

修士学位論文

LHC-ATLAS 実験における液体アルゴンカロリメータの
アップグレードに向けたファームウェアの研究開発

Firmware development for the upgrade of the liquid argon
calorimeter at the LHC-ATLAS experiment

東京大学大学院
理学系研究科 物理学専攻
田中研究室

宇野 健太

2017年 1月

目次

第 1 章	序論	1
1.1	本研究の背景	1
1.2	本研究の目的	4
1.3	本論文の構成	5
第 2 章	LHC-ATLAS 実験	6
2.1	LHC, ATLAS について	6
2.2	ATLAS 検出器概要	7
2.2.1	内部飛跡検出器	7
2.2.2	カロリメータ	9
2.2.3	ミュオンスペクトロメータ	11
2.3	トリガーシステム	12
2.4	TTC システム	13
2.5	ATLAS アップグレード計画	13
第 3 章	LAr カロリメータアップグレード	15
3.1	現在の読み出し方法	15
3.1.1	フロントエンドエレクトロニクス	16
3.1.2	バックエンドエレクトロニクス	17
3.1.3	L1Calo トリガーシステム	20
3.2	アップグレード後読み出し方法	20
3.2.1	Layer Sum Board (LSB)	21
3.2.2	LAr Trigger Digitizer Board (LTDB)	21
3.2.3	LAr Digital Processing Board (LDPB)	22
3.2.4	Feature Extractor module (FEX)	23
3.3	Field Programmable Gate Array (FPGA)	24
3.3.1	Adaptive Logic Module (ALM)	24
3.3.2	Random Access Memory (RAM)	24
3.3.3	Digital Signal Processor Block (DSP Block)	26
第 4 章	デモンストレータプロジェクト	28
4.1	プロジェクト概要	28
4.2	LTDB デモンストレータ	29
4.3	LDPB デモンストレータ	29
4.4	バックエンド FPGA ファームウェア	30

4.4.1	Front FPGA ファームウェア	31
4.4.2	Back FPGA ファームウェア	31
4.5	Adc readout モジュールのファームウェア	32
4.5.1	Channel mixing issue	38
4.5.2	latency shift (pulse shift) issue	41
4.6	デモンストレータプロジェクトのまとめ	44
第 5 章	LATOME プロジェクト	45
5.1	LATOME Board	45
5.2	LATOME ファームウェア	45
5.2.1	Low Level Interface (LLI)	47
5.2.2	Input stage	48
5.2.3	Configurable remapping	49
5.2.4	User Code	51
5.2.5	Output summing	51
5.2.6	TDAQ/Monitoring	51
5.2.7	IPBus controller	52
5.2.8	TTC	52
5.3	User Code	52
5.3.1	Filter の設計	52
5.3.2	Optimal filter 後の適切な選択条件	54
5.3.3	τ の選択条件の設計	60
5.3.4	エネルギー, 係数とペDESTALフォーマット	60
5.3.5	サチュレーションした波形の同定方法	67
5.3.6	コンフィグレーションの変更方法	71
5.3.7	Baseline Correction	72
5.3.8	ブロックダイアグラム	75
5.4	ファームウェア動作試験	76
5.4.1	インプット、チェッカーテストベンチ	76
5.4.2	シミュレーションによる動作確認	76
5.4.3	ボードによる試験	79
5.5	統合試験	82
第 6 章	結論と今後に向けて	84
6.1	結論	84
6.2	今後に向けて	84

目 次

1.1	トリガーレベルでの電磁カロリメータの読み出す領域。	2
1.2	電子収集効率 95% 下でのエネルギー閾値とトリガーレート	2
1.3	Trigger Tower と Super Cell	3
1.4	Super Cell 読み出しで使用可能になる変数	4
1.5	LAr カロリメータの読み出し経路	5
2.1	LHC 加速器	6
2.2	ATLAS 検出器	7
2.3	内部飛跡検出器	7
2.4	パレル部の SCT モジュール	8
2.5	それぞれの粒子に対してのカロリメータの応答	9
2.6	ATLAS 電磁カロリメータのアコーディオン構造	10
2.7	パレル部 ($\eta=0$) における LAr カロリメータの層構造	10
2.8	フォワードカロリメータの構造	11
2.9	MDT の構造	12
2.10	Run2 のトリガーシステムの流れ	13
2.11	LHC のアップグレード計画	14
3.1	液体アルゴンカロリメータの構造	15
3.2	カロリメータから得られる信号	16
3.3	現在の液体アルゴンカロリメータの読み出しシステム	16
3.4	検出器から得られる信号とシェイピング後の信号	17
3.5	L1Calo トリガーシステム	20
3.6	パレル部での 1 ビットの重み	22
3.7	LDPB	23
3.8	ATCA プラットフォーム	23
3.9	L1Calo アップグレードの全体図	24
3.10	Arria10 の ALM の配線図	25
3.11	StratixIV の ALM の配線図	25
3.12	Arria10 シリーズのメモリ機能	26
3.13	Arria10 シリーズの DSP Block の配線図 (18×19 モード)	27
4.1	デモンストレータ導入後の読み出しシステム	28
4.2	LTDB のデータフォーマット	29
4.3	LTDB デモンストレータ	29

4.4	ABBA ボード	30
4.5	Front FPGA ファームウェアの全体図	31
4.6	Back FPGA ファームウェアの全体図	31
4.7	adc readout モジュールの設計図	32
4.8	adc readout モジュールのインプットデータシーケンス	32
4.9	サーキュラーバッファのデータフロー	33
4.10	address FIFO のデータフロー	34
4.11	ABBA が取得する波形データ	35
4.12	ABBA の Header と Trailer 一覧	35
4.13	L1A ADC FIFO の読み出しシーケンス	36
4.14	64 ビットパケットフォーマット	36
4.15	Packet FIFO のデータシーケンス	37
4.16	IP bus からの request 信号と reply 信号	37
4.17	Channel mixing が現れたデータ	38
4.18	セットアップスラック	39
4.19	ホールドスラック	39
4.20	Front FPGA ファームウェアの全経路でのワーストセットアップスラック の一覧	40
4.21	Front FPGA ファームウェアの全経路でのワーストホールドスラックの一覧	40
4.22	パルスシフトが起きた時の波形	41
4.23	インプットデータシーケンス	41
4.24	近いタイミングで2つの ttc accept 信号が入った時の address FIFO の振舞い	42
4.25	連続した2つ ttc accept 信号の事象	43
4.26	改善後のファームウェアのデータフロー	43
5.1	LATOME Board	45
5.2	LATOME Board と Arria10 FPGA との接続の略図	46
5.3	LLI の全体図	46
5.4	LATOME ファームウェア全体図	47
5.5	パラレル変換された時のデータフォーマット	48
5.6	Input stage モジュールの全体図	48
5.7	出力のデータシーケンス	49
5.8	Configurable remapping でのチャンネル並び替え	50
5.9	バレル部でのマッピングの一例	50
5.10	エンドキャップ部でのマッピング一例	51
5.11	Optimal filter のファームウェア設計	53
5.12	Optimal filter のデータフロー	54
5.13	L1Calo でのエネルギー再構成率	55
5.14	S-Frame に使用した理想的な波形 g	56
5.15	カロリーメータの信号	56
5.16	5 GeV と 20 GeV の目的事象に対する E_T と τ の広がり	57
5.17	分布の広がりエネルギー依存性	57

5.18	τ の選択条件を課した時の目的事象の信号効率	58
5.19	$\mu=80$ におけるノイズのシミュレーション	59
5.20	フェイク事象の発生確率	59
5.21	τ の選択条件の設計	60
5.22	multi linear region エンコーディング (eFEX の場合)	62
5.23	multi linear region エンコーディングと平方根のエンコーディングの振舞い	63
5.24	R のエネルギー依存性	64
5.25	multi linear region エンコーディング (jFEX の場合)	65
5.26	エネルギーフォーマット	65
5.27	Filter 計算の流れ	66
5.28	DSP Block の出力からエネルギーを抜き出す方法と誤った抜き出しの例	67
5.29	レンジングチェックの設計	67
5.30	デモンストレータから得られたサチュレーションした波形	68
5.31	$E_T - E_T \cdot \tau$ の相関図	68
5.32	サチュレーションした信号に対する処理	70
5.33	2.5 GeV 以上でサチュレーションしてない Super Cell の η 分布	70
5.34	BCAV 再同定ファームウェア	71
5.35	デュアルクロック、デュアルポート RAM	71
5.36	係数、ペDESTAL のコンフィグレーション設計	72
5.37	IP Controller からのフォーマット	72
5.38	ペDESTAL のバンチ依存性	73
5.39	MET のトリガーレートのバンチ番号依存性	73
5.40	ベースラインコレクションの設計	74
5.41	User Code Block	75
5.42	評価用 User Code ファームウェア	77
5.43	ファームウェア中でのパラメータ定義の例	78
5.44	シミュレーション結果	79
5.45	Arria10 評価ボード	79
5.46	実験のセットアップ	80
5.47	取得できた信号	81
5.48	LAr カロリメータ波形を入れた試験	81
5.49	ファームウェア出力値の精度比較	82
5.50	チェッカー付き LATOME ファームウェア	83
5.51	ファームウェア統合状況	83

表 目 次

1.1	各読み出し方法の比較	3
2.1	ATLAS 実験の予定	14
3.1	それぞれの領域に導入される LTDB 数とそのチャンネル数	21
4.1	adc readout モジュールに入る TTC 信号	33
5.1	各モジュールに要求されるデザイン段階のレイテンシ (1 BC=25 ns)	47
5.2	ファームウェア内で使用する DSP Block の使用率	54
5.3	シミュレーションに用いた熱ノイズとパイルアップ値	57
5.4	平方根ファームウェアを実装した場合のレイテンシと最大動作周波数	62
5.5	ベースラインコレクションを実装する際の様々なパラメータ	74
5.6	User Code のレイテンシ	76
5.7	User Code のパラメータ一覧	77
5.8	シミュレーションのマイルストーン	78
5.9	チェッカー付き User Code ファームウェアのリソース	80
5.10	240 MHz クロックの最大動作周波数	80
5.11	統合したファームウェアのレイテンシとリソース	82

概要

欧州原子核研究機構にある Large Hadron Collider (LHC) は、世界最高の重心系エネルギーをもつ加速器である。ATLAS 実験は現在、この LHC 加速器で重心系エネルギー 13 TeV の Run2 実験を行っている。これと並行して、エネルギーとルミノシティを共に増強する予定の Run3 実験 (2021 年頃から開始予定) に向けて、検出器や読み出し回路の改善に関する研究が行われている。

LHC における全断面積は 100 mb と高く、陽子バンチの衝突を 25 ns 毎に起こすことから、全ての事象を記録出来ない。そのため、ATLAS 実験は 2 段階のトリガーシステムを設けている。1 段目のトリガー (L1 トリガー) は、ハードウェアベースで全事象に対して粗い選別を行い、電磁カロリメータの場合はエネルギーに閾値をかけている。しかしながら、ルミノシティの高輝度化につれて背景事象は多くなるため、L1 トリガーにおいて信号事象との識別が困難になっていく。これは、信号事象の収集効率の低下につながる。そのため、Run3 実験では従来よりも 10 倍細かい読み出しへ変更する。これにより、トリガーレベルで背景事象との識別能力が向上し、信号事象の収集効率が保てる。この読み出しを実現するための一つとして、10 倍細かくした信号のエネルギーと時間を高速に算出することが挙げられる。本研究は、これを達成する FPGA ファームウェアを開発した。

効率的なエネルギー計算を考案するために、本研究は ATLAS 検出器の一部に導入されているデモンストレータを使用した。このデモンストレータにある FPGA で、10 倍細かくした信号が Run2 実験で収集されている。しかし、この FPGA ファームウェアのテストベンチが作成されていなかったため、信号取得の際に起きる問題の原因を特定できず、効率良い取得ができなかった。そのため、本研究では作成されていなかったテストベンチを開発し、問題の原因を特定した。それを元に改善を施したファームウェアは、安定的な信号取得を可能にした。

得られた信号から効率的なエネルギー計算方法を考案し、そのファームウェアを開発した。このファームウェアは 240 MHz で動作しており、処理時間は 112.5 ns である。この結果は、ATLAS の L1 トリガーからくる要求値 (150 ns) を満たしている。また、開発したファームウェアは FPGA に実装可能なリソース量で、257.6 MHz の最大動作周波数をもっている。その後、ボードを用いて様々な試験を行い、ファームウェアの安定した動作を確認した。本研究によって、新しい読み出しに対応した FPGA ファームウェアの第 1 版が完成した。

第1章 序論

素粒子とその相互作用を記述する理論である標準模型は、現在の素粒子物理学の基礎となっている。2012年にヒッグス粒子が欧州原子核研究機構 (CERN) の Large Hadron Collider (LHC) で発見され、標準模型で予想される全ての粒子は実験的に発見された。しかしながら、暗黒物質の存在や階層性問題など標準模型では説明できない現象も数多く残っている。この標準模型を超える物理現象を見出すため、世界最高重心系エネルギーを誇る陽子・陽子衝突型円形加速器 LHC において ATLAS 実験を行っている。

ATLAS 実験は、陽子・陽子衝突事象を詳細に解析するために様々な検出器を用いている。また、効率的なデータ収集のために検出器からの情報を用いて事象 (イベント) を高速に取捨選択を行うトリガーシステムをもつ。本研究は液体アルゴン (LAr) 電磁カロリメータのトリガー読み出しの増強計画に根ざしたものである。LHC の高輝度化に伴って背景事象が増加し、興味ある事象との識別が困難になる状況が予想される。このアップグレード計画は読み出し手法の改善であり、具体的には新しいエレクトロニクスを導入である。本研究の目的は、このエレクトロニクスに不可欠な、FPGA に実装されるファームウェア開発とその評価である。

この章では、本研究の背景と目的について簡単に述べる。

1.1 本研究の背景

2015年に行っている LHC-ATLAS Run2 実験は陽子陽子衝突 (全断面積は約 100 mb) をおこし、そこで生成される事象の測定を行っている。その頻度は 40 MHz (25 ns) であり、全ての事象を記録することが困難である。そのため、ATLAS 実験は 2 段階のトリガーシステムを備えており、興味ある事象を選別し記録している。主にハードウェアで処理する 1 段目のトリガー (L1 トリガー) は粗い選別を行い、ソフトウェアで処理できるほどのデータ量まで減らす。その後、2 段目のトリガー (HLT) はオフラインデータ解析と同程度の性能をもつ選別をソフトウェアで行い、最終的に 1 kHz で記録する。

LAr 電磁カロリメータは粒子の通過する縦方向に 4 つの層をもつ。現行の L1 トリガーレベルでは、 $\Delta\eta \times \Delta\phi = 0.1 \times 0.1$ で測定した入射粒子のエネルギーを 4 層全て足し合わせて読み出している (図 1.1 左)。このセグメント方法は、Trigger Tower と呼ばれている。L1 トリガーのレートを落とす手段の 1 つに、Trigger Tower で読み出したエネルギーに閾値をかける。電子や光子の事象に対してのトリガーレートの要求値は 20 kHz である。これを達成するために、エネルギー閾値は 25 GeV に設定される。しかしながら、今後の実験はルミノシティを増加させる予定であるため、1 回の衝突で反応する陽子の数 (パイルアップ) が増える。そのため、従来のエネルギー閾値を超える事象数が増え、トリガーレート (トリガー発行頻度) が上昇する。実際は、要求されるトリガーレートを満たすためにエネルギー

ギー閾値を上げる。2021 年から開始される予定の Run3 環境下で電子や光子の収集効率 95 %に保った時をシミュレーションすると、トリガーレート 20 kHz で 34 GeV 程度のエネルギー閾値になる (図 1.2 青点)。この場合、Z や W から崩壊・生成された電子など興味ある事象が L1 トリガーを通らず、その収集効率が減少してしまう。

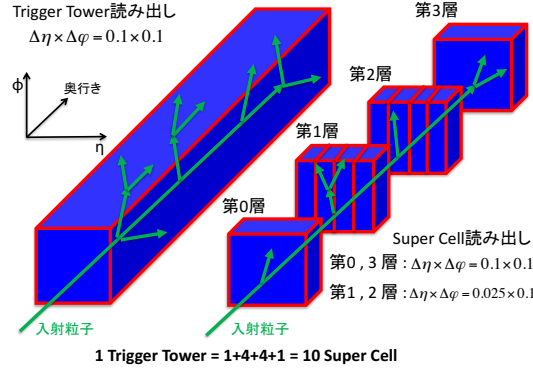


図 1.1: トリガーレベルでの電磁カロリメータの読み出す領域 ((左)Trigger Tower、(右)Super Cell)。Super Cell での第 1 層と第 2 層は、 η 方向を細かく読み出す。

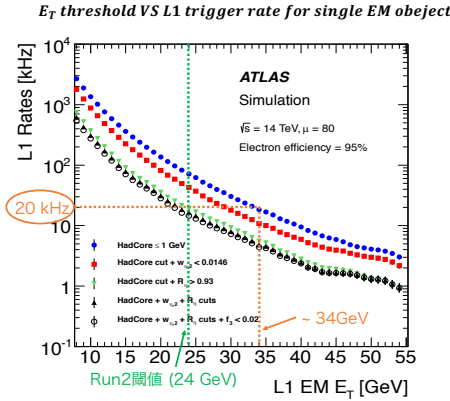


図 1.2: 電子収集効率 95% 下でのエネルギー閾値とトリガーレート [1]。14 TeV かつ平均パイルアップ数 $\mu=80$ においてのシミュレーションであり、青点が Trigger Tower 読み出しにおける関係である。ハドロンカロリメータに落としたエネルギーを使用した変数 (HadCore) を用いると、34 GeV 程度のエネルギー閾値が必要になる。Super Cell 読み出しで数式 (1.1)~(1.3) が新たに使用でき、最終的に黒丸で示される 23 GeV 程度のエネルギー閾値が得られる。

そこで、読み出し方法を、図 1.1 右のような構成に変更する。このセグメントは Super Cell と呼ばれ、LAr カロリメータの 4 層をそれぞれ個別に読み出し、特に第 1, 2 層は $\Delta\eta \times \Delta\phi = 0.025 \times 0.1$ として読み出す (表 1.1、図 1.3)。特別な領域を除き、1 個の Trigger Tower は 10 個の Super Cell に対応している。

電子や光子とジェットの識別効率を上げるために、Super Cell 読み出しで可能な 3 つの変数を定義する。

$$R_\eta = \frac{E_{T, \Delta\eta \times \Delta\phi=0.075 \times 0.2}^{(2)}}{E_{T, \Delta\eta \times \Delta\phi=0.175 \times 0.2}^{(2)}} \quad (1.1)$$

表 1.1: 各読み出し方法の比較

		Elementary Cell	Trigger Tower		Super Cell	
Layer		$\Delta\eta \times \Delta\phi$	$n_\eta \times n_\phi$	$\Delta\eta \times \Delta\phi$	$n_\eta \times n_\phi$	$\Delta\eta \times \Delta\phi$
0	Presampler	0.025×0.1	4×1	0.1×0.1	4×1	0.1×0.1
1	Front	0.003125×0.1	32×1	0.1×0.1	8×1	0.025×0.1
2	Middle	0.025×0.025	4×4	0.1×0.1	1×4	0.025×0.1
3	Back	0.05×0.025	2×4	0.1×0.1	2×4	0.1×0.1

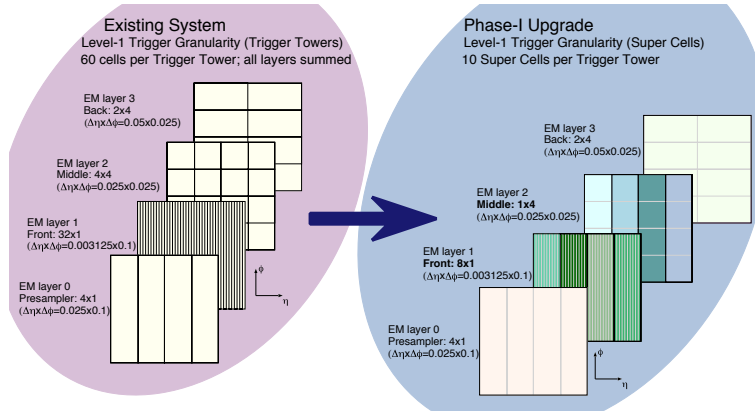


図 1.3: Trigger Tower と Super Cell[1]

$$w_{\eta,2} = \sqrt{\frac{\sum(E_T^{(2)} \times \eta^2)_{\Delta\eta \times \Delta\phi=0.075 \times 0.2}}{E_{T,\Delta\eta \times \Delta\phi=0.075 \times 0.2}^{(2)}} - \left(\frac{\sum(E_T^{(2)} \times \eta^2)_{\Delta\eta \times \Delta\phi=0.075 \times 0.2}}{E_{T,\Delta\eta \times \Delta\phi=0.075 \times 0.2}^{(2)}}\right)^2} \quad (1.2)$$

$$f_3 = \frac{E_{T,\Delta\eta \times \Delta\phi=0.2 \times 0.2}^{(3)}}{E_{T,\Delta\eta \times \Delta\phi=0.075 \times 0.2}^{(1)} + E_{T,\Delta\eta \times \Delta\phi=0.075 \times 0.2}^{(2)} + E_{T,\Delta\eta \times \Delta\phi=0.2 \times 0.2}^{(3)}} \quad (1.3)$$

R_η は、第2層で最もエネルギーを落とした Super Cell を中心にして $\Delta\eta \times \Delta\phi = 0.075 \times 0.2$ でクラスタを組んだエネルギーと、 $\Delta\eta \times \Delta\phi = 0.175 \times 0.2$ のエネルギーとの比である。電子や光子のシャワーの横方向の広がり $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2} \cong 0.08$ である。そのため、第二層に落としたエネルギーは $\Delta\eta \times \Delta\phi = 0.075 \times 0.2$ にほとんど含まれており、分布は1付近にピークをもつ。一方、ジェットの広がり $R \cong 0.8$ 程度で、 $\Delta\eta \times \Delta\phi = 0.075 \times 0.2$ の領域ではすべてのエネルギーを捉えられない (図 1.4 (1))。また、 $w_{\eta,2}$ は、第2層の $\Delta\eta \times \Delta\phi = 0.075 \times 0.2$ で見られるシャワーの広がり ΔR を示した変数であり、同様の理由で分布に違いが見える (図 1.4 (2))。

f_3 は、第3層の $\Delta\eta \times \Delta\phi = 0.2 \times 0.2$ エネルギーと、それに第1,2層の $\Delta\eta \times \Delta\phi = 0.075 \times 0.2$ を加えたエネルギーの比である。電子や光子は第2層までにエネルギーをほとんど落とすので、 $E_{T,\Delta\eta \times \Delta\phi=0.2 \times 0.2}^{(3)}$ が小さくなる。よって、 f_3 の分布は0付近にピークをもつ (図 1.4 (3))。一方、ジェットはハドロンカロリメータまで通過するので、電子や光子と比較すると、 f_3 は大きな値をとり得る。これらの変数を用いると電子や光子の識別能力が向上し、エネルギー閾値を上げずにトリガーレートが保てる (図 1.2 黒丸)。

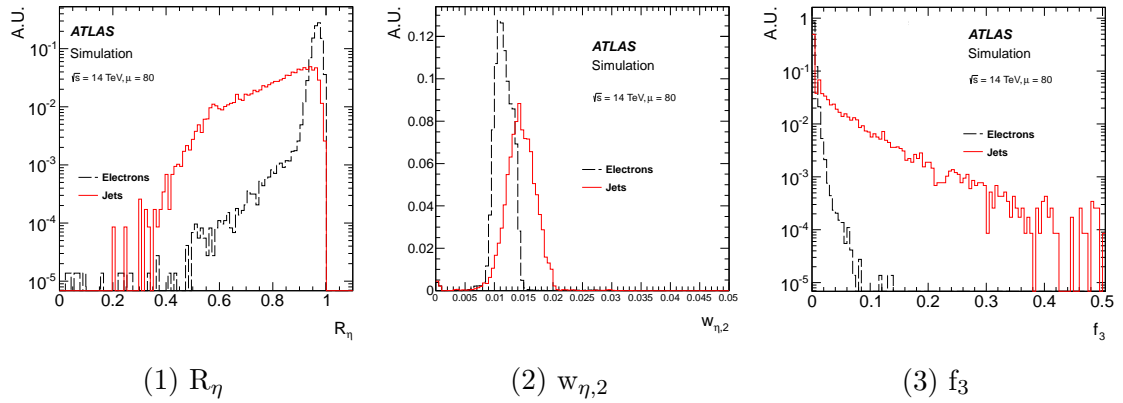


図 1.4: Super Cell 読み出しで使用可能になる変数 [1]。黒が電子、赤がジェットの分布である。

このため、ATLAS LAr グループは Super Cell 読み出しを可能にするための研究開発を行っている。

1.2 本研究の目的

図 1.5 は、LAr カロリメータの読み出し経路を示している。従来のトリガー経路の場合、信号は Front-end Board から Tower Builder Board (TBB) へ流れる。L1 トリガーで使用する Trigger Tower 信号は、TBB で足し合わせされる。一方、アップグレード後のトリガー経路の場合、信号は Front-end Board から新しく導入する LAr Trigger Digitizer Board (LTDB) へ流れる。LTDB は、Super Cell 信号を作成し、アナログデジタル変換 (AD 変換) 後に 80 m 後方の LAr Digital Processing System (LDPS) に光ファイバーを用いて伝送する。LDPS では全ての Super Cell 信号をエネルギーに変換し、L1Calo トリガーシステムへ伝送する。これらの通信速度は 50 Tbps 程度になるため、高速通信可能なボードと低レイテンシでエネルギー変換を行う FPGA ファームウェアの開発が必要になる。

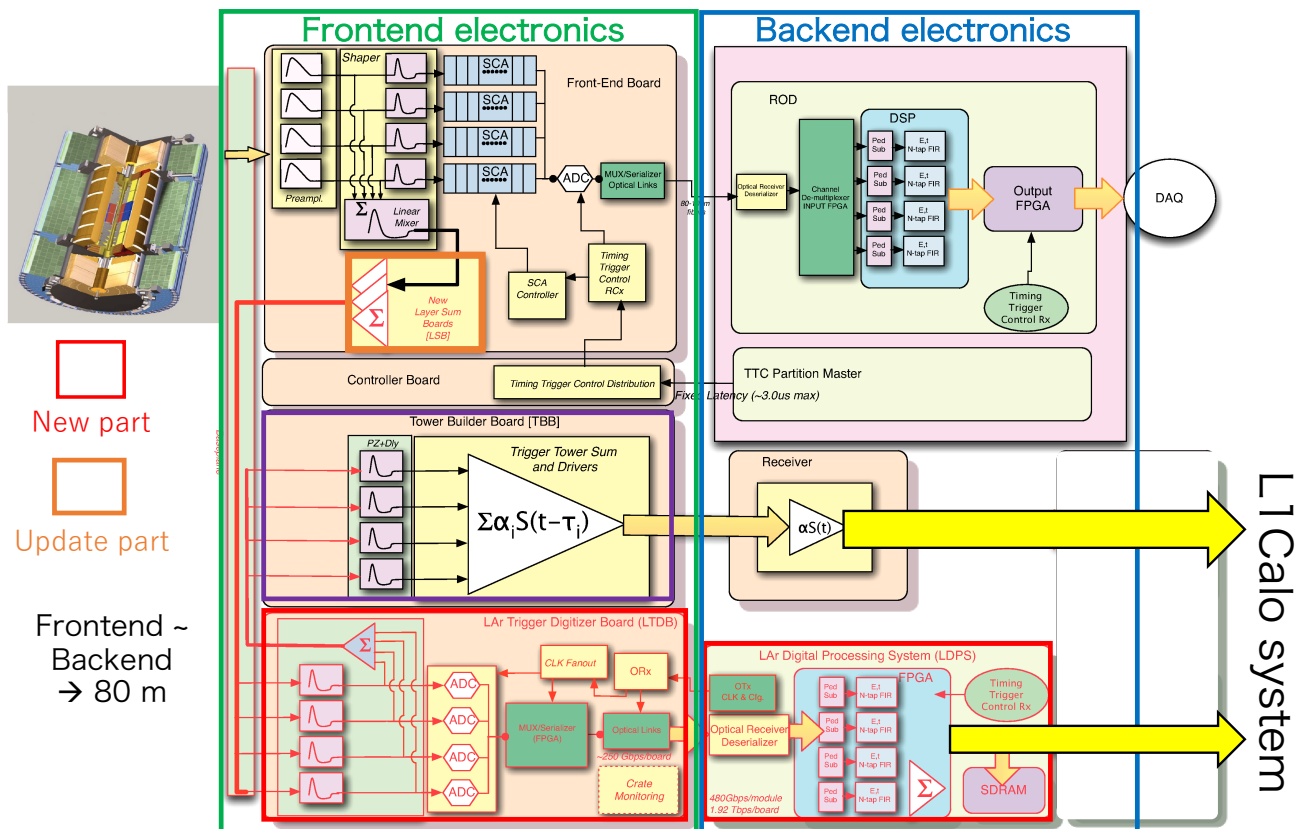


図 1.5: LAr カロリメータの読み出し経路 [1]。緑で囲まれた部分がフロントエンド、青で囲まれた部分がバックエンドエレクトロニクスである。フロントエンドからバックエンドまでは、約 80 m 離れている。紫で囲まれた部分が、Tower Builder Board であり、アナログ信号を L1Calo へ伝送している。アップグレード後、赤く囲まれた LTDB と LDPS が新しく導入される。

本研究の目的は、LDPS で主要な役割を担うエネルギー計算の FPGA ファームウェアを開発することである。私は、3章で説明するフィルタリングアルゴリズムを用いて、全 Super Cell のエネルギーを計算できるファームウェアの考案、設計、ハードウェアでの動作試験方法および試験を一から全て主導的に行った。本研究で開発したファームウェアは、ATLAS の要求するレイテンシや FPGA のリソースの要求値を満たしている。また、LDPS で使用するボードを使用した試験も行った。このファームウェアは実際に、新しい読み出し対応のファームウェアの一部に組み込まれている。

1.3 本論文の構成

本論文では、第 2 章で LHC ATLAS 実験について述べる。第 3 章では、本研究の核である液体電磁カロリメータの読み出しについて述べる。第 4 章では、LTDB, LDPS のデモンストレータボードを使用した Super Cell 信号の収集について説明する。第 5 章は、デモンストレータボードで得られた Super Cell 信号を基にした FPGA ファームウェアの開発について纏めている。最後に、6 章で纏めと今後についてを述べる。

第2章 LHC-ATLAS実験

LHCは、スイスのジュネーブ郊外の地下50~175 mの場所に位置し、円周約27 kmの加速器である(図2.1)。LHCは4つの衝突点があり、それぞれの衝突点の中にA Toroidal LHC ApparatuS (ATLAS), Compact Muon Solenoid (CMS), A Large Ion Collider (ALICE), LHCb検出器が設置されている。日本はATLAS実験に参加しているため、本論文はATLAS検出器に焦点を当てて説明する。

2.1 LHC, ATLASについて

ATLAS実験は陽子と陽子を衝突させている。2011年に7 TeV、2012年に8 TeVの重心系エネルギーをもったRun1実験が行われた。2015年からは13 TeVでRun2実験が行われており、2018年まで続ける。このような高いエネルギーでの実験が可能な理由として、衝突に使われている陽子が他の粒子に比べて容易にエネルギーを上げられるからである。これにより、前人未達のエネルギー領域で新しい物理現象を探索できる。しかし、陽子陽子衝突は、パイルアップ事象が多いため注意が必要である。パイルアップにはin time pileupとout of pileupの2種類ある。In time pileupは同じバンチ交差中に生じる別の衝突由来の粒子、out of pileupは異なるバンチ交差で起きた衝突由来の粒子をさす。1回の衝突で起きる平均パイルアップ数は、陽子の非弾性散乱断面積と衝突頻度から推定できる。

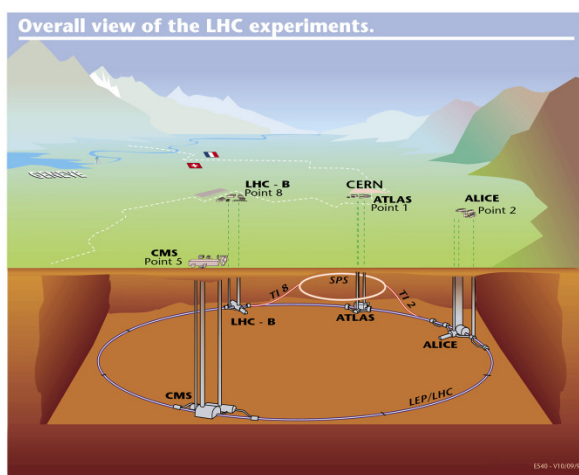


図 2.1: LHC 加速器 [2]

2.2 ATLAS 検出器概要

ATLAS 検出器は、長さ 44 m、高さ 22 m、重さが約 7000 トンある円形構造を持った大型汎用検出器であり、大きく 3 つの検出器に分けられる (図 2.2)。ATLAS は、検出器の衝突点を原点とした右手系を使用している。x 軸方向は衝突点から LHC リング中心方向、y 軸方向は垂直方向、z 軸はビームパイプに沿った方向と定義されている。円柱座標系 (r, ϕ) は xy 平面で使用され、r は衝突点からの動径方向、 ϕ はビームパイプまわりの方位角と定義される。また、擬ラピディティ η は $\eta = -\ln \tan(\frac{\theta}{2})$ と定義される。

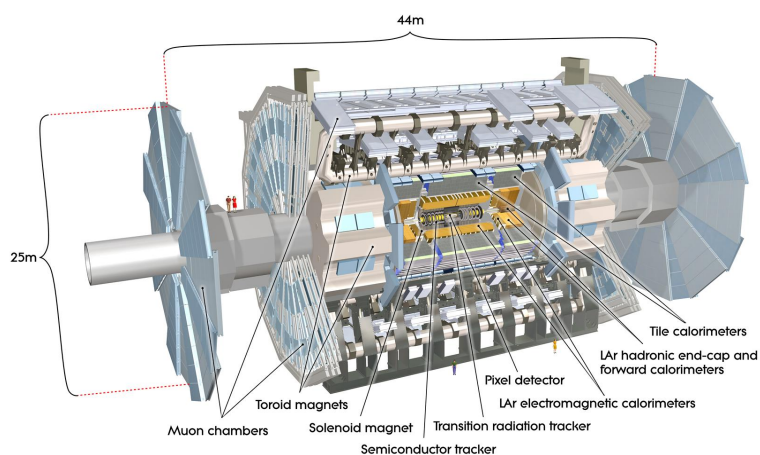


図 2.2: ATLAS 検出器 [2]

2.2.1 内部飛跡検出器

内部飛跡検出器は、シリコンピクセル検出器 (Pixel), シリコンストリップ検出器 (SCT), 遷移輻射検出器 (TRT) で構成される。

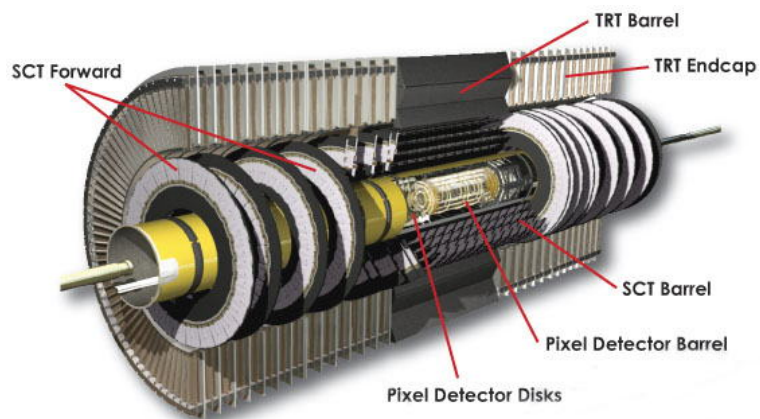


図 2.3: 内部飛跡検出器 [2]

シリコンピクセル検出器

衝突点に一番近いこの検出器は4層構造を持ち、それぞれピクセル化されている。最内層以外のピクセル検出器は $|\eta| \leq 2.5$ まで覆っており、そのサイズはビーム軸方向とその垂直方向に対して $400 \times 50 \mu\text{m}^2$ である。最内層にあるピクセル検出器はInsertable b-Layer (IBL)と呼ばれ、 $r=33 \text{ mm}$ の位置にある。IBLは $|\eta| \leq 3.0$ まで覆っており、そのサイズは $250 \times 50 \mu\text{m}^2$ である。これらは非常に細かく分割されているため、1ピクセルあたりのヒット占有率が減る上に位置分解能も良い。これにより、bメソンや τ 粒子特有の2次崩壊点を検出できる。

シリコンストリップ検出器

この検出器は、シリコンピクセル検出器の外側に4層配置されている。図2.4はパレル部のSCTモジュールで、ストリップ間隔が $80 \mu\text{m}$ 程度、ストリップ長が 128 mm である。768本のストリップを埋め込んだセンサーを 40 mrad つけて表裏に張り合わせることで、2次元座標点を得られる。

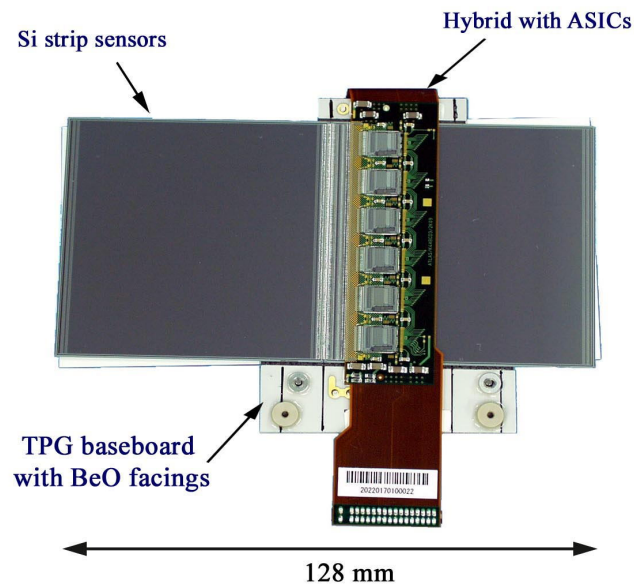


図 2.4: パレル部の SCT モジュール [2]

遷移輻射検出器

内部飛跡検出器の一番外側にあるこの検出器は、ストロー型ドリフトチューブの3層構造であり、 $\eta \leq 2.0$ を覆っている。直径 4 mm 程度のストローが並べられおり、その間にキセノン、酸素、二酸化炭素の混合ガスが充満している。荷電粒子の通過によって、キセノンガス中で粒子の電離が見られる。また、真空と混合ガス中の誘電率の違いからX線輻射が発生する。輻射の全光量はローレンツ因子に比例するため、質量の軽い電子が入射する

場合はこの X 線を検出できる。一方、 π^\pm が入射する場合は遷移輻射はほとんど発生しない。そのため、電子の π^\pm の分布が異なる。

2.2.2 カロリメータ

ATLAS 検出器は、電磁カロリメータとハドロンカロリメータの 2 つを用いている。どちらの検出器も測定器中の物質との相互作用によってカスケードシャワーを発生させ、そのエネルギーを測定する (図 2.5)。

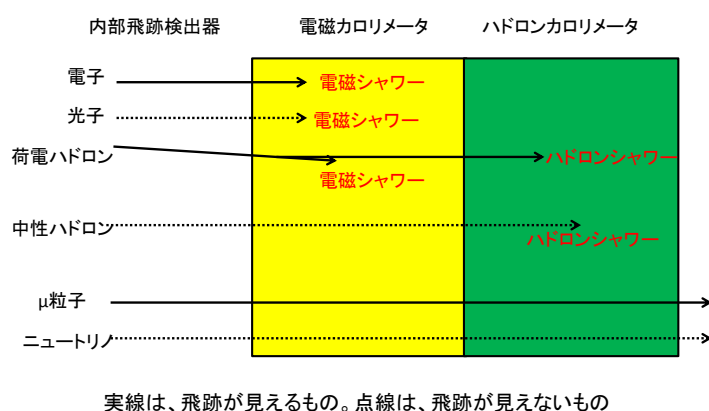


図 2.5: それぞれの粒子に対してのカロリメータの応答。 μ 粒子とニュートリノの反応の違いは、カロリメータの後ろにある μ 粒子検出器で確認できる。

電磁カロリメータ

ATLAS 電磁カロリメータは、鉛の吸収層と液体アルゴンの検出層を用いたサンプリング型カロリメータを採用しており、 $|\eta| \leq 3.2$ を覆っている。アコーディオン構造をとることで、 ϕ 方向の不感領域がなくなる (図 2.6)。電子や光子が入射すると、鉛の吸収層で電磁シャワーを起こす。電磁シャワーで生成された電子 (陽電子) や光子は、吸収層で検出される。液体アルゴンは荷電粒子の数と応答曲線の間で良い線型性をもつため、エネルギーを求めることができる。また、液体アルゴンは十分な放射線耐性をもっている。

液体アルゴン電磁カロリメータは、第 0 層から第 3 層までの 4 層の構造をとっている (図 2.7)。第 0 層は $|\eta| \leq 1.8$ にのみ設置されており、内部飛跡検出器やソレノイド等で電磁シャワーを起こした事象のエネルギー補正を行う。第 1 層は η 方向に細かく分割されている。 π^0 は物質と相互作用して $\pi^0 \rightarrow \gamma\gamma$ と崩壊するため、この層で単一の光子と π^0 の識別ができる。第 2 層は奥行きが長いので、電磁シャワーのほとんどのエネルギーがこの層でおとす。第 3 層は、高エネルギー電子を検出する。それぞれの分割領域は、表 1.1 に記述している。

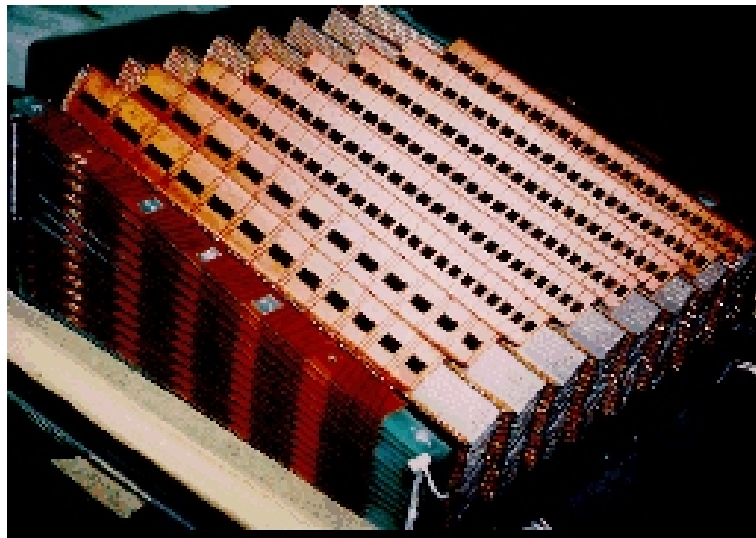


図 2.6: ATLAS 電磁カロリメータのアコーディオン構造 [3]

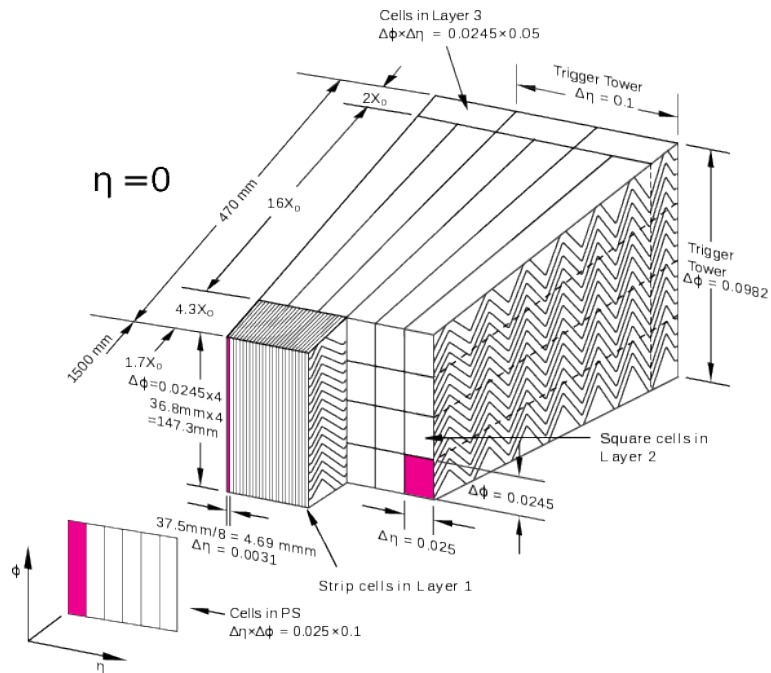


図 2.7: バレル部 ($\eta=0$) における LAr カロリメータの層構造 [3]

ハドロンカロリメータ

ハドロンカロリメータもサンプリング型カロリメータを採用しており、電磁カロリメータの外側に配置される。吸収層と検出層に使用する物質は、 η によって異なる。 $|\eta| \leq 1.7$ は、鉄とプラスチックシンチレータ、 $1.7 \leq |\eta| \leq 3.2$ は銅と液体アルゴンを使用している。

フォワードカロリメータ

フォワードカロリメータは、 $3.2 \leq |\eta| \leq 4.9$ の領域に設置され、3層構造をもつ。1層目に電磁カロリメータ (銅と液体アルゴンのサンプリング型)、2層目と3層目にハドロンカロリメータ (銅、タングステンと液体アルゴンのサンプリング型) が設置されている。また、フォワードカロリメータはビームパイプに近いので、アコーディオン構造でなくストロー構造が採用されている (図 2.8)。

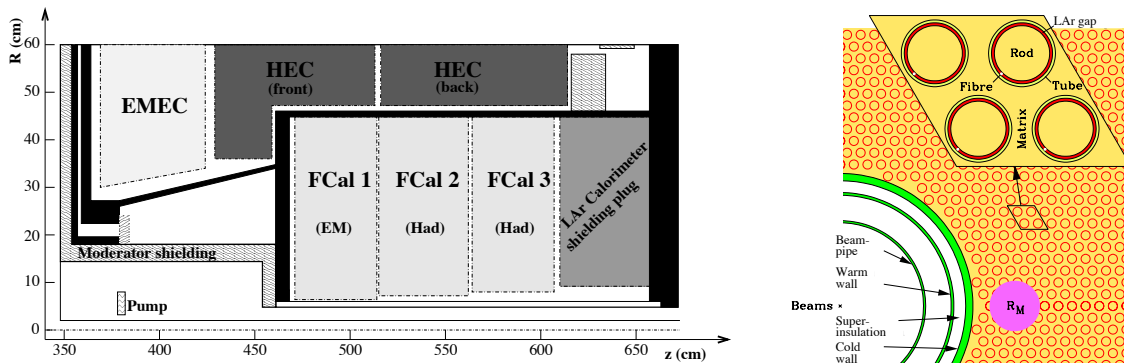


図 2.8: エンドキャップ領域でのカロリメータの配置 (左) とフォワードカロリメータのストロー構造 (右)[4]

2.2.3 ミューオンスペクトロメータ

μ 粒子の質量を考えると、制動放射の断面積が電子よりも十分に小さくなるため、 μ 粒子は電磁カロリメータで止まらず通過する性質をもつ。そのため、ミューオンスペクトロメータは ATLAS 検出器の一番外側に配置されており、Big Wheel (BW) と Small Wheel (SW) で構成されている。BW と SW はそれぞれ精密測定用とトリガー判定用の検出器をもっており、その検出器はバレル部とエンドキャップ部で異なる。

Monitored Drift Tube (MDT)

精密測定用に使用されるこの検出器は、 $\eta \leq 2.0$ の領域に配置され 3 層構造を持っている (図 2.9)。最内層は 8 つのドリフトチューブで、残りの層は 6 つのドリフトチューブで構成される。ドリフトチューブの直径は約 30 mm、その位置分解能は $80 \mu\text{m}$ である。また、アルゴンと二酸化炭素を混合したガス (ArCO_2) を使用している。

Cathod Strip Chamber (CST)

CST は $|\eta| \geq 2.0$ の領域に設置される高ヒットレート環境下で動作する精密測定用検出器で、MWPC を使用している。ドリフト時間が MDT よりも短く (30 ns 以下)、高レート耐性があり、位置分解能は約 $60 \mu\text{m}$ である。

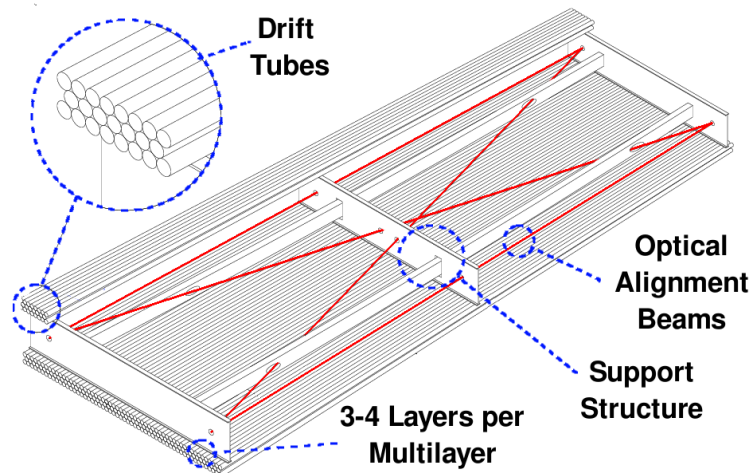


図 2.9: MDT の構造 [5]

Resistive Plate Chamber (RPC)

RPC はバレル用のトリガー検出器で、 $|\eta| \leq 1.05$ の領域に設置される。2 mm 程度の薄いガス層を平面電極で挟んでおり、時間分解能は 1.5 ns 程度である。また、10 mm 程度の位置分解能をもっている。高抵抗の電極を使用しているため、大きな電流が流れても一時的な電圧降下によって放電を抑制できる。

Thin Gap Chamber (TGC)

TGC は、エンドキャップ用のトリガー検出器 ($1.05 \leq |\eta| \leq 2.7$) である。直行しているアノードワイヤーとカソードストリップを用いることで、2次元の位置情報を読み出すことが可能である。電極間が狭いため、時間分解能が 4 ns 程度ある。また、位置分解能は 5 mm 程度である。

2.3 トリガーシステム

ATLAS 実験は 40 MHz (25 ns) で衝突し、その事象はトリガーシステムで記録するかを判断される。ATLAS は 2 つのトリガーシステムをもつ (図 2.10)。

Level1 トリガー (L1 トリガー)

1 段目にあるトリガーシステムは L1 トリガーと呼ばれ、カロリメータとミュオン検出器の情報のみを使用したハードウェアベースで行われる。衝突が起きてから $2.5 \mu\text{s}$ で取捨選択するので、検出器の大まかな情報を用いた粗い選別を行い、最終的に 100 kHz 程度レートにする。この大まかな情報は ROI (Regions of Interest) と定義され、後段の High Level トリガーに送られる。

High Level トリガー (HLT)

L1 トリガーで選別された事象に対し、ソフトウェアベースでよりオフラインに近い選別を行う。また、HLT では、L1 トリガーで使用されなかった飛跡検出器の情報も使用する。約 550 ms 処理時間で、1 kHz 程度のトリガーレートで記録する。

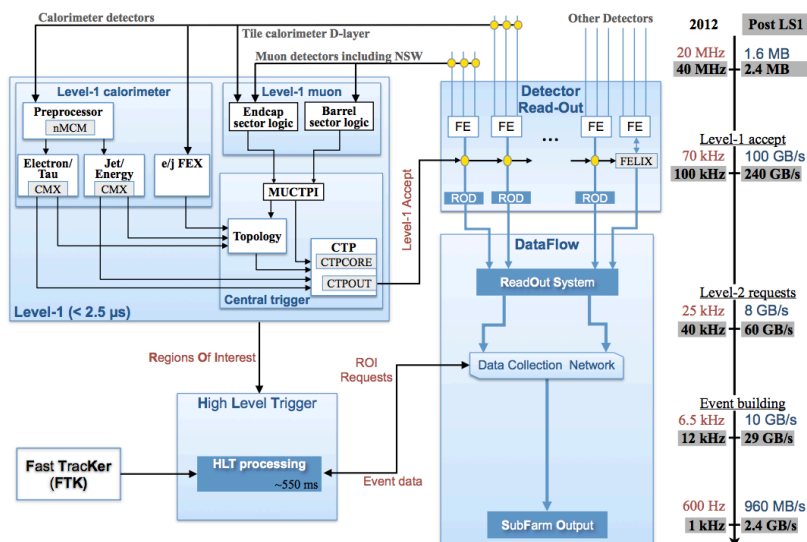


図 2.10: Run2 のトリガーシステムの流れ [6]。図中の 2012 は Run1、PostLS1 は Run2 を表している。

2.4 TTC システム

トリガーが通った事象の情報を正しく記録するため、LHC、各検出器間の同期をとる。そのための信号を扱うシステムが、TTC である。TTC は L1 トリガーの判定を行い、各検出器にその判定信号である Level1 accept 信号 (L1A) とどの条件で L1A が発行されたかを示す Trigger Type 信号を分配する。また、LHC のビームはバンチ構造をとっており、計 3564 バンチある。それぞれのバンチを識別するために、BCID (Bunch Crossing Identification) 信号も分配する。

2.5 ATLAS アップグレード計画

Run2 実験が終わると、設計値が重心系エネルギー 14 TeV、瞬間ルミノシティ $3 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ の Run3 実験に向けて、約 2 年間の Long shutdown 期間にはいる。この期間を Phase-1 アップグレードと呼び、Run3 実験に向けて新しい検出器の導入や読み出し回路のアップグレードが行われる。液体アルゴン電磁カロリメータでは検出器自体の入れ替えは行わず、読み出しエレクトロニクスを改善を施し入れ替える。これにより、より高輝度化する Run3 実験でも質の良いデータ収集ができると期待されている。ミュオンスペクトロメータでは、SW を新しい検出器 (New Small Wheel) に交換する。これにより、フェイ

クミュオン抑制効率が改善される。2026年からは、瞬間ルミノシティ $5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ の予定である High-Luminosity LHC (HL-LHC) を行う。これは約 10 年間稼働し、 3000 fb^{-1} の統計をためることを目標としている。そのためのアップグレード期間として、2024年からの約 2 年間の Long shutdown 期間で内部飛跡検出器の入れ替えなどが、行われる予定である (図 2.11)。

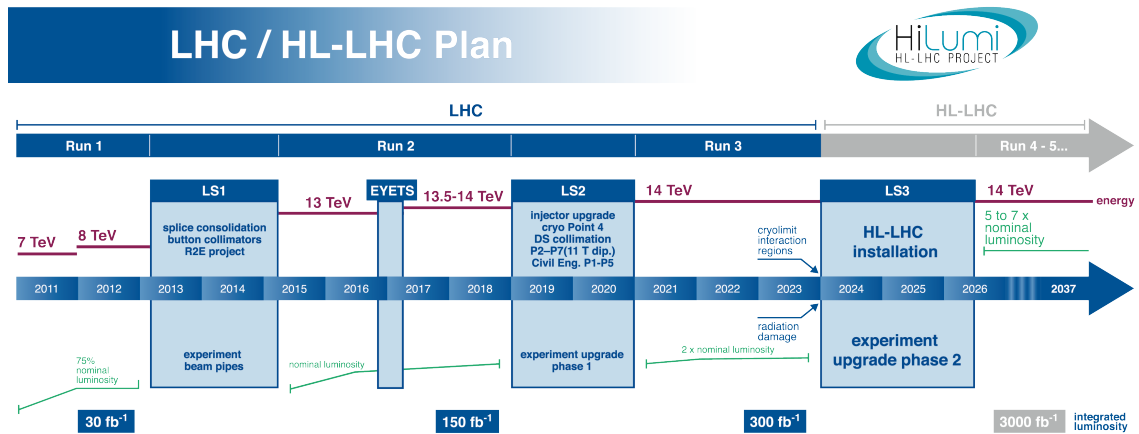


図 2.11: LHC のアップグレード計画 [7]

表 2.1: ATLAS 実験の予定

	Run1	Run2	Run3	HL-LHC
実験期間 [年]	2011-2012	2015-2018	2021-2023	2026-2036
重心系エネルギー [TeV]	7, 8	13 (14)	14	14
最大瞬間ルミノシティ [$\text{cm}^{-2} \text{ s}^{-1}$]	0.7×10^{34}	1.33×10^{34}	3.0×10^{34}	5.0×10^{34}
積分ルミノシティ [fb^{-1}]	25	100	300	3000
バンチ間隔 [ns]	50	25	25	25
平均パイルアップ数 [個]	10-35	25	80	140

第3章 LArカロリメータアップグレード

この章では、現在のLArカロリメータの読み出しとそのアップグレードの概要を述べる。

3.1 現在の読み出し方法

臨界エネルギー以上の光子や電子がカロリメータに入射する場合、ほとんど対生成 (光子の場合) と制動放射 (電子の場合) プロセスが起きる。これは、その粒子の臨界エネルギーに到達するまで引き続き起こるため、電磁シャワーとなる。図 3.1 中の鉛は、このプロセスを起こすために使用される。また、2つの鉛の間に、2 mm 程度の液体アルゴン (LAr) と3本の電極がある。電磁シャワー内の電子や陽電子がLArを通過する際に、液体アルゴン中の原子を電離させる。電離して生成された電子は、2kV程度に印加された電極に近づく。電離電子は液体アルゴン中をドリフトするため、最終的に600 ns程度の三角波信号が得られる (図 3.2)。また、この三角波信号の波高は電離した電子の数と比例関係にあるため、波高からLAr中で損失した粒子のエネルギーが求められる。

このようなサンプリング型は全吸収型に比べ、エネルギー分解能は劣るがセグメントを自由に分割できる。これにより良い位置分解能が得られ、電磁シャワーを2次元的に読み出せる。

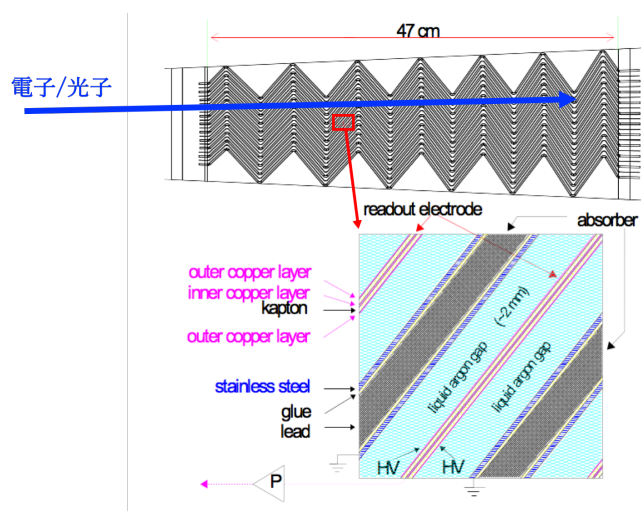


図 3.1: 液体アルゴンカロリメータの構造 [8]。衝突から生成された電子 (光子) は、鉛を通過する際に電磁シャワーを起こす。電磁シャワーで生成された荷電粒子は、鉛の間にあるLAr中の原子を電離させる。生成された電離電子は2 kV程度に印加された電極に近づくため、電圧の変化が測定できる。

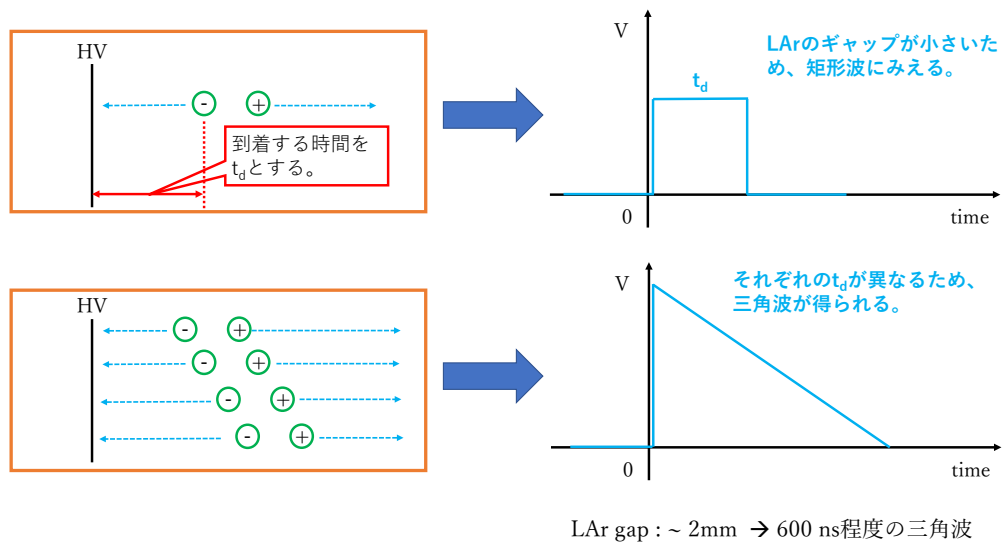


図 3.2: カロリメータから得られる信号。1 個の電子が電極に近づく時、電圧は矩形波になる (上)。矩形波の長さは、電子のドリフト時間に依存する。一方、電離した電子が複数ある時は三角波になる (下)。その理由は、それぞれの電子のドリフト時間が異なるからである。また、三角波の波高は電離電子の数に比例するため、波高から損失エネルギーが求められる。

3.1.1 フロントエンドエレクトロニクス

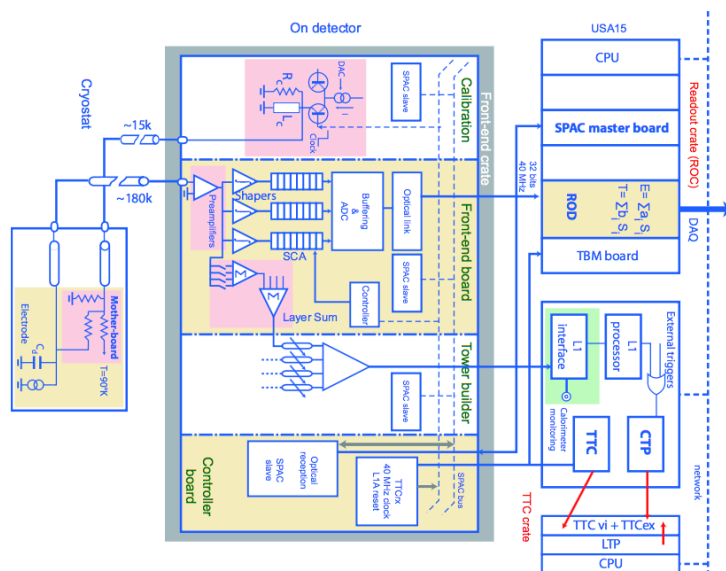


図 3.3: 現在の液体アルゴンカロリメータの読み出しシステム [1]。図 1.5 中から、現在の読み出しシステムだけを抜き出している。

図 3.3 は、現在の読み出しシステムを示している。Front-end board (FEB) に入った信号は、最初にプリアンプで増幅される。増幅された信号はバイポーラフィルタ $CR - (RC)^2$

をかけ、信号とノイズ比が良いバイポーラ波形にする (図 3.4)。バイポーラ波形にすることで、信号の重なり具合に依らずベースラインは常に一定になる。また、バイポーラ波形の波高は、三角波の波高に比例している。そのため、エネルギーの算出はバイポーラ波形の波高を測定すればよい¹。また、時定数は 13 ns に設定している。これは、熱ノイズとパイルアップの寄与の重ね合わせが最小になるように選ばれている。前者は時定数が大きいほど減少し、後者は時定数が小さいほど減少する。システムの経路は 2 つあり、メイン経路の他にトリガー用経路がある。トリガー用経路において、信号は FEB から Tower Builder Board (TBB) へ送られる。TBB では Trigger Tower 信号にして、アナログ信号のまま L1Calo システムへ送る。メイン経路にいく信号は、Switched Capacitor Array (SCA) で 40 MHz でサンプルされる。その後、L1 トリガーからアクセプトされた事象に対してアナログデジタル変換 (AD 変換) を行い、光ファイバーでバックエンドへ送る。

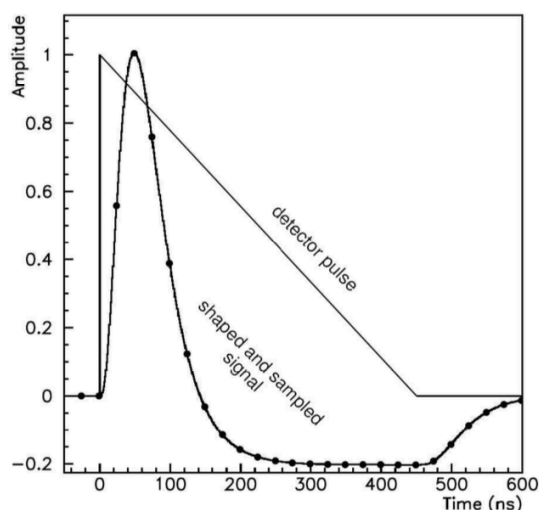


図 3.4: 検出器から得られる信号とシェイピング後の信号 [8]

3.1.2 バックエンドエレクトロニクス

バックエンドは L1 アクセプト信号を受信し、Digital Signal Processors (DSP) を用いたデジタル信号処理で各セルに落としたエネルギーと正しい時間を求める。その算出方法は、Optimal filter と呼ばれるフィルタリングアルゴリズムを使用する。

Optimal filter の原理

カロリメータから送られる信号は規格化された理想的な信号 g の相似形であり、エネルギーの大きさと波高の大きさは比例している。

¹LHC の衝突頻度 (25 ns) に比べて LAr 信号は非常に長い (600 ns 程度)。つまり、別のバンチから生成された信号 (outof pileup) の波形と重なり合った信号が測定される。バイポーラ波形にすることで、ベースラインが outof pileup に依らず常に一定となり、波高とエネルギーの線型性も保たれる。

得られた信号 S の波高を A とすると時刻 t_i の入力信号は、

$$S_i = Ag(t_i - \tau) \quad (3.1)$$

$$\cong Ag(t_i) - A\tau \frac{dg(t)}{dt} \Big|_{t=t_i} + n_i \quad (3.2)$$

$$= Ag(t_i) - A\tau \dot{g}(t_i) + n_i \quad (3.3)$$

となる。 τ は、理想的な信号からの位相のずれを表す。第 2 項では、 S を τ の 1 次まで Taylor 展開した。その際に、2 次以上の影響は小さいものとして無視し、熱ノイズとパイルアップノイズの影響をあらわす n_i を付加した。ここで、係数 a_i と b_i を用いて、

$$u = \sum_{i=0}^{N-1} a_i S_i \quad (3.4)$$

$$v = \sum_{i=0}^{N-1} b_i S_i \quad (3.5)$$

を定義する。まず、多数回の試行において、(3.4) と (3.5) の期待値は

$$\langle u \rangle = \sum_{i=0}^{N-1} \{A \langle a_i g(t_i) \rangle - A\tau \langle a_i \dot{g}(t_i) \rangle + \langle a_i n_i \rangle\} \quad (3.6)$$

$$\langle v \rangle = \sum_{i=0}^{N-1} \{A \langle b_i g(t_i) \rangle - A\tau \langle b_i \dot{g}(t_i) \rangle + \langle b_i n_i \rangle\} \quad (3.7)$$

となる。多数回試行において、ノイズ以外は固定の値であると仮定する。また、ノイズの期待値は 0 ($\langle n_i \rangle = 0$) になる。これらを利用すると、

$$\langle u \rangle = A \sum_{i=0}^{N-1} a_i g(t_i) - A\tau \sum_{i=0}^{N-1} a_i \dot{g}(t_i) \quad (3.8)$$

$$\langle v \rangle = A \sum_{i=0}^{N-1} b_i g(t_i) - A\tau \sum_{i=0}^{N-1} b_i \dot{g}(t_i) \quad (3.9)$$

となる。 $\langle u \rangle = A$, $\langle v \rangle = A\tau$ と定義すると、以下の係数の条件が得られる。

$$\sum_{i=0}^{N-1} a_i g(t_i) = 1 \quad (3.10)$$

$$\sum_{i=0}^{N-1} a_i \dot{g}(t_i) = 0 \quad (3.11)$$

$$\sum_{i=0}^{N-1} b_i g(t_i) = 0 \quad (3.12)$$

$$\sum_{i=0}^{N-1} b_i \dot{g}(t_i) = -1 \quad (3.13)$$

次に、この条件を利用して u と v の分散を計算する。

$$\text{Var}(u) = \langle u^2 \rangle - \langle u \rangle^2$$

$$\begin{aligned}
&= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \langle \{Aa_i g(t_i) - A\tau a_i \dot{g}(t_i) + a_i n_i\} \{Aa_j g(t_j) - A\tau a_j \dot{g}(t_j) + a_i n_j\} \rangle \\
&- \langle \sum_{i=0}^{N-1} \{Aa_i g(t_i) - A\tau a_i \dot{g}(t_i) + a_i n_i\} \rangle \langle \sum_{j=0}^{N-1} \{Aa_j g(t_j) - A\tau a_j \dot{g}(t_j) + a_i n_j\} \rangle \\
&= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} a_i a_j \langle n_i n_j \rangle = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} R_{ij} \tag{3.14}
\end{aligned}$$

同様に、

$$\text{Var}(v) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b_i b_j \langle n_i n_j \rangle = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b_i b_j R_{ij} \tag{3.15}$$

R_{ij} は、ノイズの自己相関関数である。ここで、Optimal filter は $\text{Var}(u)$ と $\text{Var}(v)$ を最小にする係数 a_i と b_i を要求する。これは、ノイズの影響を最小限にできる。Lagrange 乗数 $\lambda, \kappa, \mu, \rho$ とし、

$$I_u = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} R_{ij} a_i a_j - \lambda \left(\sum_{i=0}^{N-1} a_i g(t_i) - 1 \right) - \kappa \sum_{i=0}^{N-1} a_i \dot{g}(t_i) \tag{3.16}$$

$$I_v = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} R_{ij} b_i b_j - \mu \sum_{i=0}^{N-1} b_i g(t_i) - \rho \left(\sum_{i=0}^{N-1} b_i \dot{g}(t_i) + 1 \right) \tag{3.17}$$

と定義する。Lagrange 未定乗数法から、 $\frac{\partial I_u}{\partial a_i} = 0$ と $\frac{\partial I_v}{\partial b_i} = 0$ が成立する。これを用いると、以下の表式が得られる。

$$\sum_{j=0}^{N-1} R_{ij} a_j = \lambda g(t_i) - \kappa \dot{g}(t_i) \tag{3.18}$$

$$\sum_{j=0}^{N-1} R_{ij} b_j = \mu g(t_i) - \rho \dot{g}(t_i) \tag{3.19}$$

V を行列 R の逆行列とすると、係数 a と b の行列式は以下で書ける。

$$\mathbf{a} = \lambda \mathbf{V} \mathbf{g} + \kappa \mathbf{V} \dot{\mathbf{g}} \tag{3.20}$$

$$\mathbf{b} = \mu \mathbf{V} \mathbf{g} + \rho \mathbf{V} \dot{\mathbf{g}} \tag{3.21}$$

また、式 (3.20) と式 (3.21) の両辺に左から $\mathbf{g}^T, \dot{\mathbf{g}}^T$ をかけると、4つの未定乗数が算出される。式 (3.10) ~ 式 (3.12) が算出する時に利用される。

$$\lambda \mathbf{g}^T \mathbf{V} \mathbf{g} + \kappa \mathbf{g}^T \mathbf{V} \dot{\mathbf{g}} = \mathbf{g}^T \cdot \mathbf{a} = 1 \tag{3.22}$$

$$\mu \mathbf{g}^T \mathbf{V} \mathbf{g} + \rho \mathbf{g}^T \mathbf{V} \dot{\mathbf{g}} = \mathbf{g}^T \cdot \mathbf{b} = 0 \tag{3.23}$$

$$\lambda \dot{\mathbf{g}}^T \mathbf{V} \mathbf{g} + \kappa \dot{\mathbf{g}}^T \mathbf{V} \dot{\mathbf{g}} = \dot{\mathbf{g}}^T \cdot \mathbf{a} = 0 \tag{3.24}$$

$$\mu \dot{\mathbf{g}}^T \mathbf{V} \mathbf{g} + \rho \dot{\mathbf{g}}^T \mathbf{V} \dot{\mathbf{g}} = \dot{\mathbf{g}}^T \cdot \mathbf{b} = -1 \tag{3.25}$$

カロリメータのエネルギーを求めるためには、Optimal filter の係数が必要である。全ての領域に対してキャリブレーションパルスから理想的な信号 g とノイズの自己相関関数

R_{ij} を測定し、それらから係数を算出する。Run2 のバックエンドでは、 $N=4$ の Optimal filter を実装している。これ以外にも、Run3 以降の実装を目指して、過去の点を使用した 24 点の Extended Optimal filter 等の研究開発も行われている。

3.1.3 L1Calo トリガーシステム

TBB からアナログの Trigger Tower 信号が送られてくる。このアナログ信号は、Pre-processor Module において 40 MHz のサンプリングでデジタル変換される。その後、有限インパルス応答 (FIR Filter) と Look-Up-Table (LUT) を用いて、デジタル化された信号からトリガー用のエネルギーが算出される。算出されたエネルギーは、Cluster Processing Module (CPMs) と Jet Energy Modules (JEMs) に送られ、電子 (光子) や τ 粒子等を特定する (図 3.5)。

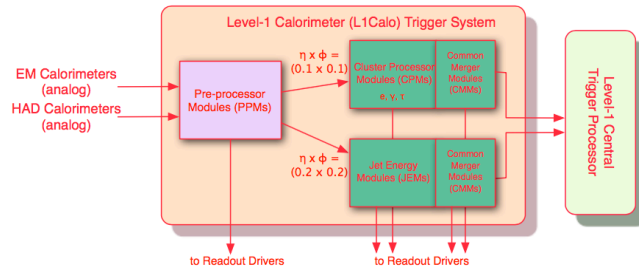


図 3.5: L1Calo トリガーシステム [1]

3.2 アップグレード後読み出し方法

Trigger Tower から Super Cell の読み出しへ変更するために、LAr カロリメータ用と L1Calo 用のエレクトロニクスが新しく導入される。また、フロントエンドに導入されている LSB の入れ替えも行われる。それ以外にも LTDB と LDPB のデモンストレータはバレル部の一部領域に導入されており、Run2 データを収集している。デモンストレータについては、4 章で詳細を述べる。

フロントエンドエレクトロニクス

- Layer Sum Boards (LSB)
- LAr Trigger Digitizer Board (LTDB)

バックエンドエレクトロニクス

- LAr Digital Processing Board (LDPB)

L1Calo エレクトロニクス

- Feature Extractor module (FEX)

3.2.1 Layer Sum Board (LSB)

Super Cell 信号を生成するために、カロリメータの第 1 層と第 2 層の信号を $\Delta\eta \times \Delta\phi = 0.025 \times 0.1$ の領域で足し合わせたアナログ信号が LTDB へ送られる。

3.2.2 LAr Trigger Digitizer Board (LTDB)

LTDB は Super Cell 信号を 12 ビット (0 ~ 4095) のフラッシュ ADC を用いて 40 MHz クロックでサンプリングし、LDPB へ光ファイバーを用いて伝送する。また、バックアップとして Run2 までと同様に TBB に Trigger Tower のアナログ信号を伝送する役割をもつ。1 つの LTDB で最大 320 個程度の Super Cell の信号を処理する必要があるため、1 本の光ファイバーは 5.12 Gbps で通信を行う (表 3.1)。

表 3.1: それぞれの領域に導入される LTDB 数とそのチャンネル数。チャンネル数は Super Cell の数に対応している。

LTDB Type	LTDB 数 per region	Channel 数 per LTDB
EMB	64	290
EMEC Std	32	312
EMEC Spec0	8	240
EMEC Spec1	8	160
HEC	8	192
FCal 0	2	192
FCal 1	2	192
Total	124	34048

デジタル化のダイナミックレンジを決める際に、デジタル化によるサチュレーションと量子化誤差の問題を考慮に入れなければならない。 E_T を算出するために、それぞれの Super Cell の $\sin\theta$ が掛けられる。これにより、サチュレーションするエネルギーは η 依存性をもつ。1 ビットの重みを全ての Super Cell で固定すると、以下の問題点が挙げられる。

- 1 ビットの重みを大きい値にとると、デジタル化によるサチュレーションは防げるが、量子化誤差が大きくなる。
- 1 ビットの重みを小さい値にとると、量子化誤差は小さくなるが、デジタル化によるサチュレーションが起き得る。

これを避けるために、1 ビットの重みを 1 mV にする。1 ビットの重みが η 依存性をもつが、各 Super Cell で適切な量子化誤差が設定できる (図 3.6)。また、増幅後の最大電圧は 3.3 V であるため、デジタル化によるサチュレーションは起きない。

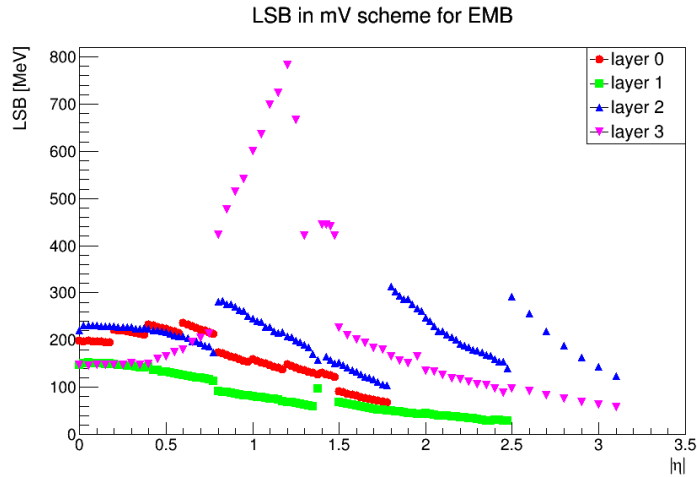


図 3.6: パレル部での 1 ビットの重み。y 軸は 1ADC による E_T の値を示している。

3.2.3 LAr Digital Processing Board (LDPB)

LDPB は光ファイバーを通じて、LTDB から約 34000 の Super Cell 信号を受け取る。その後、FPGA 上でフィルタリングアルゴリズムを適用し、L1Calo へ転送する。全ての Super Cell 信号を処理するために、31 枚の LDPB が使用される。図 3.7 は 1 枚の LDPB の簡易図である。4 枚の AMC (Advanced Mezzanine Card) がこの中に搭載されており、Super Cell 信号のエネルギー変換の役割を担っている。96 本の光ファイバー (48 本の RX、48 本の TX) が 1 枚の AMC と接続されており、LTDB や FEX との通信をする。Super Cell のエネルギーだけでなく、 $\Delta\eta \times \Delta\phi = 0.1 \times 0.1, 0.2 \times 0.2$ で足し合わせたエネルギーも送るので、FEX へ送る通信量は 50 Tbps 程度²になる。現在は、光ファイバー 1 本の転送速度を約 11.2 Gbps を目指して開発が行われている。この通信速度を達成するトランシーバーを計 96 本分搭載させる必要があるため、ALTERA (INTEL 社) からの最新バージョンである Arria10 FPGA が使用されている。

LDPB は、VME に代わり新たに使用する Advanced Telecommunications Computing Architecture (ATCA) に準拠している。図 3.7 の右半分はモニタリング用として使用されており、ATCA のゾーン 2 を介して AMC からの信号を転送する役割をもつ。

Advanced Telecommunications Computing Architecture (ATCA)

ATCA は高速通信を支えるプラットフォームであり、カロリメータで VME の代わりに新しく導入される (図 3.8)。1 台の ATCA で最大 14 枚の ATCA 規格ボードが備え付けられ、バックプレーンを介してそれらと 10 Gbps の通信ができる。バックプレーンに、ゾーン 1 からゾーン 3 と呼ばれるインターフェイスがある。

²FEX へ送るプロトコルは、シリアライズされる前に 32 ビットフォーマットになる。このデータクロックは 320 MHz のため、光ファイバー 1 本あたりに求められる最低通信速度は $320 \text{ MHz} \times 32 = 10.24 \text{ Gbps}$ である。実際に、通信を行う光ファイバーは 1 つの AMC に対して 40 本なので、 $10.24 \times 40 \times 4 \times 31 \sim 50 \text{ Tbps}$ 程度になる。

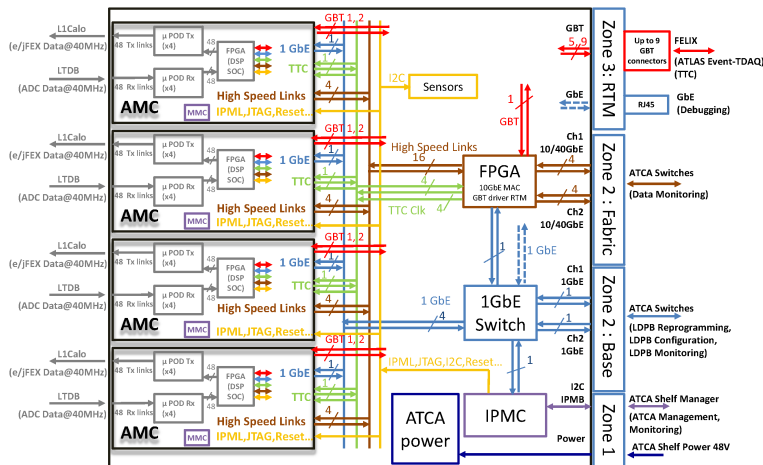


図 3.7: LDPB[1]

3.2.4 Feature Extractor module (FEX)

アップグレード期間に、FEX と呼ばれるモジュールが新しく導入され、Super Cell のエネルギーから粒子の特徴を考慮した同定を行う。FEX の中に 3 種類のサブモジュールがあり、電子や光子を同定する eFEX、ジェットや大きい消失横運動量エネルギー (MET) を同定する jFEX、そして Large-R ジェットを同定する gFEX に分かれている (図 3.9)。LDPB は、それぞれのモジュールに適したエネルギーを送る。



図 3.8: ATCA プラットフォーム [9]。この ATCA は 14 スロットあり、高さ 12 U の (~53 cm) である。

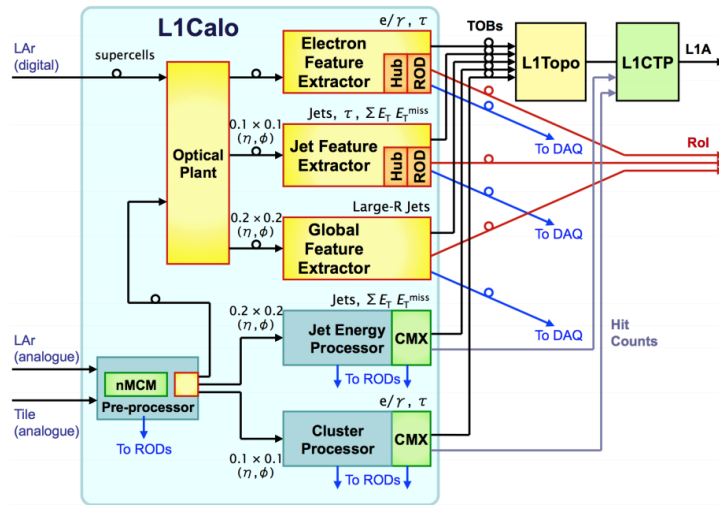


図 3.9: L1Calo アップグレードの全体図 [10]

3.3 Field Programmable Gate Array (FPGA)

この節では、LDPB の中で重要な役割をはたす FPGA について述べる。FPGA は Random Access Memory (RAM) や Look-up Table (LUT) などを組み込んでおり、ASIC と異なり製造後に回路の再設計が可能である。これによって、ハードウェアの欠点である低い柔軟性が克服され、回路制作が容易になった。近年の技術発展で高性能かつ高集積化が進んでおり、高エネルギー物理学実験でも多く使用されるようになってきた。INTEL 社の ALTERA と XILINX 社が主に FPGA を製造している。LDPB は ALTERA の FPGA を使用するため、本論文では ALTERA の FPGA に焦点をあてて説明する。

3.3.1 Adaptive Logic Module (ALM)

ALM は、LUT と加算器とレジスタで構成されている。ALTERA は、この ALM を最小単位ブロックとして実装している。ALM の構成要素や配線は ALTERA のシリーズごとに変わっている。図 3.10 と図 3.11 は、ALTERA シリーズの StratixIV と Arria10 の ALM の配線図である。ユーザーの意図に合わせて、LUT の構成を変更できる。例えば、1 つの ALM は 2 個の 3 入力の論理和と 1 個の 6 入力の論理和の両方に対応しているため、組み合わせ回路を効率的に実装できる。Arria10 の ALM はレジスタを 4 つ所有しており、結果的により多様な LUT にするための配線ができる。具体的には、Arria10 の ALM は、StratixIV でサポートされていない 7 入力の LUT を構成できる。

3.3.2 Random Access Memory (RAM)

ALTERA の FPGA は、小規模メモリブロックと大規模メモリブロックを実装しており、デザインの要求によってそれらを使い分けられる。小規模メモリブロック (~640 ビット) は MLAB と呼ばれ、大規模メモリブロックはビット数に応じて M9K, M10K, M20K な

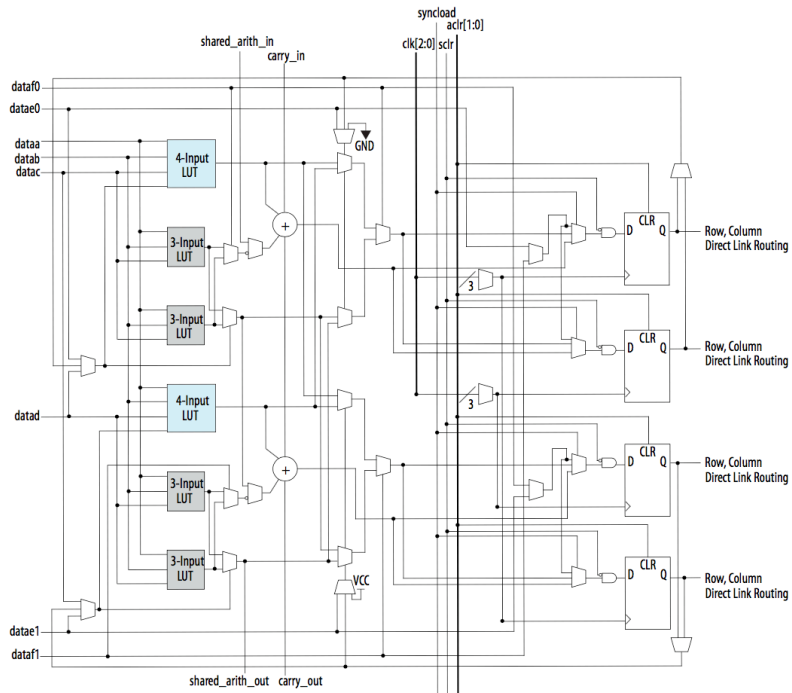


図 3.10: Arria10 の ALM の配線図 [11]。1 つの ALM にレジスタは 4 つある。

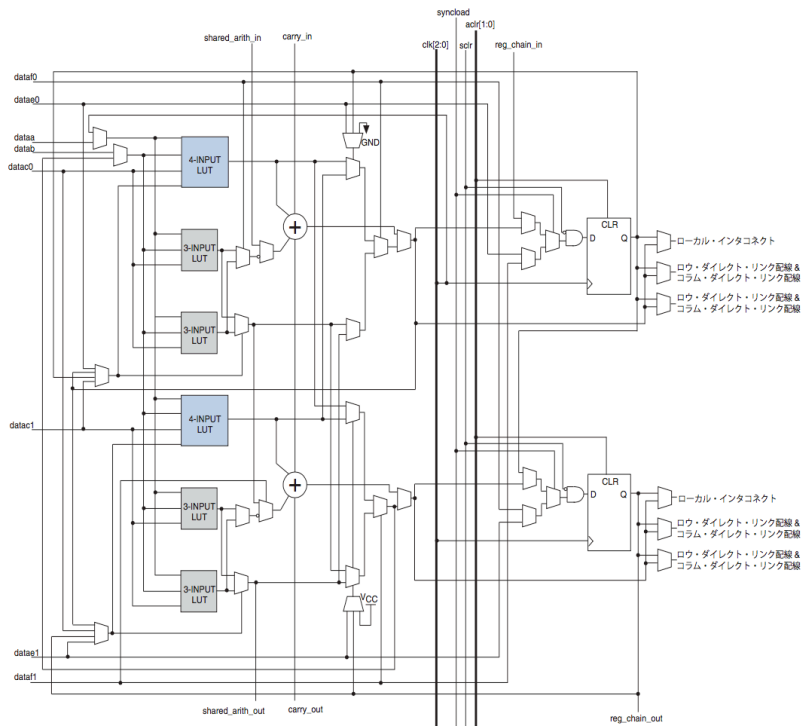


図 3.11: StratixIV の ALM の配線図 [11]。1 つの ALM にレジスタは 2 つある。

どと呼ばれる。メモリの大きさ以外にも、MLAB は出力ポートを 2 つ配線できないなど、複雑なモードに対応していない (図 3.12)。しかし、MLAB はサイズの小さい FIFO (浅い FIFO ともいう) やシフトレジスタなどの実装に最適である。大規模メモリブロックのソースは MLAB よりも限られているので、サイズの大きさが不要ないデザインに実装することは非効率である。

機能	M20K	MLAB
最大動作周波数	730 MHz	700 MHz
トータル RAM ビット数 (パリティ・ビットを含む)	20,480	640
パリティ・ビット	サポートあり	サポートあり
バイト・イネーブル	サポートあり	サポートあり
バック・モード	サポートあり	—
アドレス・クロック・イネーブル	サポートあり	サポートあり
シングル・デュアル・ポートの異なるデータ幅	サポートあり	—
トゥルー・デュアル・ポートの異なるデータ幅	サポートあり	—
FIFO バッファの異なるデータ幅	サポートあり	—
メモリ初期化ファイル (.mif)	サポートあり	サポートあり
混合クロック・モード	サポートあり	サポートあり
完全同期メモリ	サポートあり	サポートあり
非同期メモリ	—	フロースルー読み取りメモリ動作向け。
パワーアップ・ステート	出力ポートはクリアされます。	<ul style="list-style-type: none"> 登録済みの出力ポートクリア 未登録の出力ポートにメモリ内容の読み取り
非同期クリア	出力レジスタと出力ラッチ	出力レジスタと出力ラッチ
書き込み/読み取り動作のトリガ	立ち上がりクロック・エッジ	立ち上がりクロック・エッジ
同一ポートの Read-During-Write	出力ポートは"new data"または"don't care"に設定されません。	出力ポートは"don't care"に設定されます。

図 3.12: Arria10 シリーズのメモリ機能 [11]

3.3.3 Digital Signal Processor Block (DSP Block)

DSP Block は乗算専用配線されたブロックであるため、高速に乗算処理ができる。様々なフィルタに対応するために加算器も組み込まれており、デジタル信号処理のファームウェアには必須である。DSP Block 内の配線は各シリーズごとに異なるため、使用できるモードもシリーズに依存する。本論文では、多くのモードの中から本研究で使用した Arria10 シリーズの 18×19 固定小数点ストリックモード (18×19 モード) についてのみ述べる。

図 3.13 は、Arria10 シリーズの DSP Block の配線図である。18×19 モードは、1 個の DSP Block で 2 個の乗算器³を使用できる。乗算器の前に Pre-Adder 機能が備わっており、2 つの入力値の加算あるいは減算ができる。2 つの乗算結果に対しての加算器と、別

³27×27 固定小数点ストリックモードの場合は、1 つの DSP Block で 1 個しか乗算器を使用できない

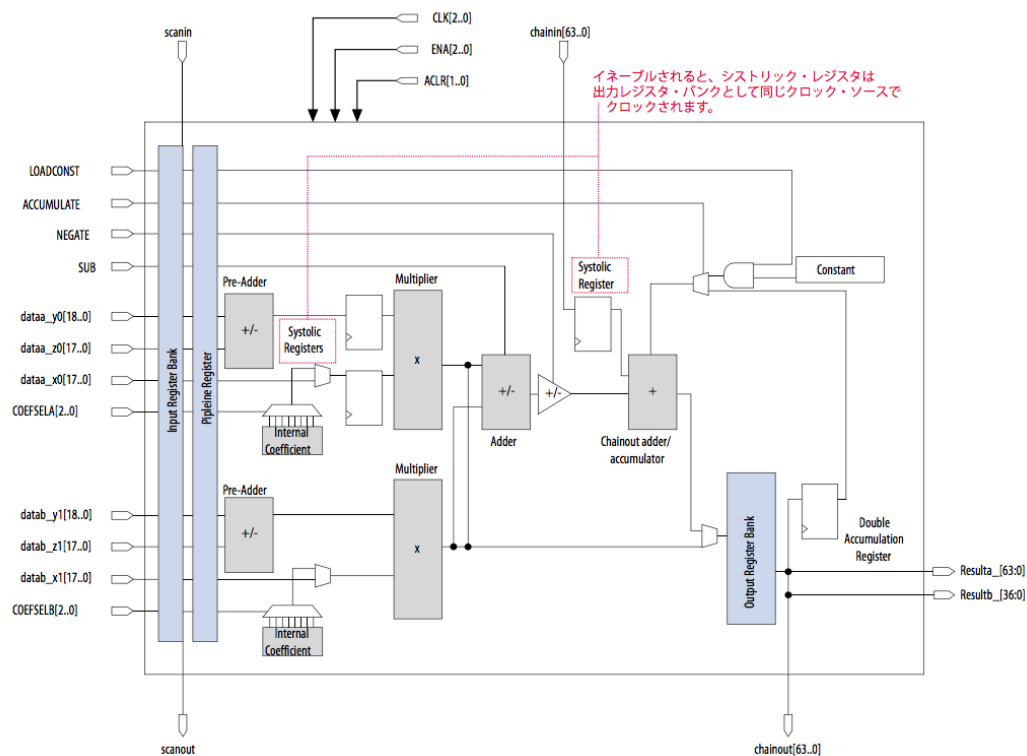


図 3.13: Arria10 シリーズの DSP Block の配線図 (18×19 モード) [11]

の DSP Block から来た値に対しての加算器が、乗算器の後に備わっている。DSP Block は 3 つのレジスタバンクをもっており、このバンクの使用数で DSP Block のレイテンシが決まる。DSP Block 内の動作安定性を最大限に引き出すためには、全てのレジスタバンクは使用されるべきである。シストリックモードを選択すると、レジスタバンク以外に 3 つのシストリックレジスタが接続される。また、隣の DSP Block からの入力ポートとして chainin 信号、隣の DSP Block への出力ポートとして chainout 信号も接続される。これにより、1 つの DSP Block で計算された値を chainout へ出力させ、別の DSP Block の chainin に入力させることで複数の DSP Block を使用したデジタル信号処理が可能である。

第4章 デモンストレータプロジェクト

LTDB と LDPB のデモンストレータがバレルの一部領域に導入されており、Run2 で Super Cell 読み出しの試験が行われている。これらは Run3 のための試験であるため、L1 トリガーに直接関わらない上に Run2 実験の DAQ システムとは独立に運用している。本研究の目的であるエネルギー計算ファームウェアを開発するために、実際の Super Cell 信号を収集することは重要である。この章では、デモンストレータプロジェクトの概要と私が研究開発に携わったバックエンドデモンストレータファームウェアについて述べる。

4.1 プロジェクト概要

このプロジェクトの大きな目的は2つある。

- Super Cell 信号の収集
- ボードのインストールや運転における試験

Super Cell 信号を収集することで、ノイズやフィルタリングアルゴリズムの評価ができる。また、問題点を知ることで、Run3 用のボードやファームウェア等の改善ができる。これらの目的のために、2014年8月から $\Delta\eta \times \Delta\phi = 1.4 \times 0.4$ の領域 ($0 \leq \eta \leq 1.4$, $1.8 \leq \phi \leq 2.2$) にデモンストレータを導入した (図 4.1)。

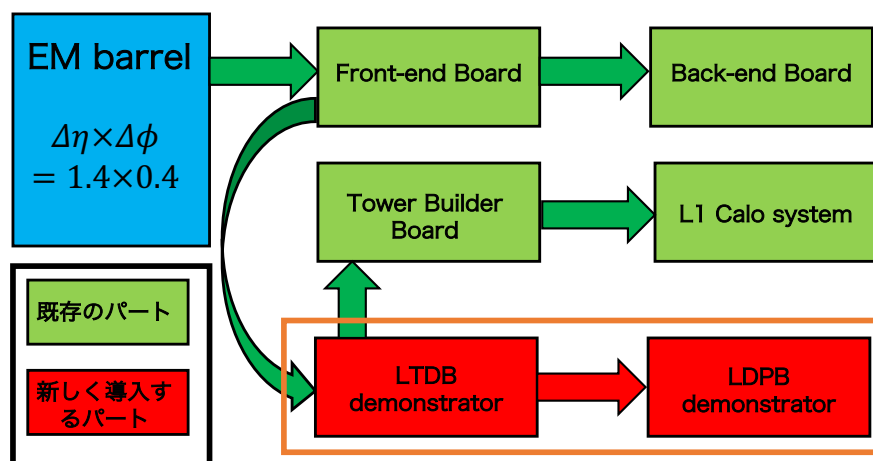


図 4.1: デモンストレータ導入後の読み出しシステム

4.2 LTDB デモンストレータ

このデモンストレータは Tower Builder Board にアナログ信号を、LDPB にデジタル化した Super Cell 信号を送る。LDPB への伝送は光ファイバーを用いており、4.8 Gbps の通信速度でシリアル伝送する。また、図 4.2 のデータフォーマットに 8b10b のエンコーディングが施される。図 4.3 は実際に導入された LTDB デモンストレータで、8 つの Super Cell 信号が含まれている。

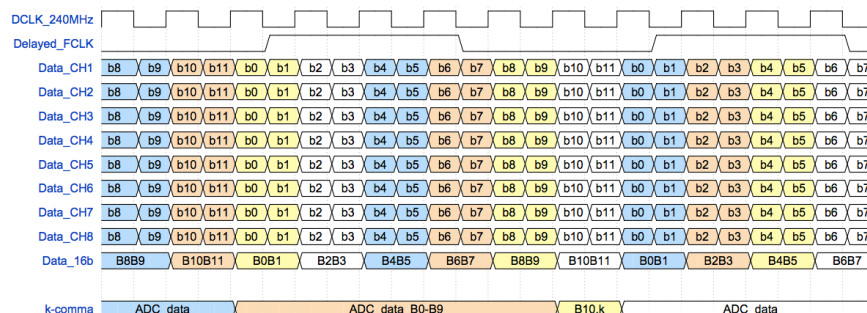


図 4.2: LTDB のデータフォーマット [12]。Data.CH1 から Data.CH8 が 8 種類の Super Cell 信号に対応している。Super Cell 信号は 12 ビットで送られ、b0 から b11 がそれぞれに対応している。一番下の k-comma 信号は 8 ビットのパターン信号である。転送等で問題があった場合、それぞれのチャンネルの b11 が k-comma 信号に置き換えられる。これによって、同期の確認ができる。

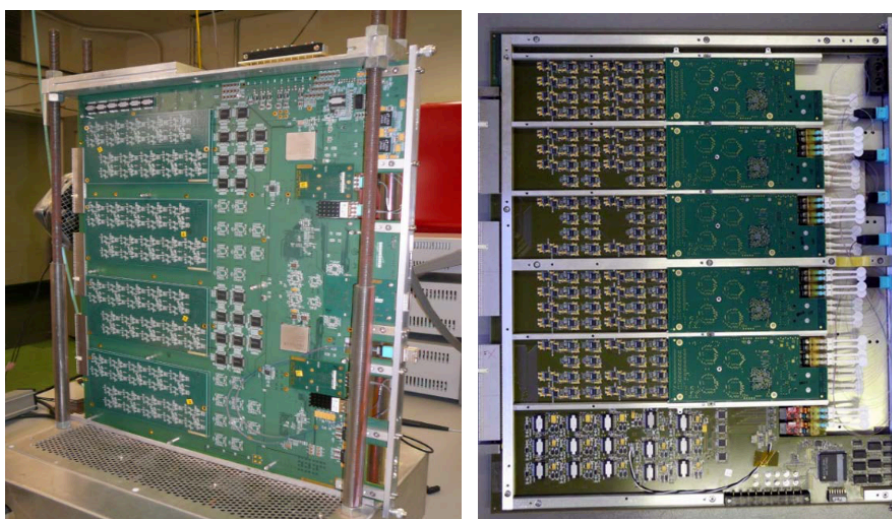


図 4.3: LTDB デモンストレータ。(左) アメリカにあるブルックヘブン国立研究所 (BNL) が製作した LTDB、(右) フランスにある LAL/Saclay 研究所が製作した LTDB。

4.3 LDPB デモンストレータ

LDPB デモンストレータは ATCA test Boards for Baseline Acquisition (ABBA) と呼ばれ、ATCA 規格で制作されている (図 4.4)。このボードの左に実装されている 4 組の

Avago pPod (以後 pPod という) が、LTDB からのデータを受信する。pPod の数は、図 3.7 中の AMC と同じである。つまり、1 個の ABBA は、1 個の LDPB で使用される AMC に対応している。

受信したデータは、2 つの FPGA (Front FPGA) で一時的に記録される。TTC から送られる信号の中に、デモンストレータ用のトリガー信号がある。この信号が Front FPGA へ入ると、記録しているデータの読み出しが行われ、後ろの FPGA (Back FPGA) へ送られる。Back FPGA は、Front FPGA からのデータに適切なフォーマットを付加し、ATCA のゾーン 2 を介して PC とイーサネット通信を行う。3 個の FPGA は全て ALTERA の StratixIV を使用している。

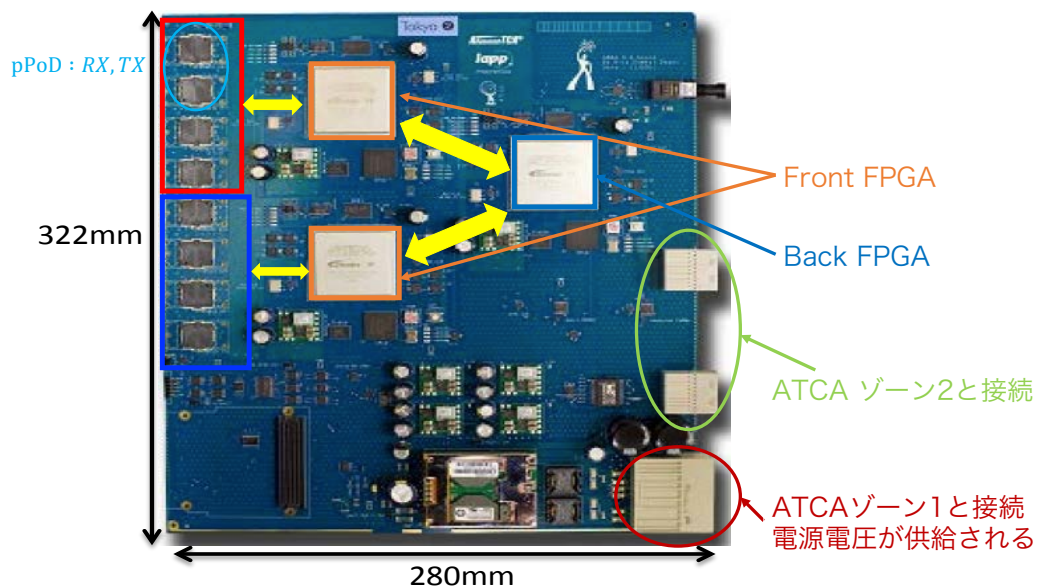


図 4.4: ABBA ボード。フランスにある LAPP 研究所で設計が行われた。日本でも制作が行われ、この図が日本で製作されたボードである。

4.4 バックエンド FPGA ファームウェア

Front FPGA と Back FPGA の役割は異なるため、2 種類のファームウェアが ABBA に必要である。私が主導的に行ったことは、どちらのファームウェアのテストベンチをモジュール毎に開発したことである。このテストベンチの作成によって、あらゆるパターンでファームウェアの動作検証をシミュレーションできた。これにより、Back FPGA ファームウェアが安定であることを示した。Front FPGA ファームウェアでは、実験中に起きていた問題を特定でき、修正を施すことによって安定したファームウェアを提供した。

この節では Front FPGA ファームウェアと Back FPGA ファームウェアの役割を簡単に説明する。これらのファームウェアの中で、最も重要で改善を施した adc readout モジュールについては次節で詳細に述べる。

4.4.1 Front FPGA ファームウェア

2つのFront FPGAは、それぞれ2組のpPod (24本の光ファイバー) と接続されている。この内、20本光ファイバーのみ使用している。Front FPGAは、大きく分けて図4.5中の4つのモジュールに分けられる。Optical interface モジュールはpPodから送られる信号をデシリアライズさせ、各Super CellのADCデータを抜き出す。抜き出されたADCデータは、adc readout モジュールで後述するとおり適切な処理を行う。その後、back fpga interface モジュールで、信号はシリアライズされる。また、TTCから送られる信号からL1A等を正確に抜き出すことをTTC decoding モジュールは行う。

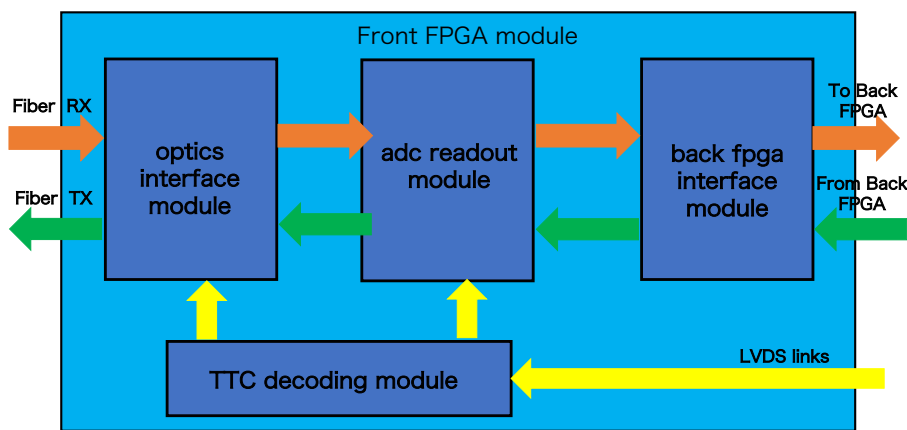


図 4.5: Front FPGA ファームウェアの全体図

4.4.2 Back FPGA ファームウェア

PCとイーサネット通信するために、2つのFront FPGAから受け取ったデータにイーサネットフレームが付加される(図4.6)。それぞれのモジュールは正常に動作しており、非常に安定したファームウェアが得られている。

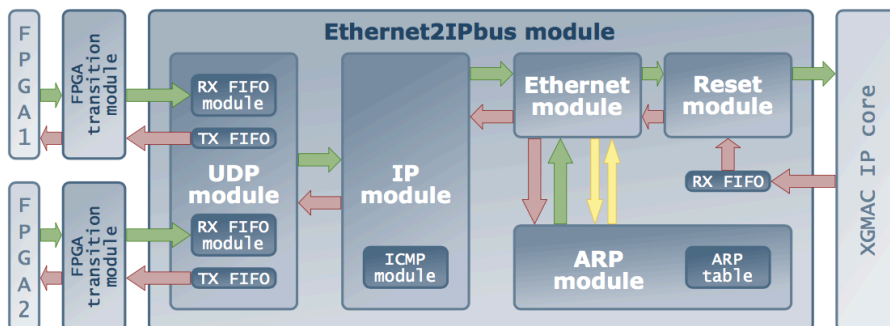


図 4.6: Back FPGA ファームウェアの全体図

4.5 Adc readout モジュールのファームウェア

ここでは、本研究で開発したテストベンチを用いて改善したファームウェアの動作検証、信号収集時に起きた問題の原因とその改善方法について述べる。

図 4.7 はこのモジュールの設計図である。1 個の Super Cell 信号は、12 ビットの ADC データである。1 本の光ファイバーに含まれる 8 つの Super Cell 信号は、 2×4 (パラレル数 \times シリアル数) のデータシーケンスでこのモジュールへ入る (図 4.8)。

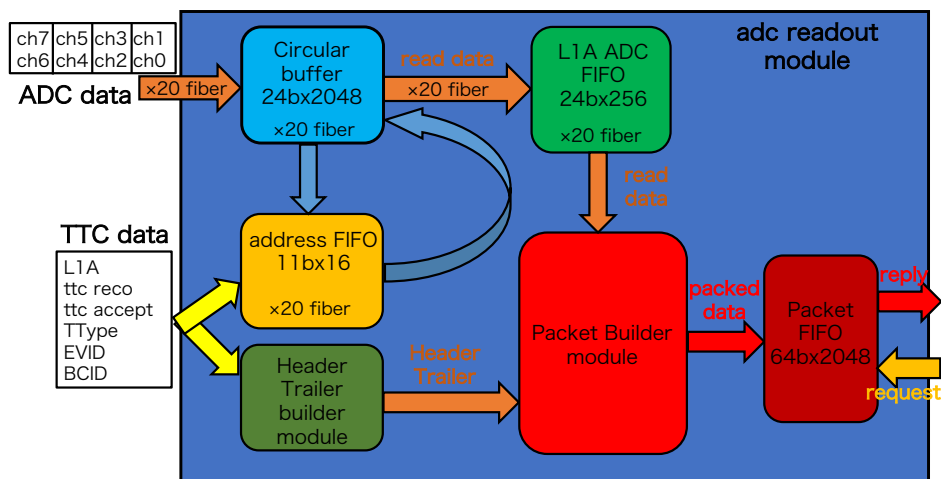


図 4.7: adc readout モジュールの設計図

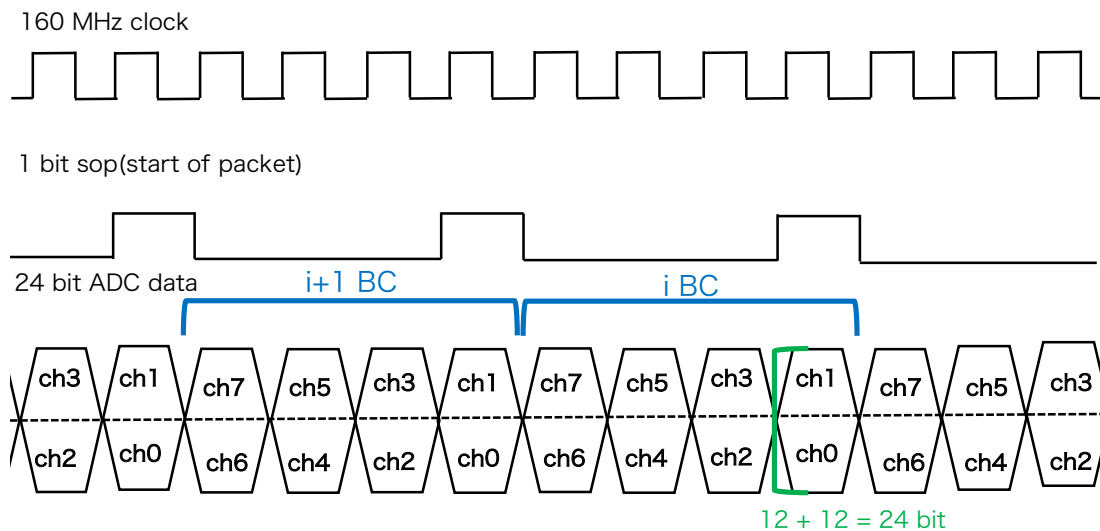


図 4.8: adc readout モジュールのインプットデータシーケンス。2 チャンネルが並列に入るため、24 ビットの ADC データがインプットになる。

表 4.1: adc readout モジュールに入る TTC 信号

信号名	意味
L1A	Level1 Accept (L1A) 信号
BCID	バンチを識別する番号
EVID	事象を識別する信号
TType	どのトリガー L1A が発行されたトリガーを示す信号
ttc reco	L1A を持つ事象の BCID, EVID, TType を全て受け取った時に発行される信号
ttc accept	デモンストレータ用のトリガーが通った時に発行される信号

また、表 4.1 の信号が、TTC からこのモジュールに入ってくる。デモンストレータ領域に $E_T \geq 50$ GeV の EM オブジェクトがある場合、ttc accept 信号が発行される。このエネルギー閾値は、ファームウェアが安定してきた 2016 年 8 月から 25 GeV へ変更された。以下、これらの信号を入れた時の各サブモジュールの役割の検証結果を述べる。

Circular buffer 図 4.8 の 24 ビット ADC データは、それぞれの光ファイバー毎にサーキュラーバッファ (リングバッファ) に記録される。このサーキュラーバッファの読み出しは、ttc accept 信号が来た時のみ開始される (図 4.9 上)。読み書き動作どちらもアドレスは 1 つずつ増加していく特徴をもち、ttc accept 信号をもたない ADC データはサーキュラーバッファ内で自動的に上書きされる (図 4.9 中央、下)。

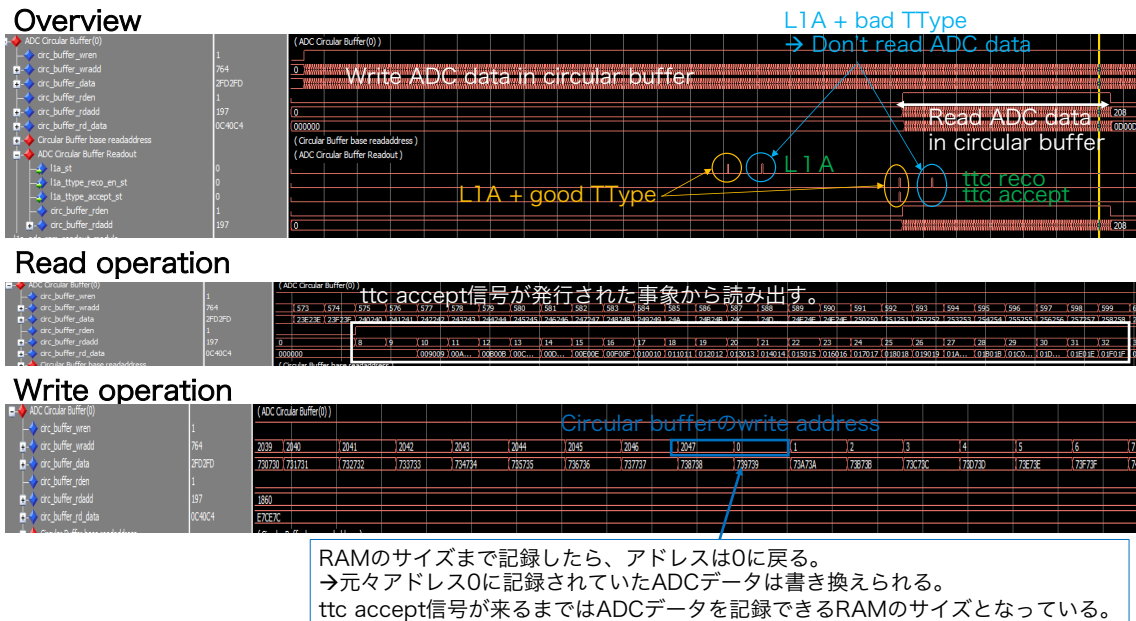
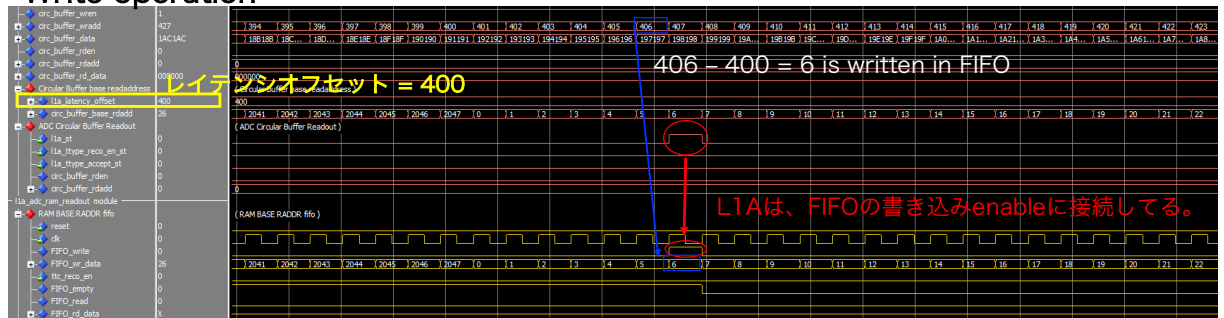


図 4.9: サーキュラーバッファのデータフロー。L1A と ttc reco や ttc accept 信号は同時に来ないが、L1A が来た順番に、その事象の ttc reco と ttc accept 信号は送られてくる。この図の場合、最初の L1A は ttc accept 信号を持っているが、2 番目の L1A は持っていない。そのため、1 番目の事象で記録された ADC データが読み出される。読み出しの詳細は、address FIFO モジュールで説明される。

Address FIFO サークュラーバッファからデータを読み出すために、L1A をもった事象のアドレスが FIFO に書き込まれる。実際にサーキュラーバッファから読み出す ADC データは、この FIFO で記録されるアドレスから読み出される。波形が来る前の数 Bunch Crossing (BC) でペDESTALを計算したいので、L1A 信号を持つ事象のアドレスからオフセットだけ差し引いたアドレスが FIFO に書き込まれる (図 4.10 上)。L1A 信号が来るレイテンシは固定なので、この信号が来た時のサーキュラーバッファの write address 信号からレイテンシ分を考慮すれば、FIFO に書き込みたいアドレスは算出できる。L1A 信号から約 1.2 μ s 後に来る ttc reco 信号で FIFO からアドレスが読み出される。その際に ttc accept 信号も来ていれば、読み出されたアドレスはサーキュラーバッファの read address 信号に使用される (図 4.10 下)。その read address 信号は、読み出されたアドレスを基準に波形全体を取得できる程度のサンプル数まで生成される (図 4.11)。いくつかの L1A をもつ事象を残すために、FIFO の深さが決定されている。

Write operation



Read operation

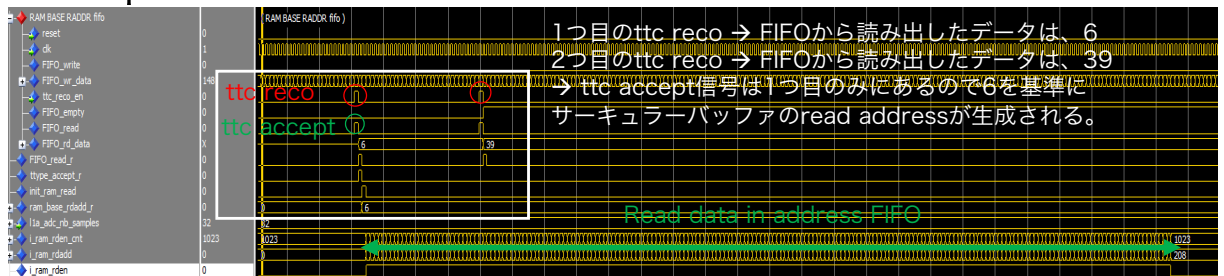


図 4.10: address FIFO のデータフロー。この図の場合、レイテンシオフセットは 400 である。FIFO に記録するアドレスは、サーキュラーバッファの write address 信号から 400 を差し引いた値になっている。L1A が来た時のみ FIFO の書き込みのイネーブル信号 (FIFO write) が立てられる。ttc reco 信号が来たら FIFO に記録されていたアドレスが読み出され、ttc accept 信号の有無によってサーキュラーバッファの read address 信号を生成するか判断する。

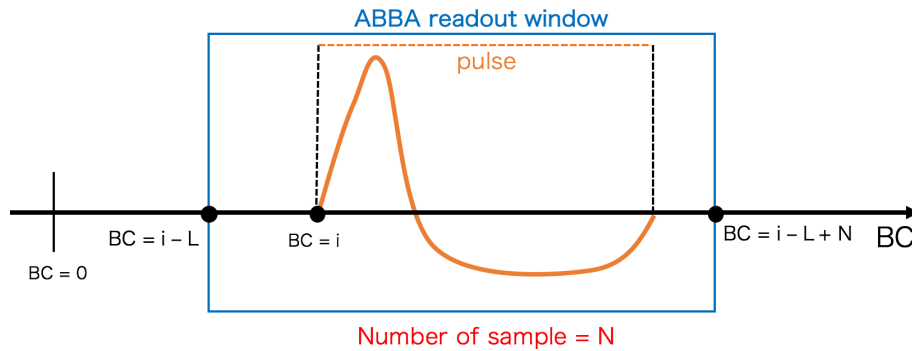


図 4.11: ABBA が取得する波形データ。TTC は、 i バンチに発行されている。LAr の波形の長さ
と最初の数 BC でペDESTAL を計算したいことを考慮して、データの読み出しは $i-L$ BC から始
める (典型的な値 : $L = 10, N = 50$)。

Header Trailer builder module Bunch Crossing Identification (BCID) や Event Identification (EVID) 等の TTC から信号は、ADC データと共に Back FPGA へ送られる。Back FPGA へ送る際のデータフォーマットは図 4.12 のように 64 ビット幅であり、この中に TTC からの信号が組み込まれる。このモジュールは、このフォーマットの作成を行う。

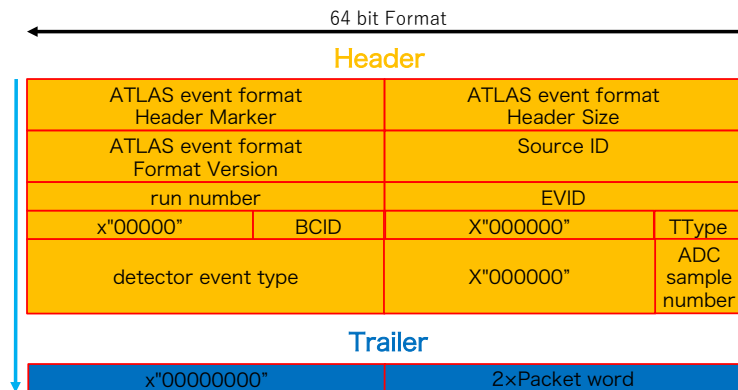


図 4.12: ABBA の Header と Trailer 一覧。図中の $x"0000"$ という表記は 16 進数表示であり、この
場合 0 が 16 ビット続くことを意味する。PC 側との兼ね合いで 64 ビットフォーマットは、最後に
2 分割される。そのため、Trailer の $2 \times \text{Packet word}$ は、32 ビットのワード数を示している。

L1A ADC FIFO 計 20 個のサーキュラーバッファから読み出された ADC データは 24 ビットであり、Circular buffer モジュールは 24 ビット \times 20 = 480 ビットを平行に処理していた。しかし、480 ビットをシリアルライズして Back FPGA へ送ることは不可能であるため、平行数を減らすフォーマット変換が施される⁴。そのため、それぞれの ADC データは L1A ADC FIFO に一時記録される。Packet Builder モジュールの中で光ファイ

⁴送信時の width は、16 ビットである。データクロックは 156.25 MHz を用いており、Back FPGA への通信速度は 3.125 Gbps である。

バー 2 本分のチャンネルごとに変換が行われている。変換が終わると、次の 2 つの FIFO を読み出す信号が Packet Builder モジュールから L1A ADC FIFO へ送られる (図 4.13)。

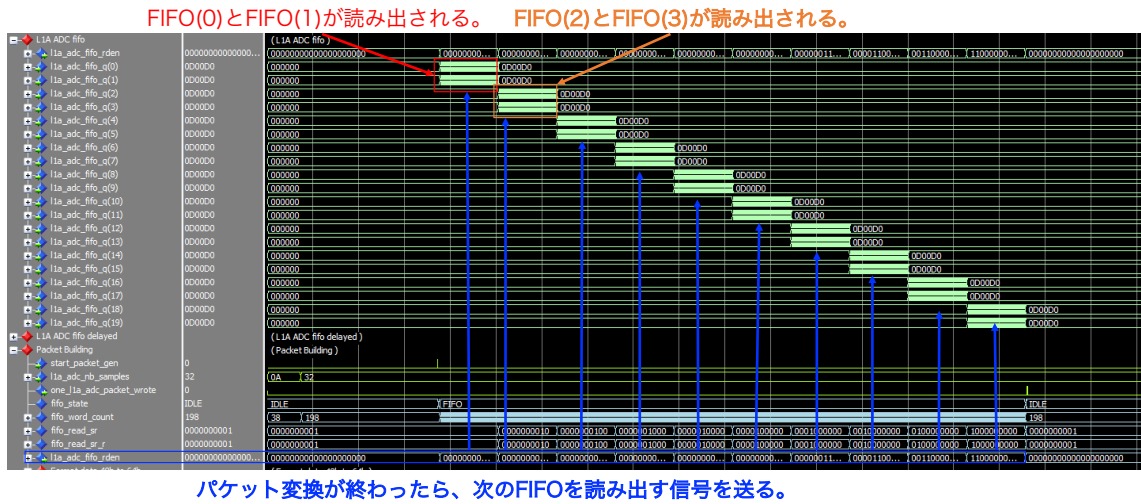


図 4.13: L1A ADC FIFO の読み出しシーケンス

Packet Builder module 2 つの L1A ADC FIFO から計 48 ビットの ADC データが図 4.14 のように 64 ビットへと変換される。この変換は全ての FIFO に対して行われ、ADC データの前に図 4.12 のヘッダー、後にトレータを付加することで、1 つの読み出しパケットが構成される。

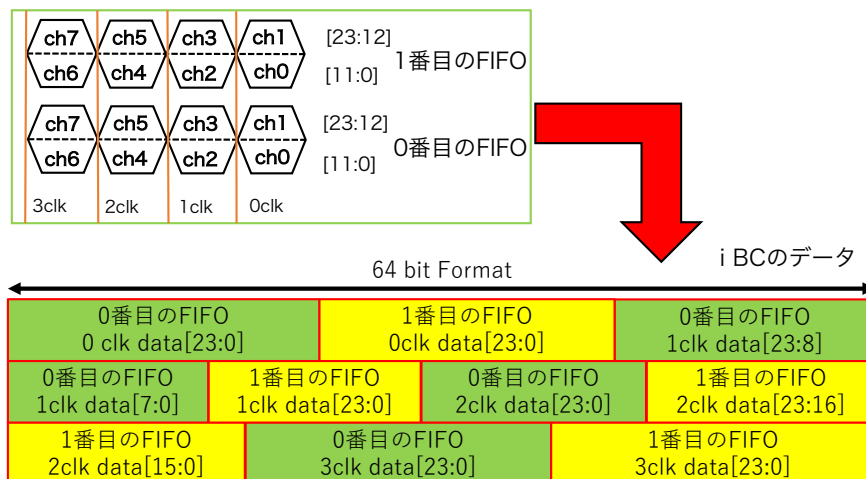


図 4.14: 64 ビットパケットフォーマット

Packet FIFO Packet Builder モジュールで作成されたパケットは、この FIFO に記録される。PC からデータを要求する信号が来た時に、FIFO からパケットが読み出され、back fpga interface モジュールへ送られる。パケット内の全てのデータが FIFO に記録された時、one l1a adc packet wrote 信号が立てられる (図 4.15)。これは、ABBA から読み出すべきパケットがあるということを示している。ファームウェアのリソースの観点から、1 パケットのみ adc readout モジュールは記録する。そのため、記録しているパケットを読みさない限り、次のパケットは作成されない。全てのパケットデータを FIFO から読み出すと、one l1a adc packet 信号が立てられる。この信号のあと、ファームウェア内でパケット作成が再開される。

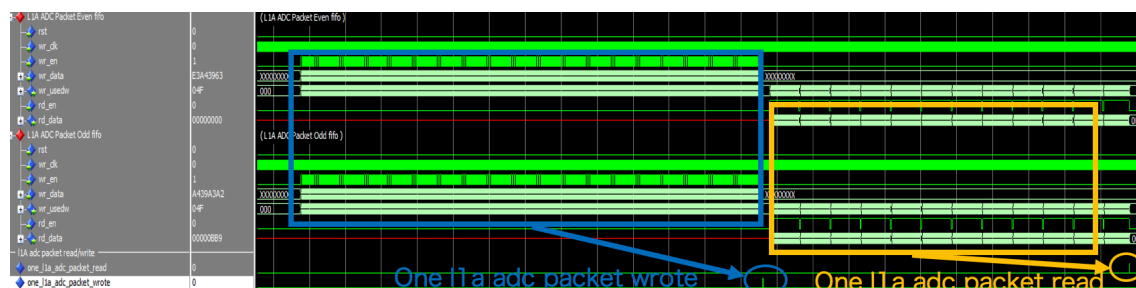


図 4.15: Packet FIFO のデータシーケンス PC 側のソフトウェアとの兼ね合いで、2 つの 32 ビット幅の FIFO (L1A ADC Packet Even FIFO と L1A ADC Packet Odd FIFO) がある。全てのパケットが書き込まれた時、one l1a adc packet wrote 信号が立てられている。同様に読み出された時、one l1a adc packet read 信号が立てられている。

図 4.16 は、実際に PC からデータを要求する信号が来た時の応答信号である。PC 側のソフトウェアとの兼ね合いで、64 ビットを 2 分割している。

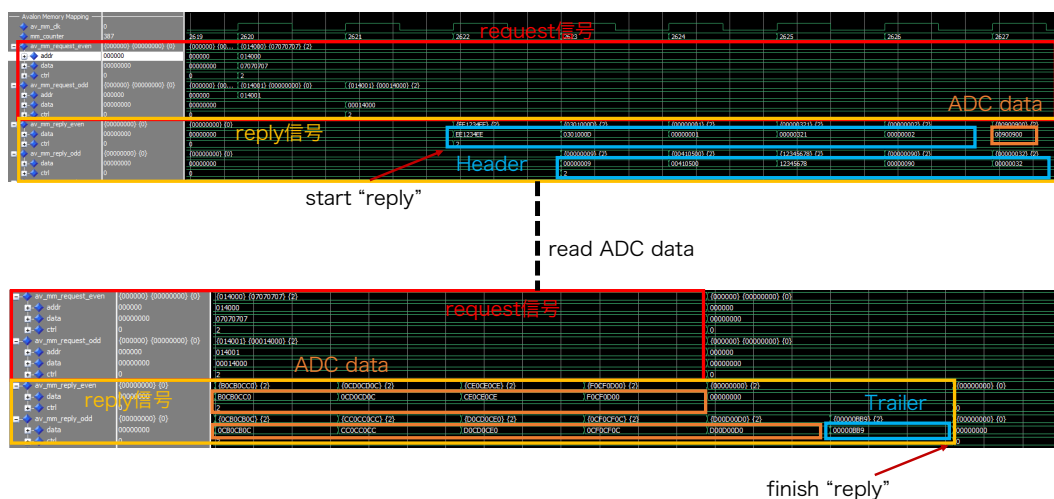


図 4.16: IP bus からの request 信号と reply 信号。PC 側からデータの要求を表すアドレス信号が定義されており、その信号が来たらパケットデータを転送する。reply 信号は Packet FIFO に入っていたデータを全て読み出す。

以下では、adc readout モジュールに関連した 2 つの問題点とその改善手法を述べる。

4.5.1 Channel mixing issue

ABBA の導入初期において、Front FPGA ファームウェアに対するコンストレインファイルが十分でなかった。コンストレインファイルは、クロックの動作周波数や FPGA の入出力の遅延情報を記述するファイルである。コンストレインファイルで記述した要求を満たすように配線が行われるため、このファイルは安定したファームウェアの作成において必要である。従来のコンストレインファイルでは、動作周波数を記述していないクロックや入出力ポートのコンストレンをかけていなかった。この時のファームウェアで取得したデータが図 4.17 であり、他チャンネルのペDESTAL と突然混じり合う事象が度々見られた。

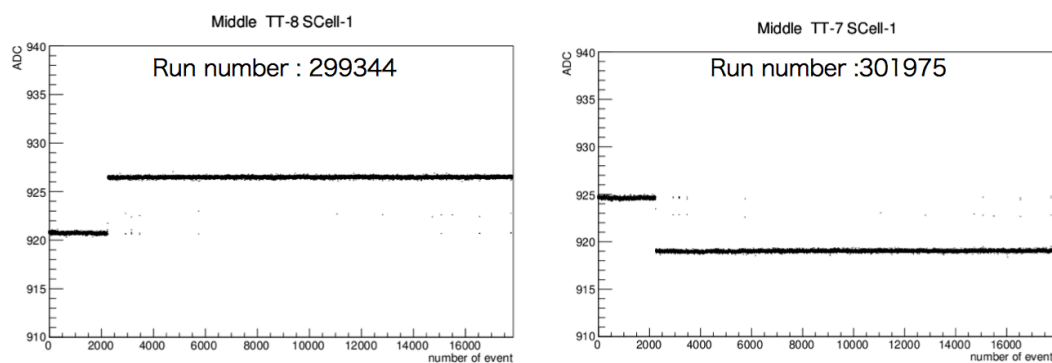


図 4.17: Channel mixing が現れたデータ。イベント毎にペDESTAL の値を計算している。Run 中は同じペDESTAL を持つはずだが、別のチャンネルと混ざってペDESTAL の値が変わる。

そこで、コンストレインファイルを完成させ、全ての経路に対してタイミングの指標となるセットアップスラックとホールドスラックを調べた。セットアップスラックは、次のクロックエッジまでにデータが到達しているかを表す指標である。この値が正であれば、次のクロックエッジで安定したデータが受け取れることを示している (図 4.18)。ホールドスラックは、クロックエッジに対してどの程度データを保持しているかを表す指標である。この値が正であれば、次のクロックで出力予定のデータと混じり合わない (図 4.19)。

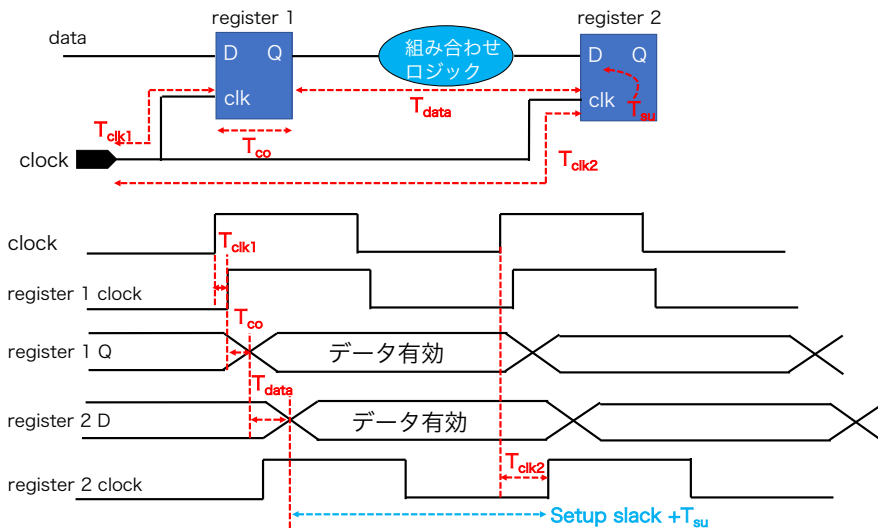


図 4.18: セットアップスラックのタイミングチャート。この図では、register1 と register2 の間に組み合わせロジックがある。register2 にデータが入力される時間は、register1 から出力された時間よりも組み合わせロジックの処理にかかる時間 T_{data} だけ遅れる。そのデータは、次のクロックの立ち上がりまでに有効でなければいけない。そうでなければ、次のクロックで register1 から送られてくるデータを出力できない。セットアップスラックは、次のクロックの立ち上がりから register2 の入力のデータ有効開始時間の差に相当する (実際は、 T_{su} というオフセットもこの差に入ってくる)。

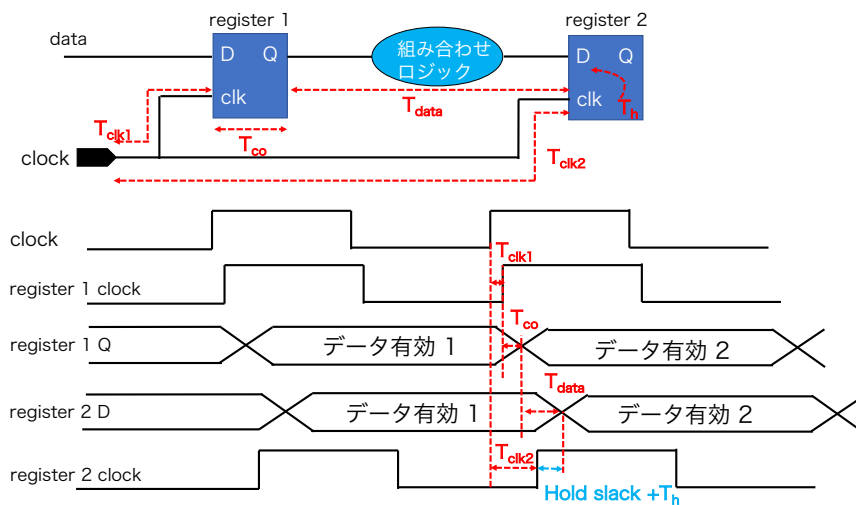


図 4.19: ホールドスラックのタイミングチャート。register2 の 2 つ目のクロックは、データ有効 2 を出力させる。そのため、このクロックが立ち上がる時には、register2 のデータ有効 1 はデータ有効 2 に切り替わっていないといけない。そうでなければ、そのクロックでデータ有効 1 とデータ有効 2 が混ざり合ってしまう。

現在の Front FPGA ファームウェアのスラックは図 4.20 と図 4.21 であり、コンストレーンを完全にかけたファームウェアはどの経路に対してもタイミング違反がないことを示せた。これにより、全てのタイミングを満たすファームウェアが作成できた。このファームウェアに改善して以降、図 4.17 のデータは現れなくなった。

	Clock	Slack
1	rst_and_clk_blk:ppll_right_160to120 gen_pll_inst:auto_generated pll1 clk[0]	0.757
2	altera_reserved_tck	37.694
3	clk_50	2.792
4	clk_100_left	4.597
5	data_clk	15.187
6	gxbr3_refclk~input~INSERTED_REFCLK_DIVIDER clkout	6.479
7	n/a	7.528
8	optics_interface\gen_gxb_blk:0:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component central_clk_div0 coreclkout	0.364
9	optics_interface\gen_gxb_blk:0:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs0 clkout	1.226
10	optics_interface\gen_gxb_blk:0:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs1 clkout	3.973
11	optics_interface\gen_gxb_blk:0:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs2 clkout	4.405
12	optics_interface\gen_gxb_blk:0:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs3 clkout	4.468
13	optics_interface\gen_gxb_blk:1:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component central_clk_div0 coreclkout	0.470
14	optics_interface\gen_gxb_blk:1:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs0 clkout	4.086
15	optics_interface\gen_gxb_blk:1:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs1 clkout	4.494
16	optics_interface\gen_gxb_blk:1:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs2 clkout	3.845
17	optics_interface\gen_gxb_blk:1:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs3 clkout	4.232
18	optics_interface\gen_gxb_blk:2:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component central_clk_div0 coreclkout	2.174
19	optics_interface\gen_gxb_blk:2:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs0 clkout	4.308
20	optics_interface\gen_gxb_blk:2:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs1 clkout	4.517
21	optics_interface\gen_gxb_blk:2:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs2 clkout	4.502
22	optics_interface\gen_gxb_blk:2:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs3 clkout	3.774
23	optics_interface\gen_gxb_blk:3:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component central_clk_div0 coreclkout	0.246
24	optics_interface\gen_gxb_blk:3:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs0 clkout	4.470
25	optics_interface\gen_gxb_blk:3:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs1 clkout	4.533
26	optics_interface\gen_gxb_blk:3:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs2 clkout	4.418
27	optics_interface\gen_gxb_blk:3:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs3 clkout	3.477
28	optics_interface\gen_gxb_blk:4:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component central_clk_div0 coreclkout	2.318
29	optics_interface\gen_gxb_blk:4:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs0 clkout	2.561
30	optics_interface\gen_gxb_blk:4:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs1 clkout	3.727
31	optics_interface\gen_gxb_blk:4:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs2 clkout	3.595
32	optics_interface\gen_gxb_blk:4:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs3 clkout	3.586
33	optics_interface\gen_gxb_blk:5:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component central_clk_div0 coreclkout	2.594
34	optics_interface\gen_gxb_blk:5:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs0 clkout	4.023
35	optics_interface\gen_gxb_blk:5:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs1 clkout	4.157

図 4.20: Front FPGA ファームウェアの全経路でのワーストセットアップスラックの一覧。全てのクロックに対して、正のスラックが得られている。

	Clock	Slack
1	rst_and_clk_blk:ppll_right_160to120 gen_pll_inst:auto_generated pll1 clk[0]	0.191
2	xau1_ipbus_interface xau1_phy altgx_xau1_alt4gxb_component central_clk_div0 coreclkout	0.194
3	optics_interface\gen_gxb_blk:5:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs2 clkout	0.200
4	optics_interface\gen_gxb_blk:4:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs2 clkout	0.205
5	optics_interface\gen_gxb_blk:2:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs3 clkout	0.225
6	optics_interface\gen_gxb_blk:5:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs1 clkout	0.236
7	clk_50	0.242
8	optics_interface\gen_gxb_blk:3:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs3 clkout	0.254
9	data_clk	0.256
10	optics_interface\gen_gxb_blk:2:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component central_clk_div0 coreclkout	0.260
11	optics_interface\gen_gxb_blk:0:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs3 clkout	0.264
12	optics_interface\gen_gxb_blk:1:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs1 clkout	0.264
13	optics_interface\gen_gxb_blk:2:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs0 clkout	0.264
14	optics_interface\gen_gxb_blk:4:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs1 clkout	0.264
15	optics_interface\gen_gxb_blk:0:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component central_clk_div0 coreclkout	0.265
16	optics_interface\gen_gxb_blk:3:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs0 clkout	0.265
17	optics_interface\gen_gxb_blk:1:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs2 clkout	0.265
18	optics_interface\gen_gxb_blk:3:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs0 clkout	0.265
19	optics_interface\gen_gxb_blk:4:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs3 clkout	0.265
20	optics_interface\gen_gxb_blk:1:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs0 clkout	0.266
21	optics_interface\gen_gxb_blk:1:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs3 clkout	0.267
22	optics_interface\gen_gxb_blk:0:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs2 clkout	0.268
23	optics_interface\gen_gxb_blk:0:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs0 clkout	0.269
24	optics_interface\gen_gxb_blk:5:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs0 clkout	0.271
25	optics_interface\gen_gxb_blk:4:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs0 clkout	0.276
26	optics_interface\gen_gxb_blk:5:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs0 clkout	0.276
27	optics_interface\gen_gxb_blk:1:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component central_clk_div0 coreclkout	0.296
28	optics_interface\gen_gxb_blk:5:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component central_clk_div0 coreclkout	0.298
29	optics_interface\gen_gxb_blk:3:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component central_clk_div0 coreclkout	0.303
30	optics_interface\gen_gxb_blk:4:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component central_clk_div0 coreclkout	0.315
31	altera_reserved_tck	0.322
32	clk_100_left	0.324
33	optics_interface\gen_gxb_blk:2:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs1 clkout	0.333
34	optics_interface\gen_gxb_blk:3:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs1 clkout	0.333
35	optics_interface\gen_gxb_blk:3:gxb_block transceiver_8B10B_4_8Gbps altgx_brx_basic4x_8B10_4_8gbps_alt4gxb_component receive_pcs2 clkout	0.333

図 4.21: Front FPGA ファームウェアの全経路でのワーストホールドスラックの一覧。全てのクロックに対して、正のスラックが得られている。

4.5.2 latency shift (pulse shift) issue

ABBA は図 4.11 のように波形全体を収集するシステムである。しかし、波形の一部しか収集できない図 4.22 の事象も度々見られた。これはキャリブレーションパルス (テストパルス) で見られず、物理データを収集した時のみ起こっていた。これは予想と違ったファームウェアの振舞いをしているため、改善が急務であった。この問題はシミュレーションなしに解決できないため、適切なテストベンチを作成することが重要なポイントである。しかし、前述したように adc readout モジュールのテストベンチはまだ開発されておらず、誰もその詳細な振る舞いを調べることができなかった。そこで、私は adc readout モジュールのシミュレーションに必要なインプットテストベンチを作成し、パルスシフトの問題の解決に努めた。

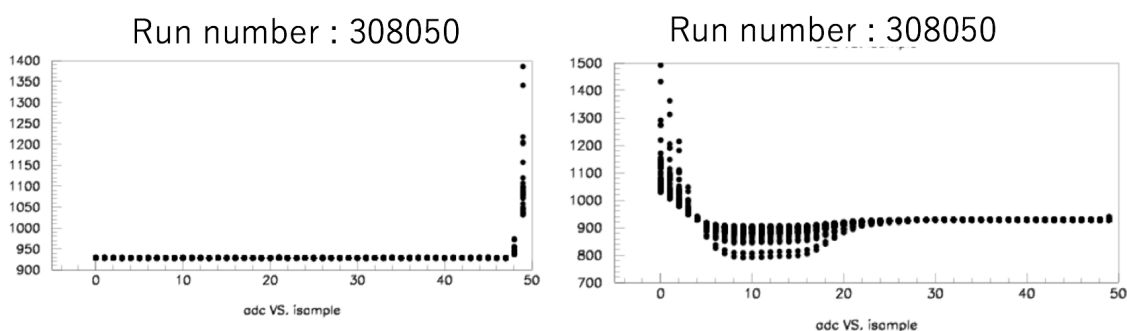


図 4.22: パルスシフトが起きた時の波形。ある光ファイバーで得られた Super Cell の信号をプロットしている。1本の光ファイバーに含まれている8つの Super Cell 信号は全てパルスシフトを起こしているが、ファイバー毎にみるとパルスシフトが起きていないファイバーもある。

シミュレーションのインプットに各チャンネルの ADC データと TTC からの信号を入れた (図 4.23)。

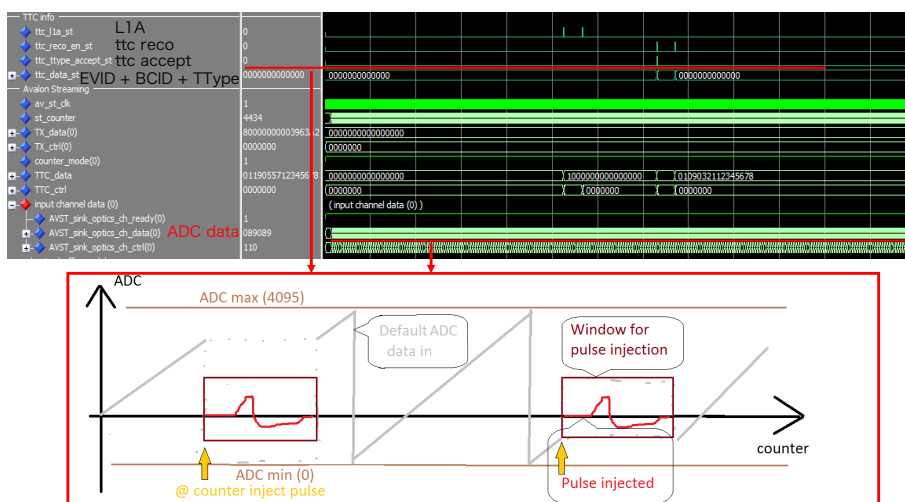


図 4.23: インプットデータシーケンス。表 4.1 の信号とそれぞれのチャンネルに適切な ADC データを入れている。ttc accept 信号の出力タイミングは、波形が収集できるように入れている。それ以外は、カウントアップデータを入れる。

あらゆるパターンにおいてこのモジュールの動作検証を行った結果、2つの ttc accept 信号が近いタイミングで入ってくる場合にファームウェアの動作が壊れることを確認した。図 4.24 は、その場合の address FIFO の振舞いを示している。ttc reco と ttc accept 信号によって FIFO に記録されていたアドレスが読み出され、サーキュラーバッファの read address 信号が生成される。サンプル数を 50 に設定しているため、1つのサーキュラーバッファから読み出すデータは図 4.8 を考慮すると、 $4 \times 50 = 200$ となる。そのため、サーキュラーバッファから読み出すデータ数を制御する信号 (i_ram_rden_cnt) は、200 から始まっている。この信号が 1 になるまで read address 信号が生成される。しかしながら、2つ目の ttc accept 信号によって、この制御信号のカウントが 200 にリセットしてしまう。これにより、結果的に予想よりも多くの ADC データがサーキュラーバッファから読み出される。図 4.14 の変換は予想したデータ数のみ行われるので、one l1a packet wrote 信号を立てても L1A ADC FIFO にデータが残ってしまう。この FIFO に残ったデータは次のパケットに組み込まれるため、図 4.25 の様に波形が分断される。

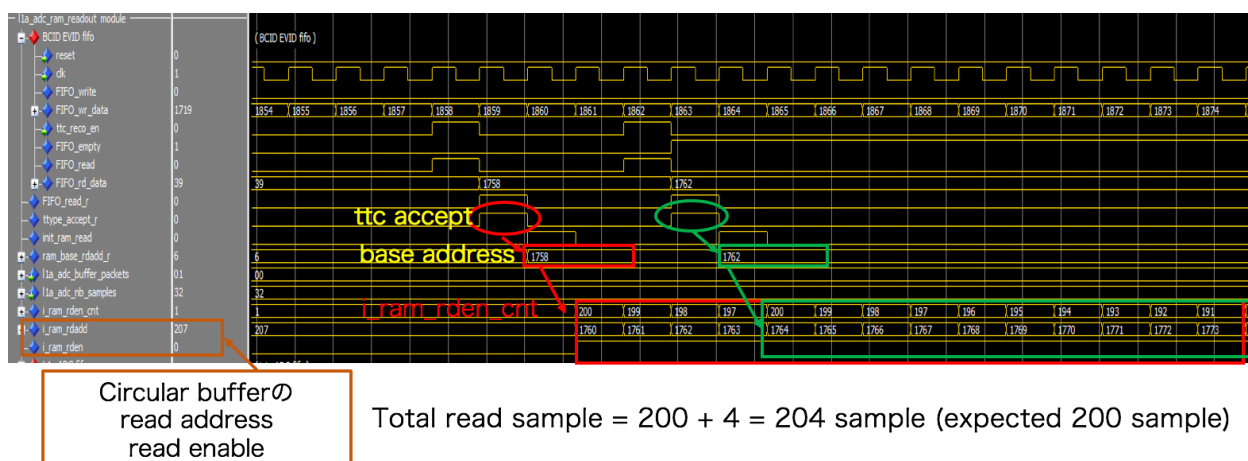


図 4.24: 近いタイミングで 2 つの ttc accept 信号が入った時の address FIFO の振舞い。2 つ目の ttc accept 信号により、i_ram_rden_cnt がリセットしている。これにより、 $200 + 4 = 204$ 個のデータをサーキュラーバッファから読み出すことになる。2 つの ttc accept 信号の時間の差で、サンプル数の読み出し量が変わる。つまり、波形のずれ具合が Run ごとに変わる。

そこで、制御信号は、カウントの減少中に他の ttc accept 信号でリセットされないようにした。そして、全ての FIFO にデータがない場合のみ、サーキュラーバッファの read address 信号が生成されるようにファームウェアを改善した (図 4.26)。改善したファームウェアでは、パルスシフトの問題は起きなくなった。

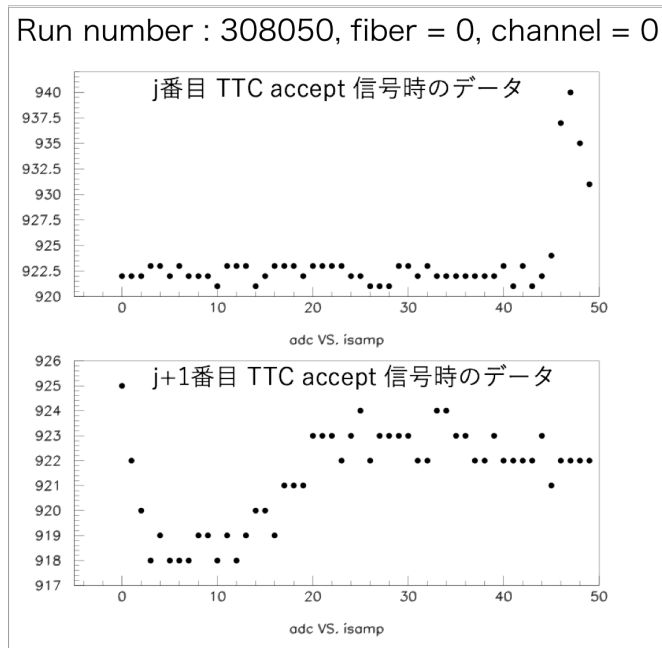


図 4.25: 連続した2つ ttc accept 信号の事象。パルスシフトにより2つの ttc accept 信号で1つの波形が分断されている。ttc accept 信号が近いタイミングで2つ起きたことでパルスシフトすることがデータ解析の観点からも示せた。

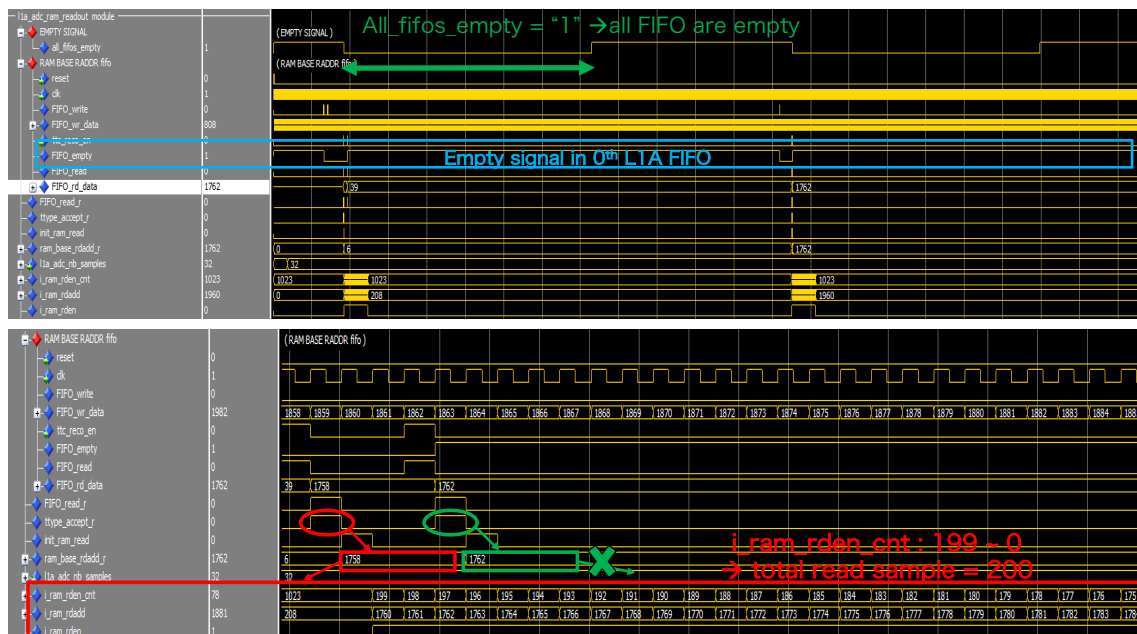


図 4.26: 改善後のファームウェアのデータフロー。図 4.24 と異なり、2 つ目の ttc accept 信号は i ram_rden_cnt 信号に影響を及ぼしていない。

4.6 デモンストレータプロジェクトのまとめ

本研究は、ABBA ファームウェアのテストベンチの作成を行った。これにより、不具合の原因を突き止めることができ、私は安定動作する ABBA ファームウェアを完成させた。このファームウェアによって、多くの Run2 データが取得できた。また、Run2 データ以外にも、キャリブレーションボードを使用したキャリブレーションデータも安定して取得できた。これは、読み出しエレクトロニクスの応答の測定や Optimal filter に必要な係数等を求めるために使用される。本研究では、5章で説明する LATOME プロジェクトのファームウェア設計に取得したデータを利用している。ファームウェア開発という点に焦点をあてると、テストベンチ開発の大切さを身をもって知ることができた。この経験は、5章で説明するファームウェア開発に大きく役立った。

第5章 LATOME プロジェクト

LDPB に実装する AMC と FPGA ファームウェアを開発するプロジェクトが現在進行中である。私はファームウェアの開発に携わっており、特に Optimal filter を用いたエネルギー計算を行うファームウェアの考案、設計から評価までを一から全て行った。本章では、その詳細を述べる。

5.1 LATOME Board

図 5.1 の AMC は共同研究を行っているフランスにある LAPP 研究所で開発されている。この AMC は、LAr Trigger Processing Mezzanize (LATOME) と呼ばれている。LATOME の右側に 4 つのコネクタが搭載されており、受信・送信用に計 96 本の光ファイバーが接続されている。図 5.2 の中央左寄りの青いコンポーネントがヒートシンクであり、その下に FPGA が搭載されている。この FPGA は INTEL 社 Arria10 シリーズ、10AX115R4F40I4SGES を用いている。この FPGA はミッドレンジの中で 2015 年に公開された最新機種であり、96 本の高速データ通信のトランシーバを持っている。また、エネルギー計算に必要な DSP Block は最大 1518 個搭載しており、業界唯一の浮動小数点にも対応している。

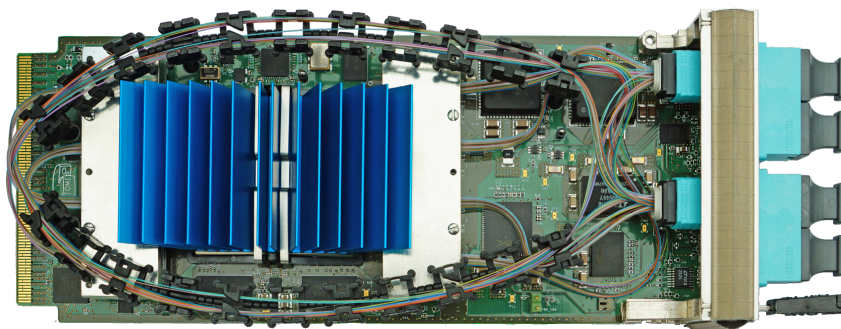


図 5.1: LATOME Board[13]。サイズは $156 \times 73.5 \text{mm}^2$ であり、これが LDPB に実装される AMC である。

5.2 LATOME ファームウェア

このファームウェアの目的は、2 つに大別される。

- 光ファイバーを 48 本 (使用するのはその内 40 本) を用いて、5.12 Gbps/本の受信と 11.2 Gbps/本の送信を達成する (図 3.7)。

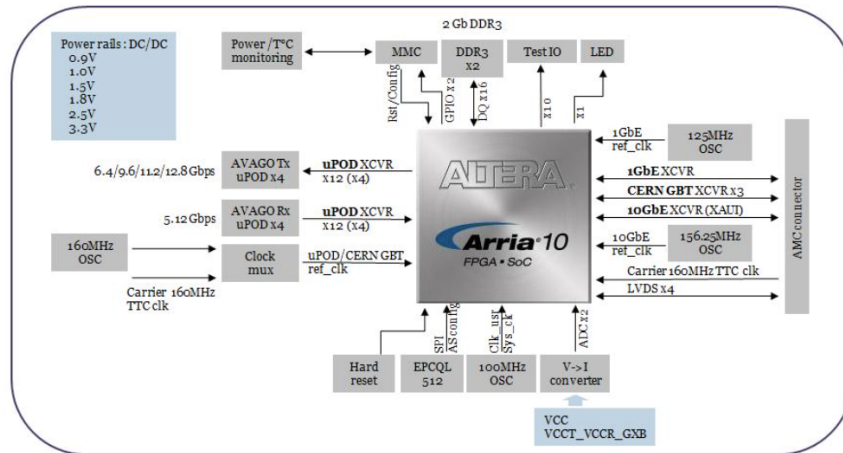


図 5.2: LATOME Board と Arria10 FPGA との接続の略図 [11]

- 受信した Super Cell の ADC データからエネルギーおよびタイミングを算出する。また、サチュレーションした信号の発生時刻の同定も行う。

このデータストリームはトリガー発行に使用されるため、いつも決まったレイテンシ（計算時間）での出力が要求される。それは、15 BC (375 ns) である。これらを満たす設計を議論した結果、図 5.3 と図 5.4 の設計図が作成された。LTDB から受け取ったデータは、Low Level Interface layer (LLI) へ入っていく。LLI で処理されたデータは、図 5.4 へ進んでいく。Low Level Interface, Input stage, Configurable remapping, User Code Output summing モジュールが ADC データの進む経路であるため、これらのレイテンシが非常に重要になる。そのため、設計段階のレイテンシの概算は表 5.1 となっている。本研究の一つの大きな目的は、実際の仕様に耐えるファームウェアを構築し、このレイテンシが実現可能であることを実証することである。

以下では、各モジュールの簡単な役割を説明する。

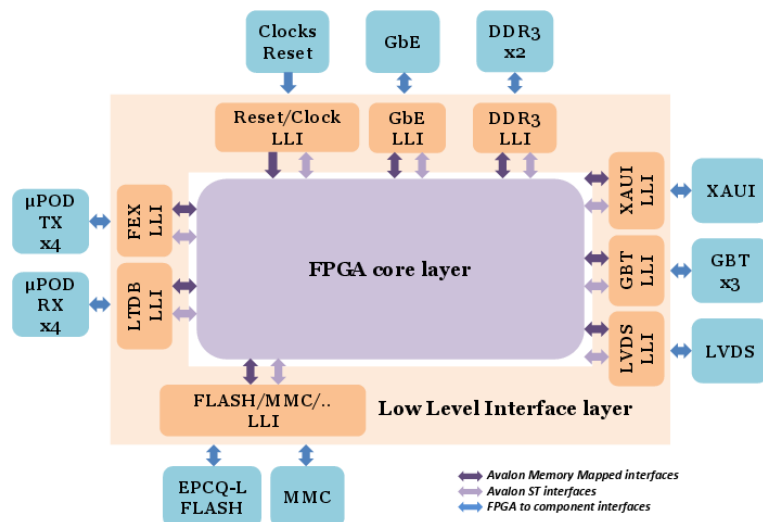


図 5.3: LLI の全体図 [14]

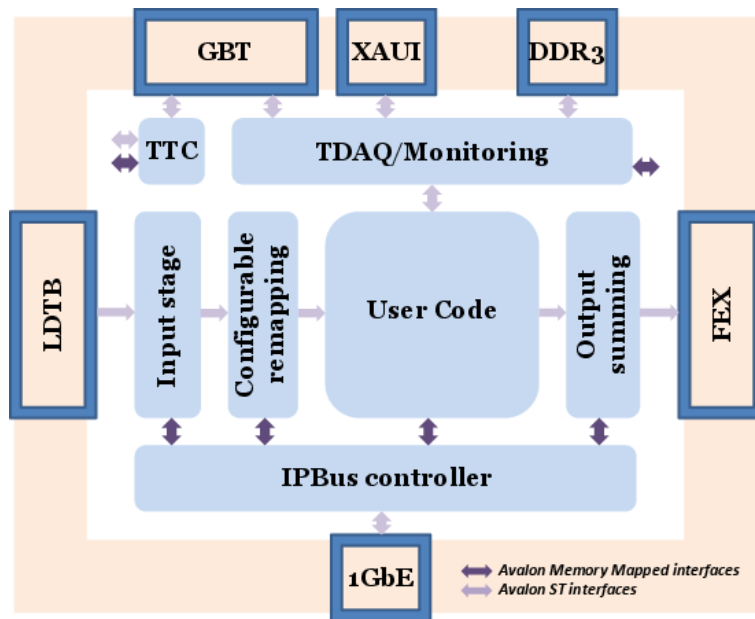


図 5.4: LATOME ファームウェア全体図 [14]

表 5.1: 各モジュールに要求されるデザイン段階のレイテンシ (1 BC=25 ns)

モジュール	レイテンシの要求値 [BC]
Low Level Interface	4.0
Input stage	3.0
Configurable remapping	1.5
User Code	5.0
Output summing	1.5
Total	15.0

5.2.1 Low Level Interface (LLI)

シリアライズされているデータはこのファームウェアに実装される LLI でデシリアライズされ、図 5.4 中の Input stage に伝送される。同様に、FEX へ送信する際も LLI でシリアライズする。このように LLI は、トランシーバや DDR3 などの外部素子と FPGA の間の入出力信号を図 5.4 中のモジュールに供給する役割を持つ。

LTDB から LDPB へ送られるデータフォーマットは、低レイテンシと再同期の高速化を満たす独自のフォーマットである (図 5.5)。それぞれのチャンネルの D0 から D11 は Super Cell の ADC データであり、D12, D13 と T0 から T15 は BCID や CRC などのコントロール信号である。また、データクロックは 320 MHz なので、図 5.5 を見ると 16 ビット × 320 MHz = 5.12 Gbps の通信速度をもつ。

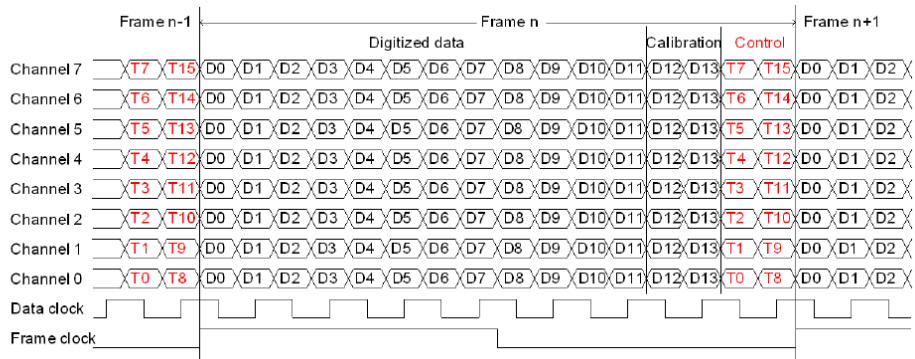


図 5.5: 平行変換された時のデータフォーマット [15]。Channel0 から Channel7 が 8 種類の Super Cell 信号に対応している。

5.2.2 Input stage

このモジュール (図 5.6) は、図 5.5 のデータフォーマットから BCID とそれぞれの Super Cell の ADC データを正しく抜き出す役割をもつ。また、CRC を計算しデータの有意を確認する。それ以外に、このモジュールは、ファイバーごとにレイテンシの異なる信号を 1 つの 320 MHz クロックで同期させる。その同期されたデータは、図 5.7 で定義される stream と呼ばれるデータ群で次のモジュールに送られる。また、ファームウェアの試験を行うために、テストパターンを生成するオプションもつける。

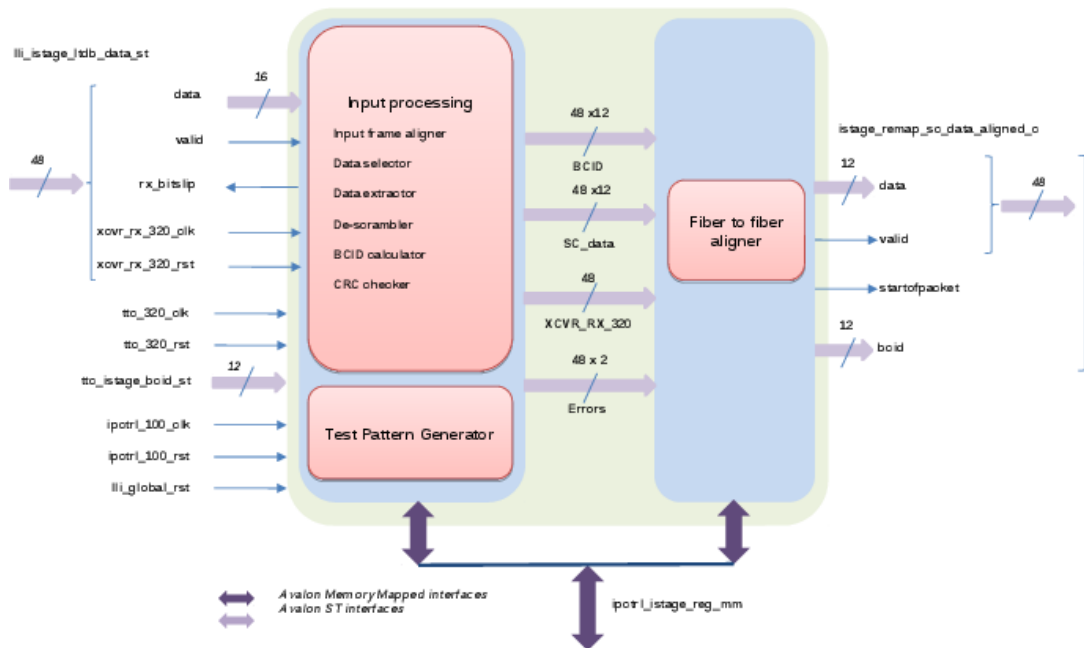


図 5.6: Input stage モジュールの全体図 [14]。各チャンネルごとに図 5.5 の 16 ビットが、平行に送られてくる。Input processing サブモジュールは、ADC データや BCID 等を正しく抜き出す。その後、Fiber to fiber aligner サブモジュールで全てのファイバーを 1 つの 320 MHz クロックと完全に同期させる。

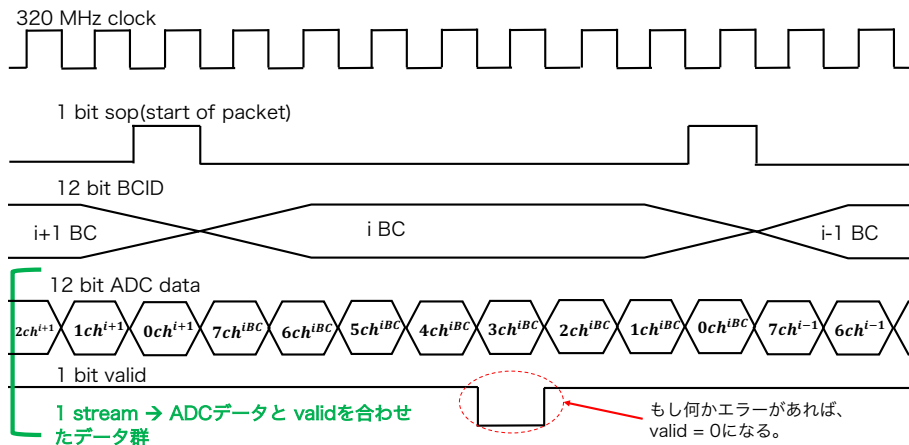


図 5.7: 出力のデータシーケンス。基本的には、データとそのデータの有意性を示す valid 信号を 1 つのデータ群としている。

5.2.3 Configurable remapping

このモジュールは、Input stage から送られてきたデータを並び替える。次の User Code の演算処理を 320 MHz クロックにて実行することが困難であるため、このモジュールが導入されている。クロックの周波数をおとすと、パラレル数が増える。しかし、FPGA 内で使用するメモリやロジックの数は、パラレル数の増加に伴い増える。これは、FPGA 内の消費電力や温度に影響を及ぼす。クロックの周波数とパラレル数を考慮した結果、パラレル数を 62 に増やし、240 MHz クロックで動作させる (図 5.8)。

並べ替え後 (6 チャンネル \times 62 stream) は、8 チャンネルの ADC データが 48 stream あった並べ替え前と比較すると、総チャンネル数が異なる ($372 \neq 384$)。しかし、LATOME に入る最大の Super Cell 数は 320 のため、複数のチャンネルにデータが入っていない。そこで、このモジュールは、PC からマッピング情報を受け取り、並べ替えの前後で消えるチャンネルは必ず ADC データを持っていないようにする。

並べ方に関しては、実装するロジックを考慮して決められている。例えば、User Code で実装するサチュレーションを同定するロジックは、Trigger Tower にある全ての Super Cell を使用する。そのため、Trigger Tower 内の Super Cell は、並列処理されている他の stream にて出力させる (図 5.9)。これにより、サチュレーションを同定するロジックのレイテンシが最小限に抑えられる。これに加えて、エンドキャップの一部領域は 1 Trigger Tower が 11 個の Super Cell で構成される。その領域に対しても考慮した結果、パラレル数を 62 にした (図 5.10)。これ以外にも、並び替えは、Output summing モジュールで行うエネルギーの足し合せにかかるレイテンシを小さくする考慮もされている。

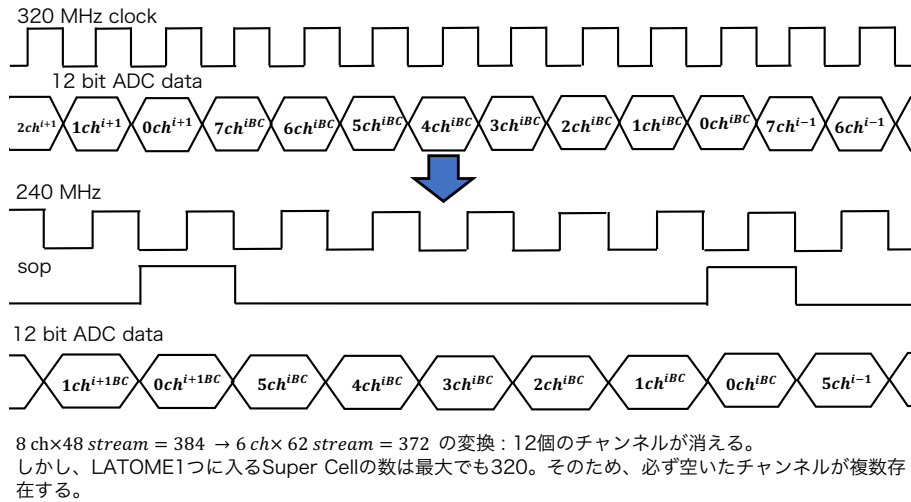


図 5.8: Configurable remapping でのチャンネル並び替え

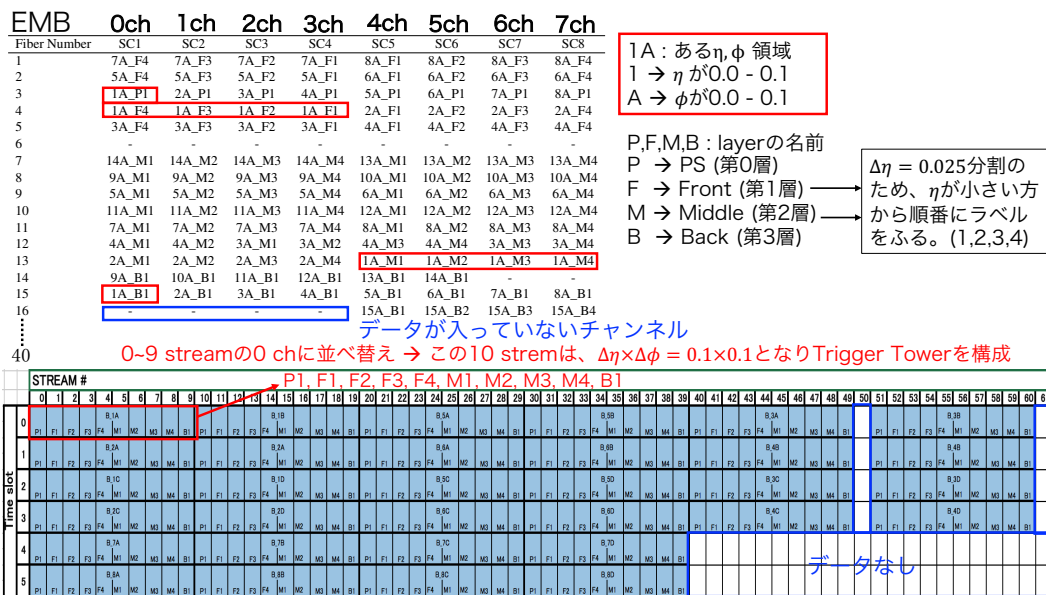


図 5.9: パレル部でのマッピングの一例。上の表は、パレル部のあるファイバーとチャンネルのマッピングを示している。3,4,13,15 本目のファイバーに $\eta=0.0 \sim 0.1$, $\phi=0.0 \sim 0.1$ 領域の Super Cell が配置されている。これらのデータを受け取って、0 ~ 9 stream 目の 0 ch へ並べる。この処理は全ての領域で行われる。

EMEC

Och	1ch	2ch	3ch	4ch	5ch	6ch	7ch	
Fiber Number	SC1	SC2	SC3	SC4	SC5	SC6	SC7	SC8
1	6A_F3	6A_F4	6A_F5	6A_F6	6B_F3	6B_F4	6B_F5	6B_F6
2	5A_F5	5A_F6	6A_F1	6A_F2	5B_F5	5B_F6	6B_F1	6B_F2
3	5A_F1	5A_F2	5A_F3	5A_F4	5B_F1	5B_F2	5B_F3	5B_F4
4	3A_F4	3A_F3	3A_F2	3A_F1	3B_F1	3B_F2	3B_F3	3B_F4
5	4A_F4	4A_F3	4A_F2	4A_F1	4B_F1	4B_F2	4B_F3	4B_F4
6	7A_M1	7A_M2	7A_M3	7A_M4	7B_M1	7B_M2	7B_M3	7B_M4
7	8A_M1	8A_M2	8A_M3	8A_M4	8B_M1	8B_M2	8B_M3	8B_M4
8	10B_M4	10B_M3	10B_M2	10B_M1	10A_M4	10A_M3	10A_M2	10A_M1
9	5A_M1	5A_M2	5A_M3	5A_M4	5B_M1	5B_M2	5B_M3	5B_M4
10	9B_M4	9B_M3	9B_M2	9B_M1	9A_M4	9A_M3	9A_M2	9A_M1
11	6A_M1	6A_M2	6A_M3	6A_M4	6B_M1	6B_M2	6B_M3	6B_M4
12	3A_M1	3A_M2	3B_M1	3B_M2	3A_M3	3A_M4	3B_M3	3B_M4
13	4A_M1	4A_M2	4A_M3	4A_M4	4B_M1	4B_M2	4B_M3	4B_M4
14	3A_B1	4A_B1	3B_B1	4B_B1	5A_B1	6A_B1	5B_B1	6B_B1
15	7A_B1	8A_B1	7B_B1	8B_B1	9A_B1	10A_B1	9B_B1	10B_B1
40								

5A : ある η, ϕ 領域
 5 $\rightarrow \eta$ が 1.8 - 1.9
 A $\rightarrow \phi$ が 0.0 - 0.1

Super Cell数
 PS \rightarrow 0個
 Front \rightarrow 6個
 Middle \rightarrow 4個
 Back \rightarrow 1個

11 streamで $\Delta\eta \times \Delta\phi = 0.1 \times 0.1$ となりTrigger Towerを構成

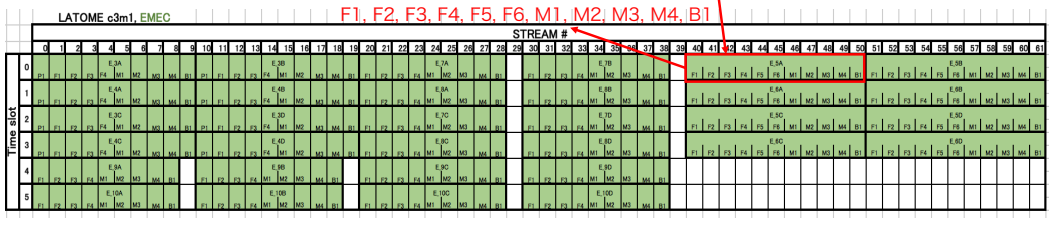


図 5.10: エンドキャップ部でのマッピング一例。1 Tower Trigger が 11 Super Cell で構成される場合、50, 61 stream 目が使用される。

5.2.4 User Code

このモジュールは、時間とエネルギーの算出を行う。これはある Super Cell において、どのバンチ (時刻) にどれだけのエネルギーを落としたかを計算することになる。また、サチュレーションした信号も通常のレイテンシで検出させなければならない。このモジュールは LATOME ファームウェアのメインパートであると同時に、私が設計・実装したパートであるため、次の節で詳しく述べる。

5.2.5 Output summing

このモジュールは、FEX に送るためのエンコーディングを行う。FEX は Super Cell のエネルギーだけではなく、Trigger Tower に落としたエネルギーと $\Delta\eta \times \Delta\phi = 0.2 \times 0.2$ のエネルギー (Trigger Tower 4 つ分) も要求している。そのため、それに適したエネルギーの足し合わせが行われる。

5.2.6 TDAQ/Monitoring

それぞれのモジュールのモニターしたい信号は、図 3.7 の右側にある FPGA を介して PC へ転送される。このモジュールは、そのためのデータフォーマットの作成等を行う役割を持つ。

5.2.7 IPBus controller

User Code で使用される係数やペDESTAL等のコンフィグレーションの読み書きは、図 5.4 中の 1 Gbps のイーサネット通信で行われる。このモジュールは、どのような変更かを判断しそれを伝える役割をもつ。

5.2.8 TTC

このモジュールは、TTC からきた信号をデコードし、BCID や Trigger Type, L1A など抜き出す役割をもつ。

5.3 User Code

このモジュールのファームウェア作成の設計・実装において重要な点は以下の 6 点であり、それぞれについて詳しく述べる。

1. Filter の設計
2. Filter 後の適切な選択条件とその設計
3. エネルギー、係数等のフォーマット
4. サチュレーションした波形の同定方法
5. モジュールで使用されるコンフィグレーションの変更方法
6. ベースラインコレクションの設計

5.3.1 Filter の設計

本研究では Optimal filter を採用し、実装した。Optimal filter の設計で、主に注意すべきことは以下の 3 つである。

- 1 stream に 6 つの Super Cell の ADC データが含まれているため、多チャンネル対応の積和計算 (図 5.8)。
- 240 MHz クロックで動作させるために、FPGA 内のリソース (DSP Block や RAM) を効率的に使用した設計
- 5 BC 以内のレイテンシ

これらを満たすために、 18×19 の固定小数点シフトモードの DSP Block を利用する (図 5.11)。このモードは隣の DSP Block とカスケード接続するため、配線遅延を最小限にできる上に、積和のロジックが全て DSP Block 内で行える。DSP Block 内の配置は変更できないため、シフトレジスタとサーキュラーバッファを利用して 6 チャンネル対応の Filter を実現させる。ADC データに 5 クロックディレイのシフトレジスタを接続させ

ると、全ての DSP Block 内の加算は同じチャンネルで行われる。また、係数とペデスタルはサーキュラーバッファ (サイズは 6) に格納しておき、その出力は各 DSP Block のインプットと接続させる。

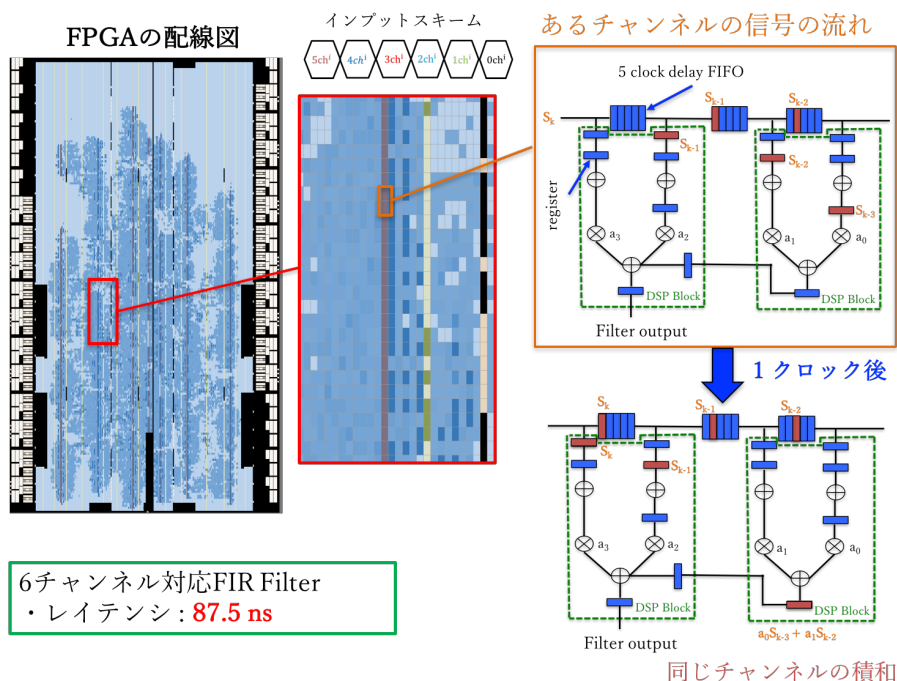


図 5.11: Optimal filter のファームウェア設計。DSP Block は FPGA 中で縦一列に並んでいるため、それを十分に利用した設計になっている。右図中の青枠は、レジスタである。DSP Block 内のレジスタの位置は固定されており、変更できない。バイパスすることは可能だが、本設計では動作安定性を考慮し、DSP Block 内の全てのレジスタを使用している。DSP Block の外側に、入力したデータを 5 クロック遅らせるシフトレジスタが配置されている。あるチャンネルの信号 S_{k-3} から S_k を橙色にした。これを見ると、他チャンネルと混ざらず積和計算ができていことがわかる。

5 クロックディレイのシフトレジスタを使用しているため、同じチャンネルのデータが同時に DSP Block に入らない。サーキュラーバッファの出力は複数のチャンネルを同時に出力できないため、本設計はペデスタルに対しても個々にサーキュラーバッファを用意した (図 5.12)。RAM のサイズは小さいため、M20K でなく MLAB を使用することで十分実装可能である。

Optimal filter は未来の信号を 3 つ使用するため、レイテンシは 3 BC は最低かかる。本設計は、計算に用いる最後の入力信号が DSP Block に入ってから 3 クロック後に、Filter の結果が出力される。そのため、図 5.11 のレイテンシは 3 BC (75 ns) + 3 クロック (12.5 ns) = 87.5 ns となる。また、4 点の Optimal filter の DSP Block の使用率は、Arria10 FPGA の全体のリソースの 20 % 以内に抑えられた (表 5.2)。さらに、今後の研究で過去の点を使用した点以上の Filter が実装されることになった場合において、本設計は DSP Block のカスケード接続を増やすだけで対応できる。そのため、ファームウェアのソースコード自体の変更はほとんどない。また、DSP Block の使用率の観点から十分に実装できることを示した。

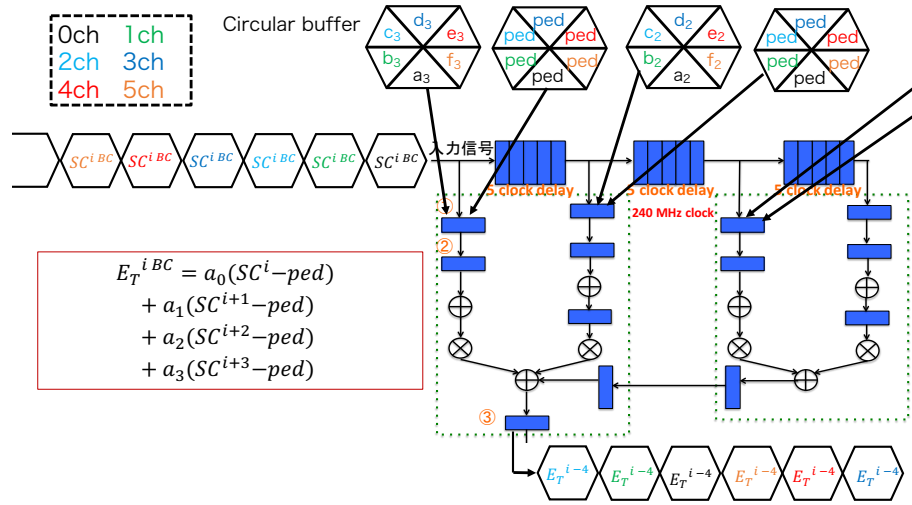


図 5.12: Optimal filter のデータフロー。各サーキュラーバッファは、それぞれの DSP Block と接続されている。5 クロックディレイシフトレジスタにより、それぞれの DSP Block に入るデータは同じチャンネルではない。(6 チャンネルのデータフローのため、1 チャンネルずつずれる。)

表 5.2: ファームウェア内で使用する DSP Block の使用率

Filter に使用するサンプル数	使用した DSP Block	DSP Block 使用率
4	248	16.3 %
6	372	24.5 %
8	496	32.7 %
10	620	40.8 %
12	744	49.0 %
14	868	57.2 %

5.3.2 Optimal filter 後の適切な選択条件

どのバンチで有意なエネルギーを落としたかを算出するために、Optimal filter 後に選択条件を課している。現在、L1Calo の選択条件は、maximum finder と呼ばれ以下で定義される。この選択条件で得たエネルギーとオフラインとの差異が、図 5.13 の赤点である。

$$A_{i-1} < A_i < A_{i+1} \quad (5.1)$$

A_i : i BC での optimal filter の値

図 5.13 をみると、約 4 GeV 以上の信号は L1Calo でほぼ全て正しいバンチで再構成できており、良い選択条件の一つと考えられていた。しかし、maximum finder は次の BC の Optimal filter の値を使用するため、最低 1 BC のレイテンシが必要になる。そのため、maximum finder に代わる低レイテンシの選択条件を考える。Arria10 FPGA は、2 種類の Optimal filter を平行に実装できる。そこで得られる τ を用いた選択条件を課してみる。具体的には以下の式である。

$$|\tau| < \tau_{\text{thre}} \quad (5.2)$$

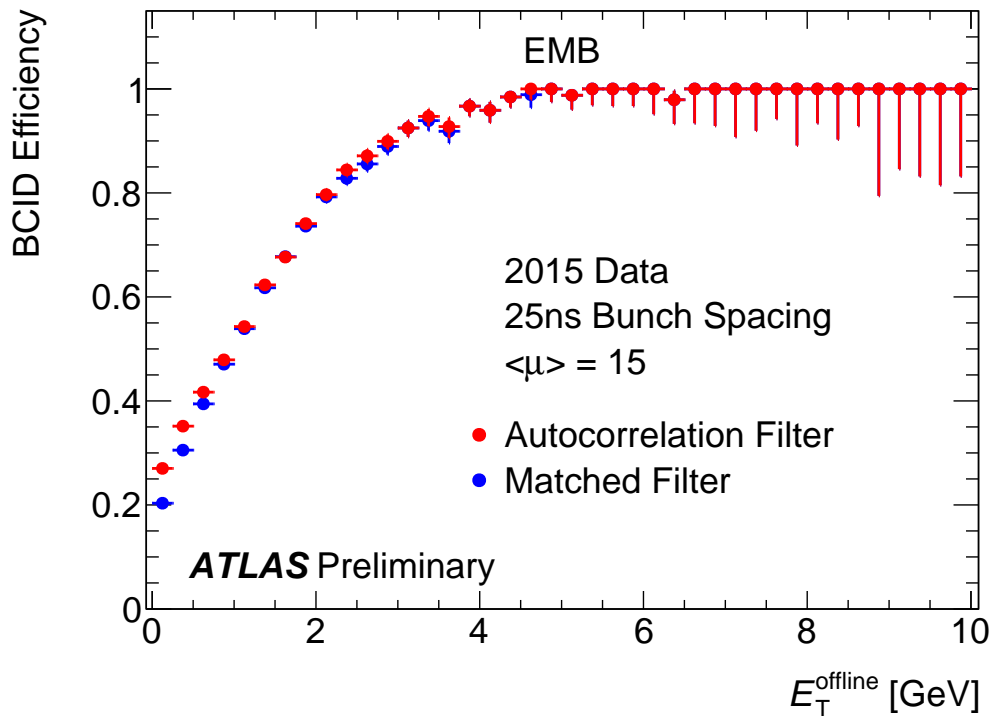
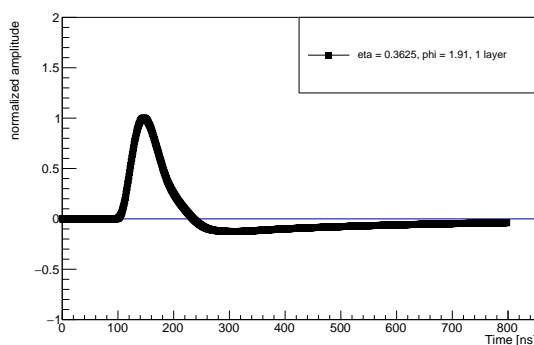
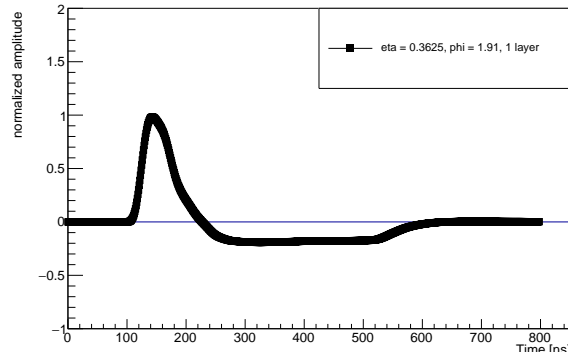


図 5.13: L1Calo でのエネルギー再構成率 [16]。横軸はオフラインエネルギーである。縦軸はそれぞれエネルギーに対し、maximum finder でエネルギーを同定できたかを示している。4 GeV 以上のエネルギーに対しては、maximum finder の選択条件を常に満たしている。青点、赤点は係数の算出方法に違いがあり、パイルアップの影響を最小限にしている方法が赤点である。

この選択条件を評価するために、S-Frame[17] と呼ばれるシミュレーションツールを用いた。このツールは電子や光子の目的事象の大きさが変数になっている。また、パイルアップ事象と熱ノイズは、大きさ以外にも発生確率を変数としてもっており、様々な環境を想定した波形シミュレーションができる。このツールは1つの Super Cell の性能評価のみに使用されるため、複数のセルを利用した R_η やトリガーレートの評価はできない。今回は、デモンストレータから取得したキャリブレーション波形の1つを使用した (図 5.14)。



(1) キャリブレーション波形



(2) 物理波形

図 5.14: S-Frame に使用した理想的な波形 g 。キャリブレーション波形と実験から得られる物理波形で信号の通過する経路が異なり、波形に違いがみえる。そのため、それを考慮した変換を行うことで、右の物理波形が得られた。

図 5.14 右の波形に波高の重み付けを行うことで、あらゆるエネルギーの波形が得られる (図 5.15)。また、Run3 で予想される熱ノイズとパイルアップを考慮してシミュレーションを行った (表 5.3)。10000 個の 5 (20) GeV をもった目的事象に対して、エネルギーの

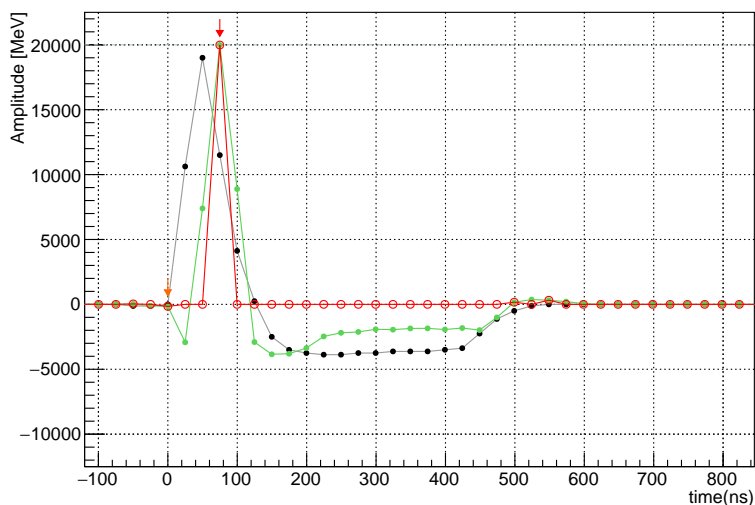


図 5.15: カロリメータの信号 (20 GeV)。デモンストレータから得られた理想的な波形 g に 20 GeV の波高分が重み付けされている。図中の黒点は、その信号を 40 MHz でデジタル化した信号である。緑が Optimal filter 後の出力値である。これは、波高だけでなくアンダーシュートの部分なども出力している。そのため、選択条件を課する (図中の赤点)。赤点は、正しく 20 GeV のエネルギーのみを捉えられている。

入力値に対する出力値の相対的なずれと τ の分布の結果が図 5.16 である。理想的な場合、図 5.16 中の (1)(2) は共に 0 になるべきだが、どちらも一定の広がりを持った分布が得られた。ノイズやパイルアップが入力波形に影響を及ぼし理想的な波形からずれてしまうこ

表 5.3: シミュレーションに用いた熱ノイズとパイルアップ値

	パターン
パイルアップ	0.42 /BC の発生確率で大きさ 625 MeV ($\mu=80$ 程度に対応)
熱ノイズ	mean = 0 MeV, $\sigma=160$ MeV のガウス分布

とで、このような振る舞いがみられる。つまり、目的事象のエネルギーが大きいほどその寄与が小さくなり、広がりは小さくなるはずである。そこで、あらゆるエネルギーに対して図 5.16 と同様の図を作成し、その広がりを示す標準偏差を求めた (図 5.17)。どちらの分布も予期した振る舞いがみられる。

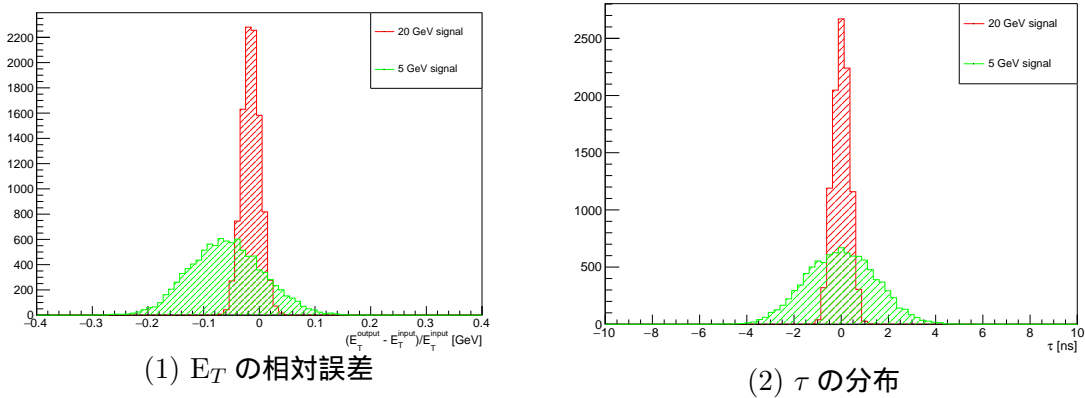


図 5.16: 5 GeV と 20 GeV の目的事象に対する E_T と τ の広がり。 $E_{out}=E_{in}$ と $\tau=0$ が理想的だが、ノイズやパイルアップの影響により一定の広がりをもった分布となる。

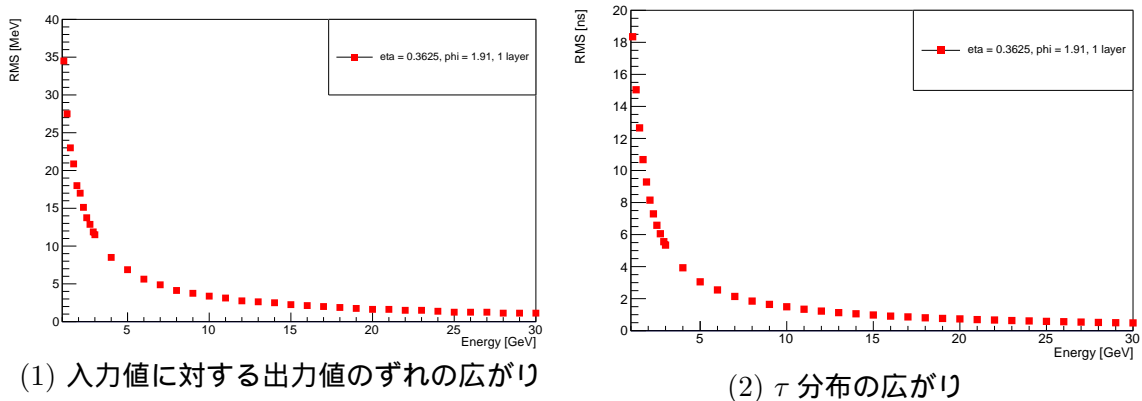


図 5.17: 分布の広がりのエネルギー依存性。どちらの分布も目的事象のエネルギーが低いと RMS が広がることを示している。

分布の広がりがエネルギーに依存するため、 τ_{thre} を決める時に以下の点を考慮に入れる

必要がある。ここで述べるフェイク事象とは、目的事象やパイルアップ事象が入射していないBCで選択条件を満たした事象を指す。

- τ_{thre} が大きくなると、目的事象の検出効率は大きくなり、フェイク事象も増大する。
- τ_{thre} が小さくなると、フェイク事象は減少するが、目的事象の検出効率は小さくなる。
- ファームウェアに実装するため、 τ_{thre} は2の累乗が望ましい。

これらを満たすために、下のようにエネルギー領域ごとに τ_{thre} を変更させた。領域の境界線は、図 5.13 と同様に約 4.0 GeV で目的事象の 100 % の検出効率になるように決めた。図 5.18 は、選択条件後の目的事象の検出効率である。

$$\begin{cases} |\tau| < 2.0 & : 10 \text{ GeV} \leq E_T \\ |\tau| < 8.0 & : -1 \text{ GeV} < E_T < 10 \text{ GeV} \end{cases} \quad (5.3)$$

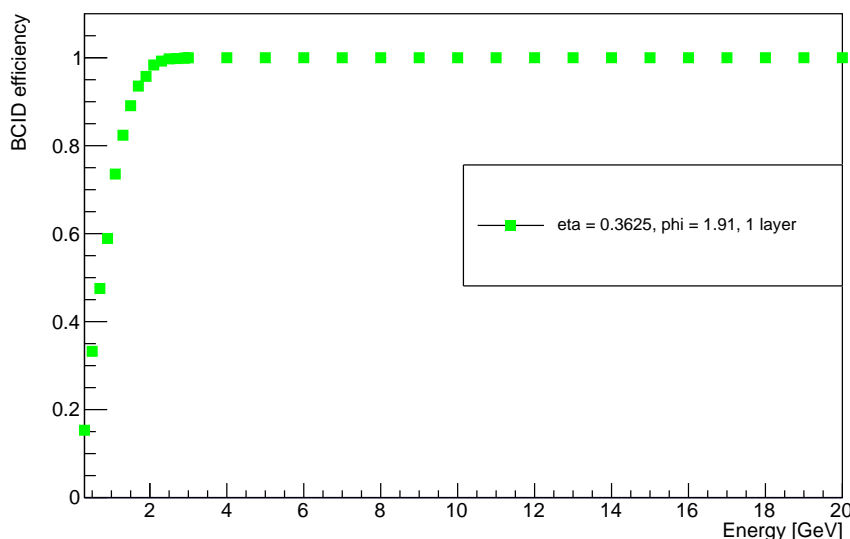


図 5.18: τ の選択条件を課した時の目的事象の信号効率

式 (5.3) は、-1 GeV 以上の負エネルギーも考慮している。これは、最近のシミュレーション結果から、次のことが確認できたからである。正のエネルギーだけをとると、熱ノイズもパイルアップノイズも加算されていき、トータルノイズは高くなる。しかし、負エネルギーをとることで打ち消しあう項も出て来て、トータルノイズが小さくなる (図 5.19)。

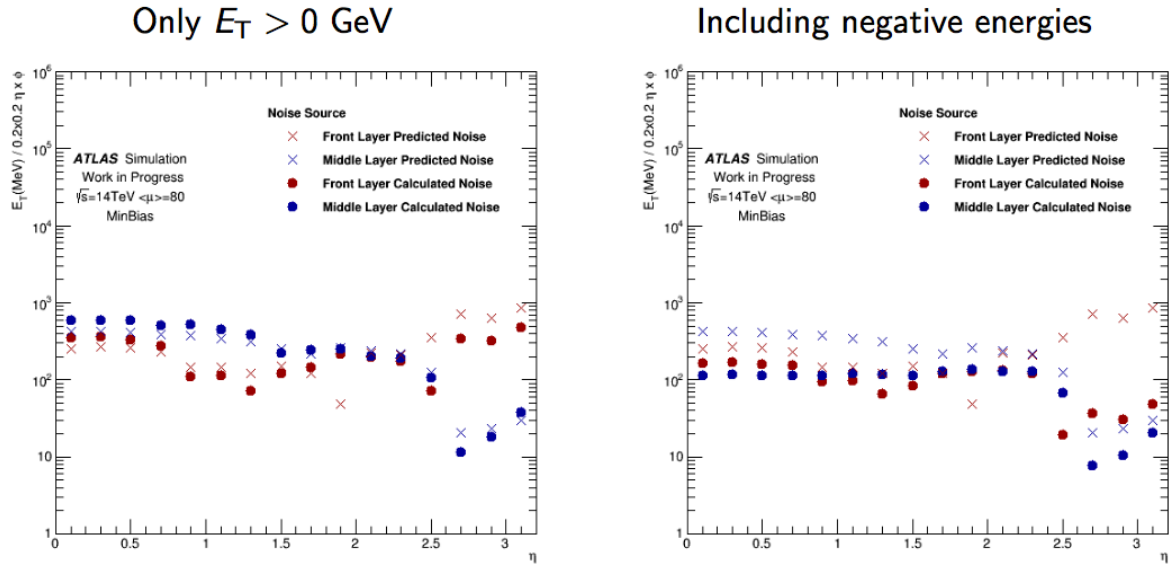


図 5.19: $\mu=80$ におけるノイズのシミュレーション。左図が正エネルギーのみの結果で、右図が負エネルギーも考慮した結果である。負エネルギーを考慮して算出したノイズ値 (Calculated Noise) は、左図よりも小さい。

この選択条件によるフェイクレートは 10 % 程度である。フェイクのほとんどが、1 GeV 以下のエネルギーである。このフェイクレートに対して、トリガーレートにどの程度影響を及ぼすかが今後の課題である。また、同時にこのフェイクレートの改善手法も考えていく。

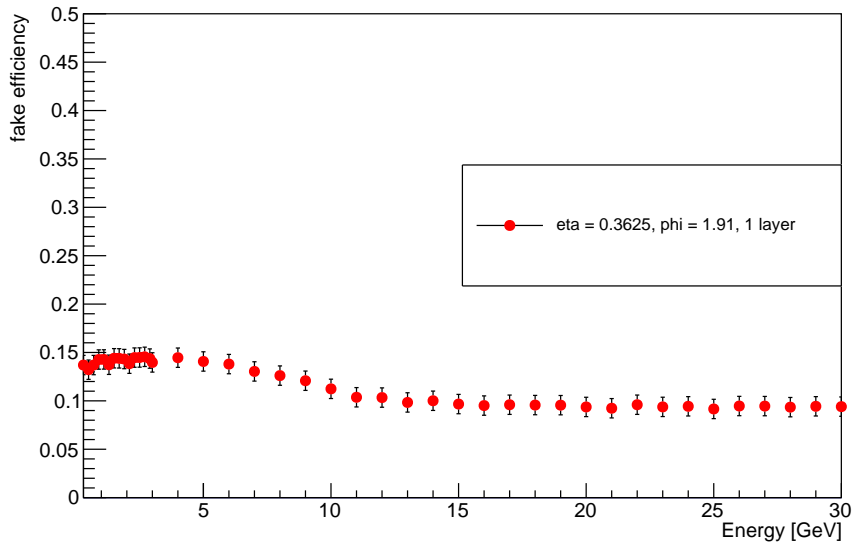


図 5.20: フェイク事象の発生確率。これらのフェイクのエネルギーは 1.0 GeV 以下であるため、single EM object に対してのエネルギー閾値に影響はない。しかし、MET 等の計算で影響を及ぼす可能性がある。

5.3.3 τ の選択条件の設計

Optimal filter から E_T と $E_T \cdot \tau$ が得られるため、式 (5.3) の評価を行うためにファームウェアの中で除算が必要である。しかし、除算のファームウェアは単純でない上に、低レイテンシでの処理が難しい。そのため、除算を必要としない以下の式をファームウェアに実装する。

$$\begin{cases} -2E_T < E_T \cdot \tau < 2E_T & : 10 \text{ GeV} \leq E_T \\ -8E_T < E_T \cdot \tau < 8E_T & : 0 \text{ GeV} \leq E_T < 10 \text{ GeV} \\ 8E_T < E_T \cdot \tau < -8E_T & : -1 \text{ GeV} \leq E_T < 0 \text{ GeV} \end{cases} \quad (5.4)$$

$2(8)E_T$ は、1(3) ビットシフトを E_T に適応することで求められる。負の処理はビット反転を利用する。これを基に、LUT とレジスタのみで τ の選択条件を設計した (図 5.21)。レジスタは図 5.21 中の青枠なので、この設計のレイテンシは3クロックである。また、この選択条件を満たしたエネルギーと共に、Bunch Crossing Assignment Valid flag (BCAV_E flag) を送る。これは、そのバンチに有意なエネルギーをもつ信号があることを示す。

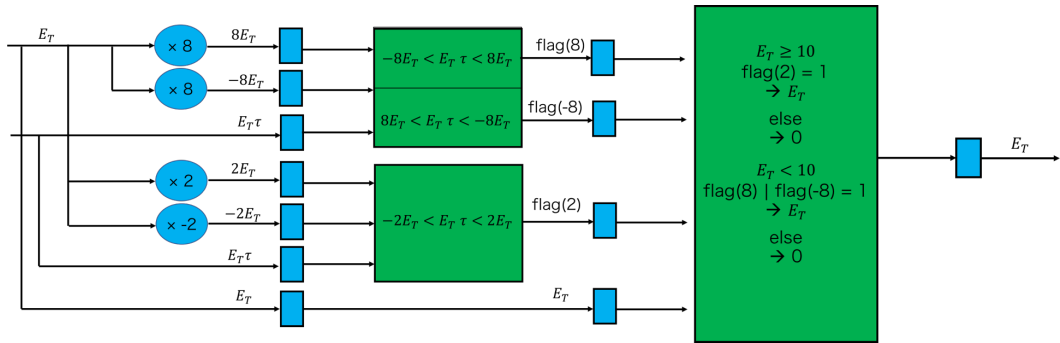


図 5.21: τ の選択条件の設計。ビットシフトを利用して、Filter 後の出力値の-2, 2, -8, 8 倍を算出する。その後、式 (5.4) の 3 つの選択条件の結果が flag として出力される。最後のクロックで、Filter 後のエネルギーと flag のの結果を使用して、最終的なエネルギーを出力する。

5.3.4 エネルギー、係数とペDESTALフォーマット

User Code 内のフォーマットは、LATOME ファームウェアが FEX へ送るフォーマットと密接に関わっている。FEX グループと議論し、eFEX には 10 ビット、jFEX と gFEX には 13 ビットを用いて伝送することにした。LATOME ファームウェアでは、この制限の中でカロリメータ本来の分解能をおとさないエンコーディングフォーマットを実装する。

ATLAS LAr カロリメータの分解能は以下で表される。

$$\frac{\sigma}{E_T} = \frac{a}{\sqrt{E_T}} \oplus \frac{b}{E_T} \oplus c \quad (5.5)$$

第 1 項は確率的な項であり、どの程度サンプリングできるかに関係する。ATLAS では、 $a = 0.1$ 程度である。第 2 項は熱ノイズやパイルアップノイズなどからの寄与を表しており、第 3 項は検出器本来が持つ分解能を表している。設計値では、 $c = 0.7\%$ 程度である。第

1項と第2項はエネルギーが大きくなると小さくなるため、高エネルギーで第3項が主に効いてくる。

ここで、 $V=k\sqrt{E_T}$ と変換した場合、 V の誤差は

$$\sigma_V = \frac{k\sigma}{2\sqrt{E_T}} \quad (5.6)$$

$$= \frac{k}{2\sqrt{E_T}} (a\sqrt{E_T} \oplus b \oplus cE_T) \quad (5.7)$$

$$= \frac{k}{2} \left(a \oplus \frac{b}{\sqrt{E_T}} \oplus c\sqrt{E_T} \right) \quad (5.8)$$

と表される。ここで第1項のみ考え、FEXへ送るビット幅(n)と最大エネルギー E_T^{max} を用いて $k=\frac{2^n-1}{\sqrt{E_T^{max}}}$ と定義する。量子化誤差は一様分布の誤差と同等の $\frac{1}{\sqrt{12}}$ であり、それも考慮すると σ_V は

$$\sigma_V = \sqrt{\left(\frac{ak}{2}\right)^2 + \frac{1}{12}} \quad (5.9)$$

と表される。この表式を用いて、 E_T の誤差を計算すると

$$E_T^R = \left(\frac{V}{k}\right)^2 \quad (5.10)$$

$$\sigma_{E_T^R} = 2\frac{\sigma_V}{k^2} \quad (5.11)$$

$$\rightarrow \frac{\sigma_{E_T^R}}{E_T^R} = 2\frac{\sigma_V}{k^2} k\sqrt{E_T^R} \frac{1}{E_T^R} \quad (5.12)$$

$$\begin{aligned} &= \frac{2\sigma_V}{k\sqrt{E_T^R}} \\ &= \frac{a}{\sqrt{E_T^R}} \sqrt{\left(1 + \frac{1}{3(ka)^2}\right)} = \frac{a}{\sqrt{E_T^R}} R \end{aligned} \quad (5.13)$$

$$R = \sqrt{\left(1 + \frac{1}{3(ka)^2}\right)}$$

FEXへ送るエネルギーはL1Aを発行するために使用される。また、ビット数に制限があることを考えると、情報は高エネルギーで粗く、低エネルギーで細かく送られるべきである。そのため、ノイズの寄与が小さい領域において、式(5.13)の表式はFEXへ送る際のカロリメータの分解能と近似できる。この表式の特徴として、通常の分解能に R の重みがついている。例えば、 $E_T \sim 200$ GeVの領域を10ビットで送ることを考えると、 $R \sim 1.006$ となり分解能の悪化が1%以下に抑えられる。しかしながら、平方根のファームウェアは複雑になりやすく、240 MHz以上で動作するためには4、5クロック必要である。また、平方根のファームウェアの最大動作周波数をシミュレーションすると、図5.4が得られた。全体のファームウェアに実装した場合、図5.4の最大動作周波数はより小さい値になり得る。FEX側で考えると、 V から E_T に戻すロジックも簡単でない。そのため、平方根のエンコーディングは直接実装せず、その近似であるmulti linear region エンコーディング手法を開発した。

表 5.4: 平方根ファームウェアを実装した場合のレイテンシと最大動作周波数

入力のビット幅	出力のビット幅	レイテンシ (240 MHz クロック)	最大動作周波数
20	10	3 クロック	185 MHz
20	10	4 クロック	250 MHz
20	10	5 クロック	280 MHz
22	11	3 クロック	205 MHz
22	11	4 クロック	211 MHz
22	11	5 クロック	250 MHz

multi linear region エンコーディング

CODE	Energy, MeV	Bin width, MeV	
0	< -775	railed	#1
1	[-775 ... -750]	25	
...	#2
94	[1 550 ... 1 575]	25	
95	[1 575 ... 1 600]	25	
96	[1 600 ... 1 650]	50	
97	[1 650 ... 1 700]	50	#3
...	
190	[6 300 ... 6 350]	50	
191	[6 350 ... 6 400]	50	
192	[6 400 ... 6 500]	100	
193	[6 500 ... 6 600]	100	#4
...	
382	[25 400 ... 25 500]	100	
383	[25 500 ... 25 600]	100	
384	[25 600 ... 25 800]	200	#5
385	[25 800 ... 26 000]	200	
...	#6 COARSE
766	[102 000 ... 102 200]	200	
767	[102 200 ... 102 400]	200	
...	

Basic parameters:

10 bits

E_MIN = -800 MeV

dE_MIN = 25 MeV

E_MAX = 200 GeV / ~1 TeV (COARSE)

dE_MAX = 400 MeV / ~100 GeV (COARSE)

CODE	Energy, MeV	Bin width, MeV	
...	#5
768	[102 400 ... 102 800]	400	
769	[102 800 ... 103 200]	400	
...	#6 COARSE
1010	[199 200 ... 199 600]	400	
1011	[199 600 ... 200 000]	400	
1012	[200 000 ... 302 400]	102 400	
1013	[302 400 ... 404 800]	102 400	#6 COARSE
...	
1018	[814 400 ... 916 800]	102 400	
1019	[916 800 ... 1 019 200]	102 400	
1020	>= 1 019 200	railed	#6 COARSE
1021	RESERVED		
1022	INVALID		
1023	SATURATED		

図 5.22: multi linear region エンコーディング (eFEX の場合)

図 5.22 が、eFEX 用の multi linear region エンコーディングである。200 GeV 以上のエネルギーは粗い分解能で送ることで、それ以下のエネルギーの分解能を上げることができる。実際に、 $n=10$ と $E_T^{max}=200$ GeV の場合の V と良い近似になっている (図 5.23)。multi linear region エンコーディングの式は、以下で表せる。

$$M = \frac{E_T + 775}{25} + 1 \quad (-775 \leq E_T \leq 1600 \text{ MeV}) \quad (5.14)$$

$$M = \frac{E_T - 1600}{50} + 96 \quad (1600 \leq E_T \leq 6400 \text{ MeV}) \quad (5.15)$$

$$M = \frac{E_T - 6400}{100} + 192 \quad (6400 \leq E_T \leq 25600 \text{ MeV}) \quad (5.16)$$

$$M = \frac{E_T - 25600}{200} + 384 \quad (25600 \leq E_T \leq 102400 \text{ MeV}) \quad (5.17)$$

$$M = \frac{E_T - 102400}{400} + 768 \quad (102400 \leq E_T \leq 200000 \text{ MeV}) \quad (5.18)$$

$$M = \frac{E_T - 200000}{102400} + 1012 \quad (200000 \leq E_T \leq 1019200 \text{ MeV}) \quad (5.19)$$

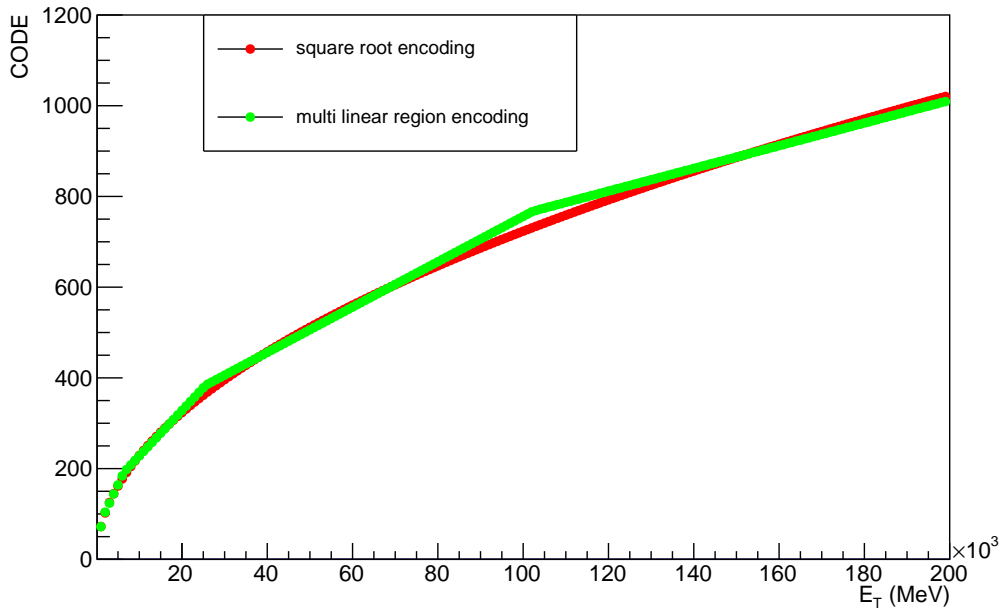


図 5.23: multi linear region エンコーディングと平方根のエンコーディングの振舞い

M の整数部が、図 5.22 中のコードと対応している。他のエネルギー領域でも、同様に表せる。図 5.23 中の Bin width は 1 ビットのもつエネルギーであり、25, 50 100, 200, 400, 102400 MeV の 5 種類を使用している。これらは、25 MeV の 1, 2, 4, 8, 16, 4096 倍となっており、ビットシフトを利用してエンコーディングを構成できる。これが、このエンコーディングの最大の特徴である。平方根の時と同様に、カロリメータの分解能の第 1 項のみ注目し、エンコーディングによる分解能の悪化を示す R を算出する。その際、定数部分を C1, C2、1 ビットのもつエネルギーを x とする。

$$M = \frac{E_T - C1}{x} + C2 \implies \sigma_M = \frac{\sigma}{x} \quad (5.20)$$

$$\begin{aligned} E_T = x(M - C1) + C2 \implies \sigma_{E_T} &= x\sigma_M \\ &= x\left(\sqrt{\frac{a^2}{x^2}E_T + \frac{1}{12}}\right) \\ &= \sqrt{a^2E_T + \frac{x^2}{12}} \end{aligned} \quad (5.21)$$

$$\frac{\sigma}{E_T} = \frac{a}{\sqrt{E_T}} \sqrt{1 + \frac{x^2}{12a^2E_T}} \quad (5.22)$$

$$\implies R = \sqrt{1 + \frac{x^2}{12a^2E_T}} \quad (5.23)$$

この時の R は平方根エンコーディングと異なり、エネルギー依存性をもつ。1.6 GeV 以上で $R \leq 1.01$ が得られ、ほとんど分解能の悪化は発生しない (図 5.24)。jFEX に対しても同様に、multi linear region を基にしたエンコーディングを行なった (図 5.25)。gFEX の詳細なエンコーディングはまだ議論中であるが、1 ビットのもつ最小のエネルギーは 12.5 MeV と決めた。これらのエンコーディングの中で、x の最小値は 12.5 MeV である。そのため、ビットシフトのみでこれら全ての multi linear region エンコーディングが行えるように、Optimal filter 後のエネルギーフォーマットは図 5.26 にした。 $E_T \cdot \tau$ の場合、選択条件ロジックで 3 ビットシフトが使用されることを考慮し、21 ビットフォーマットにする。

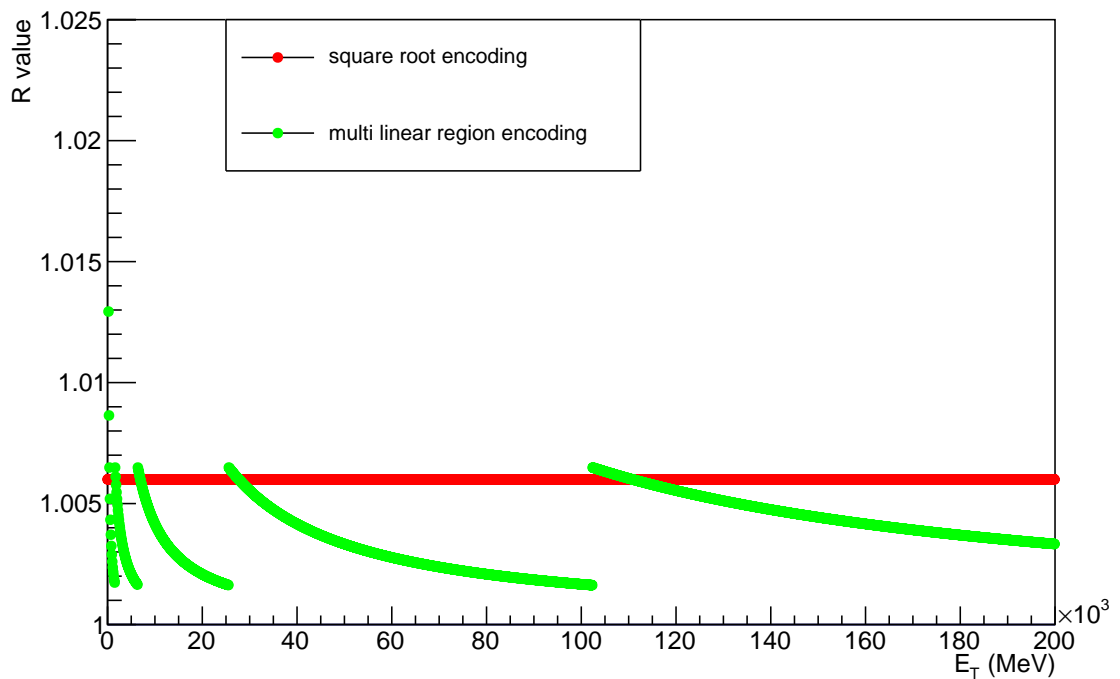


図 5.24: R のエネルギー依存性。square root エンコーディングで R の値は一定だが、multi linear region エンコーディングでは変動する。

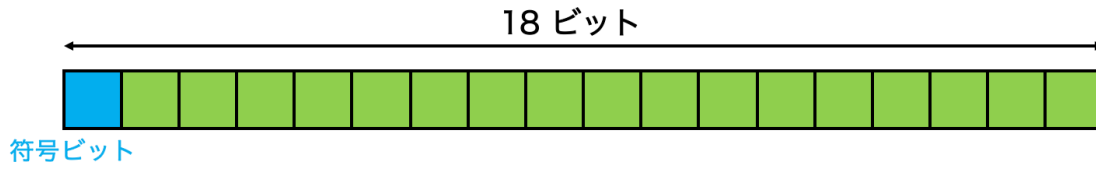
CODE	Energy, MeV	Bin width, MeV	
0	< -3 175	<i>railed</i>	linear region #1
1	[-3 175 ... -3 150]	25	
...	
382	[6 350 ... 6 375]	25	linear region #2
383	[6 375 ... 6 400]	25	
384	[6 400 ... 6 450]	50	
385	[6 450 ... 6 500]	50	linear region #3
...	
766	[25 500 ... 25 550]	50	
767	[25 550 ... 25 600]	50	#4
768	[25 600 ... 25 700]	100	
769	[25 700 ... 25 800]	100	
...	#5
1534	[102 200 ... 102 300]	100	
1535	[102 300 ... 102 400]	100	
1536	[102 400 ... 102 600]	200	#6
1537	[102 600 ... 102 800]	200	
...	
3070	[409 200 ... 409 400]	200	#7
3071	[409 400 ... 409 600]	200	
...	

Basic parameters:

12 bits (Et) + 1 bit for Saturation flag
 E_MIN = -3.2 GeV E_MAX = 800 GeV
 dE_MIN = 25 MeV dE_MAX = 400 MeV

CODE	Energy, MeV	Bin width, MeV	
...	#5
3072	[409 600 ... 410 000]	400	
3073	[410 000 ... 410 400]	400	
...	#6
4046	[799 200 ... 799 600]	400	
4047	[799 600 ... 800 000]	400	
4048	>= 800 000	<i>railed</i>	
4049 ... 4094	RESERVED		
4095	INVALID		

図 5.25: multi linear region エンコーディング (jFEX の場合)



1ビットが持つエネルギー : 12.5 MeV
 →最大エネルギー : ~ 1.6 TeV
 →最小エネルギー : ~ -1.6 TeV

図 5.26: エネルギーフォーマット。12.5 MeV のデジタル化で 1 TeV 程度を考慮するためには、18 ビットが必要である。

次に図 5.26 のフォーマットに変換する方法を考える。それぞれの ADC データは mV scheme となっている。LATOME ファームウェア内で余計なロジックを実装しないために、ファームウェアに実装する係数に mV scheme の ADC から 12.5 MeV に変換させる因子を組み込ませる。また、ペDESTAL の分解能を維持するために、ペDESTAL には 3 ビットの固定小数点を持たせる。これらを基にしたデータフローは、図 5.27 である。

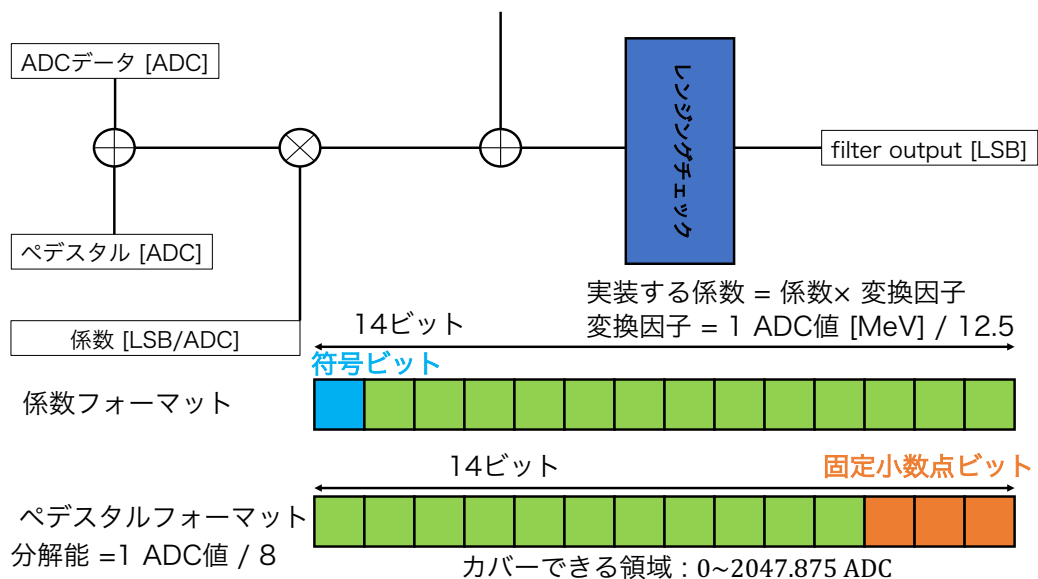


図 5.27: Filter 計算の流れ。最初に ADC からペダスタルが差し引かれる。ここで、ペダスタルは 3 ビットの固定小数点を導入している。ADC データとペダスタルの固定小数点の位置を合わせるために、ADC データの下位 3 ビットに 0 を取り付けて計算を行う。その後、変換因子が組み込まれた係数と乗算を行うことで、そのエネルギーは 12.5 MeV でデジタル化される。係数が n ビットの固定小数点をもっている場合、出力は $n+3$ ビットの固定小数点をもつ。

係数とペダスタルが 14 ビットのため、Filter の出力値が 18 ビットで表現できない値になり得る。これを考慮するために、レンジングチェックと定義したロジックを導入した。

レンジングチェック

DSP Block の出力から 18 ビットを抜き出す。その際に、予想しない波形が LATOME に入ること、図 5.28 で示した例が起り得る。これを防ぐためにレンジングチェックは、18 ビットで表現できないほど大きい (小さい) 値は、18 ビット最大値 (0) としてみなす (図 5.29 上)。 $E_T \cdot \tau$ の場合、21 ビットで表現できない値は τ が 0 から大きく離れているため、選択条件を通さないように 21 ビットの最大値として処理される (図 5.29 下)。2 つのレンジングチェックにより、正確な計算結果が構築できた。

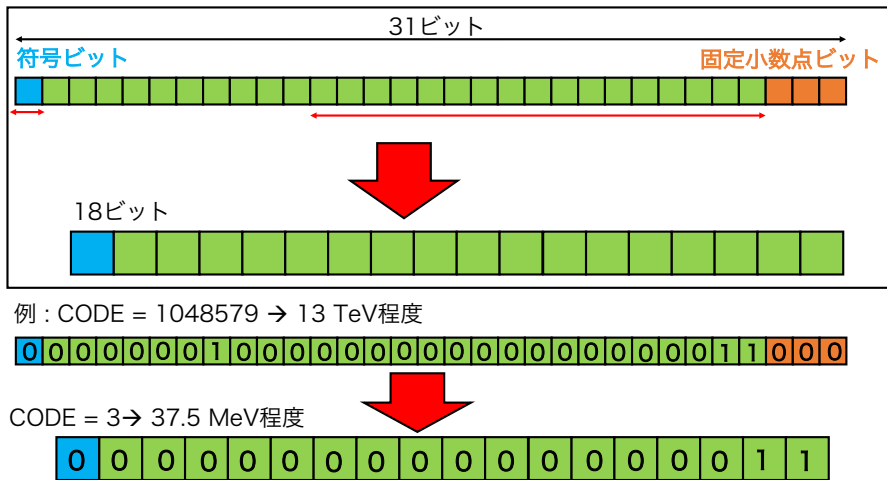


図 5.28: DSP Block の出力からエネルギーを抜き出す方法と誤った抜き出しの例。基本的に、DSP Block から固定小数点を抜いた下位 17 ビットと符号ビットが抜き出される。もしこの範囲を超える値が DSP Block で得られた場合、間違ったエネルギーが出力される。図の場合、13 TeV の出力結果であるが、抜き出しの失敗によって 37.5 MeV のエネルギーとして FEX へ送られる。

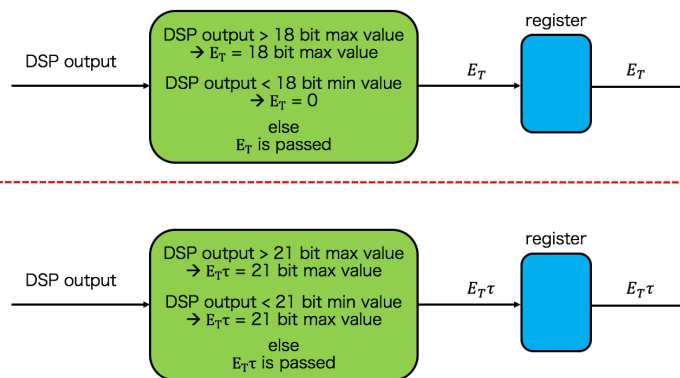


図 5.29: レンジングチェックの設計

5.3.5 サチュレーションした波形の同定方法

mV scheme を採用しているため、デジタル化によるサチュレーションは発生しない。フロントエンドで増幅された信号は最大 3.3 V の波形となるが、高エネルギーの信号が入射されると正しく増幅されずに予想した形の波形にならない (図 5.30)。Optimal filter は波形の形を仮定しているため、正しいエネルギーを同定できない。しかし、高エネルギーの事象は新粒子起源の可能性をもち、絶対に逃してはならない信号である。

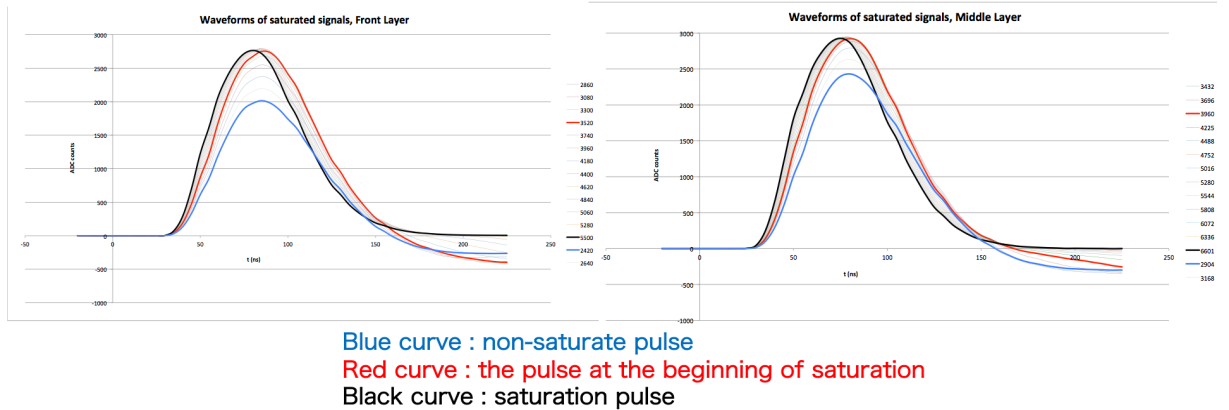


図 5.30: デモンストレータから得られたサチュレーションした波形 [18]。左が第 1 層、右が第 2 層で、どちらも $\eta=0$ 付近の Super Cell の信号である。青がサチュレーションしていない波形、赤と黒がサチュレーションした波形である。ピークの値はサチュレーションし、パイポーラのネガティブ部分が減っていく特徴がある。

サチュレーションした波形に 2 種類の Optimal filter を適応すると、特に $N-1 \sim N+1$ BC で通常の波形と振る舞いがかわることがわかる (図 5.31)。サチュレーション前は、 E_T の増加と共に $E_T \cdot \tau$ も増加し、線形性が見られる。しかし、サチュレーションするとこの線形性は崩れる。 N BC での τ は 0 から大きく離れてしまい、サチュレーションした事象は τ の選択条件を満たさず、逃してしまう。

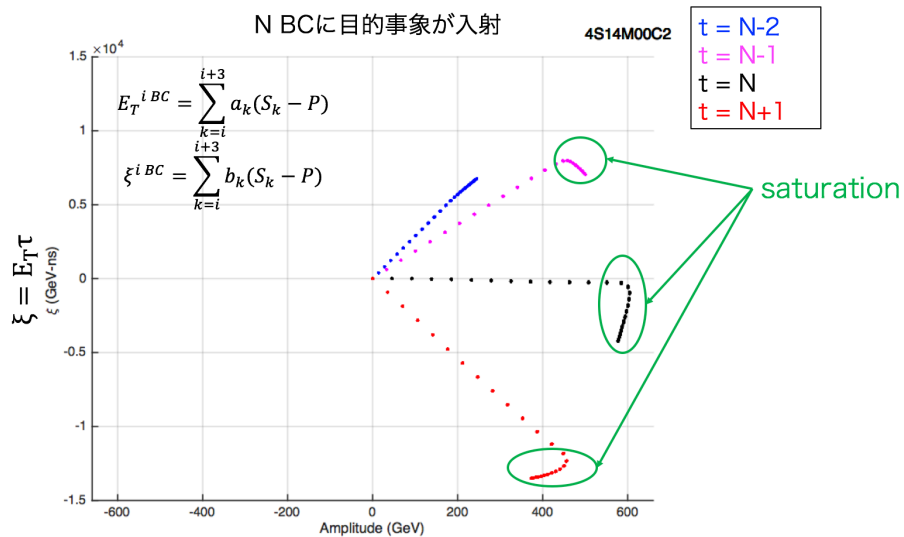


図 5.31: $E_T - E_T \cdot \tau$ の相関図 [18]。 $\eta=0$ 付近で第 2 層の Super Cell 信号である。横軸が E_T 、縦軸が $\xi = E_T \cdot \tau$ であるため、傾きが τ の線形性が見られる。BC ごとに得られる τ の値は異なるため、 $N-2 \sim N+1$ BC で異なる直線が見られる。図中の緑枠がサチュレーションを起こしている部分であり、線形性が崩れている。

サチュレーションした信号を記録するために、どのバンチにその信号が入射されたかを同定する事は非常に重要な点である。また、サチュレーションで波形が変わるため、その前後に通常の波形が来ても正しくエネルギーを求められない可能性がある。そのため、サ

チューレーションの影響があるバンチという情報も残しておくべきである。そこで、私は図 5.31 を利用して、ファームウェア内では saturation flag とサチューレーションした波形に対する $BCAV_S$ flag を実装した。このフラグの条件は、以下である。

$$\cdot \xi_{min}^{N-1 BC} < \xi^{N-1 BC} < \xi_{max}^{N-1 BC} \quad , \quad E_T^{N-1 BC} > E_{thre}^{N-1 BC} \quad (5.24)$$

$$\implies \text{saturation flag} = 1 \text{ and } BCAV_S = 0$$

$$\cdot \xi_{min}^{N BC} < \xi^{N BC} < \xi_{max}^{N BC} \quad , \quad E_T^{N BC} > E_{thre}^{N BC} \quad (5.25)$$

$$\implies \text{saturation flag} = 1 \text{ and } BCAV_S = 1$$

これに加えて、サチューレーションの影響があることを示すために、saturation flag は立ち始めてから 2 BC は無条件で立て続ける。この条件式で与えられている定数は、Super Cell 毎に図 5.31 を作成して算出する。サチューレーションによる影響があり (saturation flag = 1)、バンチのエネルギーが同定できていない場合 ($BCAV_E = 0$ & $BCAV_S = 0$)、さらに Trigger Tower 内の Super Cell のエネルギーを利用してバンチのエネルギーが同定できるか調べる。

BCAV 再同定方法

BCAV は、選択条件による同定手法 ($BCAV_E$) とサチューレーション波形の同定手法 ($BCAV_S$) の 2 つを用いて、

$$BCAV = BCAV_E | BCAV_S$$

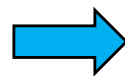
と表せる。 $BCAV_S$ において、N-1 BC の条件を満たすが N BC の条件を満たさなかった場合は、 $BCAV_S = 0$ となり、サチューレーション波形を同定できない (図 5.32)。

シミュレーションで 700, 1500, 3500, 7000 GeV の電子を入射させ、第 2 層の Super Cell がサチューレーションした場合について考える。そのまわりの Super Cell (Trigger Tower 内の Super Cell) が、2.5 GeV 以上のエネルギーを持ちサチューレーションしていないかを調べた (図 5.33)。7000 GeV の電子に対しても、最低 1 つはこの条件を満たす Super Cell が存在する。これは、高エネルギーの粒子が他の Super Cell にも影響を及ぼすことを示しており、これを利用することで高エネルギー粒子がどのバンチにエネルギーを落としたかを同定できる。そのため、このロジックをファームウェアに実装した (図 5.34)。Configurable remapping で Trigger Tower 内の Super Cell はパラレルになっているため、2 クロックのレイテンシで処理できる。また、これは LUT とレジスタのみで構成されたため、容易に実装可能である。

サチューレーションした波形や BCAV 再同定方法については、シミュレーションやデモンストレータのデータを用いた評価がまだ十分でない。具体的には、BCAV 再同定方法を導入した時に誤って BCAV を立ててしまう頻度やエネルギーの条件 (2.5 GeV) 等である。これらは今後の課題である。

$$\xi_{min}^{N-1 BC} < \xi^{N-1 BC} < \xi_{max}^{N-1 BC}$$

$$E_T^{N-1 BC} > E_T^{N-1 BC}$$



- saturation flag = 1
- $BCAV_S = 0$

$$\xi_{min}^{N BC} < \xi^{N BC} < \xi_{max}^{N BC}$$

$$E_T^{N BC} > E_T^{N BC}$$



- saturation flag = 1
- $BCAV_S = 1$



- saturation flag = 1
- $BCAV_S = 0$



- N-1 BCのサチュレーションの条件を満たしているが、N BCのサチュレーションの条件を満たさない場合
- サチュレーションパルスはN BCに来たが、たまたまN BCで条件が通らなかった。
- サチュレーションパルスは来ていないのに、たまたまN-1 BCで条件を通ってしまった。

図 5.32: サチュレーションした信号に対する処理

N BC で条件を満たさない場合、N-1 BC の影響で saturation flag は立つ。

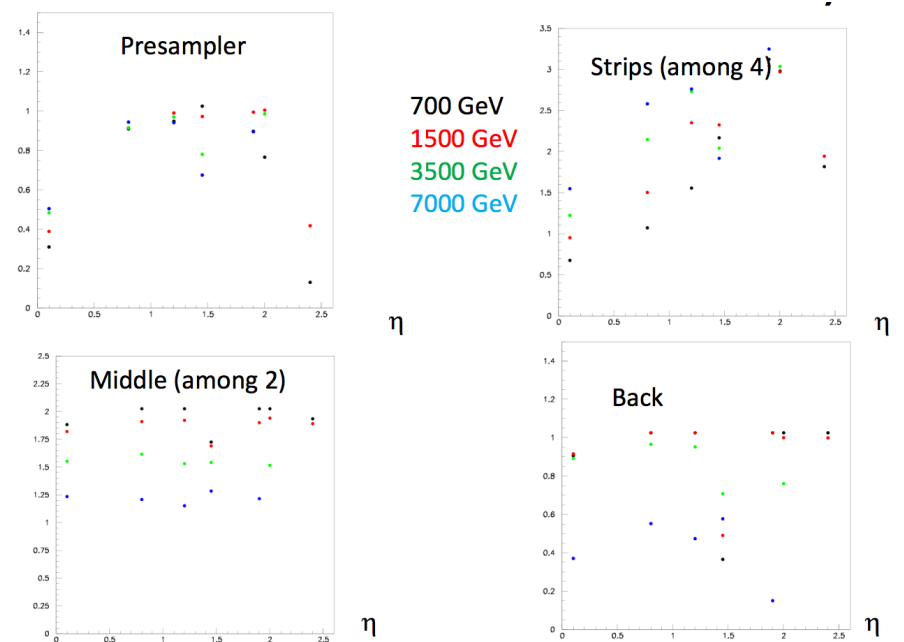


図 5.33: 2.5 GeV 以上でサチュレーションしていない Super Cell の η 分布 [19]。 $\eta=0.1, 0.8, 1.2, 1.45, 1.9, 2.0, 2.4$ に 700, 1500, 3500, 7000 GeV の電子を入射している。 y 軸は、その場合の 2.5 GeV 以上かつサチュレーションしていない Super Cell 数である。

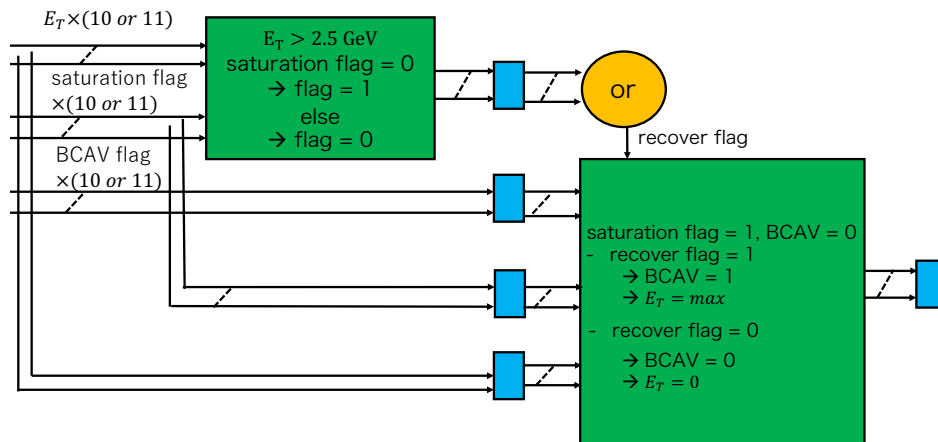


図 5.34: BCNAV 再同定ファームウェア。saturation flag = 1 かつ BCNAV = 0 の場合に、BCNAV 再同定を行う。

5.3.6 コンフィグレーションの変更方法

ファームウェアが FPGA に実装された後、係数やペDESTAL は PC から IP Controller を介してそれぞれ設定される。User Code 内で IP Controller に関わる経路は、100 MHz クロックで動作する。係数やペDESTAL を記録しているサーキュラーバッファと IP Controller が通信するならば、RAM が 100 MHz と 240 MHz の 2 つのクロックドメインを持つデュアルクロックでなければならない。さらに、IP Controller からの要求でデータを読み出すために、240 MHz とは別に 100 MHz の出力を持つ必要がある。これらの機能を持っているデュアルクロック、デュアルポート RAM は、M20K のみ使用できる (図 5.35)。しかし、サーキュラーバッファに M20K を使用すると、RAM の使用率が大幅に増える。そのため、私は M20K RAM と MLAB の両方を使用し、IP Controller と通信を行う設計を開発した。(図 5.36)。

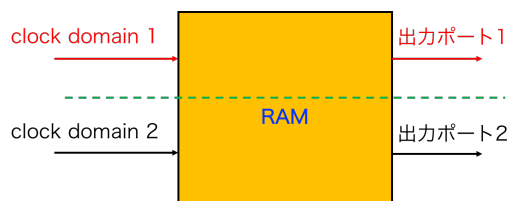


図 5.35: デュアルクロック、デュアルポート RAM。2 つのクロックで動作し、入出力とも 2 つある。User Code の場合、240 MHz クロックは DSP Block のインプットと接続され、常に係数等を供給する。そのため、自由な読み出しを行うためには必ずもう一つのポートが必要である。図 3.12 から、デュアルポート RAM は、M20K のみサポートしている。

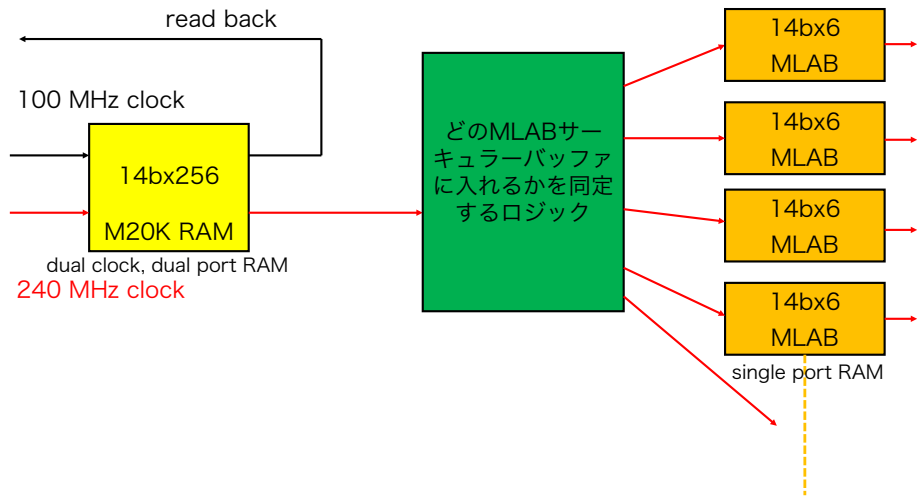


図 5.36: 係数、ペダスタルのコンフィグレーション設計。1つの積和計算に M20K RAM は1つ使用される。IP Controller から 100 MHz クロックで送られてくるデータは、M20K RAM に格納される。M20K RAM に格納されたデータは、240 MHz クロックでそれぞれ正しい MLAB へ配布される。どの係数かについての情報は図 5.37 で定義されたアドレスからわかる。配布するタイミングについても、IP Controller から信号が送られてくる。100 MHz クロックでの読み出しは、サーキュラーバッファでなく M20K から行われる。

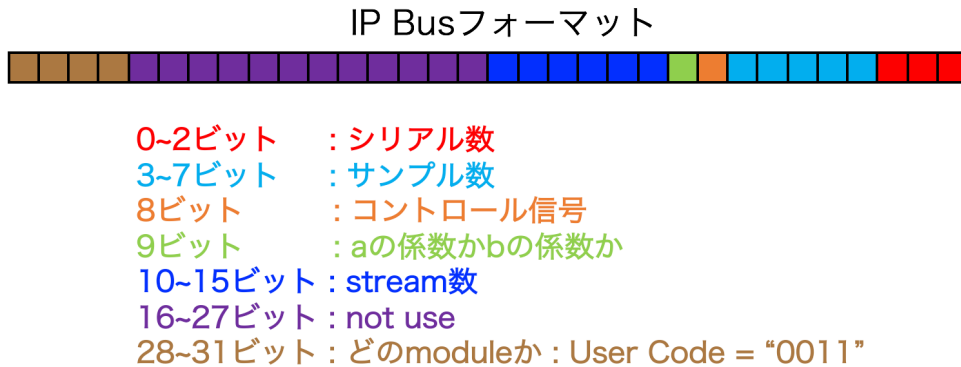


図 5.37: IP Controller からのフォーマット。下位 3 ビットは、サーキュラーバッファで格納されるアドレスに対応する。3~7 ビットは、図 5.37 中にあるどの MLAB に配布するかを決定するのに使用される。8 ビット目は、M20K RAM のデータをサーキュラーバッファへ送るコントロール信号である。9 ビット目は、 E_T か $E_T \cdot \tau$ のどちらを算出する係数かを示す信号である。10~15 ビットは、どの stream かを識別する信号である。

5.3.7 Baseline Correction

LHC は計 3564 バンチであり、陽子が詰められているバンチとそうでないバンチが混ざった構造をもつ。詰められているバンチは 72 個続き、トレインと呼ばれる。各トレインの始めにおいて、前のバンチにカロリメータの信号が入っておらず、バイポーラ波形のネガティブパートと信号が重ならない。このため、ペダスタルの平均値を計算すると他のバンチよりも高くなる傾向が見える (図 5.38)。このシフトは数 ADC であるが、この影響によってトレインの始めの MET トリガーレートが上昇する (図 5.39 (1))。トリガーレート

のバンチ依存性をなくすために、ベースラインコレクションが Run2 から行われている。これにより、MET トリガーレートのバンチ依存性はなくなった (図 5.39 (2))。

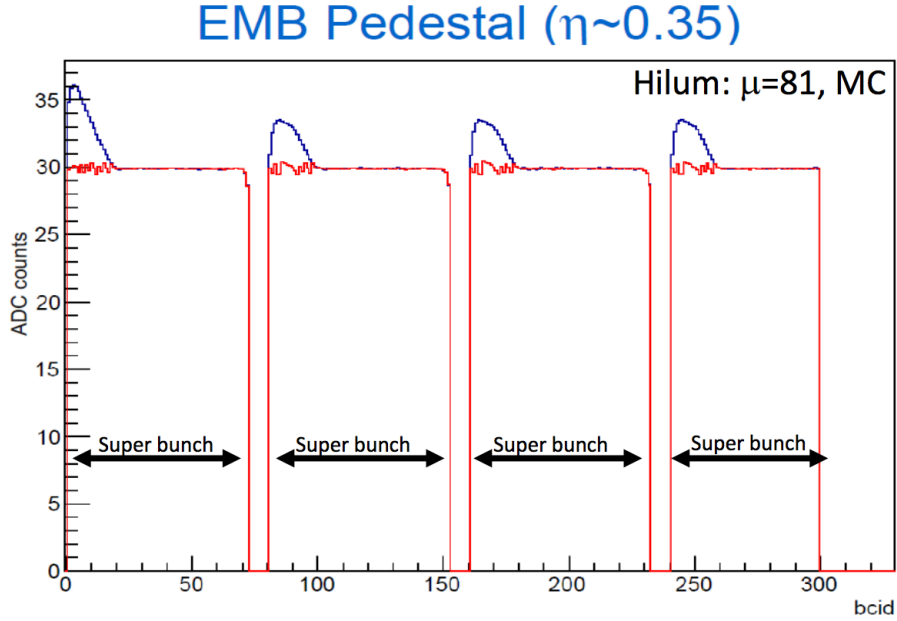
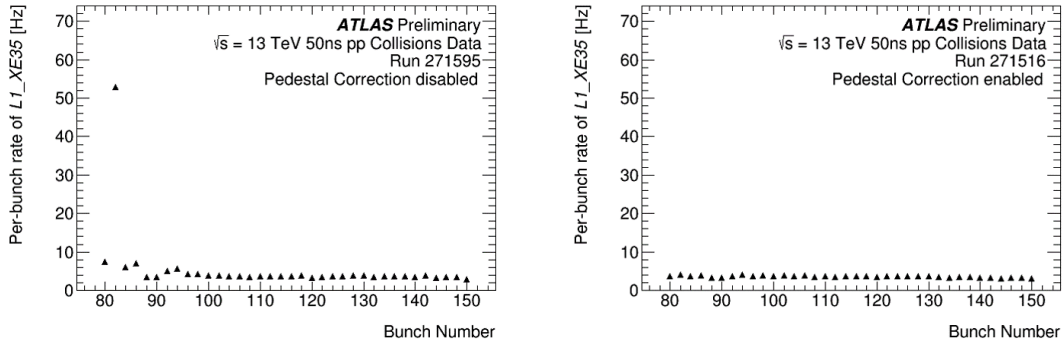


図 5.38: ペDESTALのバンチ依存性 [20]。2, 3, 4 目目のトレインの間は、空白バンチが 8 つある。一方、1 目目のバンチの前は、それよりも多く空白バンチをもっている。それによる影響がペDESTALに表れている (図中の青部分)。



(1) ベースラインコレクション前

(2) ベースラインコレクション後

図 5.39: MET のトリガーレートのバンチ番号依存性 [16]。トレインの始めの MET トリガーが大きくなっている。ペDESTALが他のバンチよりも高いため、測定されたエネルギーが小さくみえる。ベースラインコレクションにより、MET トリガーレートのバンチ依存性がなくなっている。

j BC 目のベースラインコレクションの因子 f_{corr}^j は、 2^N 個の j BC 目の Optimal filter の出力値の平均をとることで算出される。

$$f_{corr}^{jBC} = \sum_{i=0}^{2^N-1} \frac{E_{T,ith}^{jBC}}{2^N} \quad (5.26)$$

$$E_T^{corr} = E_T^{jBC} - f_{corr}^{jBC} \quad (5.27)$$

しかしながら、Run2で実装されているベースラインコレクションのロジックは、LATOMEに実装できない。図 5.40 が、Super Cell 用に設計を変更したものである。1 つ目の RAM の中にバンチごとのデータが記録され、LHC の軌道周期ごとに次々加算されていく。 2^N 回の加算後、ビットシフトを行うことで f_{corr} が算出される。この値は、2 つ目の RAM に記録される。つまり、1 stream に最低 2 つの RAM が必要である。この f_{corr} は Run 中に更新されていき、更新時間は N の値で変わる。RAM の使用率は N に依存するため、十分なサンプル数で f_{corr} を算出する事は不可能である (表 5.5)。

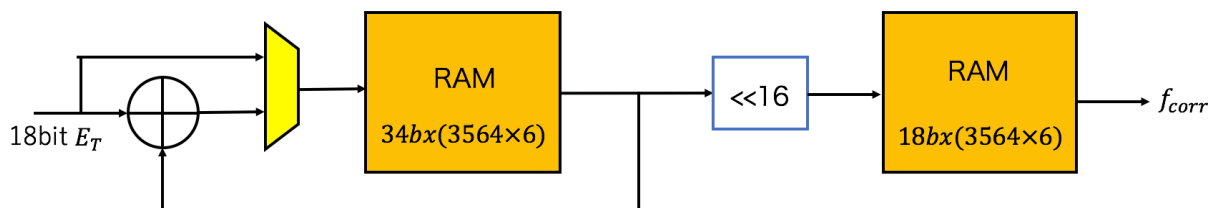


図 5.40: ベースラインコレクションの設計 ($N = 16$)。LATOME は、1 stream に 6 つの Super Cell がシリアルに入ってくるため、RAM のサイズは 3564×6 が必要である。平均をとるサンプル数に依存して、1 つ目の RAM のビット幅が変わる ($18+N$ ビット)。

表 5.5: ベースラインコレクションを実装する際の様々なパラメータ

N	RAM の使用率	更新時間
16	127 %	5.8 s
14	122 %	1.5 s
12	117 %	0.36 s
10	112 %	0.09 s
8	108 %	22.8 s
6	103 %	5.7 ms
4	98 %	1.4 ms
2	93 %	0.4 ms

そこで、LATOME 用のベースラインコレクションを実装するために 2 つのアイデアが考えられている。どちらのアイデアもまだ十分な議論がなされておらず、ベースラインコレクションは現段階で実装されていない。これも、今後の課題の 1 つである。

- LATOME ファームウェアで直接ベースラインコレクションは計算せず、 f_{corr} のみ RAM に蓄えておく。この値は PC で求められ、IP controller を介して User Code に送られる。
- f_{corr} に対し、チャンネル間の依存性やチャンネル自身の時間依存性を正確にファームウェア上で計算させる。これにより、少ない f_{corr} の計算で全チャンネルの値が決定できる。

5.3.8 ブロックダイアグラム

前節までに説明した設計を User Code に効率良く配置するために、図 5.41 のブロックダイアグラムを作成した。Configurable remapping から送られてきたデータ群は、Filtering

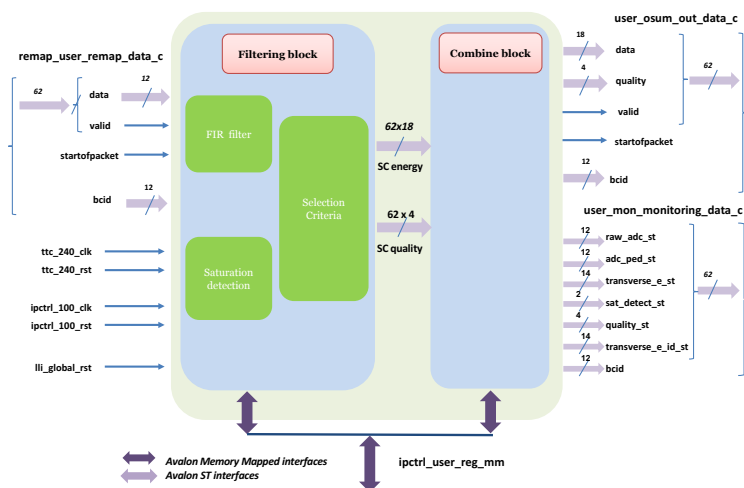


図 5.41: User Code Block[14]

block に入る。このブロックは 3 つのモジュールで構成されており、FIR Filter で E_T , Saturation detection で $E_T \cdot \tau$ が Optimal filter によって求められる。Selection Criteria は E_T , $E_T \cdot \tau$ を受け取り、 τ の選択条件とサチュレーションの同定を平行に行う。Combine block は Trigger Tower 内の Super Cell のエネルギーから BCAV を再同定を行う。

User Code は、出力信号を 2 つのモジュールへ伝送する。1 つはメイン経路である Output summing であり、 E_T と quality 信号を送る。quality 信号は 4 ビット信号として定義され、saturation flag と BCAV flag はこの中に含まれている。残りの 2 ビットは現在使用していない。もう一つは TDAQ/Monitoring であり、以下の信号を User Code は送る。

- ADC データ
- ADC データからペDESTALを差し引いた信号
- Combine block に入る前の E_T
- Combine block 後の E_T
- quality 信号
- saturation flag と BCAV flag
- BCID

また、設計した User Code のレイテンシは 4.5 BC となり、要求値 5.0 BC を満たした (表 5.6)。

表 5.6: User Code のレイテンシ

モジュール名	レイテンシ [BC]
FIR Filter / Saturation Detection	3.67
Selection Criteria	0.5
Combine Block	0.33
User Code 全体	4.5

5.4 ファームウェア動作試験

動作検証のために、各モジュールごとの試験が行われている。本論文では、User Code の試験結果を述べる。

5.4.1 インプット、チェッカーテストベンチ

User Code の動作試験に、インプットジェネレータとチェッカーのテストベンチモジュールが必要である。どちらもメモリが搭載されており、試験用データを記録しておく。ジェネレータには 2 種類のデータを入れる。

- データ、係数、ペDESTAL 全て乱数で生成された値
- LAr カロリメータの波形

チェッカーは、ジェネレータを基に PC で計算された値を RAM に格納する。User Code モジュールで処理された信号はチェッカーで格納された値と比較され、同じ場合は checker flag=1 の信号を FPGA から出力する。

乱数データを入れることで、あらゆるパターンの信号プロセスの試験が行える。しかし、これは純粋なファームウェアの動作試験のためであり、得られるエネルギー等に物理的意味はない。そのため、LAr カロリメータの波形もジェネレータとして使用する。チェッカーで比較する信号は、図 5.41 で出力される信号と DSP Block の出力値である。図 5.42 が試験用のファームウェアであり、このファームウェアを FPGA に実装して試験を行った。

5.4.2 シミュレーションによる動作確認

各ロジックの動作確認を行うために、表 5.7 で示したパラメータを設定した。これらのパラメータの変更は、ファームウェアのコンパイル前に適応される。例えば、図 5.43 は、 τ の選択条件を接続する場合としない場合のファームウェアを記述している。ファームウェア中でプリプロセッサを使用できるようにすることで、パラメータに応じたファームウェアが容易に作成できる。また、同様にパラメータに応じたサンプル数の Filter も実装できる。このような処理を可能にすることで、様々なシミュレーションができる環境を整えた。具体的には、表 5.8 で書かれているマイルストーン M1~M5 を行っている。M1 は、 τ の選択条件と BICAV 再同定ロジックをバイパスさせ、Optimal filter の計算とエネルギーの

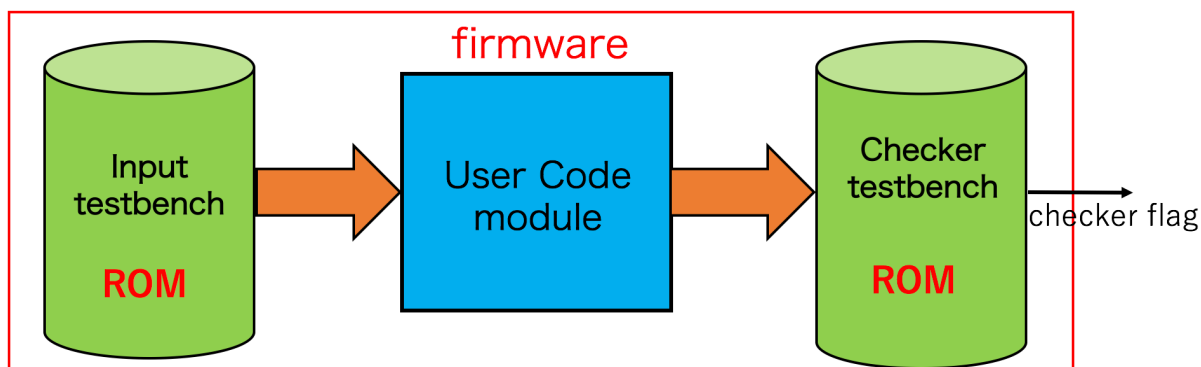


図 5.42: 評価用 User Code ファームウェア。

表 5.7: User Code のパラメーター一覧

パラメータ	デフォルト値
User Code 内に入るインプットのストリーム数	62
FIR Filter 内の filter のサンプル数	4
Saturation Detection 内の filter のサンプル数	4
DSP Block から E_T を抜き出す際の LSB	3
DSP Block から $E_T \cdot \tau$ を抜き出す際の LSB	3
τ の選択条件	true
BCAV 再同定ロジック	true
DSP Block の出力値のデバッグ	false
ベースラインコレクション	false

抜き出しを確認する。M2 では、 τ の選択条件を接続させる。M3 では、BCAV 再同定ロジックの評価をするために、stream 数を 10 にして行う。M4 では、デフォルト値での User Code ファームウェアの評価を行う。M5 では、Filter のサンプル数や抜き出すビット等を変更してもファームウェアが正しく機能しているかを確認する。ベースラインコレクションの設計はまだ完成していないため、今回の動作試験では対象外とした。ベースラインコレクションを入れた試験をすることが今後の課題である。

```

#if USER_TAU_SELECTION_ENABLE
  if(transverse_e_second_reg > Boundary_Energy)
    begin
      transverse_e_tau_selection_reg <= (high_energy_flag_reg == 1'b1) ? transverse_e_second_reg : (USER_ENERGY_WIDTH[1'b0]);
      BCAV_E_flag_reg <= (high_energy_flag_reg == 1'b1) ? 1'b1 : 1'b0;
    end
  else if(transverse_e_second_reg <= Boundary_Energy && transverse_e_second_reg > Min_Negative_Energy)
    begin
      transverse_e_tau_selection_reg <= (low_energy_flag_reg == 1'b1 || negative_energy_flag_reg == 1'b1) ? transverse_e_second_reg : (USER_ENERGY_WIDTH[1'b0]);
      BCAV_E_flag_reg <= (low_energy_flag_reg == 1'b1 || negative_energy_flag_reg == 1'b1) ? 1'b1 : 1'b0;
    end
  else
    begin
      transverse_e_tau_selection_reg <= (USER_ENERGY_WIDTH[1'b0]);
      BCAV_E_flag_reg <= 1'b0;
    end
  end
#endif
transverse_e_tau_selection_reg <= transverse_e_second_reg;
BCAV_E_flag_reg <= 1'b1;
endif

```

τ の選択条件

τ の選択条件はバイパス

ソースコードにプリプロセッサを導入

プリプロセッサによってTrueとFalse用のコードがコンパイルするときに自動生成されるようにしておく。
 → 実際使用されるコードは、True/Falseのどちらかのコード

```

//DSP Blocks
generate
  for(dsp_number = 0 ; dsp_number < USER_FILTERING_IN_FIR_FILTER_TAPS_NB / 2 - 1; dsp_number = dsp_number + 1)
    begin : dsp_block
      for(Blocks dsp_block_nb (
        accumulate ( 1'b0 ), // accumulate port
        .aclr ( {1'b0, 1'b0} ), // asynchronous clear port
        .ax ( coefficient_bank(2*dsp_number + 1) ), // coefficient
        .ay ( {1'b0, data_shift(2*dsp_number), 3'b000} ), // ADC data
        .az ( {1'b0, pedestal_bank(2*dsp_number + 1)} ), // pedestal
        .bx ( coefficient_bank(2*dsp_number) ), // coefficient
        .by ( {dsp_number == 0 ? {1'b0, data_in, 3'b000} : {1'b0, data_shift(2*dsp_number-1), 3'b000} ), // ADC data
        .bz ( {1'b0, pedestal_bank(2*dsp_number)} ), // pedestal
        .chainin ( chainout[dsp_number-1] ), // chainin port
        .chainout ( chainout[dsp_number] ), // chainout port
        .clk ( {2'b00, etc_240_clk} ), // clock
        .ena ( 3'b001 ), // enable
        .loadconst ( 1'b0 ), // load constant port
        .negate ( 1'b0 ), // negate port
        .resulta ( dsp_output[dsp_number] ), // output port
        .sub ( 1'b0 ) // sub port
      );
    end
endgenerate
DSP_Blocks_no_chainin dsp_block_no_chainin (
  .accumulate ( 1'b0 ), // accumulate port
  .aclr ( {1'b0, 1'b0} ), // asynchronous clear port
  .ax ( coefficient_bank(USER_FILTERING_IN_FIR_FILTER_TAPS_NB - 1) ), // coefficient
  .ay ( {1'b0, data_shift(2*USER_FILTERING_IN_FIR_FILTER_TAPS_NB / 2 - 1)}, 3'b000 ), // ADC data
  .az ( {1'b0, pedestal_bank(USER_FILTERING_IN_FIR_FILTER_TAPS_NB - 1)} ), // pedestal
  .bx ( coefficient_bank(USER_FILTERING_IN_FIR_FILTER_TAPS_NB - 2)} ), // coefficient
  .by ( {1'b0, data_shift(2*USER_FILTERING_IN_FIR_FILTER_TAPS_NB / 2 - 1)}, 3'b000 ), // ADC data
  .bz ( {1'b0, pedestal_bank(USER_FILTERING_IN_FIR_FILTER_TAPS_NB - 2)} ), // pedestal
  .chainin ( chainout(USER_FILTERING_IN_FIR_FILTER_TAPS_NB / 2 - 1) ), // chainin port
  .chainout ( etc_240_clk ), // chainout port
  .clk ( {2'b00, etc_240_clk} ), // clock
  .ena ( 3'b001 ), // enable
  .loadconst ( 1'b0 ), // load constant port
  .negate ( 1'b0 ), // negate port
  .resulta ( dsp_output(USER_FILTERING_IN_FIR_FILTER_TAPS_NB / 2 - 1) ), // output port
  .sub ( 1'b0 ) // sub port
);

```

サンプル数でDSP Blockがアサインされる数が変わる。

図 5.43: ファームウェア中でのパラメータ定義の例。上が τ の選択条件ファームウェアで、下が Optimal filter のソースコード。 τ の選択条件がイネーブルされているコードとバイパスされるコードが記述されている。パラメータの値に応じて、どちらのファームウェアがコンパイルされるか決定できる。Optimal filter においても、パラメータに応じた DSP Block の接続数が決定される。

表 5.8: シミュレーションのマイルストーン

パラメータ	M1	M2	M3	M4	M5
User Code 内に入るインプットの stream 数	1	1	10	62	62
FIR Filter 内の filter のサンプル数	4	4	4	4	6 ~ 14
Saturation Detection 内の filter のサンプル数	4	4	4	4	6 ~ 14
DSP Block から E_T を抜き出す際の LSB	3	3	3	3	3 ~
DSP Block から $E_T \cdot \tau$ を抜き出す際の LSB	3	3	3	3	3 ~
τ の選択条件	false	true	true	true	true
BCAV 再同定ロジック	false	false	true	true	true
DSP Block の出力値のデバッグ	true	true	true	true	true

M1~M5 の全ての設定に対して、図 5.44 のシミュレーション結果が得られた。インプットジェネレータからチェッカーを通して checker flag を出力するまでに、レイテンシがかかる。そのため、リセット直後、checker flag の値は 0 である。レイテンシ後の checker flag は 1 へ変わり、以降はずっと 1 のままである。図 5.44 中の error counter は主に長時間の試験時に使用され、試験時に何個のエラーが起こったかを計測する。RAM に格納されている全てのデータに対して、正しく処理できていることが確認された。

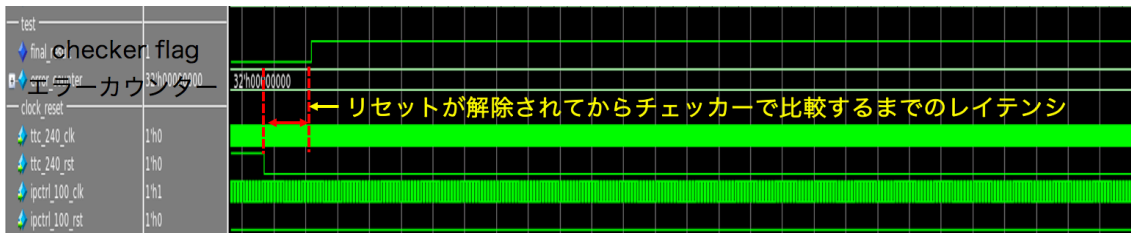


図 5.44: シミュレーション結果。エラーカウンターと checker flag から、すべての信号に対してエラーがないことを示している。

5.4.3 ボードによる試験

シミュレーションレベルで User Code モジュールは正常に動作しているので、ボードでも同様の振る舞いするかを確認する。ボードは、図 5.1 と ALTERA が提供している評価ボード (図 5.45) を用いる。LATOME ボードと評価ボードに搭載されている FPGA は同じ Arria10 シリーズであるが、トランシーバ用のピン数に違いがあるため全く同じ FPGA ではない。

試験は、LATOME と Arria10 評価ボードの両方で行っている。

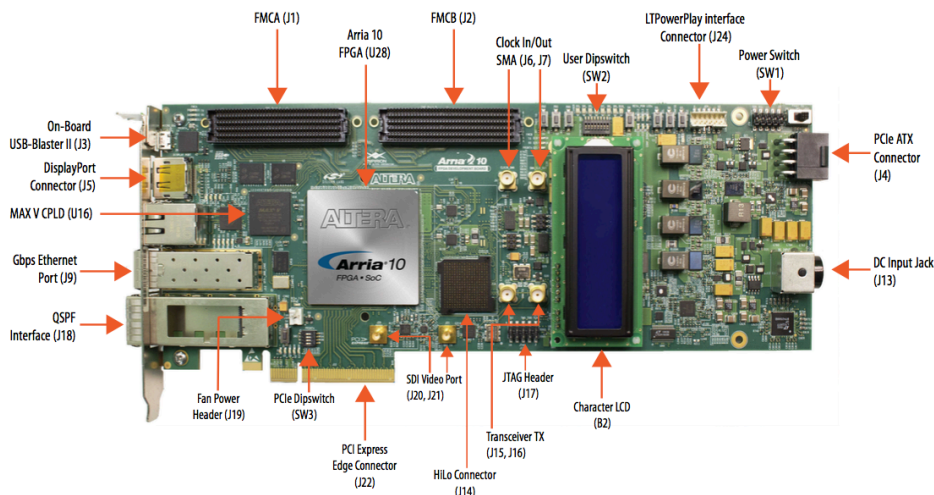


図 5.45: Arria10 評価ボード [11]。厳密には、LATOME と FPGA と異なる。しかし、同じ Arria10 シリーズであるため、User Code ファームウェアの試験はできる。

それぞれの FPGA に実装するファームウェアのリソースは、表 5.9 である。この結果は 4 点の Optimal filter を実装した場合であり、サンプル数によって、DSP Block の使用率は変動する (表 5.2)。サーキュラーバッファの数も同時に変動するが、全体のリソースと比較すると MLAB の使用率はあまり変わらない。

240 MHz クロックに対する最大動作周波数は、表 5.10 となった。どのサンプル数でも正スラックが得られ、240 MHz 以上の動作周波数を示している。これにより、タイミングの観点でも User Code ファームウェアの実装に問題がない。

リソース、最大動作周波数共に実装可能であることが示せたので、LATOME ボードと Arria10 評価ボードに実装する。動作検証方法はボードに JTAG を接続させ、それを介し

表 5.9: チェッカー付き User Code ファームウェアのリソース

リソース	使用率
ALM	11 %
DSP Block	16 %
メモリ	29 %
MLAB	1.4 %

表 5.10: 240 MHz クロックの最大動作周波数

サンプル数	4 点	6 点	8 点	10 点	12 点
最大動作周波数	257.6 MHz	257.8 MHz	259.29 MHz	252.61 MHz	253.44 MHz

ていくつかの信号のデバッグを行う (図 5.46)。JTAG は、FPGA ファームウェアの書き込みや基本的な信号をデバッグできる規格である。まず、乱数で生成された入力データを用いて、動作検証を行う。図 5.47 が、実際にボードから取得できたロジックアナライザの出力であり、シミュレーションと同様に checker flag=1 が得られた。約 2 時間の動作後の error counter が 0 であり、長時間の正常動作も確認した。また、システムが正しく動作しているかを検証するために、ある BC の 1 つのデータに人為的に間違った答えを入れた。これは、チェッカーが正しくエラーを検出できるかの試験である。その結果が図 5.47 下であり、checker flag = 0 と error counter が増加することを確認した。この試験は、表 5.7 で定義されたマイルストーン全てに対して行い、同様の結果が得られた。

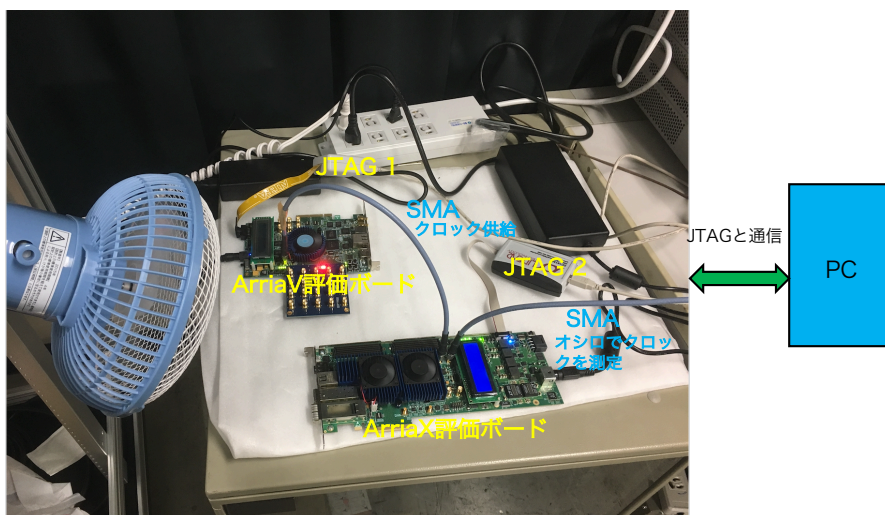
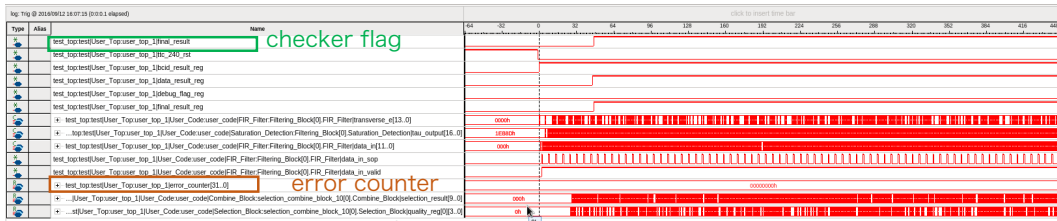


図 5.46: 実験のセットアップ。ALTERA の ArriaV 評価ボードから 160 MHz クロックが、Arria10 評価ボードに伝送される。受け取ったクロックは Arria10 ファームウェア内で 240 MHz へ分周され、User Code ファームウェアへ渡される。ファームウェアを実装するために、ALTERA 提供の JTAG とソフトウェアを利用した。JTAG を介して特定の信号のデバッグができるため、checker flag と error counter の振る舞いが PC 上で確認できる。



- ・ checker flag = 1
- ・ 約2時間のrunで、error counter = 0

あるBCの1つだけエラーを入れる：RAMに間違った答えを入れる。

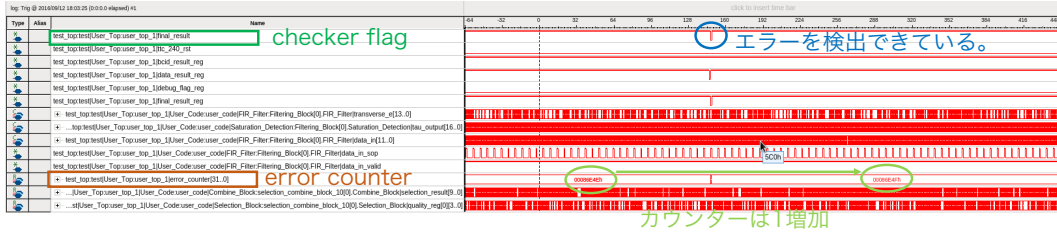


図 5.47: 取得できた信号。全ての信号が予期した振る舞いを示していることが得られ、長時間での安定した動作も確認できた。また、checker flag 等が正しくエラーを検出できるかの試験も行った。エラーを混ぜた場所で、checker flag が 0 となり、error counter の値が 1 増える。

次に、シミュレーションで LAr カロリメータの波形と Optimal filter の係数を作成し、LATOME ファームウェアに実装する。図 5.48 左はあるチャンネルのインプットデータである。インプットデータに対して、User Code から得られた出力値は図 5.48 右である。この結果は、乱数の時と同様に checker flag と error counter から正常動作を確認している。また、係数を実装する際に、ビット幅の制限から係数やエネルギーは丸められる。これによる精度の悪化をみるために、丸め前のエネルギーとファームウェアのエネルギーの相対誤差を調べ、1% 以内の精度であることを確認した (図 5.49)。

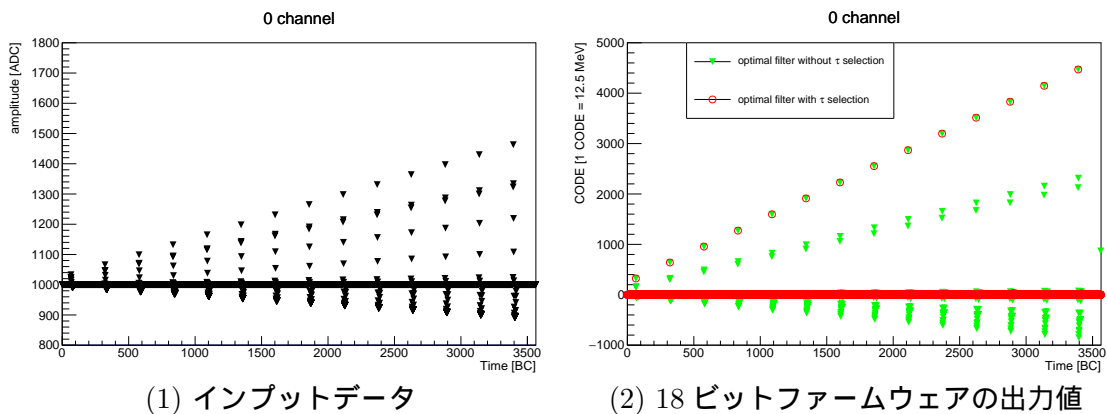


図 5.48: LAr カロリメータ波形を入れた試験。(1)は、ペダスタルを 1000 で固定した 14 種類のカロリメータ波形を入れた。(2)は、そのインプットに対してのファームウェアの出力結果である。全ての波形に対して、ピークを捉えることができています。緑点は、 τ の選択条件を false にすることで得られた。

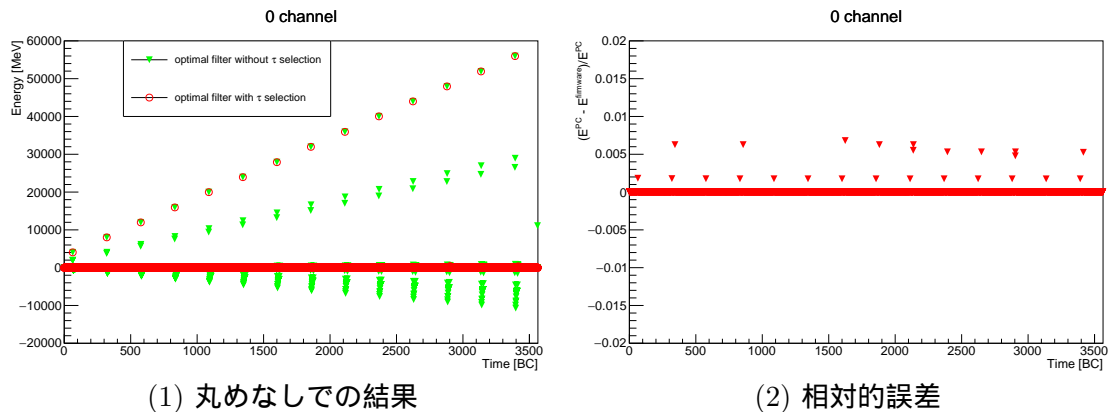


図 5.49: ファームウェア出力値の精度比較。(1) は、丸めなしを行った場合のエネルギー出力値を示している。(2) は、ファームウェアの結果との相対的誤差を示している。ファームウェアで得られたエネルギーは、1%以下の精度で丸める前と一致している。ファームウェアは、十分な精度で計算できている。

5.5 統合試験

図 5.4 からレイテンシの要請に寄与するモジュールは、LLI, Input stage, Configurable remapping, User Code, Output summing モジュールである。そのため、統合試験の最初のステップとしてこれら 4 モジュールが統合され、レイテンシやリソース等の測定を行った (表 5.11)。このファームウェアは、M1 (マイルストーン 1) ファームウェアと呼ばれる。この表から、要求値以内 (表 5.1) のレイテンシ (≤ 15 BC) と実装可能なリソース量であることがわかる。

表 5.11: 測定した各モジュールのレイテンシ (左) と統合ファームウェアのリソース使用率 (右)。L1 トリガーからの要請である 15 BC 以内のファームウェアが作成できた。また、リソースも十分 FPGA に実装可能である。

モジュール名	レイテンシ [BC]	リソース	使用率
LLI	4.94	ALM	13 %
Input stage	2.12	Total pins	39 %
Configurable remapping	1.58	Memory	7 %
User Code	4.5	DSP Block	16 %
Output summing	1.13	トランシーバ	73 %
Total	14.27	PLL	39 %

また、M1 ファームウェアにチェッカーモジュールを組み込み、ボードによる試験を行った。今回の試験では 1 本の光ファイバーを用いており、Output summing モジュールの jFEX と gFEX 用の信号にチェッカーは組み込んではいない (図 5.50)。どの程度エラーが起きるかを 24 時間かけて試験を行い、どのモジュールもエラーが起きずに正しく動作して

いる結果が得られた。今後は jFEX と gFEX のチェッカーを作成と、複数の光ファイバーでの実験を行う予定である。

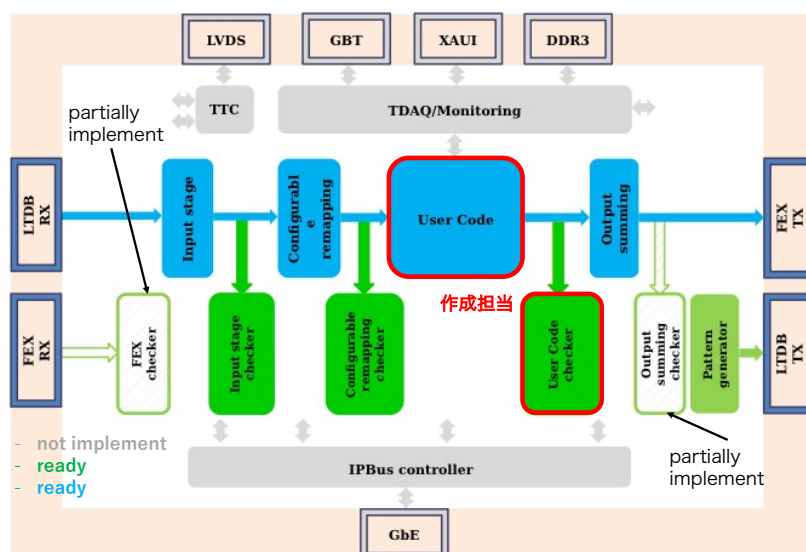


図 5.50: チェッカー付き LATOME ファームウェア。青枠と緑枠のモジュールはすでに統合が完了している。gFEX のエンコーディングはまだ確定していないため、Output summing モジュールの修正を行う可能性がある。jFEX, gFEX 以外の試験はすでに行われており、長時間での正常動作が確認された。

LATOME ファームウェアは現在、IPBus Controller と TTC モジュールの統合を行っている。2017 年 2 月初旬までに全モジュールの統合が終わる予定である (図 5.51)。その後、フロントエンドと接続し、全体のシステムテストが行われる。

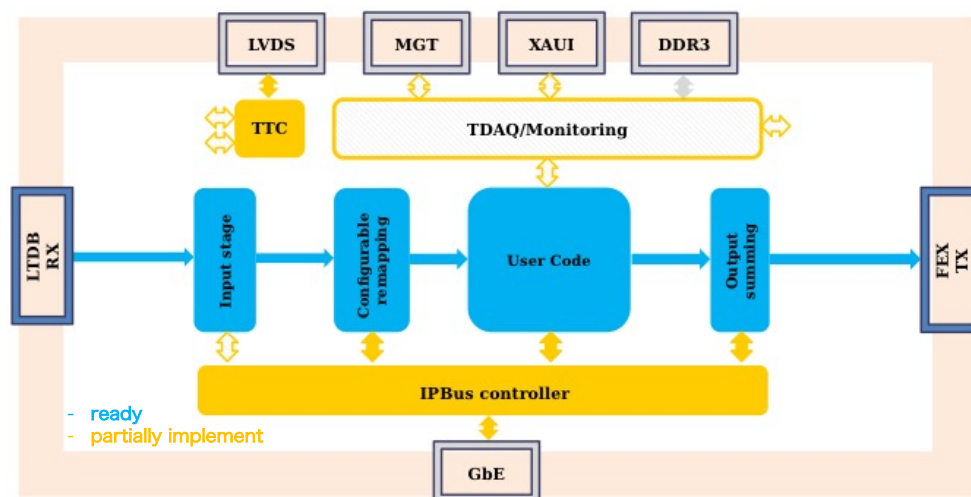


図 5.51: 2016 年 12 月時点でのファームウェア統合状況。メイン経路のファームウェアは実装は完了している。現在は、IP Controller や TTC モジュールとの統合を行っている。

第6章 結論と今後に向けて

6.1 結論

Run3以降のLHC-ATLAS実験の高輝度化により、背景事象によるトリガー発行頻度の上昇が予想される。そのため、トリガー能力の向上のため、LArカロリメータトリガー読み出しのアップグレード計画に参加している。

ATLAS日本グループはバックエンドエレクトロニクスの研究に参加し、そこで必要なFPGAファームウェアの開発を行なっている。このFPGAの主な役割は、従来よりも10倍細分化させた読み出し領域(Super Cell)のエネルギーと時間を算出することである。そのため、2つのプロジェクトに参加した。

デモンストレータプロジェクト Run2実験でSuper Cell信号を収集するバックエンドデモンストレータの運転に携わった。本研究は、膨大な数の入力・出力をもつブロックの詳細なテストベンチの作成に成功し、データ収集時に起きていた問題の原因を特定し、修正を行った。これによって改善されたファームウェアの動作は安定しており、効率的に多くの物理データを取得した。また、キャリブレーションデータも取得でき、LATOMEプロジェクトのファームウェア開発に大きく役立った。

LATOMEプロジェクト FPGAの特徴を最大限に生かし、全Super Cellのエネルギーの算出とサチュレーションした波形に対する処理が行われるファームウェアの設計および実装を一から全て行った。このファームウェアのレイテンシは112.5 nsであり、L1トリガーからの要求値を満たしている。また、リソースや最大動作周波数も調べ、LATOMEで使用されるArria10 FPGAに実装可能であることを示した。シミュレーションとボードによる様々な試験も行われ、正常動作を確認した。

本研究によって、Super Cell読み出しに対応するファームウェアの基本的な枠組みが完成した。

6.2 今後に向けて

シミュレーションによる評価の不十分な所が数多く残っている。特に、サチュレーションの同定方法はより詳細なシミュレーションを行う。また、ベースラインコレクションの設計が完成していないため、LATOMEに実装できるファームウェア開発が必要である。ハードウェアの観点では、新しく導入されるLTDBとLDPBを取り付けたシステムテストを現在行っており、2017年2月頃からLATOMEファームウェアもこのシステムテスト

に導入される。そのために、LATOME ファームウェア自体の全て統合させ、大量生産前のシステムテストを 2017 年前半に完了させる。その後は、ATLAS への実装に向けた準備に移行する予定である。

謝辞

本研究を進めるにあたり、指導教員である田中純一准教授には毎週の発表において多くの助言を賜りました。特に、学会や CERN での発表スライドに関しても何度も添削していただき、心から感謝申し上げます。江成祐二助教には、本研究の具体的な進め方に加えて、ファームウェアの実装に関して多くのご指導を賜りました。これをしると強制するでなく、自分で考えた設計の修正点などを丁寧に教えていただきました。とても有意義でやりがいのある研究テーマを与えていただき大変感謝しています。短い期間ではありましたが、山本真平特任助教にはエレクトロニクスやファームウェアについて多くの助言を頂きました。

本研究で携わった2つのプロジェクトでは、多くのメンバーにご指導賜りました。デモンストラプロジェクトのリーダーである Steffen Starz には、ABBA ファームウェアについて本当に多くの助言を頂きました。拙い英語でも理解しようとしてくれることに関しては、感謝してもしきれません。Alessandra Camplani には、TTC ファームウェアに関して多くの助言を頂き、ABBA ファームウェア試験の手助けをしてもらいました。先輩の楊易霖氏には、ABBA データの解析で大変お世話になりました。他のメンバーにも、meeting で大変お世話になりました。LATOME プロジェクトのリーダーである Nicolas.Chevillot と George Aad には、多くの助言を頂きました。特に、Nicolas.Chevillot には、ファームウェアの設計について親身に議論していただきました。Nicolas Dumont Dayot には、LATOME ボードについて説明、実機での試験の手助けをしていただきました。Bernard Dinkespiler をはじめとする CPPM のメンバーには、Arria10 評価ボードでの試験の手助けをしていただきました。その他のメンバーにも、meeting 等で大変お世話になりました。

ICEPP の方にも大変お世話になりました。難波俊雄助教には、検出器の基礎知識について多くのご指導賜りました。秘書の塩田雅子様、竹本葉子様、山浦華世様、手塚淑恵様、加瀬由美様、原田靖子様には出張の手続きをはじめとする多くの事務手続きを行っていただきました。先輩である嶺岸優司氏には、多くのことについて非常にお世話になりました。ファームウェアについての基礎の丁寧な説明だけでなく、CERN での生活のサポートまでしてもらいました。研究室の同期である井口君とは多くの苦楽を共にしました、また頑張りましょう。同じく同期である周君と清野君はよくご飯にいきました、また行きましょう。樊君、岩井君、藤倉君、中尾君、内田君、中西君や先輩、後輩達のおかげで充実した研究生活が送れました。

最後に、家族をはじめとした周りの方のおかげで、修士生活を全うできました、ありがとうございます。

参考文献

- [1] ATLAS Collaboration, ATLAS LAr Calorimeter Phase-I upgrade TDR, CERN-LHCC-2013-017
- [2] ATLAS Japan group, <http://atlas.kek.jp/sub/photos/>
- [3] ATLAS Photo LAr Calorimeter, <http://www.atlasexperiment.org/photos/calorimeters-lar-barrel.html>
- [4] ATLAS Photo FCalo, http://www.project-atlas-lucid.web.cern.ch/project-atlas-lucid/taskforce/main_10.html
- [5] ATLAS Photo MDT, http://www.atlas.mpp.mpg.de/atlas_mdt/subpages/mdt_chamber.html
- [6] Atlas Collaboration, ATLAS TDAQ System Phase-I upgrade TDR, CERN-LHCC-2013-018
- [7] ATLAS Collaboration, HL-Plan, <http://www.hilumilhc.web.cern.ch/about/hl-lhc-project>
- [8] ATLAS Collaboration, Performance of the ATLAS Liquid Argon Calorimeter after three years of LHC operation and plans for a future update, <https://inspirehep.net/record/1240499/plots>
- [9] Wikipedia, ATCA create, https://en.wikipedia.org/wiki/Advanced_Telecommunications_Computing_Architecture
- [10] R.Schwienhorst on behalf of the ATLAS collaboration, The Phase-1 upgrade of the ATLAS first level calorimeter trigger, 2016 JINST 11 C01018
- [11] ALTERA company homepage, <https://www.altera.co.jp>
- [12] CERN Twiki, Proposal format of data link from LTDB to LDPB, https://twiki.cern.ch/twiki/pub/LAr/LArDemonstrator/format_of_data_link.pdf
- [13] Nicolas.Dumont.Dayot, Preliminary Design Review of LATOME AMC A10 Carrier Board I/O Interfaces: <https://indico.cern.ch/event/376479/>
- [14] CERN Twiki, ATLAS LAr Calorimeter trigger electronics phase I upgrade: LATOME Firmware Specifications, <https://twiki.cern.ch/twiki/bin/view/LAr/LATOME>

- [15] B.Deng, M.He, J.Chen, D.Guo, S.Hou, X.Li, C.Liu, P.-K.Teng, A.C. Xiang, Y.You, J.Ye, D.Gong and T.Liu, A line code with quick-resynchronization capability and low latency for the optical data links of LHC experiments, 2014 JINST 9 P07020
- [16] CERN Twiki, Public result, <https://twiki.cern.ch/twiki/bin/view/AtlasPublic/L1CaloTriggerPublicResults>
- [17] 久島慎吾、ATLAS 実験液体アルゴンカロリメータのデジタル信号処理におけるエネルギー再構成アルゴリズムの研究開発、<https://www.icepp.s.u-tokyo.ac.jp/papers/thesis.html>
- [18] Bill Cleland, Study of Optimal Filtering Options for Treatment of Super Cell Data in the Phase-I upgrade <https://cds.cern.ch/record/2064710/files/ATL-COM-LARG-2015-032.pdf>
- [19] Philippe Schwemling, Neighbouring cell use for BCID, https://indico.cern.ch/event/464574/contributions/1139505/attachments/1196706/1739404/trigger_saturation301115.pdf
- [20] Brian Petersen, BCID Dependent Pedestal Subtractions, <https://indico.cern.ch/event/249832/contributions/1573956/attachments/432170/599744/Pedestals.pdf>