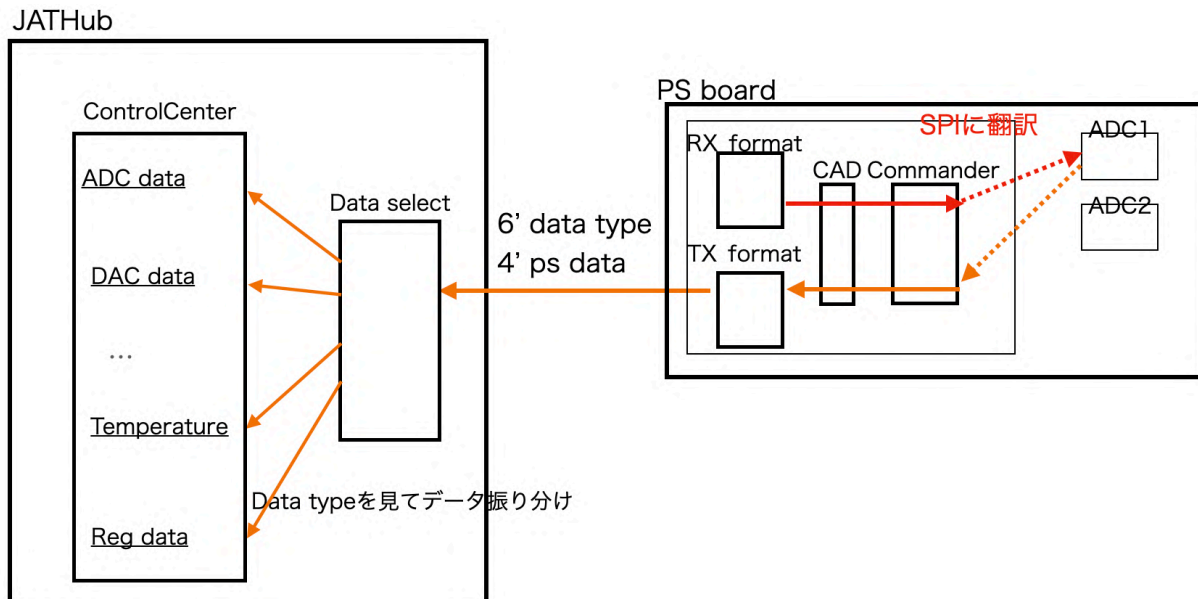


図 5.12 QAQC 用 JATHub に実装されたモニター機構。PS board から送られるモニター用データを自動でデコードし、Control Center 内のレジスタに格納する。これにより、常に最新のモニターデータが Control Center 内のレジスタに反映される。

PS board との通信

PS board の制御は SL と PS board 間で定められた通信フォーマット (図 2.23) に従って、高速光通信を介して行われる。QAQC 用 JATHub にも、このフォーマットに従ったパケット交換を行う機能を実装し、PS board のコントロールおよび LHC クロックの分配を行う。200 MHz の TX ユーザークロックで動作するステートマシンにより、40 MHz おきに 5 ワードを送信する。

PS board へ送るソフトリセット信号や TPT 信号は Control Center 内のレジスタ操作により制御する。PS board FPGA 内のレジスタ操作はワード 2、ワード 3 に定義された 16 bit の Address、Command、Data を用いて実行する。Command により書き込み/読み出し動作を決め、Address で PS board 内のレジスタアドレスを指定し、書き込み動作の場合は Data に設定した値を書き込む。

PS board 上の QSPI フラッシュメモリーへのパラメータ書き込みは、ワード 1 で定義された SPI バス制御用信号 (CS Bitmap および SCLK、SDI) を操作することで行う。PS board FPGA や各素子の状態のモニター値は、自立型制御機構により定期的に読み出される。DAC の設定値、ADC の測定値、FGPA の温度、xADC による供給電圧値などのデータは表 5.1 に示すフォーマットに従い、6 bit のデータタイプと 4 bit のデータに分割され、QAQC 用 JATHub に送信される。QAQC 用 JATHub は図 5.12 に示すように、分割されたモニターデータを再構成し、Control Center 内のレジスタに自動で分配する。これにより、常に最新のモニターデータが Control Center 内のレジスタに反映される。

高速光シリアル通信

PS board と SL 間的高速シリアル通信では、適切な BCID と Fixed latency scheme を実現するため、固定位相でのクロック分配および固定時間でのデータ転送が行われる。QAQC 用 JATHub においてもこれを達成するため、先行研究 [16] で開発された高速シリアル通信システムを実装する。これにより、JATHub や PS board にリセットや再コンフィギュレーションを施しても、両者の基準クロック間の位相関係は変わらず一定となる。この結果、

表 5.1 PS board から送信されるモニターデータのデータタイプ。

Data type [5:0]	PS Data [7:0]
0 0 0 1 0 0	ADC data [15:12]
0 0 0 1 0 1	ADC data [11:8]
0 0 0 1 1 0	ADC data [7:4]
0 0 0 1 1 1	ADC data [3:0]
0 0 1 0 0 0	DAC data [15:12]
0 0 1 0 0 1	DAC data [11:8]
0 0 1 0 1 0	DAC data [7:4]
0 0 1 0 1 1	DAC data [3:0]
0 0 1 1 0 0	xADC data [15:12]
0 0 1 1 0 1	xADC data [11:8]
0 0 1 1 1 0	xADC data [7:4]
0 0 1 1 1 1	xADC data [3:0]
0 1 0 0 0 0	SEM data [15:12]
0 1 0 0 0 1	SEM data [11:8]
0 1 0 0 1 0	SEM data [7:4]
0 1 0 0 1 1	SEM data [3:0]
0 1 0 1 0 0	Tempurature data [15:12]
0 1 0 1 0 1	Tempurature data [11:8]
0 1 0 1 1 0	Tempurature data [7:4]
0 1 0 1 1 1	Tempurature data [3:0]
1 1 1 0 0 0	Register access Address [7:4]
1 1 1 0 0 1	Register access Address [3:0]
1 1 1 1 0 0	Register access data [15:12]
1 1 1 1 0 1	Register access data [11:8]
1 1 1 1 1 0	Register access data [7:4]
1 1 1 1 1 1	Register access data [3:0]

ASD テストパルス試験に置いて、TPT 信号の発行からヒットデータが返ってくるまでのレイテンシーが一定に保たれることが保証される。図 5.13 にシステムの概要を示す。

■TX ロジック

QAQC 用 JATHub と PS board は JATHub 上の水晶発振器で生成される 160 MHz クロック^{*2}を基準クロックとして、これと同期して固定時間でデータを送受信する。

TX ロジックは 160 MHz クロックを 200 MHz に分周したものを、動作クロック (TX ユーザークロック) として利用する。TX ロジックは、200 MHz おきに 32 bit のデータ (1 ワード) を GTX トランシーバーに送信する。GTX トランシーバーは 8b/10b のプロトコルでこれを 40 bit のパラレルデータへとエンコードした後、シリアルデータへと変換する。生成されたシリアルデータは、参照クロックから生成した 4 GHz クロックに乗せられ、PS board に送信される。

^{*2} 正確には LHC バンチ交差クロック 40.079 MHz の 4 倍に相当する 160.316 MHz。

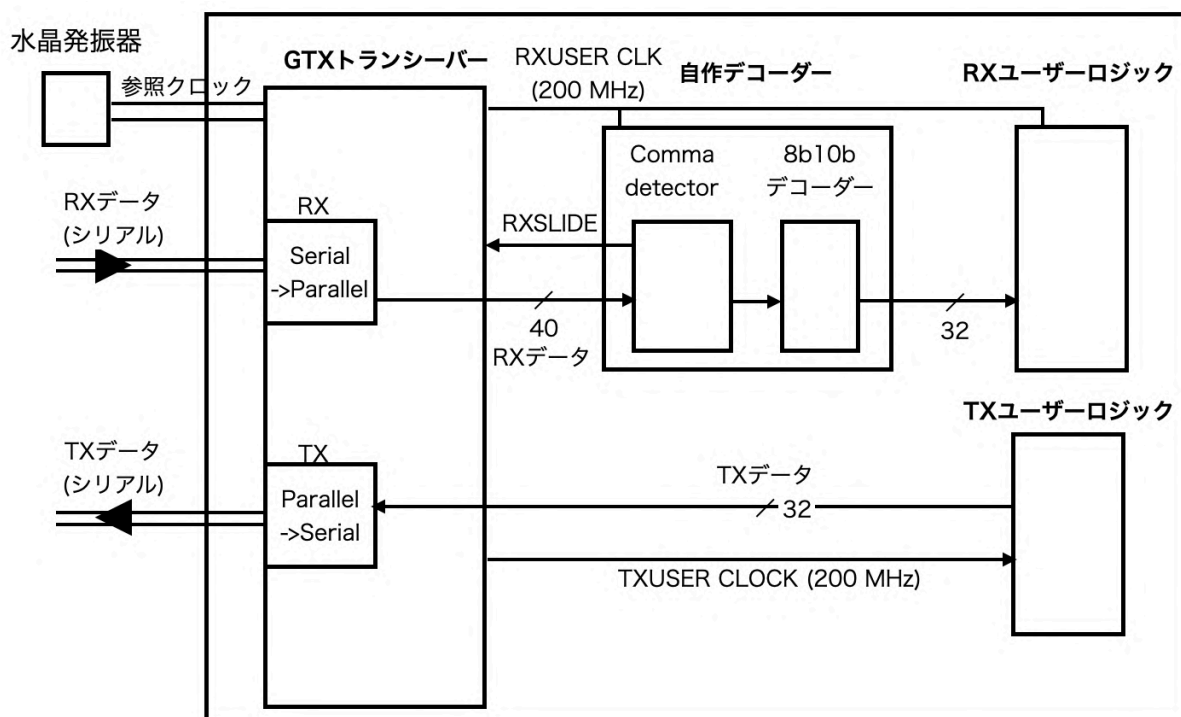


図 5.13 QAQC 用 JATHub における固定位相でのクロック分配 GTX トランシーバーの概要。GTX トランシーバーは JATHub 内部の水晶発振器で生成した 160 MHz クロックを参照クロックとして利用する。TX ユーザーロジックは GTX トランシーバーと同期した 200 MHz の TXUSER CLOCK で動作する。RX ロジックでは、PS board から受信したシリアルデータから、固定位相で RXUSER CLOCK を再構成するため、Comma detector を実装する。これにより JATHub と PS board 間におけるデータの送受信が固定時間で行われることが保証される。

■RX ロジック

QAQC 用 JATHub の RX ロジックでは、Fixed latency でのデータ受信を実現するため、受信したデータから固定位相で RX の動作クロック (RX ユーザークロック) を再構成する必要がある。その実現に重要な役割を果たすが、RX Clock Data Recovery 機構 (CDR) と Comma detector である。CDR 機構とは受信したシリアルデータの立ち上がりまたは立ち下がりのタイミングに同期してクロックを再構成する機能で、受信データから固定位相で 4 GHz クロックを再構成する。このクロックを 1/20 に分周し、200 MHz の RX ユーザークロックを作るが、その過程で合計 20 種類の位相の不定性が生じる。この中から特定の 1 つの位相を決めるのが Comma detector である。Comma データとは送信側と受信側の間で事前に取り決められた 10 bit の予約語で、PS board との通信では 40 MHz に 1 回送信するよう決める。Comma detector は Comma データが 200 bit のシリアルデータの下位 10 bit にくるまで、RX ユーザークロックをシフトする (図 5.14)。これにより 1/20 の分周においても不定性なく、RX ユーザークロックを再構成することができる。

固定時間でのヒットデータ読み出し機能

ヒットデータ読み出し機能は、PS board から受信するヒットデータをイベントごとにまとめ、Zynq PS 領域へと安定的にデータを送る機能である。QAQC 用 JATHub は光ファイバーを介して、PS board から 25 ns ごとに 256 bit のヒット信号を受け取る。ヒットデータとそのデータに割り当てられたイベント情報 (BCID、L1ID) は FPGA 上でバッファーされ (L1 Buffer)、L1A により取り出される。L1 Buffer から取り出された 1 BC 分のデータは FIFO メモリーにダンプされ、CPU の動作する任意のタイミングでデータパスとは非同期に読み出される。

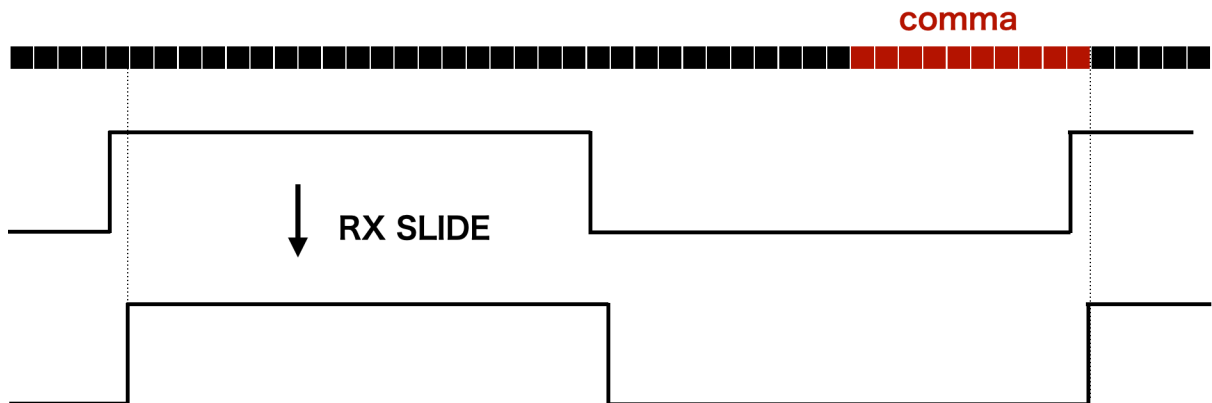


図 5.14 Comma detector の概要。Comma detector は GTX トランシーバーで再構成された 200 bit のシリアルデータの下位 10 bit が Comma ワードと一致するまで、RX SLIDE を打ち続け、RX ユーザークロックの位相をシフトさせる。これにより、受信するデータに対して、RX ユーザークロックの位相を一位に定めることができる。

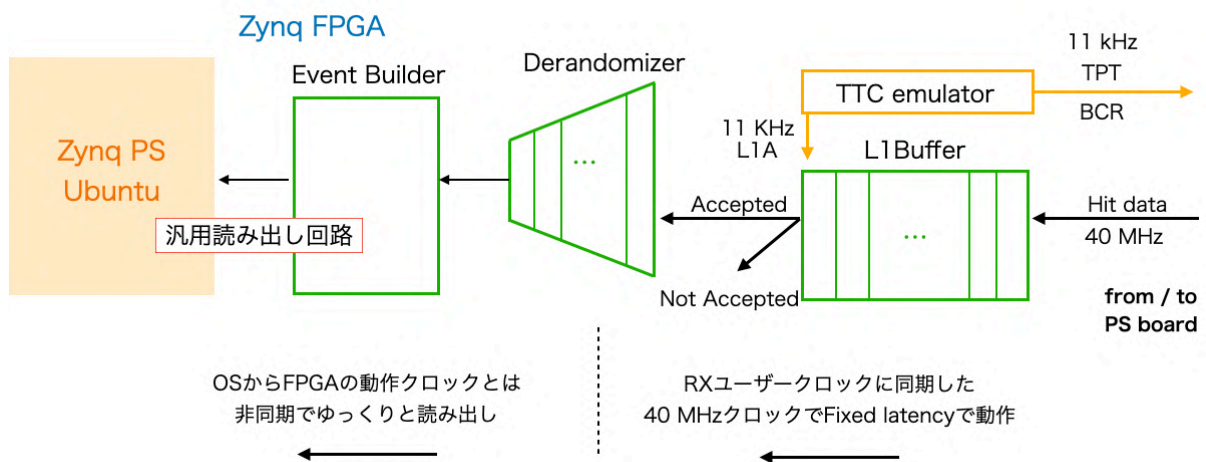


図 5.15 読み出し回路の全体像。PS board から 25 ns ごとに送られる 256 bit のヒット信号は、LHC 交差クロックに同期して L1 Buffer に格納される。TTC emulator は TPT 信号から固定レイテンシー後に L1A を発行し、テストパルスに該当するイベントを L1 Buffer から取り出す。取り出されたデータは Derandomizer に入れられ、CPU の動作する任意のタイミングでデータパスとは非同期に読み出される。

ヒットデータ読み出し機能の全体像を図 5.15 に示す。読み出し回路は L1 Buffer、Derandomizer、Event Builder で構成される。Derandomizer の入力までのデータ処理は RX ユーザークロックを基準クロックに行われ、固定レイテンシーで動作することが保証される。Derandomizer の出力以降は、LHC バンチ交差クロックとは非同期で Ubuntu の動作速度に応じて、処理される。以下に各モジュールの役割を説明する。

L1 Buffer

L1 Buffer は PS board から受信した 256 bit のヒット信号と ECRID、L1ID、BCID などのタイミング情報を合わせた 432 bit のデータを一時的に保管するためのリングバッファである。TTC emulator から L1A が出されたイベントは後段の Derandomizer に転送され、それ以外のデータはここで捨てられる。25 ns ごとに到着するヒット信号は、書き込み用ポインタが指すアドレスに格納される。書き込み用ポインタと読み出し用ポインタは LHC

バンチ交差クロックに同期して、1 ずつインクリメントする。書き込み用ポインタが BRAM の最後尾まで達した場合、次のクロックチックで再び先頭に戻る。L1 depth によって書き込み用ポインタと読み出し用ポインタのアドレスの差を設定することができ、Buffer の深さを任意の値に設定できる。また、L1A が発行されてから何 BC 分のデータを読み出すかを ReadoutBC によって設定することができる。デフォルトでは 3 に設定されており、一回の L1A に対して Previous BC、Current BC、Next BC の 3 バンチ分のデータ読み出す。ReadoutBC で設定された BC 分のデータを読み出している途中で再度 L1A 信号を受信すると、読み出しエラーを出力する。

Derandomizer

Derandomizer は、後段で行われる読み出しの処理待ちバッファであり、432 bit の入力データを 32 bit ずつ出力する。Derandomizer は FIFO を 2 つ直列に並べることで実装している。データの分割には FIFO IP のスライス機能^{*3}を利用している。Derandomizer の出力は Ubuntu の uint32_t 型と整合性のある 32 bit が好ましい。そこで 432 bit の入力データに不要な 64 bit のデータを加え、512 bit のデータを 2 つの FIFO で 1/4 ずつスライスすることで、最終的な出力を 32 bit にしている。64 bit の不要データは 1 つ目の FIFO と 2 つ目の FIFO の間で捨てられ、Ubuntu からの読み出しには関係しない。

Derandomizer へのデータ書き込みレートが Ubuntu からの読み出しレートを上回る場合には、Derandomizer の Occupancy は増大していく。その状態が続くとバッファのオーバーフローが発生し、データが欠損する。バッファオーバーフローが生じた場合の対応は後に紹介する。

Event Builder

Event Builder は Derandomizer に格納された 32 bit 幅のデータを 240 MHz のクロックチック毎に 1 ワードずつ順番に読み出し、イベントごとに所定のフォーマットに成形する。図 5.16 に 1 イベント分の読み出しフォーマットを示す。PS board から受信した 256 bit × 3 BC 分のデータに加えて、TTC emulator から発行された TTC 信号 (ECRID、L1ID、BCID) および PSB で発行された BCID が含まれる。JATHub 内で割り当てられた BCID と PS board から返ってきた BCID の差を確認することで、固定レイテンシーでのデータ通信が実現されていることを確かめることができる。

読み出し回路の性能

実装した読み出し回路における、読み出しレートの上限を概算することは、QAQC 試験を設計するにあたり重要である。例えば、ASD テストパルス試験における TPT 発行レートはこれにより決められる。Ubuntu 上でアプリケーションを実行し、FIFO に格納された 1000 イベント分のパケットを読み出すのにかかる時間を測定した。図 5.17 にその結果を示す。横軸に読み出したパケット数、縦軸に経過時間 (s) をとる。得られた測定結果を線形フィットし、その傾きから 1 パケット読み出すのにかかる時間を概算した。その結果、1 イベント分のデータを読み出すのに概ね 85 μ s がかかることがわかった。

バッファオーバーフローへの対応

万が一、バッファオーバーフローが生じた場合においてもイベントパケットが崩れることのないよう、Derandomizer にはバックプレッシャー機能を実装した。Derandomizer の Occupancy があらかじめ設定した容量閾値 (4000/4096) を超えると、L1 Buffer へ almost full 信号が送られる。almost full 信号を受け取った L1 Buffer

*3 FIFO IP では入力 bit 幅の 1/2、1/4、1/8 の bit 幅での出力が可能である。

	Forth Byte [31:24]	Third Byte [23:16]	Second Byte [15:8]	First Byte [7:0]
word [0]	Header (0xb0d0)			0x0
word [1]	ECRID (8bit)		L1ID (24bit)	
word [2]	0x0	BCID by JATHub (12bit)		JATHub ID (12bit)
word [3]	Tga 0x0 (Previous)	0x1 (RX1)	0xf (data tag)	Privious RX1 BCID by PSB (12bit)
word [4]	Previous RX1 hit data (32bit x 4)			
word [5]				
word [6]				
word [7]				
word [8]	Tga 0x0 (Previous)	0x0 (RX0)	0xf (data tag)	Privious RX0 BCID by PSB (12bit)
word [9]	Previous RX0 hit data (32bit x 4)			
word [10]				
word [11]				
word [12]				
word [13]	Tga 0x1 (Current)	0x1 (RX1)	0xf (data tag)	Current RX1 BCID by PSB (12bit)
word [14]	Current RX1 hit data (32bit x 4)			
word [15]				
word [16]				
word [17]				
word [18]	Tga 0x1 (Current)	0x0 (RX0)	0xf (data tag)	Current RX0 BCID by PSB (12bit)
word [19]	Current RX0 hit data (32bit x 4)			
word [20]				
word [21]				
word [22]				
word [23]	Tga 0x2 (Next)	0x1 (RX1)	0xf (data tag)	Next RX1 BCID by PSB (12bit)
word [24]	Next RX1 hit data (32bit x 4)			
word [25]				
word [26]				
word [27]				
word [28]	Tga 0x2 (Next)	0x0 (RX0)	0xf (data tag)	Next RX0 BCID by PSB (12bit)
word [29]	Next RX0 hit data (32bit x 4)			
word [30]				
word [31]				
word [32]				
word [33]	Number of SL data = 0x1e		Footer (0xe0d0)	

図 5.16 JATHub からのヒットデータ読み出しフォーマット。3 BC 分のヒットデータに加えて、デバッグ用に JATHub で付与した BCID や PS board で付与した BCID などが読み出される。

は、処理中のイベントのデータ出力を完遂させたのち (Previous の出力中に almost full 信号が来た場合、Current、Next のデータまで出力を終えた後)、データの出力を一時的に停止する。Ubuntu からのデータ読み出しが進み、Derandomizer の Occupancy が容量閾値を下回ると、再びデータの出力を再開する。これにより、任意のタイミングで読み出しを開始/終了してもイベントのパケットを崩すことなくデータを読み出すことができる。

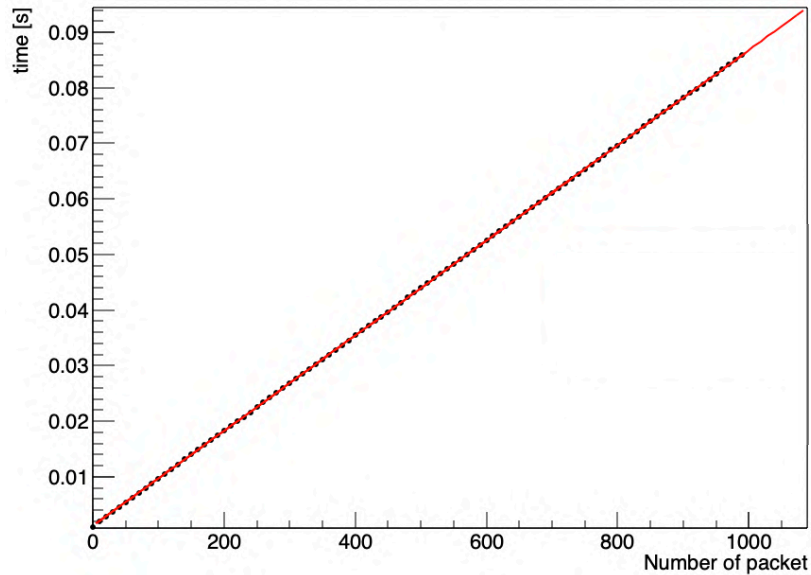


図 5.17 PS からのヒットデータ読み出し速度の測定。横軸に読み出したパケット数、縦軸に経過時間 (s) をとり、得られた測定結果を線形フィットしている。1 イベント分のパケットを読み出すのにかかる時間は $85 \mu\text{s}$ 程度である。

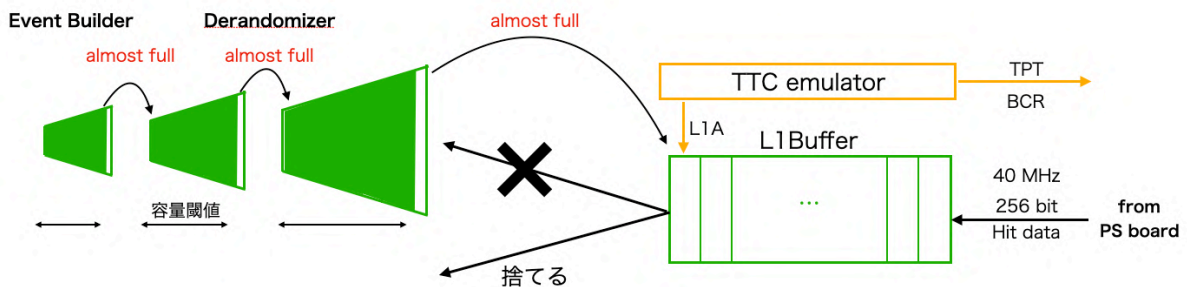


図 5.18 読み出し回路におけるバックプレッシャー機能。Derandomizer の Occupancy があらかじめ設定した容量閾値を超えると、L1 Buffer に almost full 信号が送られる。almost full 信号を受け取った Derandomizer は処理中のイベントの出力が終わった後、Derandomizer へのデータ送信を停止する。



図 5.19 QAQC 試験のための東京大学でのセットアップ。QAQC 用 JATHub は VME クレートに設置し、バックプレーンの J3 コネクタを通じて給電する。PS board および ASD には 3.3 V デジタル、3.0 V、-3.0 V アナログ電源をそれぞれ用意し、デスクトップで給電する。QAQC 用 JATHub と PS board は 3 本の光ファイバーと 2 本の Cat-6 ケーブルで接続する。1 台の PS board には 8 台の ASD を接続する。

5.3 PS board 品質保証試験のデモンストレーション

5.3.1 試験環境

東京大学に設置したテストベンチを利用して、開発した QAQC 用 JATHub の動作検証および QAQC 試験のデモンストレーションを行なった。セットアップを図 5.19 に示す。QAQC 用の JATHub は VME クレートに設置し、バックプレーンの J3 コネクタを通じて給電する。PS board には 3.3 V デジタル電源、3.0 V、-3.0 V アナログ電源を用意しデスクトップで給電する。QAQC 用 JATHub と PS board は 3 本の光ファイバーと 2 本の Cat-6 ケーブルで接続する。1 台の PS board には 16 台の ASD をシグナルケーブルで接続する。

5.3.2 試験用パラメーターの決定とデモンストレーションの結果

QAQC 試験では量産個体に対する試験に先んじて、PP ASIC の遅延、DAC の閾値電圧、L1A depth などセットアップに依存する制御パラメーターを設定しておく必要がある。以下に各パラメーターを決定するために行った、事前試験の内容とその結果を示す。

PP ASIC 遅延パラメーター

PP ASIC の遅延パラメーターは、使用するシグナルケーブルの長さに依存するパラメーターである。ASD からのテストパルス信号の立ち上がり、PP ASIC 陽子バンチ識別回路における LHC バンチ交差クロックの立ち上がりと極めて近い場合、イベントごとに付与される BCID が 1 つに定まらない可能性がある。これを防ぐため、PP ASIC の可変遅延回路の遅延パラメーターを変更し、両者の立ち上がりが十分離れるよう調整する。このパラメーターの決定のため、Delay スキャンを行なった。Delay スキャンとは、PP ASIC の遅延パラメーターを 1 step ずつ変更しながらデータを取得し、付与された BCID の変化をみるものである。

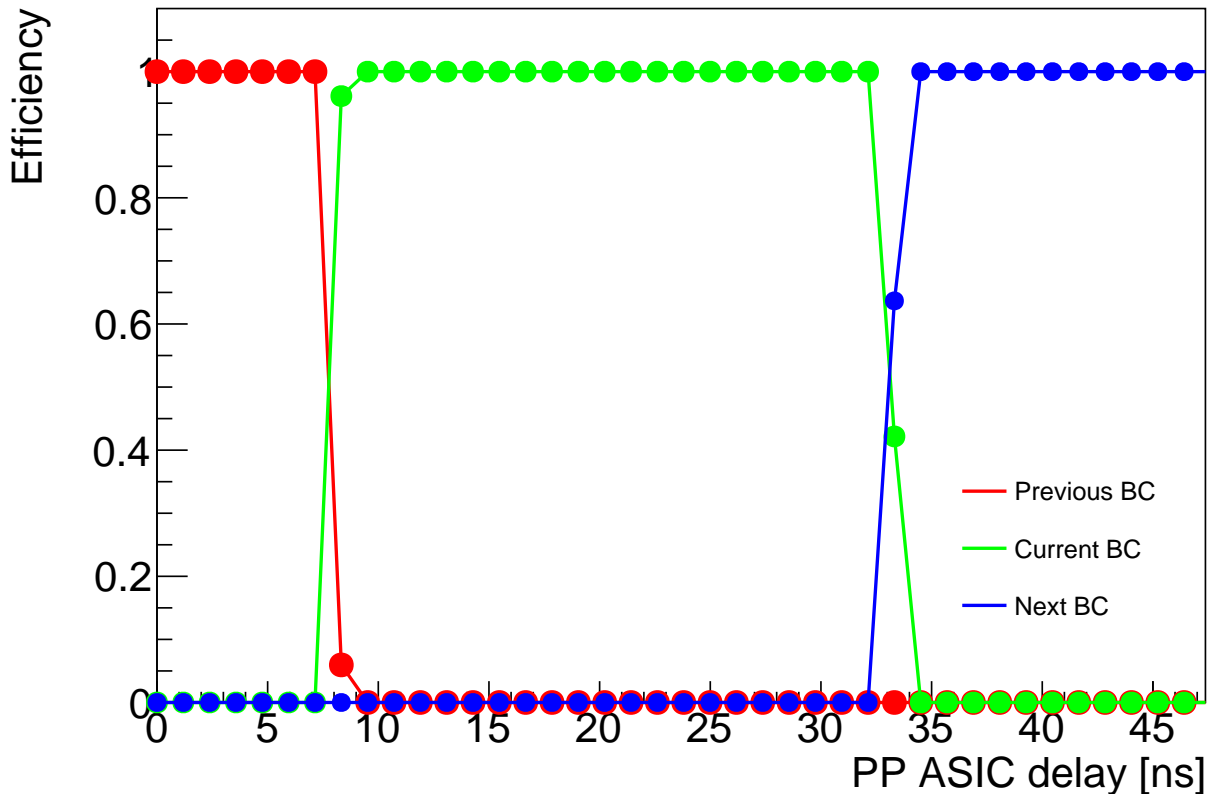


図 5.20 テストベンチで測定した Delay Curve。1 つの ASD における結果を示す。陽子バンチ識別回路の有効ゲート幅を 25 ns、可変遅延回路の刻み幅を 1.19 ns に設定した。遅延が 0 ns の時に付与される BCID が Previous BC になるよう L1 Buffer Depth を調整した。遅延を増やすことで、付与される BCID が期待通り Previous BC、Current BC、Next BC と遷移することがわかる。

1 つの ASD に対する Delay Curve を図 5.20 に示す。横軸に設定した遅延の大きさ、縦軸に Efficiency を表す。この測定では、陽子バンチ識別回路の有効ゲート幅を 25 ns、可変遅延回路の刻み幅を 1.19 ns に設定している。遅延が 0 ns の時に付与される BCID が Previous BC になるよう JATHub 内の L1 Buffer Depth を調整した。遅延パラメータを変更していくと、ヒット信号に付与される BCID が 1 つずつずれ、付与される Bunch tag が Previous BC、Current BC、Next BC と遷移していく。この試験の結果、テストパルスの立ち上がりで PP ASIC の陽子バンチ識別回路における LHC バンチ交差クロックの立ち上がりが十分に離れるよう、遅延パラメータを 20 ns と決定した。

ノイズスキャン

DAC の閾値電圧は、使用する ASD のノイズの大きさに依存して変えるべきパラメータである。ASD には $\mathcal{O}(10\text{mV})$ のノイズが乗っていることが知られており、それよりも閾値電圧を低く設定してしまうと、ノイズをヒット信号として処理してしまう。閾値電圧はノイズの値よりも高く、かつテストパルスの波高より十分低く設定する。

DAC の閾値電圧決定のためノイズスキャンを行なった。1 台の ASD に対する結果を図 5.21 に示す。ASD に印加する閾値電圧を 10 mV ずつ変えていき、ノイズレートを測定した。閾値電圧が 10 mV 程度になるとノイズによるヒットが見られるようになり、60 mV 以上の閾値電圧をかけるとノイズは消えた。ノイズレートが最大となる点が 0 mV からずれているのは、コンパレータに -30 mV 程度のオフセットが乗っているためである。この結果から本セットアップでの DAC 閾値を \pm それぞれ +90 mV、-40 mV と設定した。

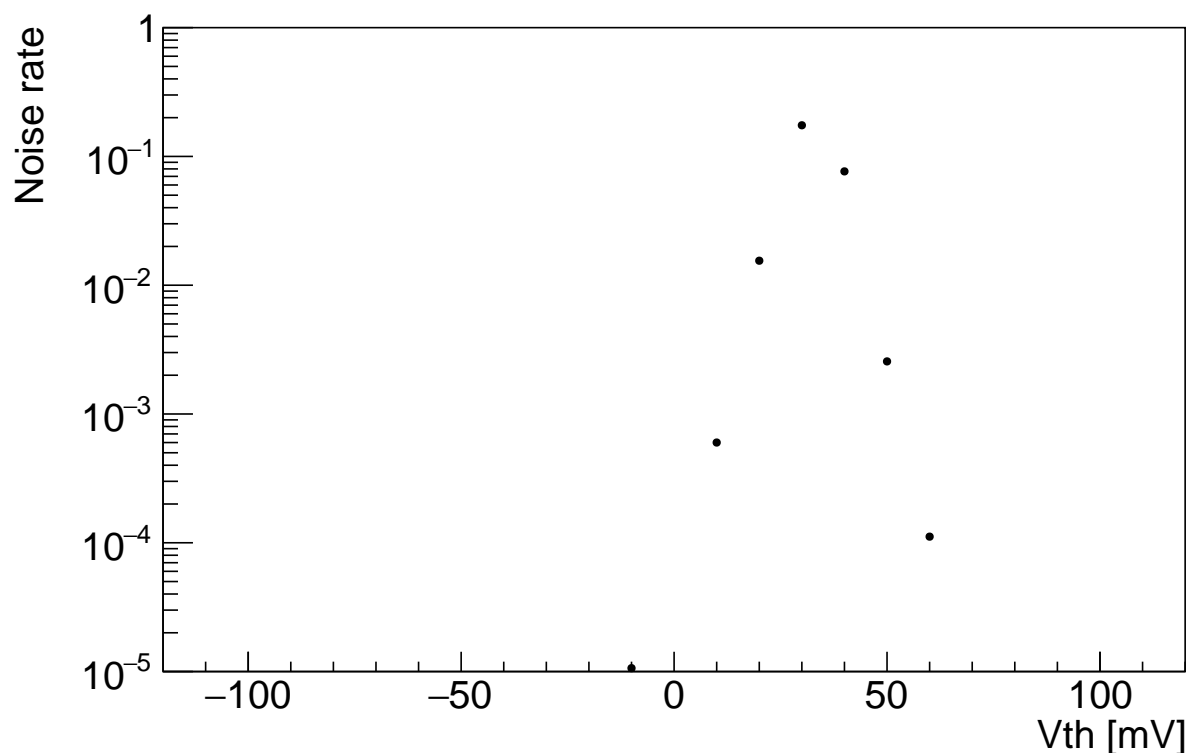


図 5.21 ノイズスキャンの結果。ASD に印加する閾値電圧を変えながら、10,000 回のテストパルスに対するノイズレートを測定した。ここでは 16 台ある ASD のうちの 1 台の結果を示している。閾値電圧が -10 mV から 60 mV の間でノイズによるヒットが見られた。

表 5.2 各試験にかかる所要時間

試験項目	所要時間
SVF プレイヤーによる QSPI ファームウェア書き込み	40 min
QSPI パラメーター書き込みおよび読み出し	2 min
リカバリー試験	1 s
電圧値のモニタリング (ADC、xADC)	2 s
Clock 位相測定	30 s
ASD テストパルス試験	10 s

デモンストレーションの結果

これらのパラメーターをもとに ASD テストパルス試験を行なった。結果を図 5.22 示す。パラメーターを固定した状態で、10,000 回テストパルスを発行し、Previous BC、Current BC、Next BC それぞれの Efficiency を測定した。結果は、期待通り、すべてのチャンネルで Current BC の Efficiency のみが 1 となっており、PS board 上の各種パラメーターの設定が正常に完了していること、また固定レイテンシーでの DAQ が実現できていることを確認した。

JTAG/Recovery/Clock monitor 試験についても試験を行い、期待通りの結果が得られた。これにより、QAQC 用 JATHub システムの実装が精度良く達成されていることを確認した。表 5.2 に各試験の所要時間を示す。

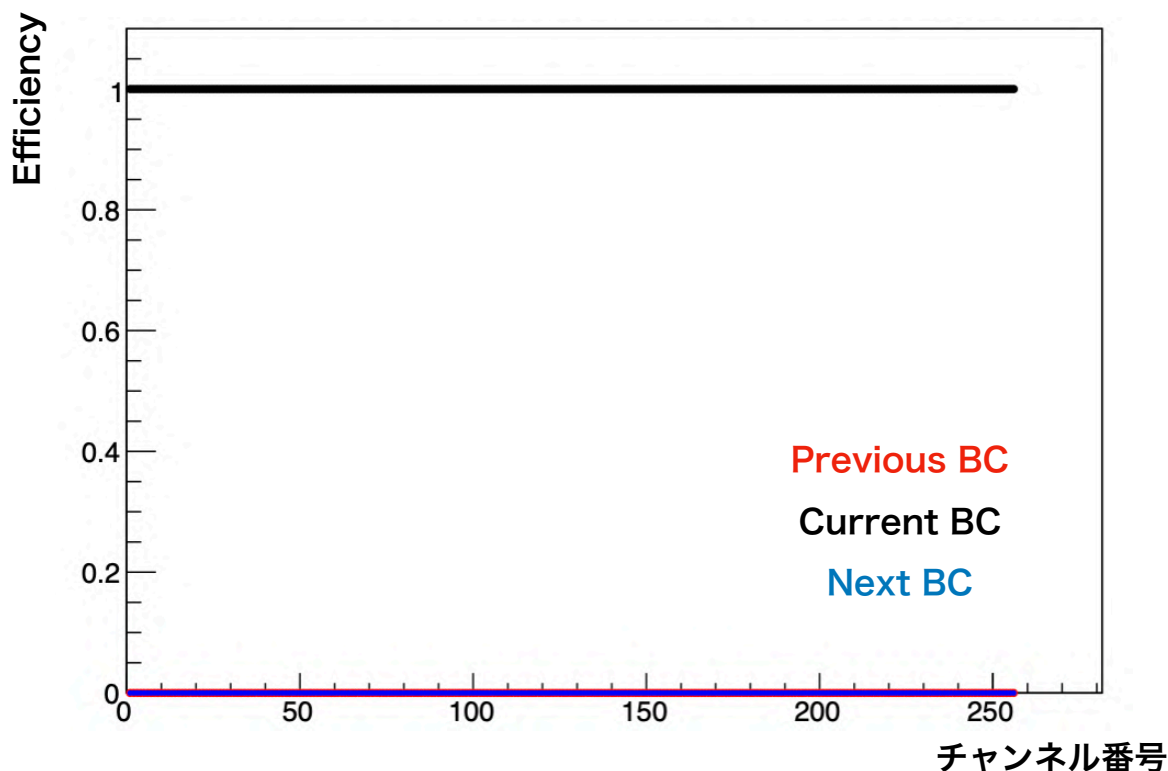


図 5.22 ASD テストパルス試験の結果。10,000 回のテストパルスに対する各チャンネルの Efficiency をプロット。PS board が担う全 256 チャンネルで Current BC の Efficiency が 1 であり、Previous BC、Next BC の Efficiency は 0 である。このことから ASD テストパルスの発行から、読み出しまで固定レイテンシーでの動作が実現されていることがわかる。

5.3.3 試験並列化のためのシステムアップグレード

QAQC 試験のデモンストレーションにより、1 枚の PS board を試験するのに、合計で 40 分以上の時間が必要であることがわかった。1400 枚の PS board に対する試験を効率的に完了させるためには、試験の高速化が求められる。単純には QAQC 用 JATHub1 台と PS board1 台のセットアップを複数用意し、並列に試験を行うことが考えられるが、イーサネット通信のための RJ コネクターと光通信のための SFP+ コネクターを両方有している JATHub は、試作段階で作成された 1 台 (試作 1 号機) しか存在しない。並列化のためには、工夫が必要となる。

そこで、本研究では VME 通信を利用した並列化システムを考案した。システムのセットアップを図 5.23 に示す。このシステムのコンセプトは、イーサネット通信ができる試作 1 号機を QAQC 試験におけるマスター (QAQC master) として利用し、QAQC master から 20 台の QAQC 用 JATHub (QAQC slave) を操作することで、20 台の PS board を並列に試験することである。試験の際には QAQC master のみを操作して、QAQC slave に対して試験開始の命令や、試験状況の監視を行う。QAQC master は TAM ボードを^{*4}中継して、QAQC slave と通信する。QAQC master と TAM は光ファイバーを介したシリアル通信を行い、TAM と QAQC slave は VME 通信を行う。

QAQC slave にはネットワークが通じていないため、slave の操作は master からのレジスタ操作で行う。QAQC slave の Ubuntu には起動と同時に試験用アプリケーションが走るよう設定し、master からの合図を起点に 1 台の

*4 Mini-Rack に導入される TGC エレクトロニクスの一つ。異なる 1/24 セクター間の陽子バンチ交差クロックの位相合わせに用いられる。

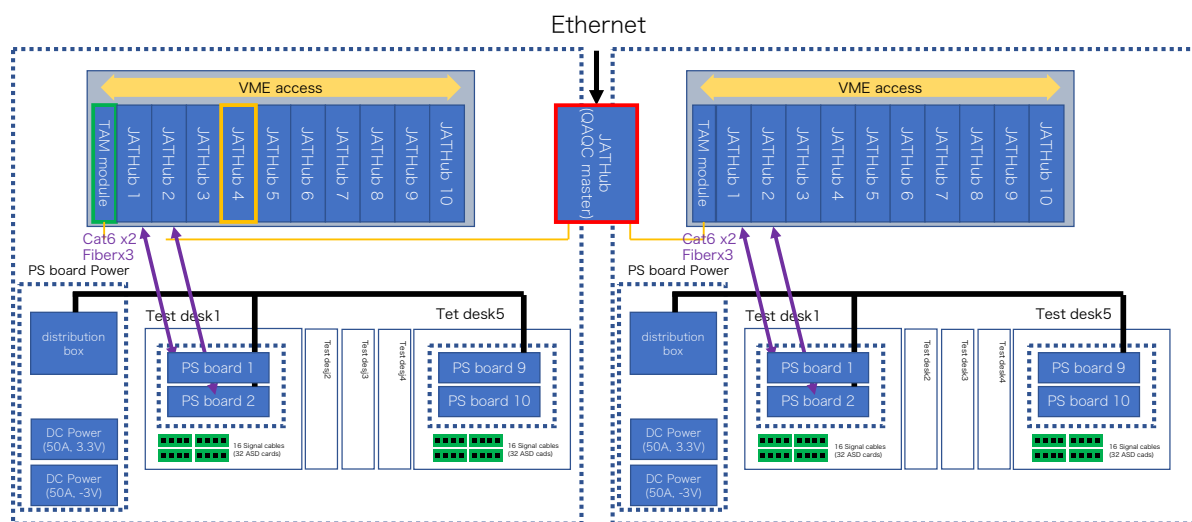


図 5.23 QAQC 試験並列化システムの概要。JATHub1 台 (QAQC slave) と PS board1 台を用いた試験システムを 20 個並列に設置する。20 台の JATHub の動作を制御するため、1 台の検査マスター JATHub (QAQC master) を用意する。QAQC master と QAQC slave の通信は TAM を介して行う。QAQC master と TAM は光ファイバーで接続し、高速光通信を行う。TAM と QAQC slave は VME バックプレーンで接続され、VME 通信を行う。

PS board に対する試験を開始する。各試験の結果は 2 bit の status bit で表され、QAQC master はそれを読み出すことで各 slave における試験の進行状況を把握する。

並列化システムについても、東京大学でテストベンチを構築した。図 5.24 にセットアップを示す。QAQC master、TAM、QAQC 用 slave1 台、PS board1 台、ASD 16 台を接続して、並列化システムの動作検証を行った。QAQC master の試験開始から、試験の終了まで期待通り動作することを確認した。

5.4 コンパクト DAQ システムとしての応用例

最後に、本研究で開発した QAQC 用 JATHub のその他の利用例について紹介する。本システムは、Linux を搭載しており拡張性に富んでいることに加え、デスクトップでも給電可能なコンパクトなシステムである。その柔軟性と利便性から汎用的な DAQ システムとして、TGC 検出器の性能評価やエレクトロニクスの試験などに幅広く利用されている。

実際の使用例を 2 つ紹介する。1 つ目は TGC EIL4 チェンバーの性能評価である。セットアップを図 5.25 に示す。この試験は、高輝度 LHC-ATLAS 実験に向けて新しく設置される TGC EI チェンバーの検査を目的としたもので、チェンバーに取り付けた ASD からの信号を PS board で処理し、QAQC 用 JATHub から読み出している。ASD テストパルスを用いたノイズレートの測定や、宇宙線ミューオンによるチャンネルごとの efficiency 測定が行われた。

2 つ目は PS board の放射線耐性試験である (図 5.26)。この試験は、ATLAS 実験室の実際の放射線環境における、PS board の SEU 発生頻度を測定するものである。UX15 に設置した PS board から送られるモニターデータの読み出しに QAQC 用 JATHub は使われている。

その他にも TAM モジュールの QAQC 試験や PS board のメンテナンス目的でも利用され、本システムは高輝度

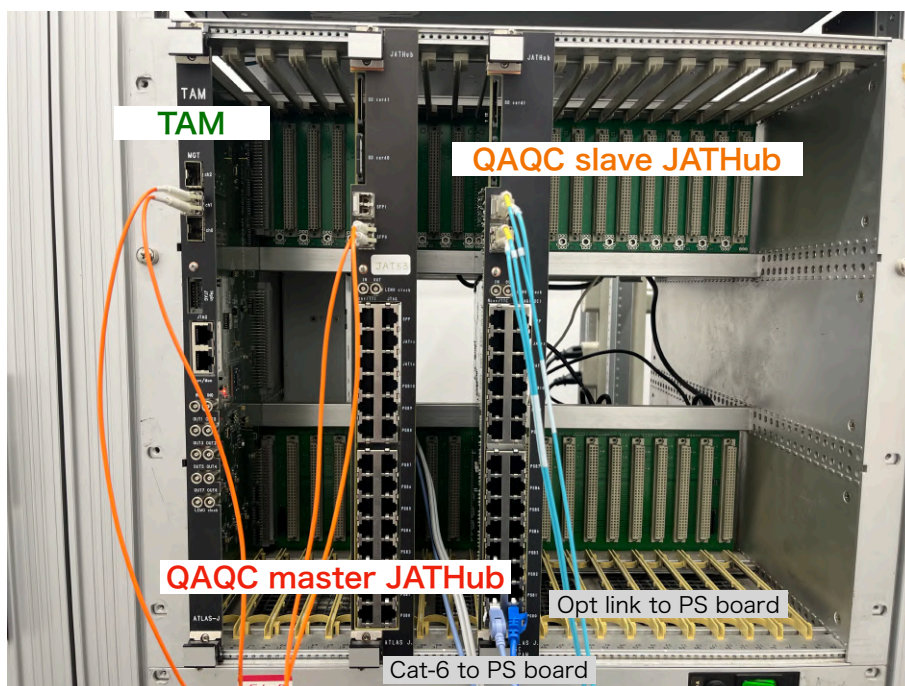


図 5.24 並列化システムのセットアップ。QAQC master、QAQC slave、TAM が VME クレートに設置されている。QAQC master と TAM は光ファイバーで接続し、QAQC slave と TAM は VME バックプレーンで接続されている。QAQC slave と PS board は光ファイバーと Cat-6 ケーブルで接続されている。

LHC-ATLAS 実験に向けた TGC エレクトロニクスの安定運用を支える、重要なインフラとして活用される。

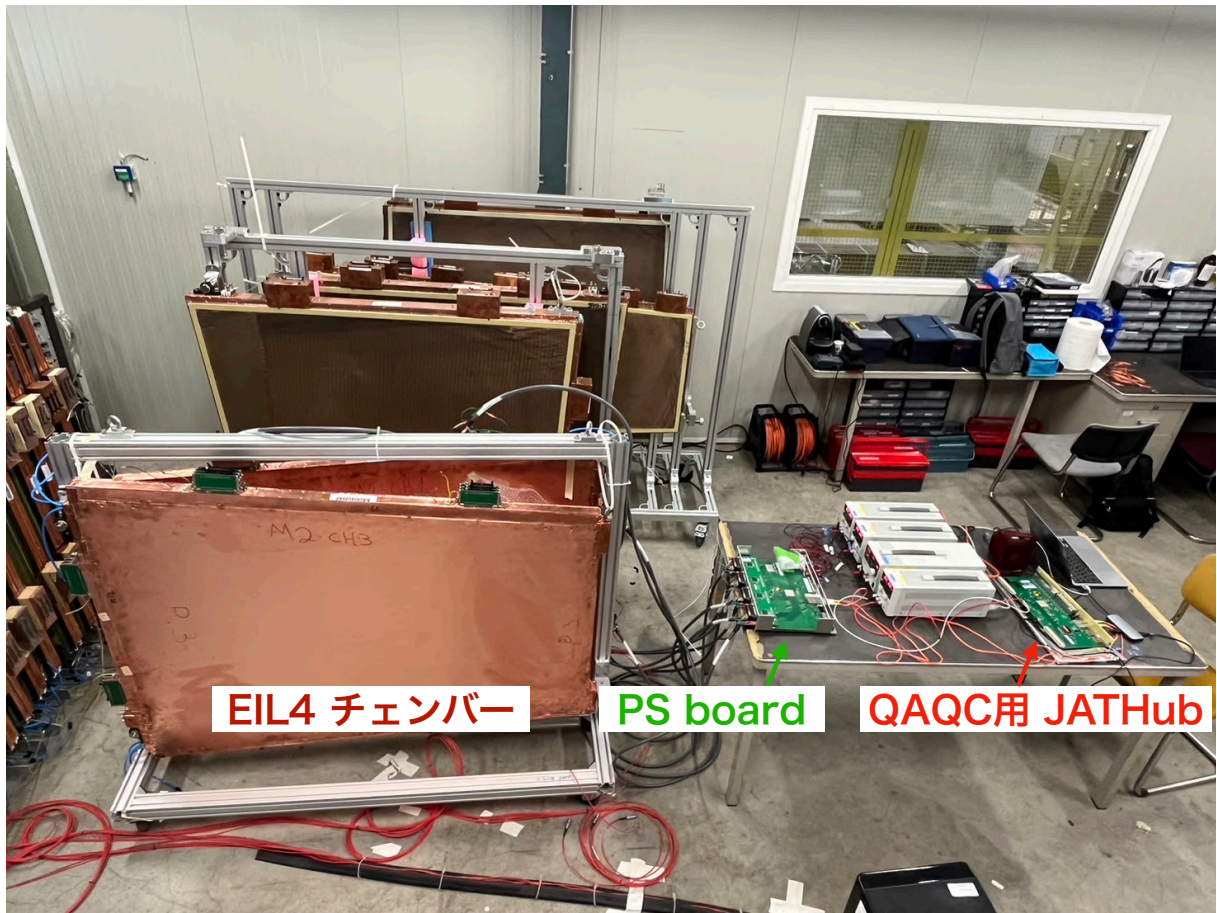


図 5.25 QAQC 用 JATHub の使用例：TGC EIL4 チェンバーの性能評価試験 [26]。高輝度 LHC-ATLAS 実験で新しく導入される、Triplet の EIL4 チェンバーの性能評価を目的として CERN で行われた。ASD を EIL4 チェンバーに接続し、PS board、QAQC 用 JATHub を用いて信号処理を行うことで、EIL4 チェンバーのチャンネルごとの応答を調べている。ノイズスキャンや宇宙線ミュオンを用いた Efficiency 測定が行われた。

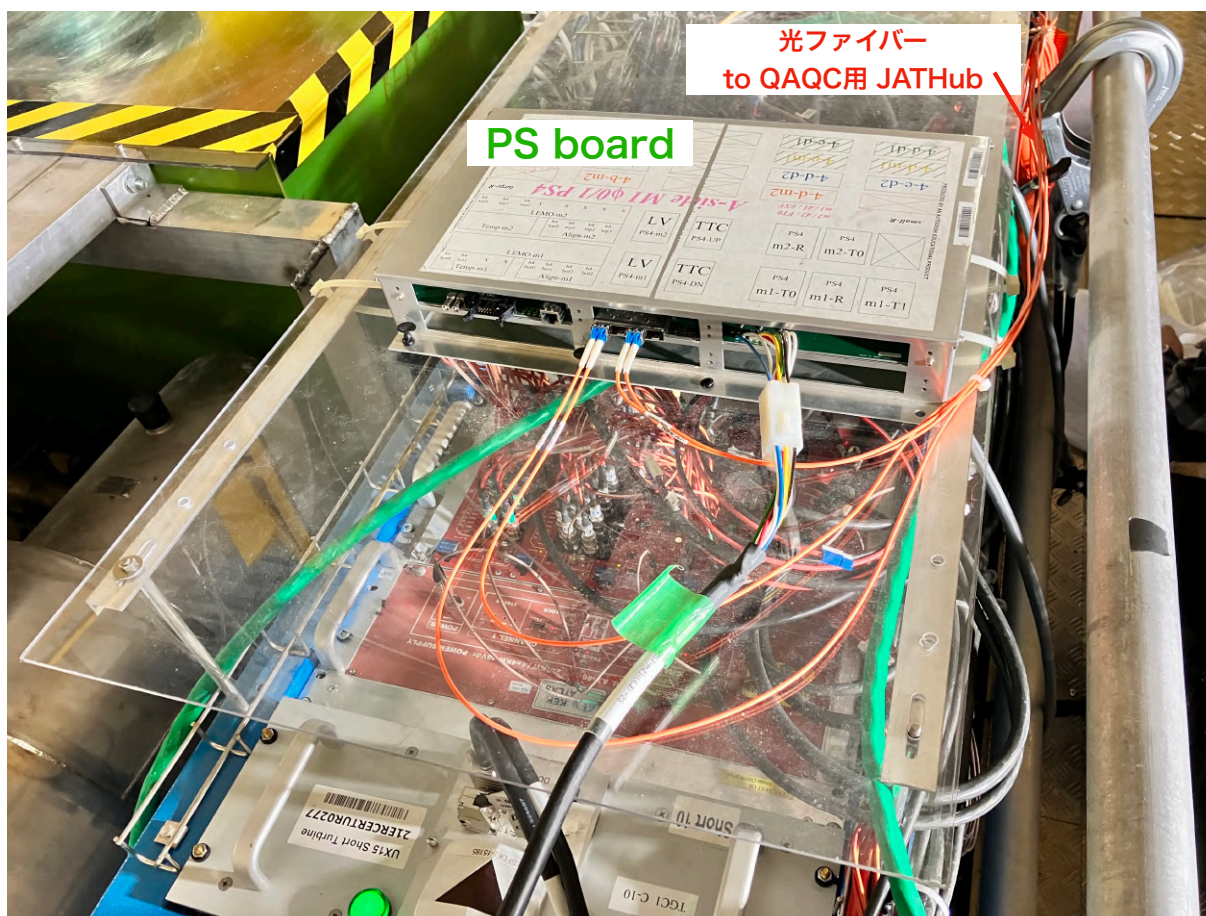


図 5.26 QAQC 用 JATHub の使用例：PS board SEU モニター [27]。ATLAS 実験室内に設置した PS board における、SEU 発生頻度を測定することを目的に CERN で行われた。PS board から送信されるモニター用データの読み出しに QAQC 用 JATHub が使われ、SEU 発生頻度の測定が行われた。

第 6 章

結論と今後の展望

2029 年に開始される高輝度 LHC-ATLAS 実験では、初段トリガーレートは現行の 100 kHz から 1 MHz に、初段トリガーレイテンシーは $2.5 \mu\text{s}$ から $10 \mu\text{s}$ に拡張される。これに伴い、TGC 検出器システムも一部を除くすべてのエレクトロニクスが刷新され、PS board からすべてのヒット信号をヒットの有無にかかわらず SL へと送信するようになる。これにより Run 3 で複数のボードで分割されていたトリガーロジックは、SL の FPGA 上に集約され、TGC BW 全 7 層のヒット信号を用いた柔軟かつ包括的なトリガーロジックが実現可能となる。

本研究では SL のトリガー論理回路の統合を完了させ、その性能評価のためのシングルボード試験システムを開発した。まず、Wire Strip Coincidence、Inner Coincidence、Track Selector の各ロジックの統合を進めた。クロックの分配やトリガーマジュール間の配線に加え、LUT のコンフィギュレーションのための機能実装も行い、トリガー回路を実機上で動かすための準備を整えた。また、統合の際に生じたタイミング違反を解決するため、パイプラインレジスタの実装方法の最適化や Track Selector のロジック最適化を行った。その結果、タイミング違反なくすべてのトリガーマジュールを統合することに成功した。

次に、SoC を活用したシングルボード試験システムを開発した。これにより、任意のデータセットに対するトリガー応答を、ハードウェア上で動作するトリガー回路を直接用いて検証することが可能となった。また、この試験システムと Bitwise シミュレーターを合わせることで、トリガー回路に対する詳細な調査およびデバッグが可能となり、トリガー開発における盤石な開発基盤を確立した。本システムは開発段階だけでなく、本番運用時のファームウェアのメンテナンスやアップグレードに際した性能評価にも活用される予定で、精度の高いミューオントリガーの運転に必要なシステムとして機能する。

今後は、本試験で明らかになった無限運動量飛跡に対する 3% 程度の Inefficiency や、シングルミューオン MC に対する局所的な不具合に対する調査および修正を行い、TGC BW Coincidence までのトリガーロジックの精度を高めていく。また、トリガーロジックの拡張の観点では、Inner Coincidence の開発を進めるとともに、本番運用に向けて堅牢性を高めていく。例えば、磁場内部検出器に不具合が生じた場合でも、トリガー効率を落とすことなく安定して運転できるよう、柔軟なトリガー運用オプションを準備する。

また本研究では、多くの要素技術を共有する形で PS board QAQC 試験に向けた、コンパクト DAQ システムを開発した。まず、PS board 上に搭載された素子とインターフェイスを網羅的に試験できるセットアップとして、JATHub を用いるのが合理的であることを見出し、試験セットアップを考案した。次に、ASD テストパルス試験と JTAG/Recovery/Monitor 試験をベンチマークに、JATHub の PS 領域と PL 領域に必要な機能を実装し、テストベンチを用いたデモンストレーションを完了させた。最後に QAQC 試験の高速化を目的とした、システム並列化を実現した。このシステムは、2024 年から開始される 1400 枚以上に及ぶ量産試験で実際に使われる。さらに、その高い拡張性とコンパクトさから、TGC 検出器の性能評価やエレクトロニクスの試験に幅広く応用されており、高輝度 LHC-ATLAS 実験における TGC システムの安定運用を支える重要なインフラとなる。今後は 2024 年 6 月から始まる PS board の量産に向けて、システムをさらに洗練させていく。

本研究で確立した、高速 FPGA や SoC を活用した次世代型の検証モデルや、汎用的な読み出しシステムは、ATLAS 実験などの高エネルギー物理実験のみに留まらず、様々なエレクトロニクスシステムに幅開く応用されることが期待される。

謝辞

本研究を遂行するにあたって、多くの方々にお世話になりました。まずは指導教員である奥村恭幸准教授には大変お世話になりました。研究の構想から、詳細な技術各論に至るまでたくさんの有用なアドバイスをいただき、本研究をスムーズに進めることができました。また、CERN 出張やシンポジウムなど多くの学びの機会を提供していただき、様々な面での成長をサポートしていただきました。石野雅也教授にも大変お世話になりました。研究に必要な環境の構築や、研究内容に対する的確な助言をいただき、研究を深めることができました。博士課程進学後もよろしく申し上げます。

日々の研究に関して多くの助言をくださった、研究室ミーティングに参加してくださっているスタッフの皆様にも心から感謝申し上げます。斎藤智之助教、増淵達也助教には現行システムの観点から様々な助言を頂き大変参考になりました。他の ICEPP の皆様にもお世話になりました。田中純一教授、澤田龍准教授にはセミナーなどで質問やコメントをいただき、研究をさらに深めることができました。ICEPP 夏の学校やワークショップでは、他にも多くの ICEPP のスタッフや先輩方にお世話になり、充実した研究生生活を送ることができました。また秘書の皆様にも、出張手続きをはじめとした様々な面で支えていただきました、感謝申し上げます。

共同で高輝度 LHC-ATLAS 実験に向けたアップグレード研究を行なっている皆様にも、大変お世話になりました。KEK の佐々木修氏には、ASD や PS board の取り扱いについて助言をいただき、大変助かりました。堀井泰之氏には PS board や SL の開発についてたくさん相談させていただきました。また、名古屋大学への出張の際にもサポートしていただきお世話になりました。戸本誠氏や前田順平氏には、TGC 会合やシンポジウムなどで研究に対するコメントを頂き、研究を進めるにあたり大変参考になりました。青木雅人氏には、TGC 検出器のエキスパートとして CERN での活動を牽引してくださり、専門的な知識を多く学ぶことができました。河本さん、辻川さん、三森さん、大隅さん、鍋山さん、橋爪さん、中川くんには SL や PS board の開発に際して、様々なアドバイスをいただき感謝しています。皆様との議論や試験を通して、共同研究としてグループで開発を進めることの面白さを実感することができました。今後ともよろしく申し上げます。

研究室の先輩方にも大変お世話になりました。杉崎さん、田中さん、青木さん、林さん、三島さん、山下さんにはエレクトロニクスや TGC システムに関して多くのことを教えていただきました。研究で行き詰まった際にはいつも相談に乗っていただき、心強い支えとなりました。同期の長坂とは、日頃から多くの議論を共にさせてもらい、良い刺激を受けさせてもらっています。これからも互いに切磋琢磨していければと思います。名古屋大の橋本くん、和田さんとは長い間研究を共に進めることができ、活発な意見交換を通じて研究を深めることができました。ありがとうございました。以上に述べた他にも、多くの方々にお世話になりました。本研究の遂行に協力していただいたこと、深く感謝申し上げます。

そして最後に、充実した研究生生活を支えてくれた家族に感謝します。

付録

A これまでに行われたトリガーロジックの性能評価研究

これまで、Inner Coincidence までのロジックはソフトウェアシミュレーターによる性能検証が完了しており、TGC BW Coincidence までのロジックは Vivado シミュレーターによる動作検証も完了している。以下に先行研究で行われた、TGC BW Coincidence までの各モジュールの性能検証の結果を示す。

ソフトウェアシミュレーターによる Wire Strip Coincidence のトリガー効率測定

図 A.1 にソフトウェアシミュレーターで測定された、Single Muon モンテカルロデータ (Monte-Carlo, MC) に対する、TGC BW コインシデンスのトリガー効率を示す。この試験では、MC サンプルに含まれる TGC ヒットチャンネルの情報をステーションコインシデンスのインプットとしている。Efficiency は以下のように定義され p_T 閾値ごとにビン分けしている。

$$\text{Efficiency} = \frac{\text{TGC BW coincidence で } p_T \text{ 閾値 20, 15, 10, 5 GeV と判断されたミューオンの数}}{\text{オフラインで再構成されたミューオンの数}} \quad (\text{A.1})$$

トリガー効率測定の結果、 p_T 20 GeV 以上のプラトー領域での Efficiency は 94 % 程度であることが確認された。また、各 p_T 閾値に対して Efficiency の立ち上がりが鋭く、効率的に p_T 閾値以下のイベントを落とすことができていることがわかる。

Vivado シミュレーターによる Wire Segment Reconstruction のトリガー効率測定

Vivado シミュレーターで測定された、Single Muon MC に対する Wire Segment Reconstruction のトリガー効率を図 A.2 に示す。MC サンプルに含まれる TGC チェンバーのヒットチャンネル情報を、Wire Station Coincidence のインプットに適したフォーマットに成形した上で Vivado シミュレーションに入力している。Efficiency は以下のように定義され、Truth Muon の η 、 ϕ ごとにビン分けされている。

$$\text{Efficiency} = \frac{\text{Wire Segment Reconstruction で } \Delta\theta \text{ を再構成できたイベント数}}{\text{Truth ミューオンの数}} \quad (\text{A.2})$$

TGC チェンバーの構造やエンドキャップマグネットとの干渉によると考えられる、数 % の Inefficiency が局所的に見られるものの、Efficiency は η 、 ϕ 全領域で 95 % 程度に達している。

Vivado シミュレーターによる Strip Segment Reconstruction の動作検証

論理回路実装された Strip Segment Reconstruction と Wire strip Coincidence の動作検証は、ソフトウェアシミュレーター出力との一致を確かめることで行われた。Strip Segment Reconstruction ではソフトウェアシミュレーターによる Strip Station Coincidence の出力を Vivado シミュレーターの入力とし、出力をソフトウェアシミュレーターと比較している。比較結果を表 A.1 に示す。エンドキャップ領域 (EC)、フォワード領域 (FW) それ

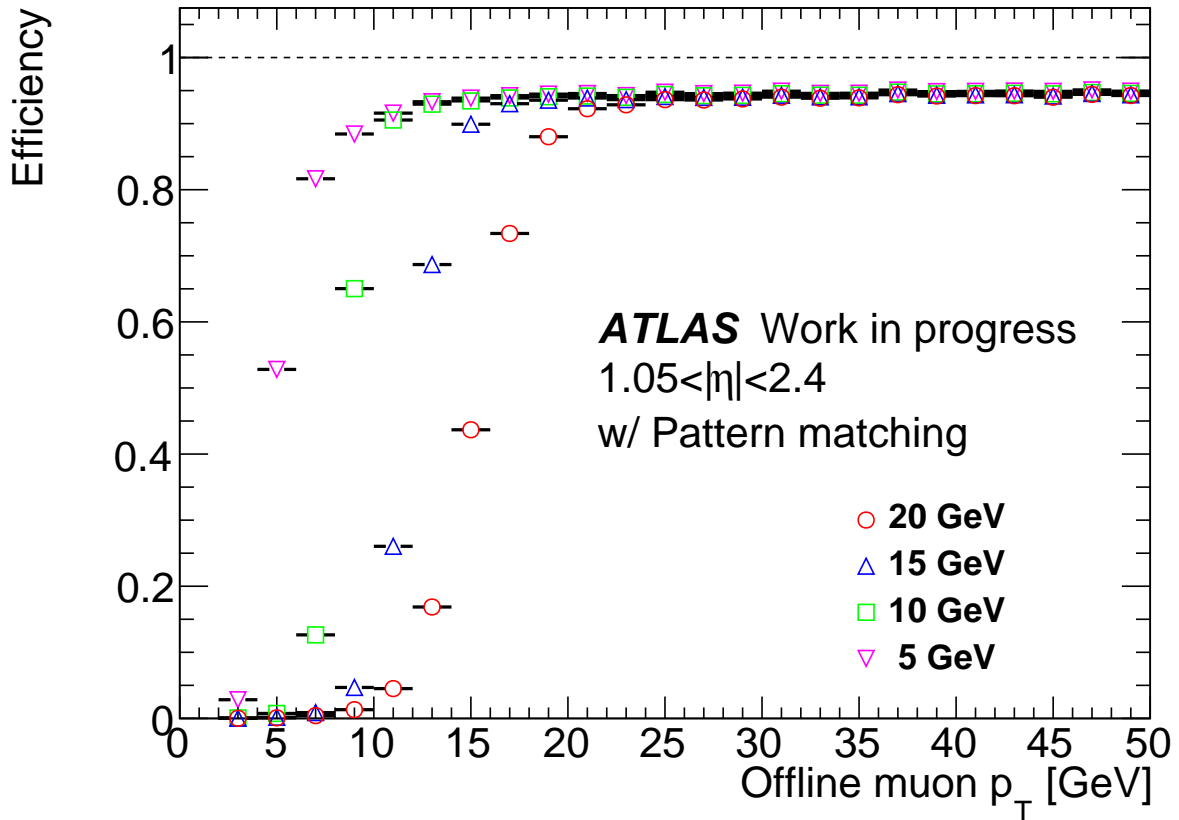
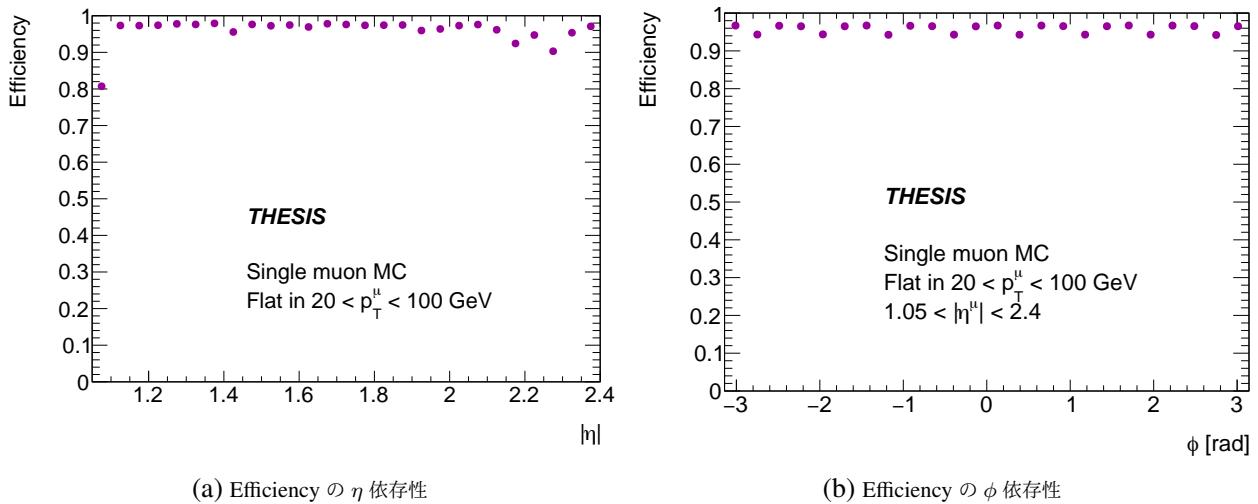


図 A.1 ソフトウェアシミュレーターで測定された、Wire Strip Coincidence のトリガー効率 [21]。20 GeV $< p_T$ のプラトー領域での Efficiency は 94 % 程度である。



(a) Efficiency の η 依存性

(b) Efficiency の ϕ 依存性

図 A.2 Wire Segment Reconstruction の検出効率 [28]。 η 、 ϕ 全領域でトリガー効率は 95 % 程度に達している。

表 A.1 Strip Segment Reconstruction 出力のソフトウェアシミュレーターと Vivado シミュレーター比較結果

	割合	
	EC	FW
飛跡情報が一致したイベント	96.8 %	97.8 %
候補の選び方の違いに由来する差異があったイベント	3.2 %	2.2 %
候補の選び方以外に由来する差異があったイベント	1.8 %	0 %

表 A.2 Wire Strip Coincidence 出力のソフトウェアシミュレーターと Vivado シミュレーター比較結果

	割合	
	EC	FW
飛跡情報が一致したイベント	98.4 %	99.9 %
飛跡情報の異なったイベント	1.6 %	0.07 %
イベントで最大の p_T 出力が異なったイベント	0.09 %	0.02 %

それぞれ出力の 96 % 程度が一致した。数 % の不一致は、Segment Reconstruction をパスした飛跡候補が複数あった場合にどれを優先的に後段に送るか、という候補選択ロジックの差異によるものと理解されている。

Vivado シミュレーターによる Wire Strip Reconstruction の動作検証

Wire Strip Coincidence の動作検証では、ソフトウェアシミュレーターによる Wire Segment Reconstruction と Strip Segment Reconstruction の出力を取り出して、Vivado シミュレーターの入力としている。比較結果を表 A.2 に示す。同様のわずかなロジックの違いによる出力の不一致は存在するが、概ねソフトウェアで設計されたロジックと同等のものを論理回路で実装できていることが確認された。

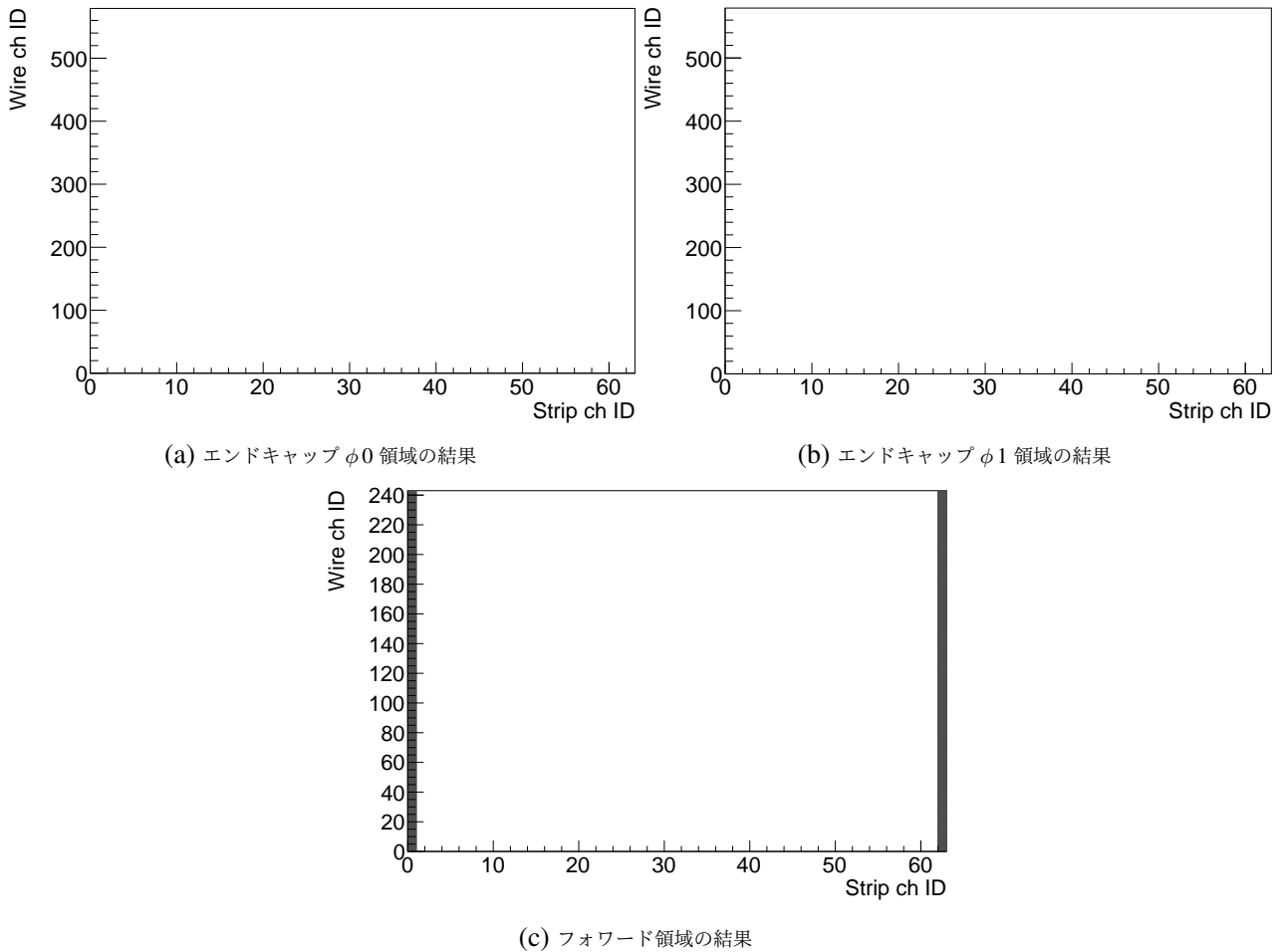


図 B.3 無限運動量飛跡に対する、Strip Segment Reconstruction の応答。横軸に M3 における Strip のスタッガード ID、縦軸に M3 における Wire のスタッガード ID をとる。各 2 次元格子点をピボットとする無限運動量飛跡をシングルボード試験システムに投入し、 $0 \leq \Delta\phi$ の Segment を再構成できた場合にはその格子点を白色、できなかった場合は黒色で塗り潰す。フォワード領域の Strip スタッガード ID 0 番と 62 番に該当する全ての点で、飛跡再構成に失敗している。

B 無限運動量飛跡試験で明らかになった問題点とその修正

無限運動量飛跡の試験で明らかになったトリガー回路の修正点とそのデバッグの過程について述べる。以下にそれぞれのモジュールにおける、修正前の結果と、それに対するデバッグの詳細を示す。

Strip Segment Reconstruction の結果

LUT 修正前の段階における、無限運動量飛跡に対する Strip Segment Reconstruction の応答を図 B.3 に示す。フォワード領域の Strip スタッガード ID 0 番と 62 番に該当する全ての点で、飛跡再構成に失敗した。この結果は、Bitwise シミュレーターでも再現された。Bitwise シミュレーターを用いて詳細な調査を行ったところ、この Inefficiency は LUT に該当するチャンネルの飛跡候補が格納されていないことが原因であると理解された。

この調査の結果をもとに LUT の生成プロセスのデバッグが行われ、この不具合は解消された。

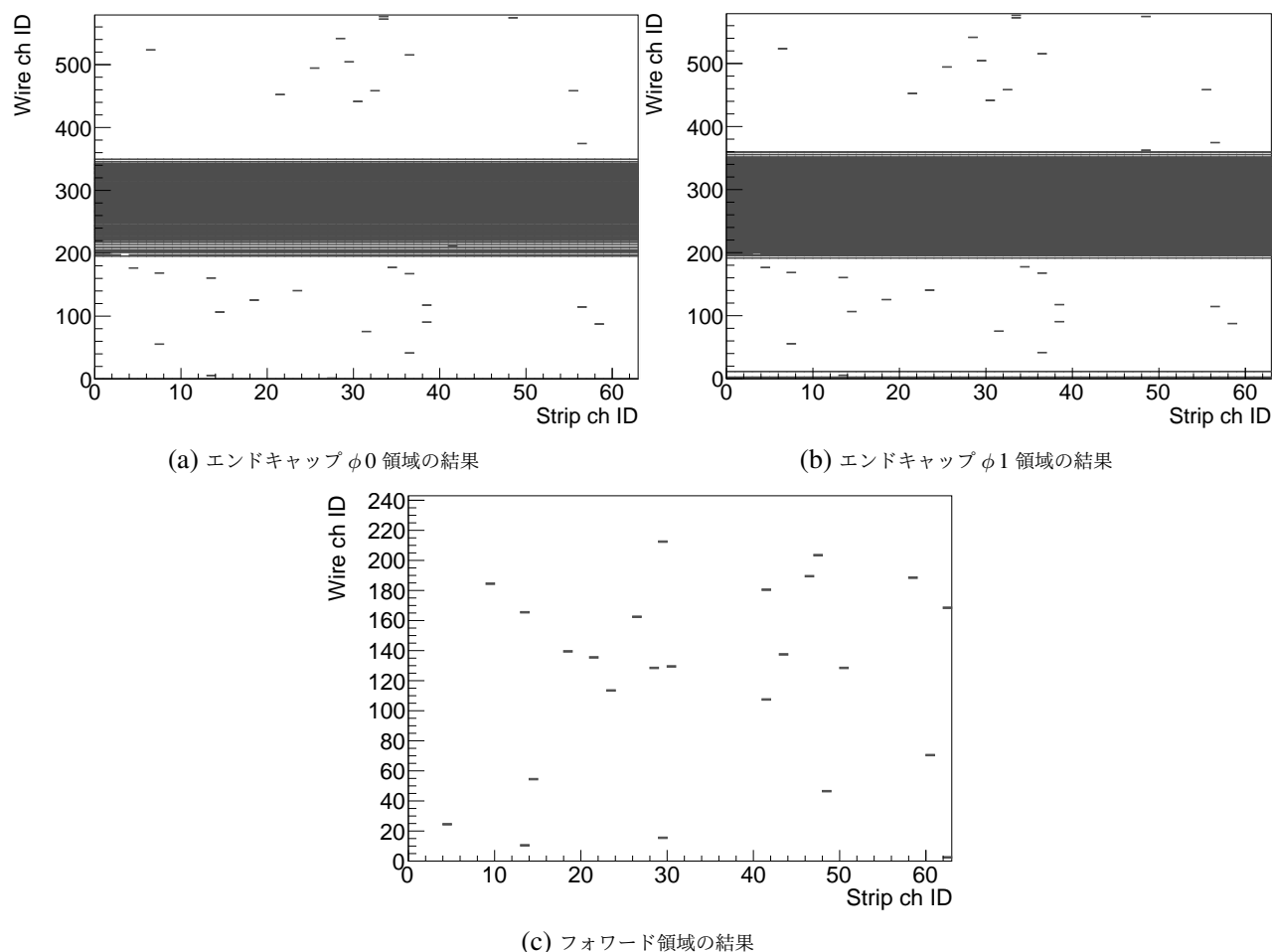


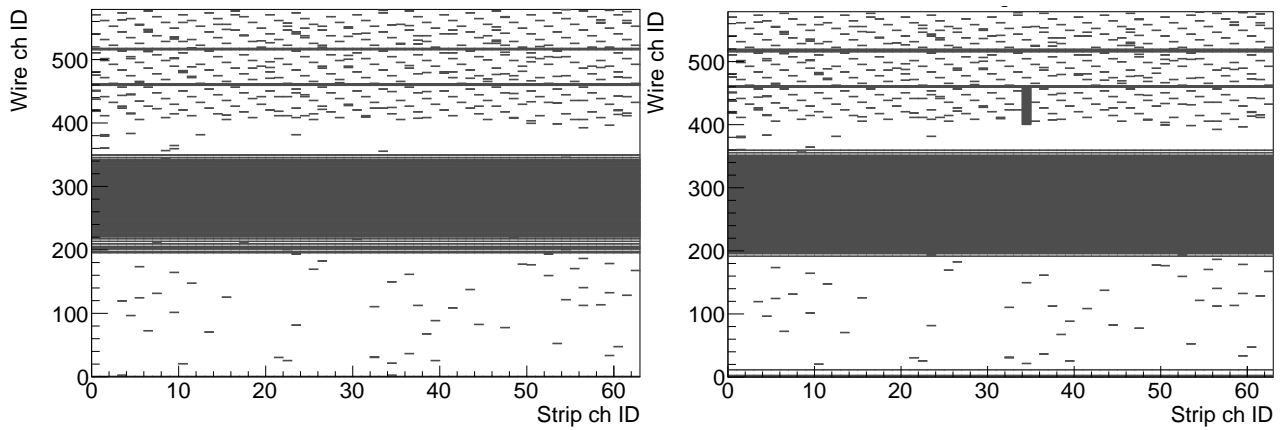
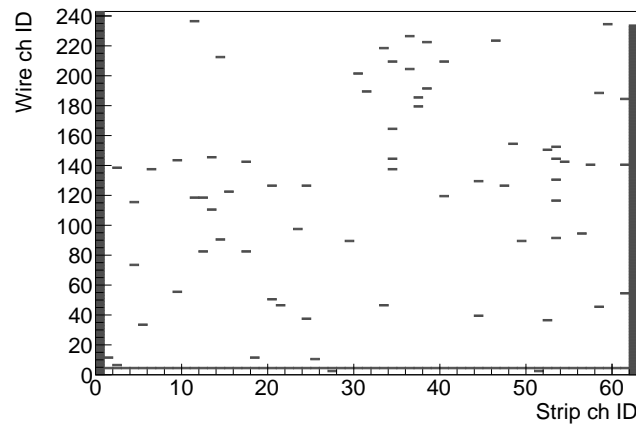
図 B.4 無限運動量飛跡に対する、Wire Segment Reconstruction の応答。横軸に M3 における Strip のスタッガード ID、縦軸に M3 における Wire のスタッガード ID をとる。各 2 次元格子点をピボットとする無限運動量飛跡をシングルボード試験システムに投入し、 $0 \leq \Delta\theta$ の Segment を再構成できた場合にはその格子点を白色、できなかった場合は黒色で塗り潰す。エンドキャップ領域の Wire スタッガード ID 200 番から 350 番あたりで大きな Inefficiency が確認された。

Wire Segment Reconstruction の結果

Wire Segment Reconstruction の結果を図 B.4 に示す。Wire Segment Reconstruction ではエンドキャップ領域の Wire スタッガード ID 200 番から 350 番あたりで大きな Inefficiency が確認された。この不具合は η ID の割り振り方の問題によるものであると理解され、ケーブルングデータベースの修正が行われた。もともと、TGC 検出器はステーションごとの η 位置分解能が等しくなるように設計されており、設置によるずれが生じた後でも通し番号的に η ID を割り振れば同じ η をカバーするチャンネルを一意に定められると考えられていた。しかし、本試験の結果により、通し番号的に η ID を割り振った場合、検出器の中央の領域でステーションごとに η 位置にずれが生じることが明らかになった。

Wire Strip Coincidence の結果

Wire Strip Coincidence の結果を図 B.5 に示す。Wire Strip Coincidence では Strip Segment Reconstruction、Wire Segment Reconstruction それぞれの Inefficiency に加えて、Wire スタッガード ID 400 番以降の領域で特定

(a) エンドキャップ $\phi 0$ 領域の結果(b) エンドキャップ $\phi 1$ 領域の結果

(c) フォワード領域の結果

図 B.5 無限運動量飛跡に対する、Wire Strip Coincidence の応答。横軸に M3 における Strip のスタッガード ID、縦軸に M3 における Wire のスタッガード ID をとる。各 2 次元格子点をピボットとする無限運動量飛跡をシングルボード試験システムに投入し、 $0 \text{ GeV} < p_T$ の Segment を再構成できた場合にはその格子点を白色、できなかった場合は黒色で塗り潰す。Strip Segment Reconstruction、Wire Segment Reconstruction それぞれの Inefficiency を引き継いでいるほか、Wire スタッガード ID 400 番以降の領域で特定の構造を持った不具合が見られる。

の構造を持った不具合が見られる。この不具合については調査と修正が進行中である。

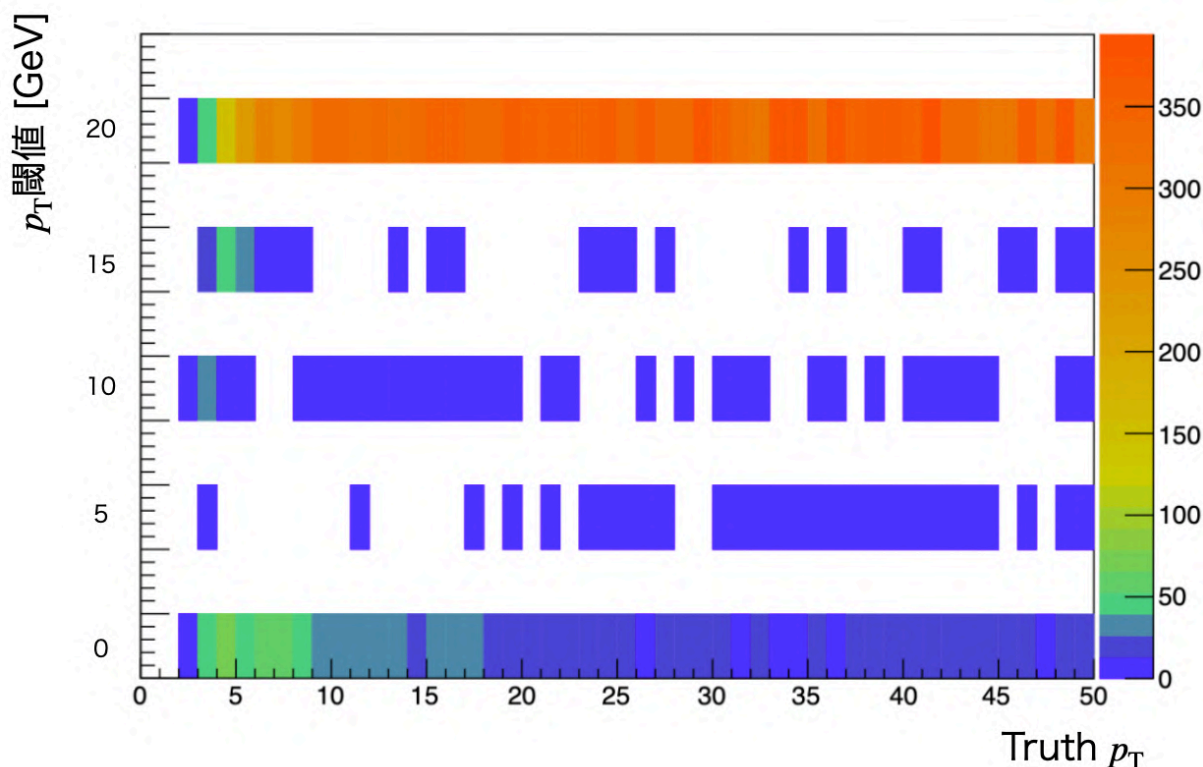


図 C.6 Wire LUT 修正前の Truth p_T と Wire Strip Coincidence で判定された p_T の関係。Truth p_T が 5 GeV や 10 GeV のイベントの多くが、 p_T 閾値 20 GeV と判定されており、正しい p_T 判定がなされていないことがわかる。

C MC データを用いた試験で明らかになった問題点とその修正

Wire Strip Coincidence

シングルミュオン MC を用いた試験によって、Wire Strip Coincidence の p_T 判定に関する不具合を発見した。図 C.6 にその時の結果を示す。この図では横軸に Truth muon の p_T 、縦軸には Wire Strip Coincidence で判定された p_T 閾値を示す。期待される挙動としては、Truth muon の p_T が 5 ~ 10 GeV のイベントはトリガーロジックによって p_T 閾値 5 GeV と判定される。しかし、実際は Truth p_T が 5 GeV や 10 GeV のイベントの多くが、 p_T 閾値 20 GeV と判定されていることがわかる。

この問題のデバッグを行ったところ、Wire Segment Reconstruction の出力として定義される $\Delta\theta$ の単位系と Wire Strip Coincidence の入力として定義される $\Delta\theta$ の単位系が異なっていることが明らかになった。Wire Segment reconstruction では $\Delta\theta$ は 4 mrad 区切りの整数値として定義され、Wire Strip Coincidence では 1.25 mrad 区切りの整数値として定義されていた。そのため、例えば Truth の $\Delta\theta$ が 10 mrad の場合、WireSegment Reconstruction の出力は"2"である。一方、Wire Strip Coincidence ではそれを $1.25 \times 2 = 25$ mrad と解釈する。これにより、Truth の p_T が小さく、 $\Delta\theta$ が大きいイベントに対しても高い p_T 閾値判定を行ってしまっていた。この不具合の修正のため、Wire Strip Coincidence の高い分解能に合わせる形で Wire LUT を作り直した。その結果を図 C.7 に示す。Wire Strip Coincidence では Truth p_T と矛盾のない p_T 閾値が判定されるようになった。

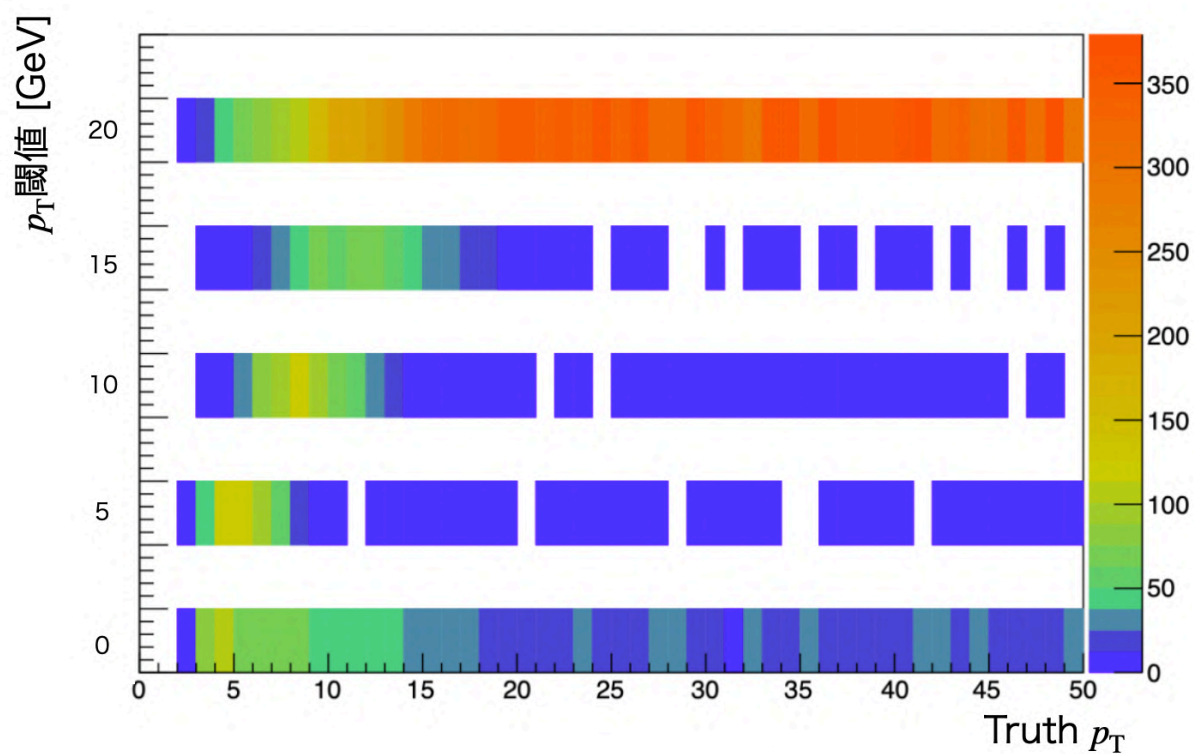


図 C.7 Wire LUT 修正後の Truth p_T と Wire Strip Coincidence で判定された p_T の関係。Wire Strip Coincidence において Truth p_T と矛盾のない p_T 閾値が出力されている。

表 D.3 Inner Coincidence を統合した後のデバイスのリソース使用状況

Name	Block	LUT (17280000)	REG (34560000)	CLB (2160000)	LUT as Memory (791040)	BRAM (2688)	URAM (1280)
SLR0 EC ϕ 1	Wire Station Coincidence	7.4	1.48	22.2	0	0	0
	Strip Station Coincidence	0	0.2	0.96	0	0	0
	Wire Segment Reconstruction	16.28	2.96	28.12	0	0	45.88
	Strip Segment Reconstruction	6.24	3.08	11.04	0.08	0	2.43
	Wire Strip Coincidence	3.56	3.76	14.56	0	37.52	0
	Total	58.24	31.04	95.28	1.08	73.68	51.56
SLR1	Inner Coincidence	67.8	19	83.64	3	28.88	50
	Track Selector	15.4	1.76	17.72	0	0	0
	Total	87.16	45.24	100	3	28.88	50
SLR2 EC ϕ 1	Wire Station Coincidence	7.4	1.48	22.2	0	0	0
	Strip Station Coincidence	0	0.2	0.96	0	0	0
	Wire Segment Reconstruction	16.28	2.96	28.12	0	0	45.88
	Strip Segment Reconstruction	6.24	3.08	11.04	0.08	0	2.43
	Wire Strip Coincidence	3.56	3.76	14.56	0	37.52	0
	Total	58.24	31.04	95.28	1.08	73.68	51.56
SLR3 FW	Wire Station Coincidence	3.84	0.64	8.96	0	0	0
	Strip Station Coincidence	0	0.04	0.16	0	0	0
	Wire Segment Reconstruction	6.4	1.28	9.6	0	0	19.84
	Strip Segment Reconstruction	1.24	0.6	2.52	0.04	0	1.24
	Wire Strip Coincidence	1.52	1.6	5.36	0	14.28	0
	Total	25.84	15.88	52.04	0.48	32.52	20.32

D タイミングバイオレーションへの対処

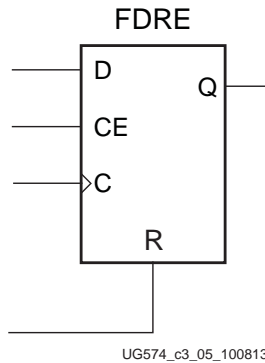
Inner Coincidence を統合した際、本研究による最適化を行う前の段階ではタイミング違反が発生していた。Inner Coincidence を統合した後のデバイスのリソース使用状況を表 D.3 に示す。表中の値は、1 つの SLR 中のリソースに対する使用量の割合を百分率で表したものである。Total にはトリガロジックだけでなく、コントロール、リードアウトのロジックも含めたリソース使用状況を示している。BRAM や URAM などのメモリーデバイスには余裕を持って実装できている一方で、Inner Coincidence を統合した SLR1 では Configurable Logic Block (CLB) の使用率が 100 % に達しており、リソースが逼迫している。SL のファームウェアはまだ開発段階であり、機能が増築されていくことを考えるとリソースの削減が必要不可欠であることが判明した。

タイミングの解析の結果を表 D.4 に示す。Setup 解析、Hold 解析どちらにおいてもタイミング違反(タイミングバイオレーション)が発生した。本研究では特に、Total Negative Slack (TNS) をタイミング違反の定量化に利用する。TNS とは、FPGA 内に含まれる全てのフリップフロップ間パス間のネガティブスラックを足し上げたものである。タイミングバイオレーションが生じている場合、ファームウェアの正常動作は保証されないため、これに関しても改善が必要である。

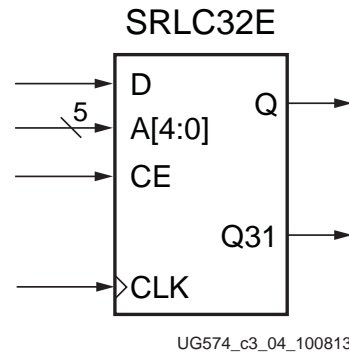
以下にタイミングバイオレーションを解決するために行った 2 つの最適化について説明する。いずれも、FPGA

表 D.4 Inner Coincidence 統合後のタイミング解析の結果

Name	Setup	Hold
Worst Negative Slack (WNS) / Worst Hold Slack (WHS)	-8.616 ns	-0.256 ns
Number of Failing Endpoints / Total Number of Endpoints	59 K / 1911 K	9K / 1911 K
Total Negative Slack (TNS) / Total Hold slack	-126, 297 ns	-388 ns



(a) FERE Primitive のロジックブロック



(b) SRLC32E Primitive のロジックブロック

図 D.8 UltraScale+ FPGA で使用されるシフトレジスタ用のプリミティブ [29]。FDRE は基本的なフリップフロップで 1bit のデータを保持する。SRL は最大深さ 32 のシフトレジスタ用の LUT で、A[4:0] でその深さを変えることができる。

で実現される論理自体に変更を加えることなく、リソースの使い方を最適化したものであり、トリガーロジックの性能に影響を与えるものではない。

D.1 シフトレジスタ実装方法の最適化

タイミングバイオレーション問題の解決に向けて、まずシフトレジスタの実装方法を検討した。一般に、FPGA でタイミング違反が生じた場合、2つのフリップフロップ間にシフトレジスタを挟む処理が行われる。これは、回路の大規模化に伴ってフリップフロップが離れた位置に配置された場合でも、シフトレジスタを中継点として挿入することで、フリップフロップ間の距離が短くなりタイミング制約を緩和することができるからである。この目的で挿入されるシフトレジスタをパイプラインレジスタと呼ぶ。SL FPGA でもこれまでタイミング違反が生じた場合に、この処置がとられて来た。たとえば、Wire Station Coincidence と Wire Segment Reconstruction の間で 2 段、Wire Segment Reconstruction と Wire Strip Coincidence の間で 2 段のパイプラインレジスタが挿入されている。

UltraScale+ FPGA ではシフトレジスタを実装するのに 2 種類のプリミティブ*1が用意されている。それぞれの概要を図 D.8 に示す。1 つ目は D Flip-Flop with Clock Enable and Synchronous Reset (FDRE) と呼ばれる基本的なフリップフロップである。1 つの FDRE は 1 bit のデータを保持するため、幅 n 、深さ m のシフトレジスタを実装するには $n \times m$ 個の FDRE が必要となる。2 つ目は Shift Register LUT (SRL) と呼ばれる、メモリー用の LUT である。最大の深さが 16 のもの (SRL16E) と 32 (SRLC32E) のものが存在し、同様のシフトレジスタを SRLC32E で実現しようとする $n \times (m/32)$ 個プリミティブが必要となる。

これまで SL FPGA ではパイプラインレジスタを実現するのに FDRE のみを利用して来た。これは一連のシフト

*1 FPGA の最小構成要素。FPGA の論理回路はプリミティブの組み合わせで構成される。代表的には LUT、Flip Flop、RAM などがある。目的の論理回路を実現するのにどのプリミティブを利用し、FPGA 内のどこに配置するかは“Vivado”ソフトウェアで自動で行われる。

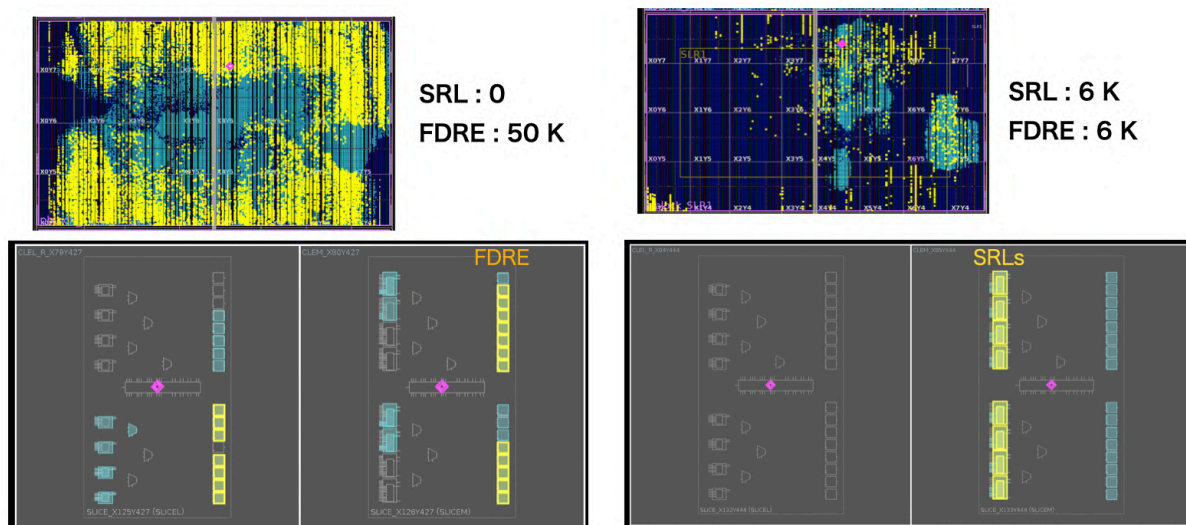


図 D.9 SRL を利用する場合と利用しない場合におけるリソース使用状況の比較

レジスタが 1 つの SRL タイルに配置されると、ロジックとロジックを繋ぐ中継点の役割を果たさず、タイミング制約を緩和するという本来の役割を果たさなくなると考えられていたからである。しかし、ロジックが大規模になり、基本的なプリミティブである FDRE の使用量が増えてくると FPGA 内に密集が発生し、逆に適切な位置にパイプラインレジスタを配置できなくなる。そこで、本研究では FDRE に加えて SRL も利用してインプリメンテーションを行い、どちらの方がタイミング制約の緩和に有効な実装方法か調査した。

図 D.9 に 2 種類の実装方法における SLR0 のプリミティブ使用状況を示す。図の黄色のセルが SLR0 内のパイプラインレジスタの実装に利用されているプリミティブである。FDRE のみを利用した場合、パイプラインレジスタは FDRE 50 K 個を用いて実装された。一方、SRL も利用した場合、パイプラインレジスタは SRL 6K 個、FDRE 6K 個を利用して実装された。図 D.11 に変更前後におけるファームウェア全体でのネガティブスラック発生状況、表 D.5 にタイミング解析の結果を示す。SRL を利用することで、TNS は -126297.769 ns から -7903.744 ns に約 1/16 に削減された。この結果は、SRL を利用したインプリメンテーションでは、より自由度の高い柔軟なロジック配置を行うことができ(フリップフロップ間の距離が遠いところには中継点としてパイプラインを設置し、逆に距離が近く不要なところでは数段のソフトレジスタを SRL にまとめて不要な密集を避けるなど)、タイミング制約が緩和したものと考えられる。

D.2 Track Selector の最適化

モジュールごとのネガティブスラックの量をみると、全体の 50% 以上が Track Selector で発生しており、このタイミングバイオレーションを解決することは、統合ファームウェア全体における制約の緩和に大きく貢献する。そこで本研究は Track Selector に必要なレイテンシーを 160 MHz 1 チック分増やす代わりに、ソーティングロジックの簡略化を行い、タイミング制約の緩和を図った。

従来のロジックと変更後のロジックの概念図を図 D.10 に示す。Track Selector は Inner Coincidence から出力された最大 128 個のミューオン飛跡候補から、 p_T の大きいものを 8 個選び出すものである。1 つのミューオン候補に対して 124 bit の情報が用意される。128 個の飛跡候補は 160 MHz の 1 チックごとに 1 対 1 で p_T 比較が行われ、合計 4 チックで 8 候補まで絞り込まれる。従来のロジックではこのソーティングのロジックにおいて 124 bit × 128 候補のデータを並列に並べた上で、124 bit のデータ同士を並び替えていた。これに対し、変更後のロジックでは Inner Coincidence から出力される 124 bit のデータのうち、ソーティングに利用する p_T 5 bit の情報だけを

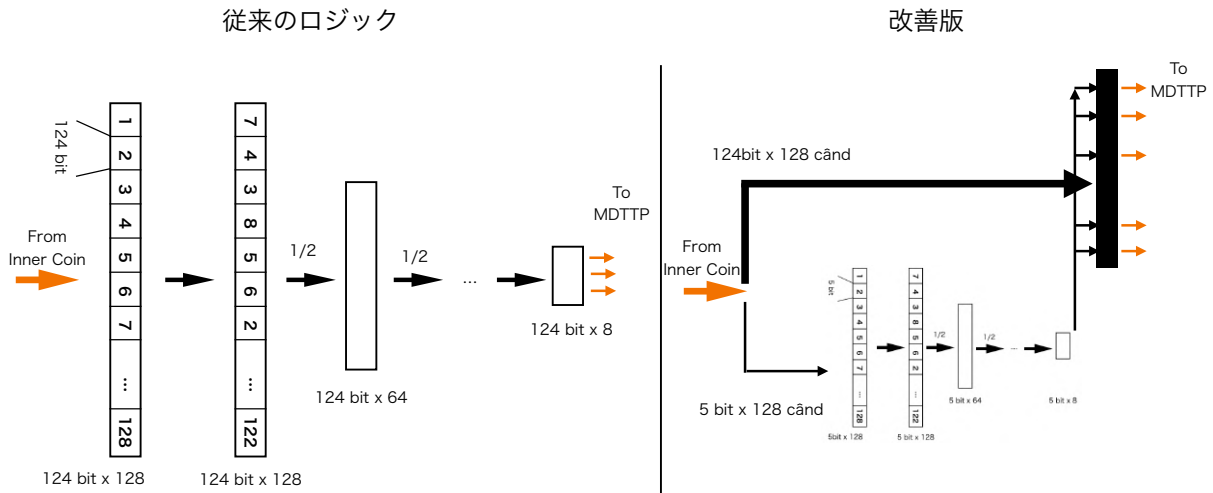


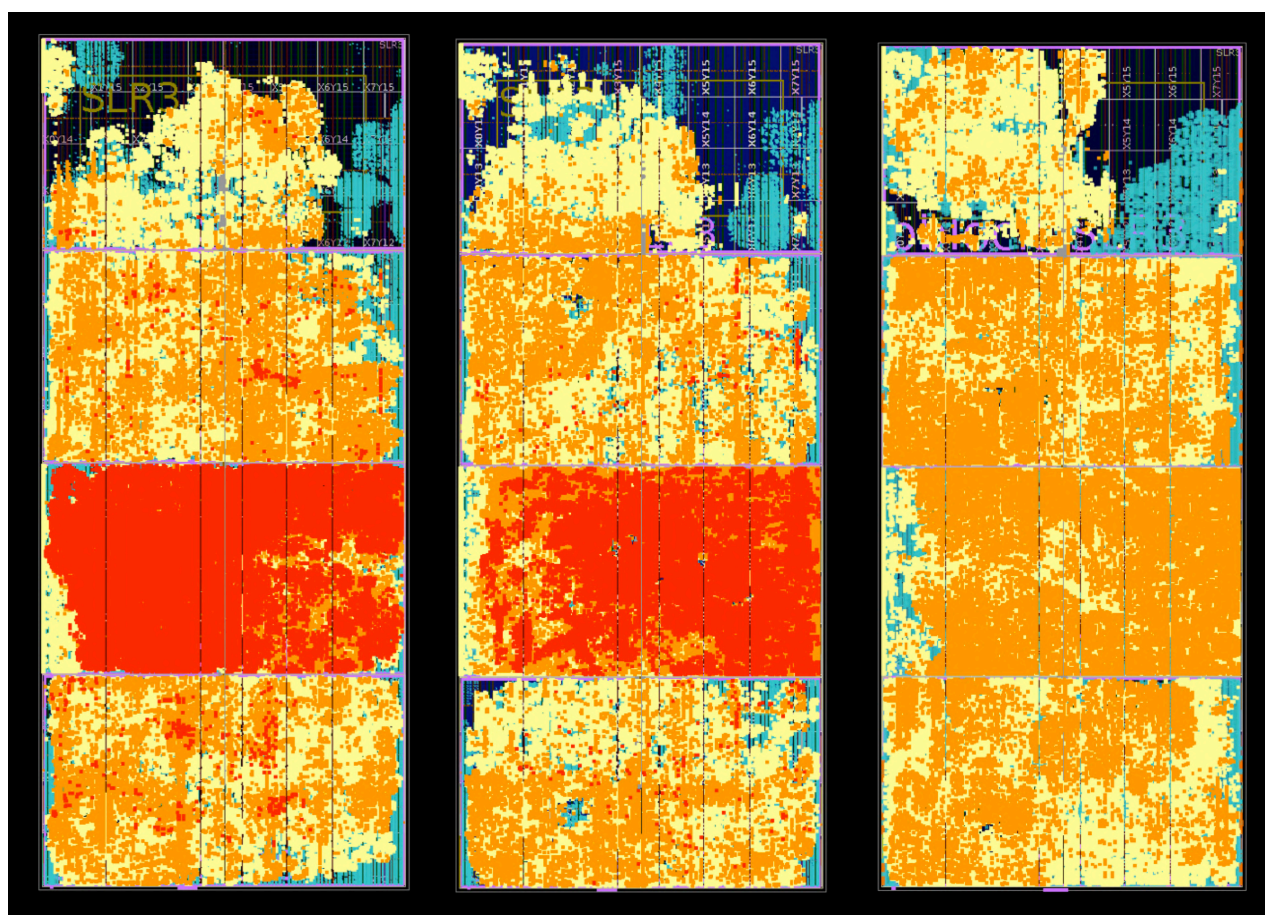
図 D.10 Track Selector の最適化前後の比較。従来のロジックでは 124 bit × 128 候補のデータを並列に並べた上で、124 bit のデータ同士を並び替えることでソーティングを実現していた。これに対し、変更後のロジックではソーティングに利用する p_T 5 bit の情報だけを用いて並び替えを行い、そのほかのデータは 128 個の飛跡候補を識別する 7 bit の ID と共に BRAM に格納しておく。

表 D.5 Track Selector の最適化前後のタイミング解析の結果

Name	最適化前		シフトレジスタ最適化		Track Selector 最適化	
	Setup	Hold	Setup	Hold	Setup	Hold
Worst Slack	-8.616 ns	-0.256 ns	- 1.164 ns	0	0	0
Number of Failing Endpoints / Total Number of Endpoints	59 K / 1911 K	9K / 1911 K	14 K / 1643 K	0 / 1643 K	0 / 1716 K	0 / 1716 K
Total Slack	-126, 297 ns	-388 ns	- 7903.744	0	0	0

取り出し並び替えを行う。そのほかのデータは 128 個の飛跡候補を識別する 7 bit の ID と共に BRAM に格納しておく。ソーティングが完了し 8 個の飛跡候補が選ばれたら、その ID を用いて BRAM からデータを取り出す。この方法では、ソーティングの後 BRAM からデータを取り出すのに 160 MHz 1 チック分レイテンシーが余計にかかってしまうが、ソーティングのために FPGA 内を遷移させる信号の量を 124 bit × 128 候補から 5 bit × 128 候補に削減することができる。これにより、FPGA で配線する必要のあるプリミティブを大幅に減らすことができ、タイミング制約の緩和する。

図 D.11 および表 D.5 に最適化前後のタイミング解析の結果をまとめる。シフトレジスタの最適化後にあった -7903.744 ns の TNS は 0 ns になっており、タイミングバイオレーションを解決することができた。



(a) 最適化前

(b) シフトレジスタ最適化後

(c) Track Selector 最適化後

図 D.11 最適化前後のネガティブスラック発生状況。赤色のセルがネガティブスラックが発生している領域を示す。オレンジ、黄色、白色の順に正のスラックが大きい。シフトレジスタの最適化と Track Selector の最適化を経て、タイミングバイオレーションを解決することができた。

引用文献

- [1] P. A. Zyla et al. The Review of Particle Physics. *Prog. Theor. Exp. Phys.*, 2020:083C01, 2020.
- [2] CERN Service graphique. Overall view of the LHC, June 2014.
- [3] The ATLAS Collaboration et al. *JINST*, 3:S08003, 2008.
- [4] HL-LHC industry CERN. Project Schedule.
- [5] Technical Design Report for the Phase-II Upgrade of the ATLAS Muon Spectrometer. Technical report, CERN, Geneva, 2017.
- [6] CERN Document Server. Installation of the first of the big wheels of the atlas muon spectrometer, a thin gap chamber (tgc) wheel, 2006.
- [7] Andrea Ventura. The Muon Trigger of the ATLAS experiment: performance and improvements for Run 3. page 654, 11 2022.
- [8] The ATLAS Collaboration. Trigger menu in 2018. Technical report, Oct 2019.
- [9] The ATLAS Collaboration. Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System. Technical Report CERN-LHCC-2017-020, CERN, June 2018.
- [10] 赤塚 俊一. LHC-ATLAS 実験 Run-3 に向けたミューオントリガーの改良. Master's thesis, 京都大学大学院 理学系研究科 物理学専攻, 2017.
- [11] 河本 地弘. 高輝度 LHC ATLAS 実験に向けた初段ミューオントリガーアルゴリズムの実装と検出器全体への拡張. Master's thesis, 京都大学大学院理学研究科 物理学・宇宙物理学専攻 物理学第二分野 高エネルギー物理学研究室, 2023.
- [12] The ATLAS Collaboration. Amplifier-Shaper-Discriminator ICs and ASD Boards. Technical report, 1999.
- [13] TGC Electronics Group. TGC Patch-Panel ASIC Design Report for Production Readiness Review. Technical report, 2019.
- [14] The ATLAS Collaboration. ATLAS Muon Spectrometer Technical Design Report. Technical report, July 1997.
- [15] 竹本亨史. ATLAS 実験ミューオントリガーシステム用 LSI の開発と統合評価テスト. Master's thesis, 東京大学大学院 理学系研究科 物理学専攻, 2004.
- [16] 青木匠. 大規模エレクトロニクスシステムにおける次世代型オペレーションモデルの研究 -自立型ステートマシンの導入及び SoC による遠隔型制御の実現-. Master's thesis, 東京大学大学院 理学系研究科 物理学専攻, 2022.
- [17] The ATLAS Collaboration Yasuyuki Horii. ATLAS-TDR-026. Technical report, April 2019.
- [18] 田中 碧人. System-on-a-Chip を用いた エレクトロニクス制御回路の開発 - 高放射線環境での大規模システムへの応用. Master's thesis, 東京大学大学院 理学系研究科 物理学専攻, 2021.
- [19] 三島 章熙. 高輝度 LHC-ATLAS 実験に向けたミューオントリガー論理回路の開発・実装・試験 — 読み出し・制御系の構築とトリガー系の統合. Master's thesis, 東京大学大学院 理学系研究科 物理学専攻, 2023.

- [20] 三野 裕哉. 高輝度 LHC ATLAS 実験に向けた初段ミューオントリガーアルゴリズムの開発およびハードウェアへの実装. Master's thesis, 京都大学大学院理学研究科 物理学・宇宙物理学専攻 物理学第二分野 高エネルギー物理学研究室, 2020.
- [21] ATLAS TDAQ PhaseII Upgrade Project. Endcap Sector Logic. Technical report, 2021.
- [22] 小林蓮. 高輝度 LHC ATLAS 実験に向けた初段ミューオントリガーアルゴリズムの改良とハードウェアへの実装. Master's thesis, 京都大学大学院 理学系研究科 物理学専攻, 2021.
- [23] K. E. Batcher. Sorting networks and their application. *Proc. AFIPS Spring Joint Computer Conference*, 32:307–314, 1968.
- [24] 山下 恵理香. 高輝度 LHC-ATLAS 実験に向けた初段ミューオントリガーの開発と検証. Master's thesis, 東京大学大学院 理学系研究科 物理学専攻, 2023.
- [25] 岡崎 健人. System-on-a-Chip を用いた組み込みシステム開発基盤の構築と FPGA 間通信の実証 — 次世代高エネルギー実験制御系への応用. Master's thesis, 東京大学大学院 理学系研究科 物理学専攻, 2022.
- [26] 和田 有咲. 高輝度 LHC-ATLAS 実験に向けた新型内層 TGC 検出器実機の性能評価とトリガー回路開発. Master's thesis, 名古屋大学大学院 理学研究科 物理学専攻, 2024.
- [27] 橋本 大輔. 高輝度 LHC-ATLAS 実験に向けた TGC 検出器前段回路量産機の最終評価と電源供給回路実機の開発. Master's thesis, 名古屋大学大学院 理学研究科 物理学専攻, 2024.
- [28] 鍋山 友希. HL-LHC ATLAS 実験に向けたミューオン飛跡検出・選別回路の開発とパイルアップ耐性評価. Master's thesis, 名古屋大学大学院理学研究科 素粒子宇宙物理学専攻 高エネルギー素粒子物理学研究室, 2023.
- [29] XILINX. UltraScale Architecture Configurable Logic Block. Technical report, 2017.