修士学位論文
LHC-ATLAS RUN3 実験に向けた 液体アルゴンカロリメータートリガー読み出し ファームウェアの開発と検証
Firmware development and validation for the liquid argon calorimeter trigger readout system at the LHC-ATLAS RUN3 experiment

東京大学理学系研究科物理学専攻
田中研究室
大石玲誉

2020 年 1 月

目次

図目次

表目次

第 1 章

序論

標準理論は 17 個の素粒子とそれらの相互作用を記述する 3 つの力「電磁相互作用」「弱い相互作用」「強い相互作用」からなる理論体系である (図 1.1)。もう一つ粒子間の相互作用である「重力相互作用」は、一般相対性理論により記述されている。標準理論は現在最も成功した理論体系である。しかしながら、ニュートリノの質量問題、暗黒物質、階層性問題など様々な未解決問題がある。LHC-ATLAS 実験では 25 ns の頻度で高エネルギーの陽子同士を衝突させ、標準理論では説明できない現象の解明、標準理論を超えた物理の探索が盛んに行われている。1 回の陽子同士の衝突あたりのデータ量は 1.6 Mbytes 程度であり、40 MHz の頻度で衝突が起こるので、それら全てを保存するには莫大なリソースが必要になり現実的ではない。しかし衝突のほとんどが陽子同士の非弾性散乱であり、興味の無い事象がほとんどを占める。そのため全てのデータに対しそのデータが有用かどうかをごく短い時間で判断、取捨選択をし、興味

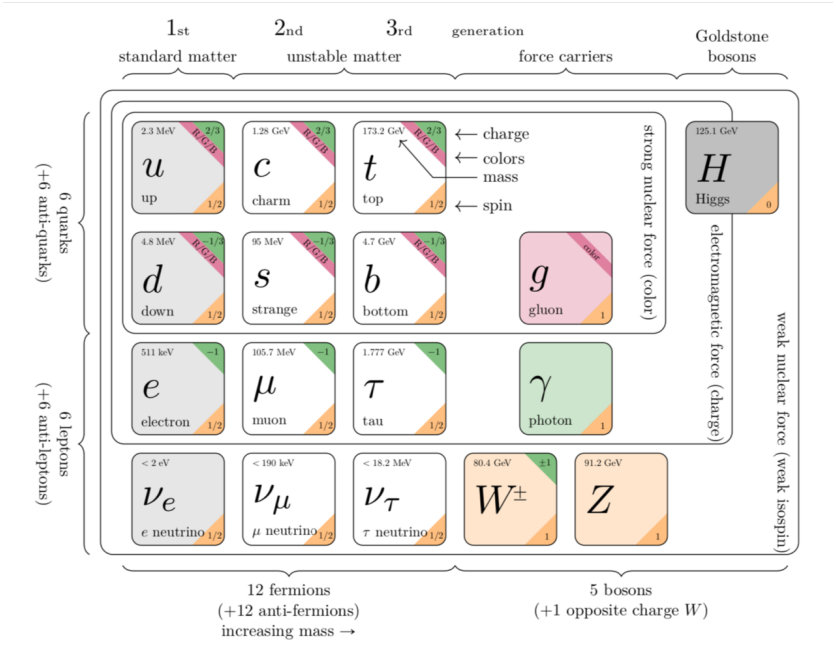


図 1.1: 標準理論で扱われる素粒子。 ヒッグス粒子が 2012 年に ATLAS, CMS 実験により発見され標準理論で予想された粒子が全て実験的に発見された [1]。

ある事象のみを収集する手段 (トリガー) が取られる。本研究では LHC-ATLAS 液体アルゴンカロリメーターのトリガーシステムの開発に関するものである。この章では本研究の背景と目的について述べる。

1.1 本研究の背景

2015 年から 2018 年まで行われた LHC-ATLAS RUN2 実験は、重心系エネルギー 13 TeV での陽子陽子衝突が 40 MHz の頻度で行われ、積分ルミノシティー 147 fb^{-1} に相当するデータ量を取得した。ATLAS 実験では効率よく興味ある事象を残しその他を破棄するために、2 段階のトリガーシステムを導入して収集頻度を大幅に削減している。1 段目のトリガー (Level 1 trigger) では主にハードウェアで高速に荒く処理し 40 MHz のものを 100 kHz まで落とす。後段のトリガー (High level trigger) では PC ファームでより詳細に選別し、最終的に 1 kHz まで収集頻度を落とす。本研究は Level 1 trigger に関する内容である。

液体アルゴン検出器の信号読み出し単位 (Elementary Cell) は非常に細かく分割されているが、それらを足し合わせた信号を用いて Level 1 trigger を行う。RUN2 では $\Delta\phi \times \Delta\eta = 0.1 \times 0.1$ をトリガー読み出しの最小単位 (Trigger Tower) として扱い、Trigger Tower 内に落とすエネルギーにより閾値をかけていた (24 GeV)。液体アルゴン検出器は電子、光子のエネルギーと位置を測定する目的で導入されており、ハドロンとの識別が重要である。RUN3 (2021∼2023 年) からはパイルアップが増加し、同じ信号背景識別能力ではトリガーレートが上がる。トリガーレートの要求値を満たしかつ同じ Trigger Tower 内に落とすエネルギーでトリガーをかけた場合は、閾値を 34 GeV にする必要があり興味ある事象を落としてしまう可能性が高くなる。そのためトリガーに用いる検出器からの読出しを細かくすることで電子、光子とハドロンの識別能力を上げる。このより細かい読み出し単位は Supercell と呼ばれ図 1.2 に示すように動径方向に 4 層構造 (presampler, front layer, middle layer, back layer) をしており、Trigger Tower は 10 個の Supercell に分けられる。4 層のうち 2 層 (front layer, middle layer) は η 方向にさらに細かく $\Delta\phi \times \Delta\eta = 0.1 \times 0.025$ として読み出す。

電子, 光子のカロリメーター内での広がりは、モリエール半径[*1]で表すことができそれはハドロンに比べ 10 分の 1 程度である。ハドロンはグルーオンとの強い相互作用により半径の大きなジェットとして観測される。そのためジェットとの識別効率を上げるために Supercell での読み出しで可能な 3 つの新たな変数 $R_\eta, \omega_{\eta,2}, f_3$ を導入する。これらの変数を用いることでシャワー形状をトリガーに用いる頃ができるため、より詳細な事象選別をトリガーで行うことができる。

R_η は式 1.1 で表される。

$$R_\eta = \frac{E^{(2)}_{T,\Delta\eta\times\Delta\phi=0.075\times0.2}}{E^{(2)}_{T,\Delta\eta\times\Delta\phi=0.175\times0.2}} \tag{1.1}$$

middle layer で最も大きなエネルギーを落とした Supercell を中心にして $\Delta\eta \times \Delta\phi = 0.075 \times 0.2$ をひとまとまりにしたエネルギーと $\Delta\eta \times \Delta\phi = 0.175 \times 0.2$ をひとまとまりにした時のエネルギーの比に対応する。電子、光子は検出器内部で $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2} \simeq 0.08$ 程度の領域にエネルギーのほとんどを

[*1] モリエール半径はその中に 90% のエネルギーが含まれている範囲と定義されており原子番号, 原子量と密度を用いて $R_m \simeq \frac{7A}{Zg}(\mathrm{g/cm^{-2}})$ と表すことができる。LHC-ATLAS 実験で用いられる吸収層は鉛 (A=207, Z=82, g=11.34g/cm^{-3}) であるので $R_m \simeq 1.55$cm である。

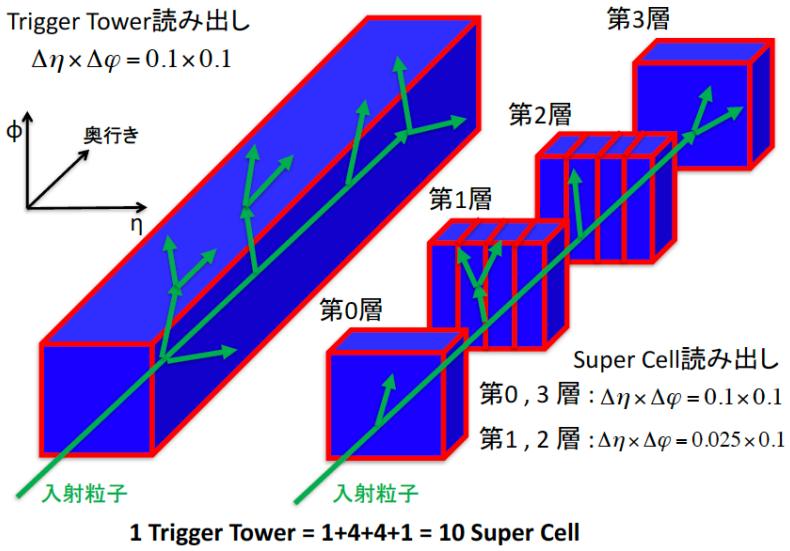


図 1.2: 右が RUN2 までのトリガー読み出し最小単位 Trigger Tower、左が RUN3 からのトリガー読み出し最小単位 Supercell。Trigger Tower は検出器の位置に依存するが概ね 10 個の Supercell で構成される。

落とすので 1 に近い値になる。一方ジェットなどは $\Delta R \simeq 0.8$ 程度とおおきな範囲にエネルギーを落とすので図 1.3 (1) に示すようにそれぞれの分布に違いが見れる。

$\omega_{\eta,2}$ は式 1.2 で表される。

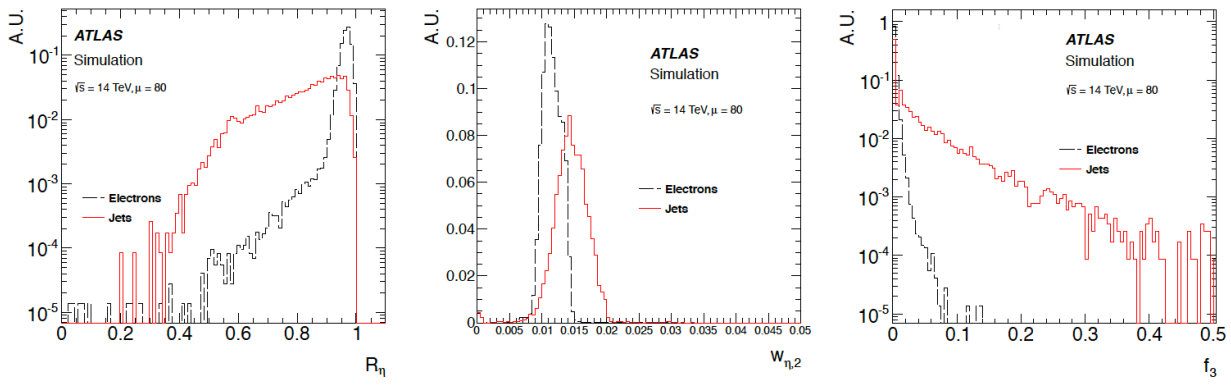
$$\omega_{\eta,2} = \sqrt{\frac{\sum \left(E_T^{(2)} \times \eta^2\right)_{\Delta\eta\times\Delta\phi=0.075\times0.2}}{E^{(2)}_{T,\Delta\eta\times\Delta\phi=0.075\times0.2}} - \left(\frac{\sum \left(E_T^{(2)} \times \eta^2\right)_{\Delta\eta\times\Delta\phi=0.075\times0.2}}{E^{(2)}_{T,\Delta\eta\times\Delta\phi=0.075\times0.2}}\right)^2} \tag{1.2}$$

$\omega_{\eta,2}$ は R_η 同様に middle layer に落としたエネルギーの広がりを示した変数であり、図 1.3 (2) に示すように分布に違いが見れる。

f_3 は式 1.3 で表される。

$$f_3 = \frac{E^{(3)}_{T,\Delta\eta\times\Delta\phi=0.2\times0.2}}{E^{(1)}_{T,\Delta\eta\times\Delta\phi=0.075\times0.2} + E^{(2)}_{T,\Delta\eta\times\Delta\phi=0.075\times0.2} + E^{(3)}_{T,\Delta\eta\times\Delta\phi=0.2\times0.2}} \tag{1.3}$$

f_3 は $\Delta\eta \times \Delta\phi = 0.2 \times 0.2$ の back layer に落としたエネルギーと、それに $\Delta\eta \times \Delta\phi = 0.075 \times 0.2$ の front layer, middle layer のエネルギーの和の比を取ったもの。電子, 光子は middle layer までででほとんどのエネルギーを落とすがジェットは電磁カロリメーターの外側にあるハドロンカロリメーターまで通過するので、図 1.3 (3) で示すように電子、光子は 0 付近にピークを持つがジェットなどのハドロンはそれより大きい値を持ちやすい。



(1) R_η (2) $w_{\eta,2}$ (3) f_3

図 1.3: Supercell 読み出しにより新しく導入できる 3 変数。黒が電子、赤がジェットの分布。全てのグラフは規格化された結果を表示している。重心系エネルギー $\sqrt{14}$ TeV、平均相互作用数 80 を仮定している [9]。

これらの変数を用いることで、電子、光子とハドロンジェットとの識別能力が向上し、トリガーレートを効率的に落とすことが可能になる。エネルギー閾値を 21.5 (20.5) GeV とし、$R_\eta \geq 0.93$ (0.94)、$\omega_{\eta,2} < 0.0146$ (0.014)、$f_3 \leq 0.02$ (0.02) とすることにより図 1.4 に示すようにトリガーレートを 20 kHz に保ったまま $Z \to e^+e^-$ 過程のトリガー効率を 90 (95)% に保つことができる。そのため Supercell を RUN3 からのトリガー用の読み出しとすることで、トリガーレート要求値を保ったままエネルギー閾値を上げることなく効率よく事象選別を行うことが可能になる。

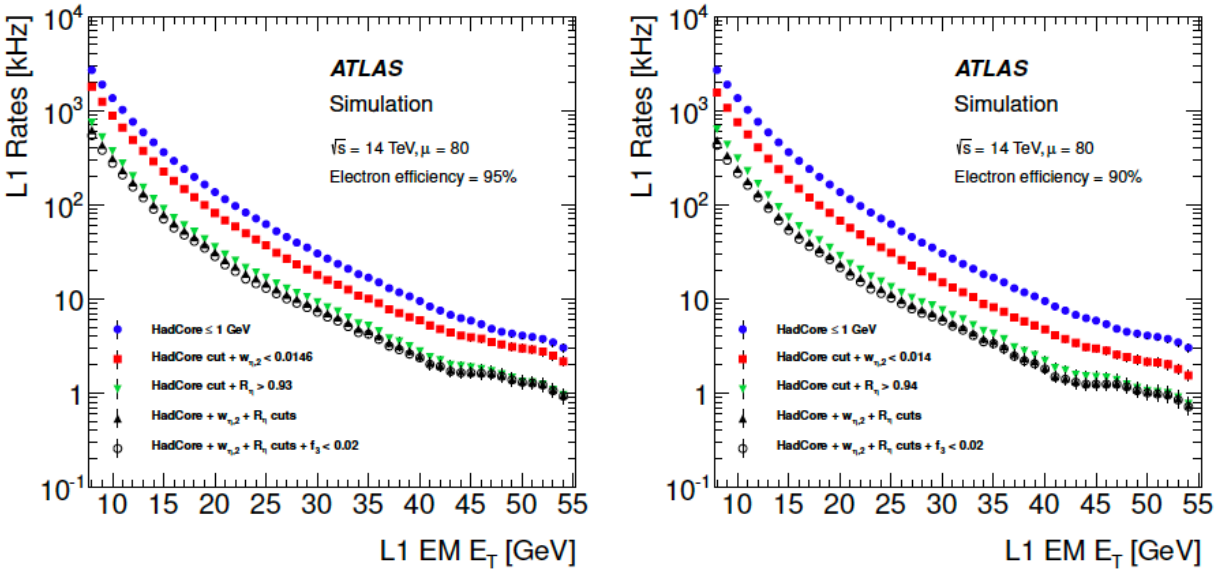
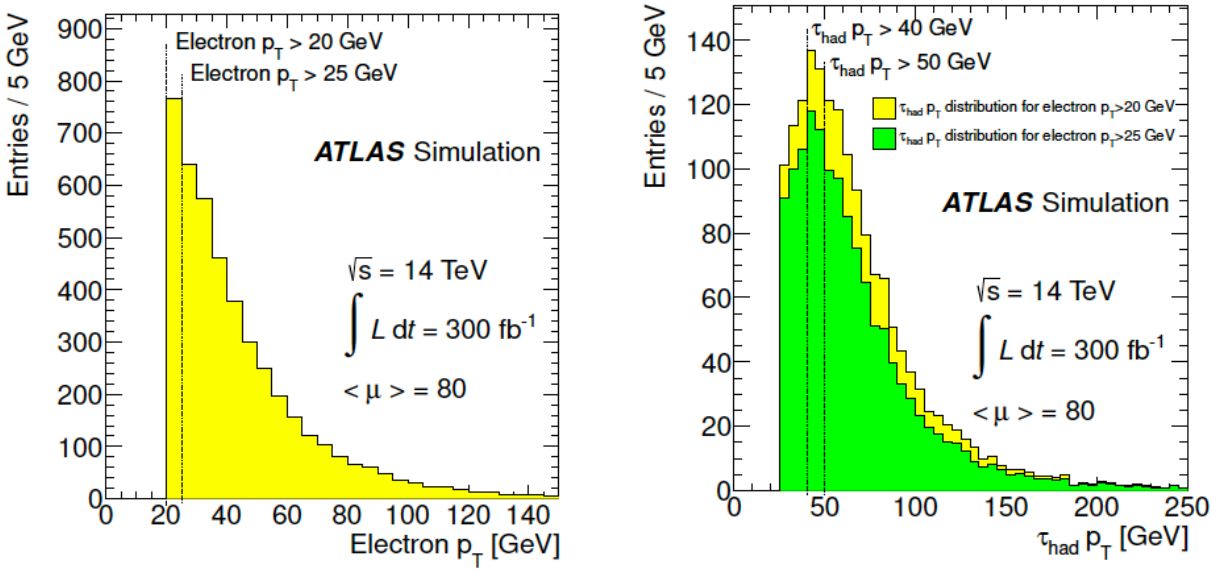


図 1.4: ルミノシティーを $L = 3 \times 10^{34}\mathrm{cm}^{-2}\mathrm{s}^{-1}$ とし、$Z \to e^+e^-$ のシミレーションで得られた電子に対する結果。左がトリガー効率 95% 右が 90% を示した図。白抜き丸が新たに定義された 3 変数を現行のシステムに加えた際のエネルギー閾値に対応する [9]。

エネルギー閾値を低いままトリガーを行うことにより低いエネルギーの電子、光子の収集を可能にする。RUN3 実験での重要なテーマとして質量 125 GeV のヒッグス粒子に関する精密測定がある。その一例として $H \to \tau\tau$ 過程で、片方の τ がレプトンに崩壊、もう片方がハドロンに崩壊する場合について述べる。レプトン崩壊する τ が電子に崩壊する場合、その電子の運動量は図 1.5a のような分布を持つ。電子への崩壊に伴い電子ニュートリノを放出するため電子が持つ運動量が小さい場合が多い。またハドロンに崩壊する τ の運動量は図 1.5b のような分布になる。



(a) 電子の運動量分布

(b) τ の運動量分布

図 1.5: $H \to \tau\tau \to e\bar{\nu}_e\nu_\tau\tau_{had}\nu_\tau$ 過程における電子と τ_{had} の運動量に対する分布。H は 125 GeV であり、それの崩壊先である τ は約 35% がレプトンに崩壊する。重心系エネルギー 14 TeV, 平均相互作用数 80 を仮定し、積分ルミノシティーを 300 fb^{-1} で規格化している。(b) はそれぞれ電子の運動量として 20 GeV 以上 (黄色)、25 GeV 以上 (緑) を要求した場合のハドロン崩壊する τ の運動量分布を表示している [9]。

この Simulation は RUN3 実験を想定して行われており、重心系エネルギー 14 TeV、平均相互作用数 80 の場合であり積分ルミノシティー 300 fb^{-1} に対応する結果である。RUN2 実験で用いられたオフラインでの条件 (電子の運動量が 25 GeV 以上、ハドロンに崩壊する τ の運動量を 50 GeV 以上) を課した場合と、トリガーのアップグレードにより実現可能な条件 (電子の運動量が 20 MeV 以上、ハドロン崩壊する τ の運動量を 40 GeV 以上) を図 1.5 内の点線で示した。この過程において電子と τ を要求した場合の取得率は表 1.1 に纏めた。ここから分かるように RUN2 で用いられた条件は RUN3 から実現可能な条件に比べ、電子、τ を要求した場合信号の取得率が 37% 減少する。この例のように低い運動量の電子、光子の取得が重要な解析において読み出し機構をアップグレードしてエネルギー閾値を下げることは物理的な観点から非常に重要である。

Trigger での要求	$P_T(e) > 25\ GeV, P_T(\tau) > 50\ GeV$	$P_T(e) > 20\ GeV, P_T(\tau) > 40\ GeV$
電子の選択効率	21.0±0.2%	25.4±0.2%
電子と τ の選択効率	4.7±0.1%	7.5±0.1%
相対度	0.631±0.015	1

表 1.1: $H \to \tau\tau$ 過程における電子の運動量、τ の運動量依存した信号の収集効率。電子の運動量の要求値を 20 GeV から 25 GeV、τ の運動量の要求値を 40 GeV から 50 GeV に変更した場合の信号取得率を示す [9]。

1.2 本研究の目的と本論文の構成

Supercell を導入し、トリガー読み出しを 10 倍細かくすることにより、10 倍の検出器からのデータを Level 1 trigger のレイテンシー要求時間 (2.5 μs) で処理しなければならない。そのため高密度、高速に処理できる新たなシステムが必要である。そのためトリガー読み出しシステムが刷新される。そのシステムの中核である LATOME Project が現在進行中である。トリガーに関する信号は検出器全体で 25 Tbps にもなり、LATOME Firmware で全ての Supercell に対する ADC データをエネルギーに変換する。エネルギーへの変換は LATOME Firmware 内の 1 つの Module である User Code を用いて行われ、私はこの Module の開発、検証を行った。

本論文は 2 章で LHC-ATLAS 実験について述べる。3 章では刷新されるトリガー読み出し機構について述べ、4 章と付録 A で LATOME Project についてまとる。5 章、6 章と付録 B ～ 付録 E では本研究の最大のテーマである User Code の開発、検証について述べる。7 章では RUN2 実データを用いた Baseline Shift の検証について述べ、8 章でまとめと今後について述べる。

第 2 章

LHC-ATLAS 実験

欧州原子核研究機構 (CERN) には図 2.1 に示す全長 27 km の陽子陽子衝突加速器 Large Hadron Collidor (LHC) がスイス、ジュネーブの地下約 100 m に設置されている。LHC には 4 つの衝突点が設置されており、A Toroidal Lhc ApparatuS (ATLAS), Compact Muon Solenoid (CMS), A Large Ion CollidEr (ALICE), Large Hadron Collider beauty (LHCb) 検出器が設置されている。ATLAS, CMS は広い領域の新物理の探索のため、ALICE は宇宙初期の強い力が支配的な物理現象の解明を目的とし特に quark-gluon plasma と呼ばれる現象の解明のため、LHCb は'bottom quark' に注目することで CP 対称性の破れに関する研究の目的のために設置されている。本研究は LHC-ATLAS 実験に関連した内容である。

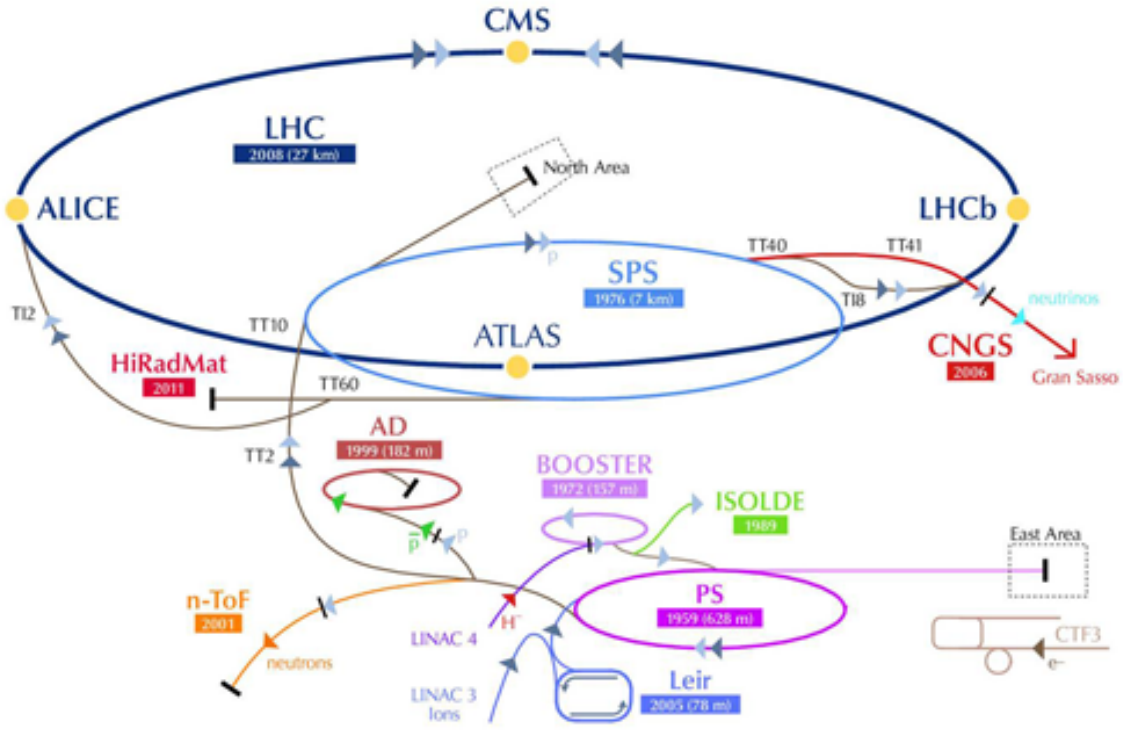


図 2.1: LHC 関連の加速器の全容 [3]。

2.1 ATLAS 検出器

ATLAS 検出器 (図 2.2) は全長 43 m 高さ 22 m の大型の検出器である。検出器からの全読み出しチャンネルはおおよそ 8800 万、ケーブル長はおおよそ 3000 km にもなる。大きく分け 3 つの検出器から構成され、さらに複雑なマグネットシステムを有する。

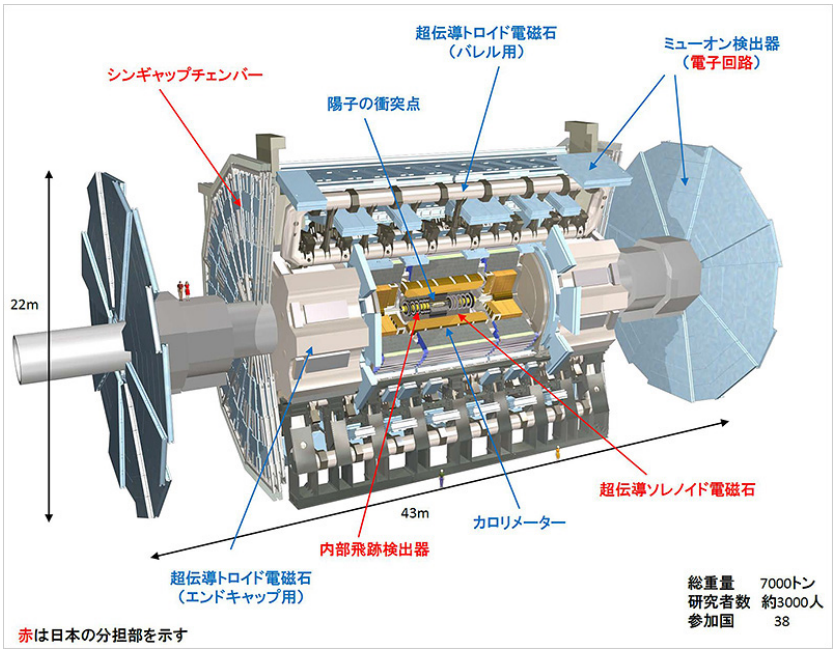


図 2.2: ATLAS 検出器 [4]。

ATLAS 実験では検出器の方向を示す変数として擬ラピディティーが用いられる。まずラピディティー y は粒子のビーム軸方向に対するローレンツ変換とその粒子の飛行方向に関係する変数であり式 2.1 のように定義される。

$$y = \frac{1}{2} ln \left(\frac{E + P_{||}}{E - P_{||}} \right) = ln \left(\frac{E + P_{||}}{\sqrt{P_T^2 + m^2}} \right) \tag{2.1}$$

ここで $E, P_{||}, P_T, m$ をそれぞれエネルギー、ビーム軸に沿った方向の運動量、ビーム軸に対し垂直方向の運動量、粒子の不変質量と定義した。次に粒子のビーム軸に対するローレンツ変換を考える。基準となる系と変換後の系の差に対応する変数 $\beta = v/c = tanh\Delta$ と γ を用いて式 2.1 を変換すると

$$\begin{aligned} y \to y' &= \frac{1}{2} ln \left(\frac{E' + P'_{||}}{E' - P'_{||}} \right) = \frac{1}{2} ln \left(\frac{(\gamma E + \beta\gamma P_{||}) + (\gamma P_{||} + \beta\gamma E)}{(\gamma E + \beta\gamma P_{||}) - (\gamma P_{||} + \beta\gamma E)} \right) \\ &= y + \frac{1}{2} ln \left(\frac{1+\beta}{1-\beta} \right) = y + \Delta \end{aligned} \tag{2.2}$$

このようにラピディティーは β 同様に系の変換に対する因子 Δ に帰する。粒子がとりうる最大のラピディティーはエネルギーに依存し、重心系では粒子のとりうる最大エネルギーは $E_{max} = \sqrt{s}/2$ である

ので

$$y_{max}^{CM} = \frac{1}{2}ln\left(\frac{\frac{\sqrt{s}}{2}+\sqrt{\frac{s^2}{4}-m^2}}{\frac{\sqrt{s}}{2}-\sqrt{\frac{s^2}{4}-m^2}}\right) = \frac{1}{2}ln\left(\frac{\left(\frac{\sqrt{s}}{2}+\sqrt{\frac{s^2}{4}-m^2}\right)}{m^2}\right) \simeq ln\frac{\sqrt{s}}{m} \tag{2.3}$$

となり粒子の質量に依存してラピディティーの最大値が変化する。そのため質量を 0 にした場合のラピディティーに対応する擬ラピディティーを用いる。擬ラピディティーは

$$\eta = \frac{1}{2}ln\left(\frac{E+Ecos\theta}{E-Ecos\theta}\right) = -ln\left(tan\left(\frac{\theta}{2}\right)\right) \tag{2.4}$$

と表され検出器内の角度方向を表すために用いられる。θ は検出器の中心点からビーム軸に対し垂直方向を 90 とし、0 度から 180 度までをとる変数である、そのため η は $\pm\infty$ をとりうる。またビーム軸を回転軸として 0 度から 360 度までをとる変数 ϕ を用いて検出器の位置を表す。

2.1.1　マグネットシステム

LHC-ATLAS 実験で用いられるマグネットは、図 2.3 に示す Central Solenoid, Barrel Toroid, Endcap Toroid の 3 つに分けられ、それらは全て超電導マグネットである。これらのマグネットは荷電粒子を曲げることによりその曲率から電荷, 運動量を測定する目的で用いられる。Central Solenoid の磁場は 2T で内部検出器で飛程を再構成する目的で導入され、長さ 5.3 m、直径は 2.4 m の単層コイルで構成される。カロリメーターの内側に導入さているため低物質量のマグネットである。Barrel Toroid と Endcap Toroid はそれぞれ 8 つに分けられビーム軸に沿って設置されている。Barrel Toroid と Endcap Toroid の各々のソレノイドは 22.5 度の差がある状態で設置されている。これらはミューオンの軌道を曲げる目的で用いられ、検出器の領域に依存して磁場の強さを変化させている。

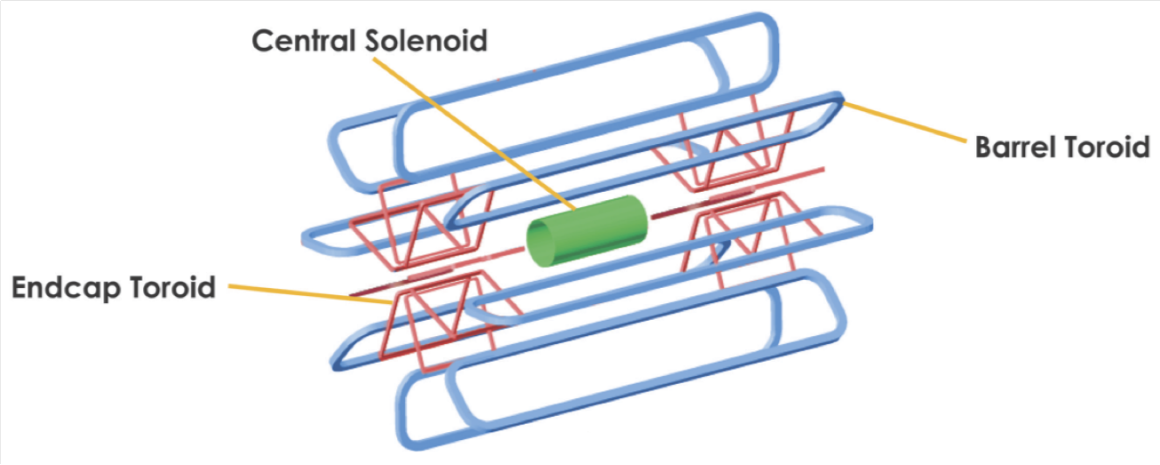


図 2.3: ATLAS 検出器で用いられるマグネットシステム [5]。

2.1.2 内部検出器

内部検出器 (図 2.4) は衝突点から最も近い位置にある検出器で、Central Solenoid からの磁場を受ける。ビーム軸に対し長さ 6.2 m、直径 2.1 m であり Barrel 領域, End cap 領域に分けられ $|\eta| \leq 2.5$ までカバーしている。Barrel 領域はシリンダー状の検出器であるのに対し、End cap はビーム軸に対し垂直な検出器の構造となっている。Central Solenoid からの磁場 2 T により荷電粒子が曲がり、曲率から運動量を測定することができる。内部検出器の構成は Pixel Detector, Semi-Conductor Tracker, Transition Radiation Tracker で構成されている。

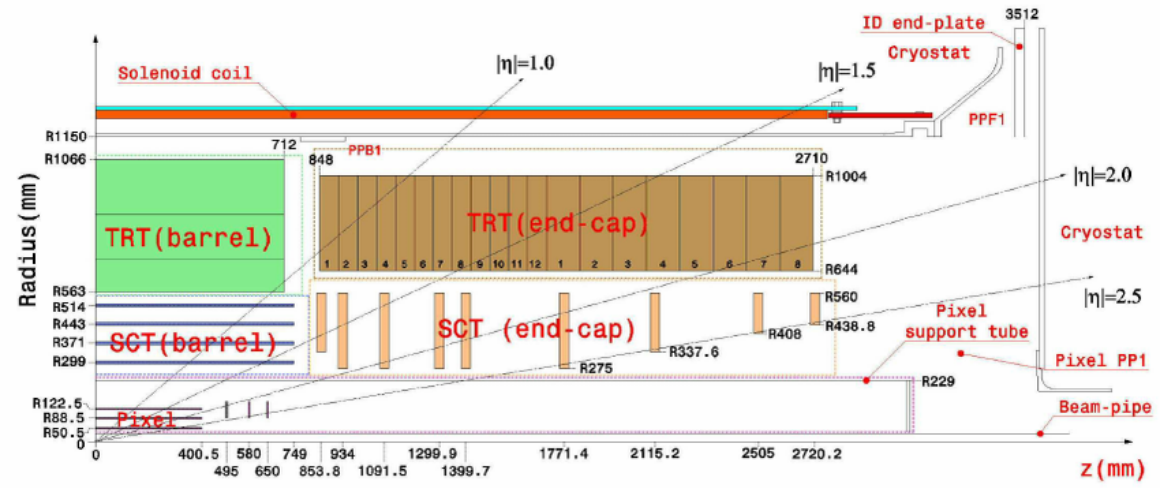


図 2.4: 内部検出器 [6]。

2.1.3 カロリメーター

カロリメーターは電子、光子に用いる電磁カロリメーターとハドロンに用いるハドロンカロリメーターに分けられる。全領域の電磁カロリメーターとエンドキャップ, フォワード領域のハドロンカロリメーターは検出層に液体アルゴンを用いており、吸収層には検出器の範囲に応じて物質 (鉛、銅、タングステン) を変えている。入射粒子は吸収層でカスケードシャワーを起こし、それらにより液体アルゴン原子から電離した電子を信号として読みだす。入射粒子のエネルギーはカスケードシャワーで生成される電子との線形性が良く、液体アルゴンは放射線耐性があるので採用された。バレル領域のハドロンカロリメーターはタイルカロリメーターと呼ばれ鉄を吸収層, プラスチックシンチレーターを検出層に用いている。カロリメーターはミューオンとニュートリノ以外の粒子を止めることができそれらのエネルギーを測定する。各カロリメーターの吸収層、検出層の物質は表 2.2 に纏めた。図 2.5 に示す液体アルゴンカロリメーターは横 12 m, 縦 8 m, 質量 4000 t の巨大な検出器である。

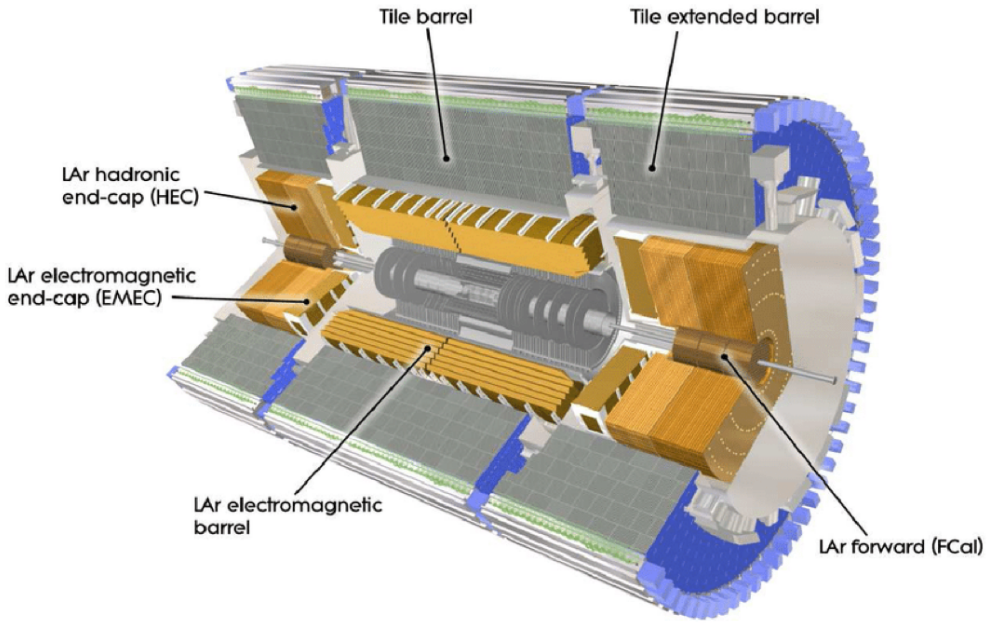


図 2.5: 黄色の検出器が液体アルゴンカロリメーター、灰色の検出器がタイルカロリメーター [7]。

LHC-ATLAS 実験で用いられるカロリメーターはすべてサンプリング型であり、サンプリング型のカロリメーターのエネルギー分解能は一般的に

$$\frac{\sigma}{E} = \frac{a}{E} + \frac{b}{\sqrt{E}} + c$$

と表せる [21]。第一項は Noise term とよばれノイズ・パイルアップの影響によるもので低エネルギーで優位。第二項は Sampling term と呼ばれ検出層、吸収層の物質やそれらの厚さなどに依存し 10∼100 GeV で優位。最後に第三項は Constant term と呼ばれ検出器の深さや不均一性などに依存し高エネルギー帯で優位に影響する。RUN3 実験は重心系エネルギー 14 TeV での陽子衝突型実験を行うため Constant term がエネルギー分解能を考えるうえで重要である。Constant term は電磁カロリメーターでは 0.7% 程度、ハドロンカロリメーターでは 10% 程度に設計されている。

液体アルゴンカロリメーター

電磁カロリメーターは $|\eta| < 4.9$ までをカバーし、3 つの検出器 LAr electromagnetic barrel (EMB), LAr electromagnetic end-cap (EMEC), LAr forward (FCal) に分けられる。EMB, EMEC はそれぞれ $|\eta| < 1.5$, $1.4 < |\eta| < 3.2$ までの領域をカバーしている。それぞれの検出器は 3 層構造 (Front layer, Middle layer, Back layer) であり、$|\eta| < 1.8$ の領域はさらにもう一層 (Presampler) あり Presampler は電磁カロリメーター以前の検出器でのエネルギー損失を補正する目的で用いられる。π^0 は 2.5×10^{-8} 秒の寿命を持ち、2 つの光子に崩壊するがそれらの光子は同方向に飛ぶので検出器の近い位置に同程度の信号を残す。それらを識別するために Front layer はほかの layer に比べ η 方向に細かく分けられている (表 2.1)。

EMB と EMEC で合計 173312 の Elementary Cell から構成されている。ハドロンカロリメーター

もサンプリング型のカロリメーターで $|\eta| < 4.9$ までをカバーしており 3 つの検出器 Tile Barrel, LAr hadronic end-cap (HEC), LAr forward (FCal) に分けられる。Tile Barrel, HEC, FCal はそれぞれ $|\eta| < 1.7$, $1.5 < |\eta| < 3.2$, $3.2 < |\eta| < 4.9$ までの領域をカバーしており、Tile Barrel, HEC はそれぞれ 5632, 3524 の Elementary Cell で構成されている。

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

表 2.1: Elementary Cell, Trigger Tower, Supercell の $\eta \times \phi$ のサイズ

LHC-ATLAS 実験で用いるフォワード領域以外の液体アルゴンカロリメーターは、図 2.6 に示すような特徴的なアコーディオン構造をしており ϕ 方向の不感領域をなくし、かつ検出器内にケーブルを這わす必要をなくすことができる。

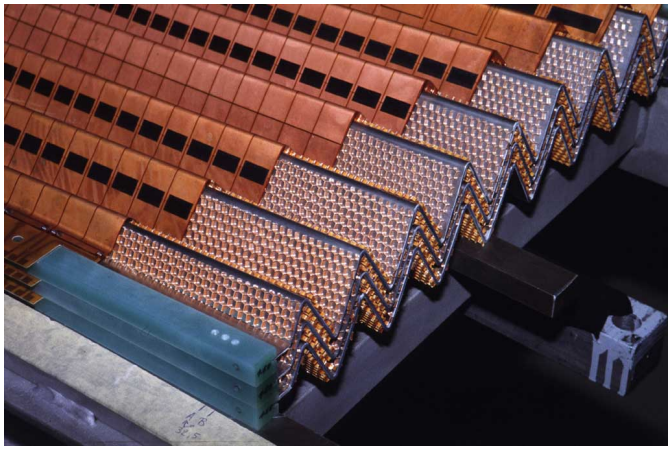
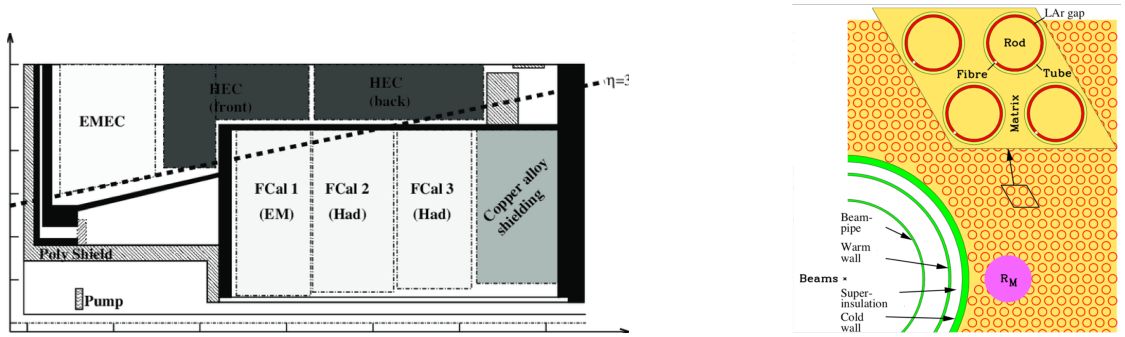


図 2.6: アコーディオン構造、衝突点から離れることにより電極間の間隔が大きくなる [42]。

フォワード領域は電磁カロリメーターが 1 層、ハドロンカロリメーターが 2 層の検出器で合計 3 層構造である。バレル領域とは異なり図 2.7 に示すようなストロー構造をしておりビーム軸に対し平行にストローが並べられており液体アルゴンが円筒形に満たされている。



(a) FCal の 3 層構造。　　**(b)** ストロー構造。

図 2.7: フォワードカロリメーター全体の構造と内部の構造 [9]。

タイルカロリメーター

タイルカロリメーターは $|\eta| < 1.7$ の領域までを 3 つの検出器がカバーする (図 2.5)。3 つ全ての検出器は 2π / 64 のおうぎ形のスライスを合わせた形状をしている (図 2.8)。吸収層のプラスチックシンチレーターの厚さは 3 mm と薄く、吸収層との比率は 4.7:1 で検出器の大きさを考慮しつつ高いレートに強く短い Interaction length を実現している。

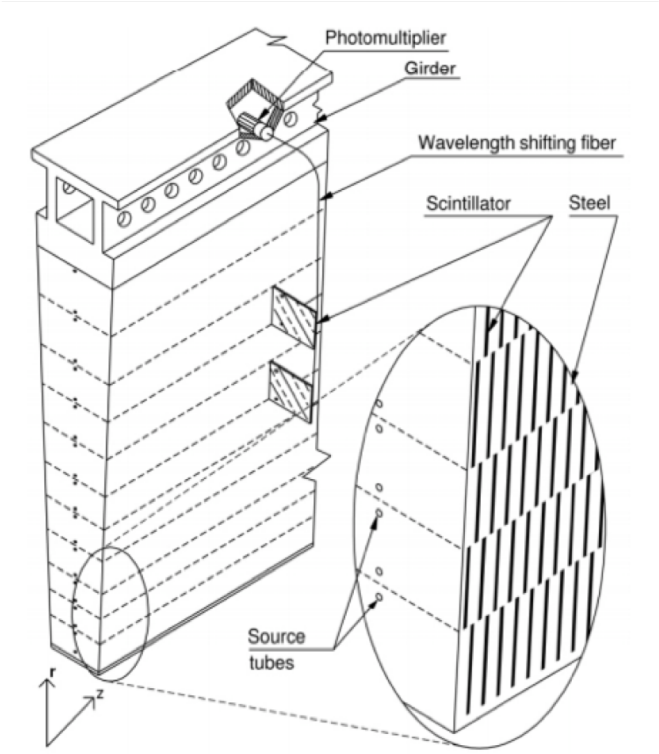


図 2.8: ハドロンカロリメーターのタイル構造 [8]。

	電磁カロリメーター		ハドロンカロリメーター		
	EMB	EMEC	Tile Barrel	HEC	FCal
$\|\eta\|$	0 - 1.5	1.4 - 3.2	0 - 1.7	1.5 - 3.2	3.2 - 4.9
吸収層	鉛	鉛	鉄	銅	銅、タングステン
検出層	液体アルゴン	液体アルゴン	プラスチックシンチレーター	液体アルゴン	液体アルゴン

表 2.2: カロリメーターの概要

2.1.4　ミューオンスペクトロメーター

ミューオンは質量が大きく制動放射の断面積が十分小さいのでカロリメーターを通り抜ける。ミューオンスペクトロメーター (図 2.9) は内部検出器、カロリメーターの外側に設置される検出器であり、Toroid マグネットにより曲げられたミューオンのトラック, 運動量を測定する。図 2.9 内の紫・オレンジ・赤の検出器は Monitored Drift Tubes (MDT)、緑は Thin-Gap Chambers (TGC)、青は Resistive-Plate Chambers (RPC)、黄色が Cathode Strip Chambers (CSC) を表している。

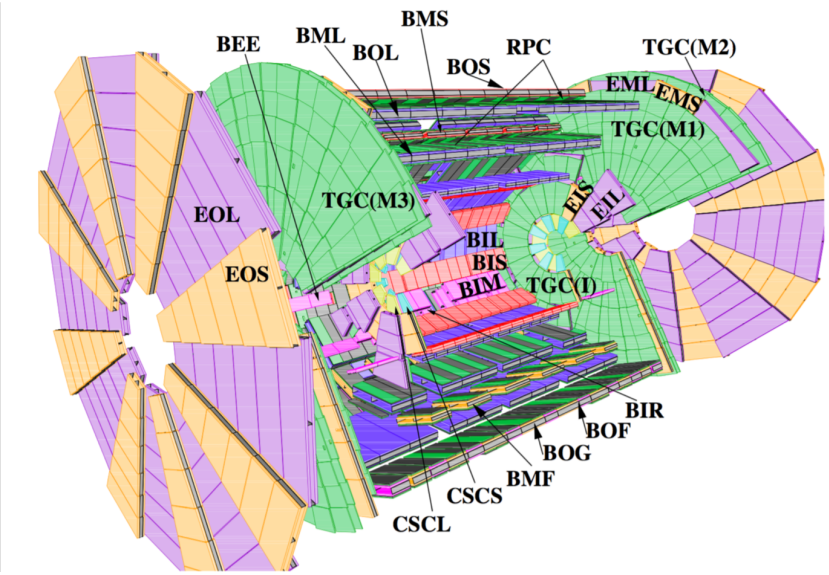


図 2.9: ミューオンスペクトロメーター。TGC, RPC はトリガー用に用いられ、それ以外の検出器でミューオンの運動量, 飛跡を算出している [2]。

2.2　トリガーシステム

LHC-ATLAS 実験では 40 MHz もの頻度で陽子衝突を起こしているが、ほとんどの陽子陽子衝突はクォークの非弾性散乱で興味ない事象が大半を占める。それら全ても保存するのは莫大なリソースが必要になり現実的でない。そのため LHC-ATLAS 実験では 2 段階のトリガーシステムを用いて興味ある事象のみを選別している。トリガーシステムは Trigger and Data Acquisition (TDAQ) システム (図

2.10) に属するシステムであり、TDAQ[10] は 3 つのステージ (Level 1 trigger, High level trigger, Data Acquisition) に分けられる。Level 1 trigger は高運動量のミューオン, 電子, 光子, ジェット, タウ粒子を限られた検出器の情報を用いて 2.5 μs 以内に荒く調査することにより 40 MHz の陽子衝突データを検出器全体で 100 kHz の収集頻度に落とす (液体アルゴンカロリメーター単体で 20 kHz)。Level 1 trigger を通過したイベントは High level trigger に送られる。Level 1 trigger はさらにおおまかなエネルギー分布の情報である Regions of Interest (RoI) を定義し High level trigger に送る。RoI は検出器のどの領域に粒子が大きく信号を残したかを示した情報である。High level trigger では Level 1 trigger の情報をより詳細に調査し、対象の粒子に依存したトリガーの方法を用いる。最終的に High level trigger により約 550 ms 程度で 100 kHz を 1 kHz の収集頻度にまで落とす。

2.3 Data Acquisition

Level 1 trigger を通った事象は全ての検出器の情報が Readout Drivers (ROD) に保存される。それらは検出器の実際のデータにデコードされ、Data Acquisition (DAQ) システムに送信される。その後 High level trigger を通ったデータは物理解析用に大容量ストレージ内に保存される。

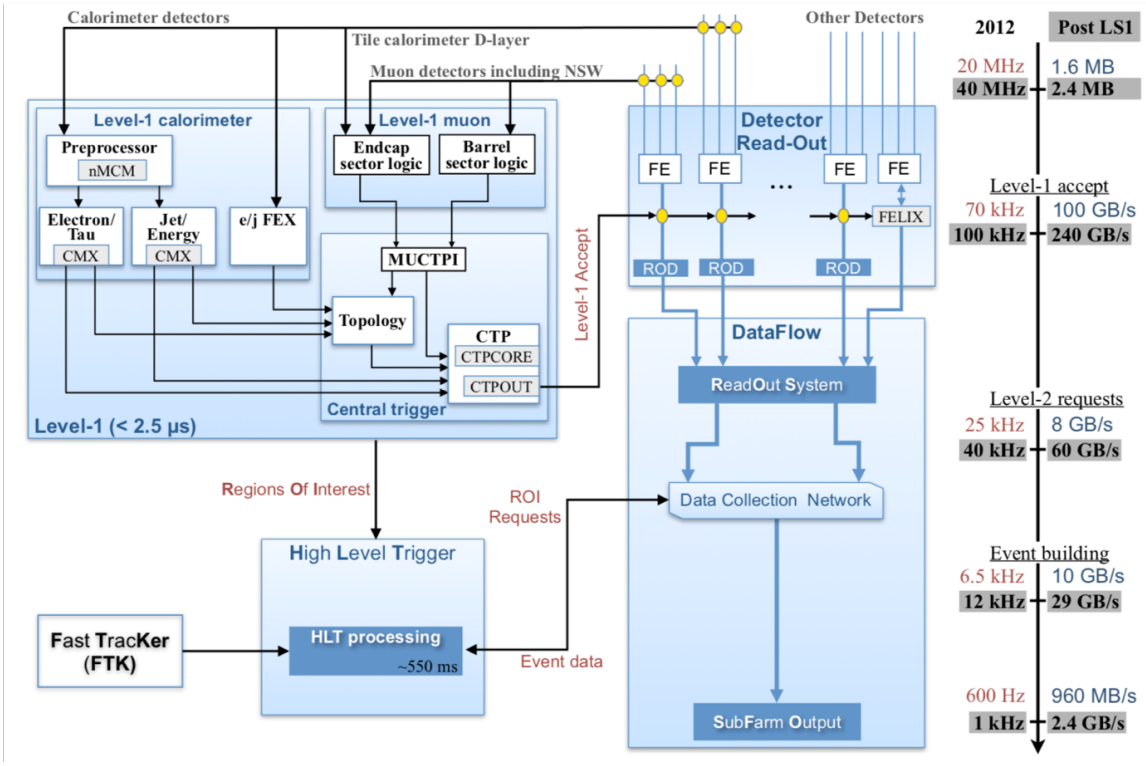


図 2.10: Trigger と Data Acquision system、Post LS1 が RUN3 に対応する。RUN3 では Level 1 trigger で検出器全体で 100 kHz の頻度にまで収集頻度を落とす要求があり液体アルゴン検出器単体で 20 kHz まで落とす [10]。

2.4 Upgrade 計画

現在 LHC は運転を停止し RUN3 からの実験に向けたアップグレードの時期にある (Phase-I Upgrade)。液体アルゴン検出器、内部検出器は検出器自体の更新は行わず読み出しの増強により対応する。ミューオンスペクトロメーターは新しい検出器 (New Small Wheel) を導入し、RUN3 からのルミノシティーの増強に備える。RUN3 は 2021 年から 2023 年までの間行われその後再び LHC の運転を停止し (Phase-Ⅱ Upgrade) RUN4, RUN5 に備える。Phase-Ⅱ Upgrade では最大瞬間ルミノシティーが L=7.5 × 10^{34} $cm^{-2}s^{-1}$ に増強され、内部検出器, カロリメーターは検出器の取り替えが行われる (図 2.11)。

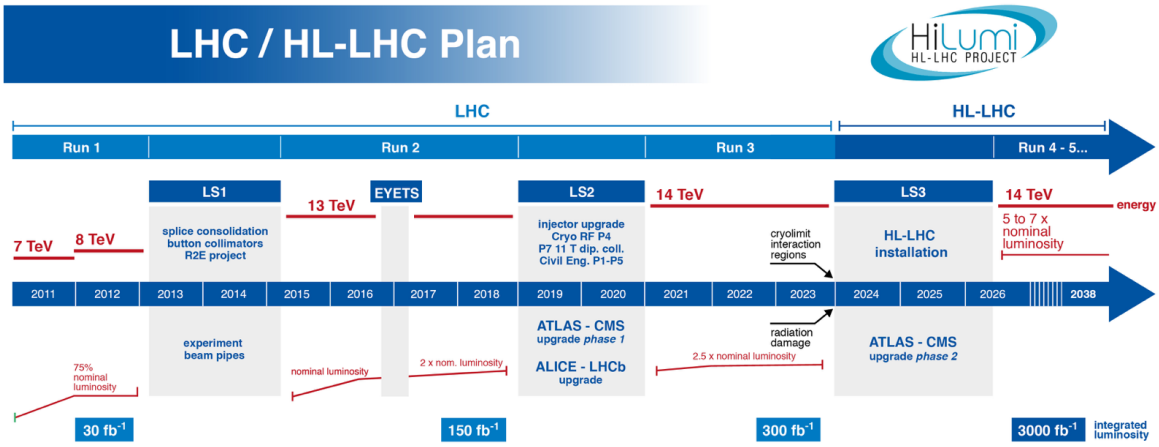


図 2.11: LHC, HL-LHC アップグレード計画 [18]。

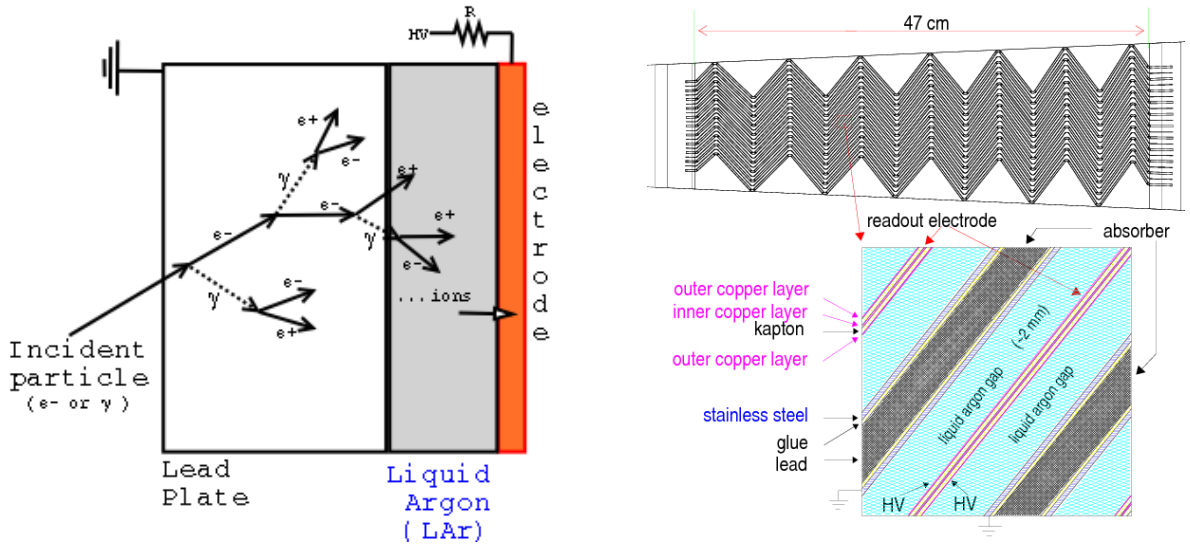
第 3 章

LAr Phase-1 アップグレード

2021 年に開始される RUN3 実験の運転に向けて、検出器信号読出しシステムのアップグレード (Phase-I Upgrade) が行われている。液体アルゴン検出器のアップグレードでは検出器自体の更新は行わず、信号読み出し回路のみ改善する。

3.1　液体アルゴン検出器からの信号と再構成

臨界エネルギー以上の電子、光子が液体アルゴン検出器に入射すると制動放射、対生成が起きる。このプロセスは臨界エネルギーに達するまで引き起こされ、図 3.1a ような電磁シャワーが観測される。図 3.1b で示されるように液体アルゴン検出器はサンプリング型のカロリメーターであり、吸収層に原子番号の大きい金属 (主に鉛)、検出層に液体アルゴンを用いている。

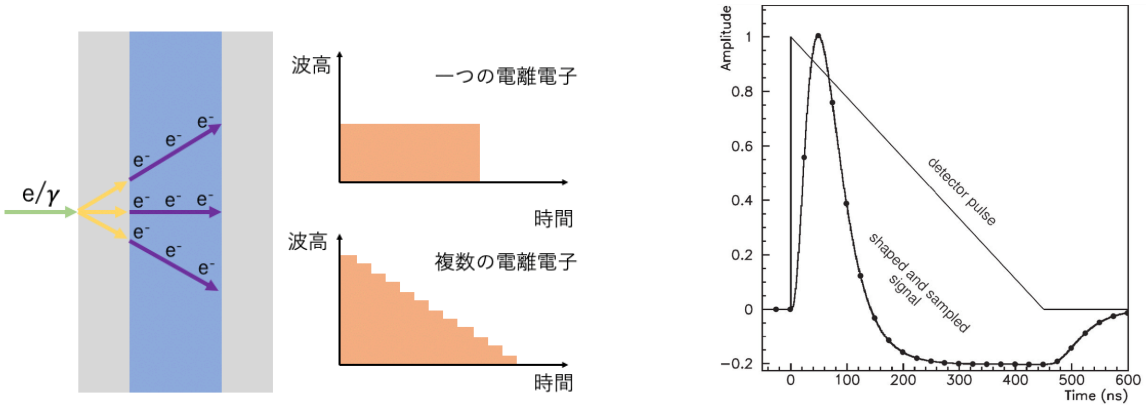


(a) 信号読み出し機構　　**(b)** 液体アルゴン検出器の構造

図 3.1: 液体アルゴンの検出器構造と信号読み出し機構 [19]。

電磁シャワーにより液体アルゴンは電離され、電離電子は液体アルゴンギャップ間に印加された 2 kV

の電圧差により移動する。その際図 3.2a のように各々の電離電子は矩形波の信号を残す。アルゴンイオンは電子に比べ易動度が小さくそれらの残す信号は無視できるほど小さい。電離電子は液体アルゴンギャップ間に均等に生成されるので、各々の電離電子が残す信号が重なり合い三角波が観測される。三角波は信号立ち上がりが 1 ns 以下で液体アルゴンギャップの間隔に依存するがおおよそ 450 ns 程度である。この三角波はアンプにより増幅され、$CR-(RC)^2$ 回路により Bipolar 波形に変換される。Bipolar 波形の特徴として波形のプラス領域の面積とマイナス領域の面積が等しく、時間積分をすると 0 になる点がある。これによりノイズやパイルアップが均等に重なり合うことでそれらの影響を無視した信号が得られる。液体アルゴンギャップは 2 mm 程度であり、検出器からの信号は図 3.2b のような約 600 ns の波形になる。そのため 1 つの入射粒子は 600 ns の波形を作り、その波形が 1 つのエネルギーに対応する。



(a) 三角波の生成原理　　**(b)** Bipolar 波形 [11]

図 3.2: 検出器からの信号は各電子の残す波形の重ね合わせにより三角波として読み出される。三角波はアンプ、シェイパーを通り Bipolar 波形に変換される。1 つの Bipolar 波形は 1 つの入射粒子が液体アルゴンカロリメーターで落としたエネルギーに対応する。

LHC には 25 ns 間隔で陽子が詰められており、それをバンチ構造という。そのため各陽子衝突事象が残す信号の重なり合いが起こり、この現象はパイルアップと呼ばれる。電離電子の数はカスケードシャワーの規模に比例し、カスケードシャワーの規模は入射粒子のエネルギーに比例するので、Bipolar 波形の波高は入射粒子のエネルギーに対し線形である。また Bipolar 波形の長さは液体アルゴンのギャップ幅に依存し各 Supercell 固有の値である。つまり同じ Supercell で得られる信号は波高が定数倍され長さが一定の波形になる。理想的な波形を g、得られた信号を S_i とし位相のズレを τ、理想的な波形と得られた信号の波高の比率を A とし S_i を g を用いて表現する。

$$\begin{aligned} S_i(t_i) &= Ag(t_i-\tau) \\ &= Ag(t_i) - A\tau \left.\frac{dg(t)}{dt}\right|_{t=t_i} - A\tau^2 \left.\frac{d^2g(t)}{dt^2}\right|_{t=t_i} \cdots \\ &\simeq Ag(t_i) - A\tau\dot{g}(t_i) + n_i \end{aligned}$$

τ は Bipolar 波形の長さに対し十分に小さいものであると考え 3 次の展開項以上を 0 に近似する。また各バンチでノイズなどの影響は完全に打ち消しあわず正確に理想的な波形の定数倍とならないことが多い。

それらの影響を n_i に全て押し付けた。ここで得られた信号の N 個のサンプリング S_0, ..., S_{N-1} の線形結合を 2 つ用意する。

$$u = \sum_{i=0}^{N-1} a_i S_i$$

$$v = \sum_{i=0}^{N-1} b_i S_i$$

ここで係数 a_i と b_i は多数回試行における u の期待値が波高 A、v の期待値が波高と位相のズレ τ の積 $A\tau$ と一致するように選ばれる。

$$\begin{aligned} A \equiv \langle u \rangle &= \sum_{i=0}^{N-1} \{\langle A a_i g(t_i) \rangle - \langle A\tau a_i \dot{g}(t_i) \rangle + \langle a_i n_i \rangle\} \\ &= \sum_{i=0}^{N-1} \{A a_i g(t_i) - A\tau a_i \dot{g}(t_i) + a_i \langle n_i \rangle\} \end{aligned} \tag{3.1}$$

$$A\tau \equiv \langle v \rangle = \sum_{i=0}^{N-1} \{A b_i g(t_i) - A\tau b_i \dot{g}(t_i) + b_i \langle n_i \rangle\} \tag{3.2}$$

任意の τ に対し以上の条件を満たすには

$$\sum_{i=0}^{N-1} a_i g(t_i) = 1 \tag{3.3}$$

$$\sum_{i=0}^{N-1} a_i \dot{g}(t_i) = 0 \tag{3.4}$$

$$\sum_{i=0}^{N-1} b_i g(t_i) = 0 \tag{3.5}$$

$$\sum_{i=0}^{N-1} b_i \dot{g}(t_i) = -1 \tag{3.6}$$

である必要がある。ここでカロリメーターの入力信号に含まれるノイズは時間平均すると 0 になると考えられる。つまり多数回の試行に置いて $\langle n_i \rangle = 0$ である。$\langle n_i \rangle = 0$ と式 (3.1)、 式 (3.2)、 式 (3.3) $\sim$ (3.6)

を用いて u の分散 $Var(u)$ と v の分散 $Var(v)$ は

$$\begin{aligned}
Var(u) &= \left\langle u^2 \right\rangle - \left\langle u \right\rangle^2 \\
&= \left\langle \sum_{i=0}^{N-1} [Aa_i g(t_i) - A\tau a_i \dot{g}(t_i) + a_i n_i] \sum_{j=0}^{N-1} [Aa_j g(t_j) - A\tau a_j \dot{g}(t_j) + a_j n_j] \right\rangle \\
&\quad - \sum_{i=0}^{N-1} [Aa_i g(t_i) - A\tau a_i \dot{g}(t_i) + a_i \left\langle n_i \right\rangle] \sum_{j=0}^{N-1} [Aa_j g(t_j) - A\tau a_j \dot{g}(t_j) + a_j \left\langle n_j \right\rangle] \\
&= \left\langle \left(A + \sum_{i=0}^{N-1} a_i n_i \right) \left(A + \sum_{j=0}^{N-1} a_j n_j \right) \right\rangle - A^2 \\
&= \left(A^2 + 2A \sum_{i=0}^{N-1} a_i \left\langle n_i \right\rangle + \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} a_i a_j \left\langle n_i n_j \right\rangle \right) - A^2 \\
&= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} a_i a_j \left\langle n_i n_j \right\rangle \\
&\equiv \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} a_i a_j R_{ij} \qquad (3.7)
\end{aligned}$$

$$\begin{aligned}
Var(v) &= \left\langle v^2 \right\rangle - \left\langle v \right\rangle^2 \\
&= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b_i b_j \left\langle n_i n_j \right\rangle \\
&\equiv \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b_i b_j R_{ij} \qquad (3.8)
\end{aligned}$$

となる。式 (3.7) 内にある $R_{ij} = \left\langle n_i n_j \right\rangle$ はノイズの自己相関関数を表している。

Optimal filter はノイズの影響を抑えるフィルタリング機構であり、$Var(u)$ と $Var(v)$ を最小にするような係数 a_i, b_i を要求する。式 (3.3) $\sim$ (3.6) のような束縛条件下での最適解を求めるために Lagrange の未定乗数法を用いる。Lagrange 乗数は $\lambda, \kappa, \mu, \rho$ を用いる。

$$\begin{aligned}
I_u &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} R_{ij} a_i a_j - \lambda \left(\sum_{i=0}^{N-1} a_i g(t_i) - 1 \right) - \kappa \sum_{i=0}^{N-1} a_i \dot{g}(t_i) \\
I_v &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} R_{ij} b_i b_j - \mu \sum_{i=0}^{N-1} b_i g(t_i) - \rho \left(\sum_{i=0}^{N-1} b_i \dot{g}(t_i) + 1 \right)
\end{aligned}$$

二組の係数 a_i, b_i に対しそれぞれ最適解を導く方程式が得られる。それぞれ最適な係数を求めるために a_i, b_i で偏微分したものを 0 とする。

$$\begin{aligned}
\frac{\partial I_u}{\partial a_i} &= \sum_{j=0}^{N-1} R_{ij} a_j - \lambda g(t_i) - \kappa \dot{g}(t_i) = 0 \\
\frac{\partial I_v}{\partial b_i} &= \sum_{j=0}^{N-1} R_{ij} b_j - \mu g(t_i) - \rho \dot{g}(t_i) = 0
\end{aligned}$$

これを行列形式を用いて係数 $\vec{a}, \vec{b}$ について書き直せば

$$\vec{a} = \lambda \boldsymbol{R}^{-1}\vec{g} + \kappa \boldsymbol{R}^{-1}\vec{\dot{g}}$$
$$\vec{b} = \mu \boldsymbol{R}^{-1}\vec{g} + \rho \boldsymbol{R}^{-1}\vec{\dot{g}}$$

となり、式 (3.3) ∼ (3.6) での条件を用いるために行列表示の式に左から $\vec{g}^T, \vec{\dot{g}}^T$ をかける

$$\vec{g}^T \cdot \vec{a} = \lambda \vec{g}^T \boldsymbol{R}^{-1}\vec{g} + \kappa \vec{g}^T \boldsymbol{R}^{-1}\vec{\dot{g}} = 1$$
$$\vec{\dot{g}}^T \cdot \vec{a} = \lambda \vec{\dot{g}}^T \boldsymbol{R}^{-1}\vec{g} + \kappa \vec{\dot{g}}^T \boldsymbol{R}^{-1}\vec{\dot{g}} = 0$$
$$\vec{g}^T \cdot \vec{b} = \mu \vec{g}^T \boldsymbol{R}^{-1}\vec{g} + \rho \vec{g}^T \boldsymbol{R}^{-1}\vec{\dot{g}} = 0$$
$$\vec{\dot{g}}^T \cdot \vec{b} = \mu \vec{\dot{g}}^T \boldsymbol{R}^{-1}\vec{g} + \rho \vec{\dot{g}}^T \boldsymbol{R}^{-1}\vec{\dot{g}} = -1$$

あとは λ, κ と μ, ρ のそれぞれに対し連立方程式を解けば

$$\lambda = \frac{\vec{\dot{g}}^T \boldsymbol{R}^{-1}\vec{\dot{g}}}{\vec{g}^T \boldsymbol{R}^{-1}\vec{g}\,\vec{\dot{g}}^T \boldsymbol{R}^{-1}\vec{\dot{g}} - \left(\vec{\dot{g}}^T \boldsymbol{R}^{-1}\vec{g}\right)^2}$$
$$\kappa = \frac{-\vec{\dot{g}}^T \boldsymbol{R}^{-1}\vec{g}}{\vec{g}^T \boldsymbol{R}^{-1}\vec{g}\,\vec{\dot{g}}^T \boldsymbol{R}^{-1}\vec{\dot{g}} - \left(\vec{\dot{g}}^T \boldsymbol{R}^{-1}\vec{g}\right)^2}$$
$$\mu = \frac{\vec{g}^T \boldsymbol{R}^{-1}\vec{\dot{g}}}{\vec{g}^T \boldsymbol{R}^{-1}\vec{g}\,\vec{\dot{g}}^T \boldsymbol{R}^{-1}\vec{\dot{g}} - \left(\vec{\dot{g}}^T \boldsymbol{R}^{-1}\vec{g}\right)^2}$$
$$\rho = \frac{-\vec{g}^T \boldsymbol{R}^{-1}\vec{g}}{\vec{g}^T \boldsymbol{R}^{-1}\vec{g}\,\vec{\dot{g}}^T \boldsymbol{R}^{-1}\vec{\dot{g}} - \left(\vec{\dot{g}}^T \boldsymbol{R}^{-1}\vec{g}\right)^2}$$

となる。得られた Lagrange 未定乗数とノイズの自己相関関数を用いれば係数 a_i, b_i が得られる。RUN3 からも RUN2 同様 4 点を用いたアルゴリズムを採用する予定であり、LHC の Physics RUN が始める前に自己相関関数を観測し係数を算出する。理想的な波形 g がエネルギーに対し規格化されている場合、波高 A はエネルギー (E_{T}) に対応し Aτ はエネルギーと位相のズレの積 ($E_{\mathrm{T}}\tau$) に対応する。この時求められる係数は Bipolar 波形のうち始めの 4 点を用いた場合に正確な E_{T}, $E_{\mathrm{T}}\tau$ が算出されるようにデザインされている。そのためその他の 4 点を用いて算出された A, Aτ は E_{T}, $E_{\mathrm{T}}\tau$ に対応しない。計算された A, Aτ から正しい 4 点を用いて計算されたものを選び出すことにより正しく値を算出することができる。この機構はエネルギー算出のアルゴリズム内に実装されており、5.3 節で詳しく述べる。

3.2　読み出し方法エレクトロニクス

1.1 節で説明したように、RUN3 実験では RUN2 に比べて 10 倍細かいトリガー読み出しを行う。そのためそれら全てを処理するシステムの開発が現在行われている。アップグレードは Front End, Back End 双方で行われる。以降では主に図 3.3 で示した RUN3 から新たに導入されるシステムについて述べる。

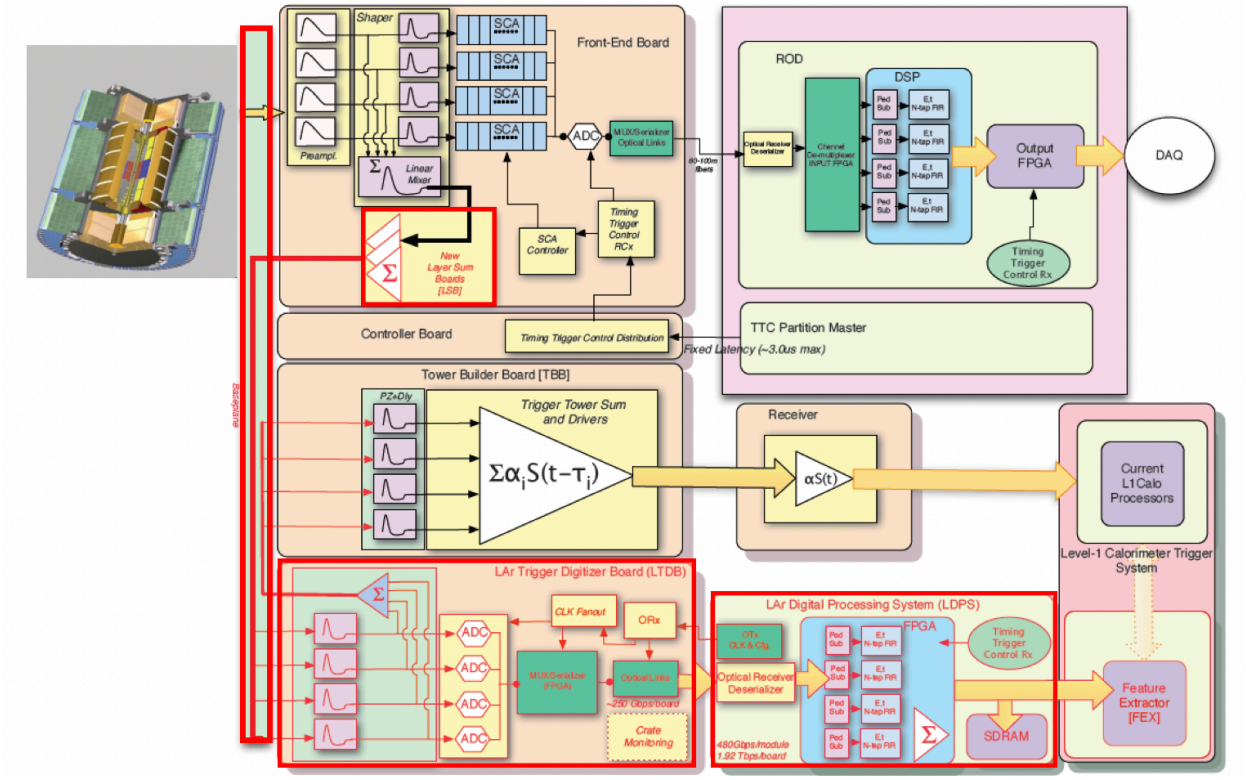


図 3.3: トリガー読み出し機構。赤で囲まれた部分が RUN3 から新たに導入される部分である [9]。

3.2.1 New Layer Sum Board

RUN2 までは Trigger Tower ($\Delta\eta \times \Delta\phi = 0.1 \times 0.1$) をトリガー読み出しとしていたため Elementary Cell で得られた信号を Layer Sum Board で Trigger Tower の領域内の信号のアナログ和を生成していた。RUN3 からは Supercell ($\Delta\eta \times \Delta\phi = 0.025 \times 0.1$) を用いたトリガーを行うので、New Layer Sum Board を導入してより細かい領域の信号を生成する。

3.2.2 LAr Trigger Digitizer Board

New Layer Sum Board により各 Supercell 毎に足し上げられたアナログ信号は、図 3.4 の LAr Trigger Digitizer Board (LTDB) で 40 MHz の頻度で 12 bit の ADC データに変換される。デジタイズは 1 mV あたり 1 ADC count とする mV Scheme を採用している。1 つの LTDB で最大 320 Supercell を処理する (表 3.1)。Supercell の数は検出器の位置に依存しており LTDB は Back End に 40 本の光ファイバーを用いて ADC データを伝送し、1 本あたりは 5.12 Gbps の通信を行う。また RUN2 で用いられたトリガー方法 (Legacy System) も RUN3 始めで用いる為、New Layer Sum Board で生成された Front layer, Middle layer の Supercell 信号を各 Layer 毎に足し合わせる。それぞれの Layer で足し上げられた信号は Tower Builder Board (TBB) で Layer 間の足し合わせが行われ Trigger Tower の領域のアナログ信号に形成される。

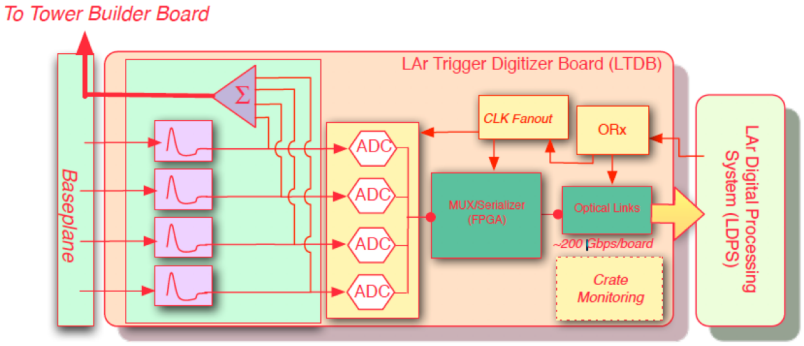


図 **3.4:** LTDB のブロックダイアグラム [9]。

LTDB Type	Supercell 数	LTDB の数
EMB	290	64
EMEC Std	312	32
EMEC Spc 0	240	8
EMEC Spc 1	160	8
HEC	192	8
FCAL 0	192	2
FCAL 1	192	2
合計	34048	124

表 3.1: 各領域の LTDB 数と LTDB が担当する Supercell 数。EMB, EMEC Std, HEC の LTDB は 1 つのクレートにつき 1 枚挿入される。一方 EMED Spec, FCAL は 1 つのクレートに 2 枚挿入され、0, 1 はそれを区別インデックスである。

3.2.3 LAr Digital Processing Board

LTDB からの 12 bit の ADC データは、図 3.5 の LAr Digital Processing Board (LDPB) でエネルギーに変換される。全部で 31 個の LDPB は 31 枚の LDPB Carrier Board と 124 枚の AMC により構成される。1 枚の LTDB は 4 枚の AMC を搭載しており、LDPB Carrier Board と AMC はそれぞれ XILINX 社、Intel 社の FPGA を搭載している。各 AMC はそれぞれ複数の LTDB とつながっており、LTDB と同様最大 320 Supercell を一度に扱うことができる。1 枚の AMC は 40 本の光ファイバーを用いてデータを受信 (5.12 Gbps/本) し、40 本の光ファイバーを用いてデータを送信 (11.2 Gbps/本) する。つまり 1 枚の AMC は約 200 Gbps の受信と約 450 Gbps の送信を担う。LDPB は ATCA 規格 [14] を採用しており、LDPB が ATCA Blade に対応し、3 つの ATCA Shelf で全ての Supercell データが処理される。

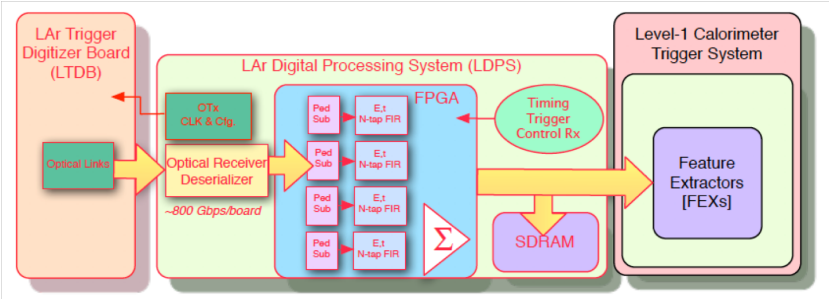
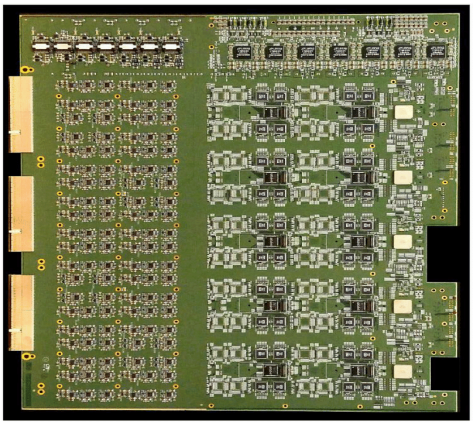
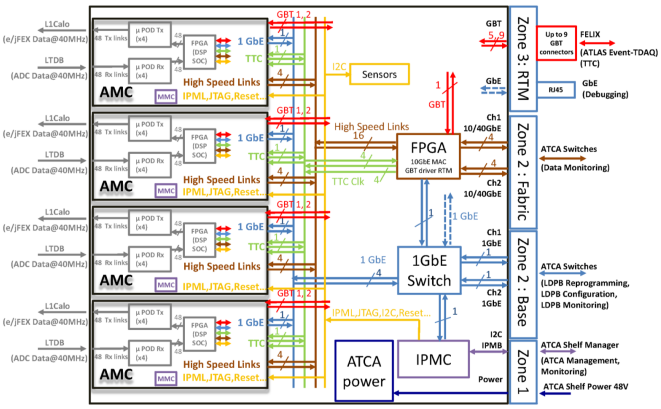


図 3.5: LDPB のブロックダイアグラム [9]。



(a) 実際の LTDB[12]



(b) LDPB の内部配線図 [13]

図 3.6: LTDB と LDPB。どちらも既に大量生産されている。

3.2.4　Trigger Feature Extractor

LDPB は各 Supercell ごとに ADC データをエネルギーに変換し、3 種類のエネルギー情報 (e, j, g FEX) を Trigger Feature Extractor(FEX)[15] に送る。表 3.2 に示すように eFEX は電子/光子/タウ粒子、jFEX はジェット、gFEX は半径の大きいジェットの同定に用いられる。各 FEX は電磁カロリメーターとハドロンカロリメーター双方の情報を用いて同定を行う。

	ターゲット	$\Delta\eta \times \Delta\phi$	カバーする領域	データ量
eFEX	電子/光子/タウ粒子	0.025×0.1	$\lvert\eta\rvert \leq 2.5$	14 Tbps
jFEX	ジェット	0.1 ×0.1	$\lvert\eta\rvert \leq 4.9$	3 Tbps
gFEX	半径の大きいジェット	0.2 ×0.2	$\lvert\eta\rvert \leq 4.9$	0.4 Tbps

表 3.2: 各 FEX の役割と領域 [15]

3.2.5　Front End Link Exchange

図 3.7 に示す Front End LInk eXchange (FELIX)[16] は RUN3 から導入されるフロントエンドエレクトロニクスと TDAQ システム間を結ぶインターフェースである。FELIX は GBT Link [*1]を用いて Trigger Timing Control (TTC) 信号を LTDB, LDPB に分配する。さらに LTDB のコンフィグレーションや、Monitoring data の読み出し、LDPB から TDAQ readout の読み出しも行う。

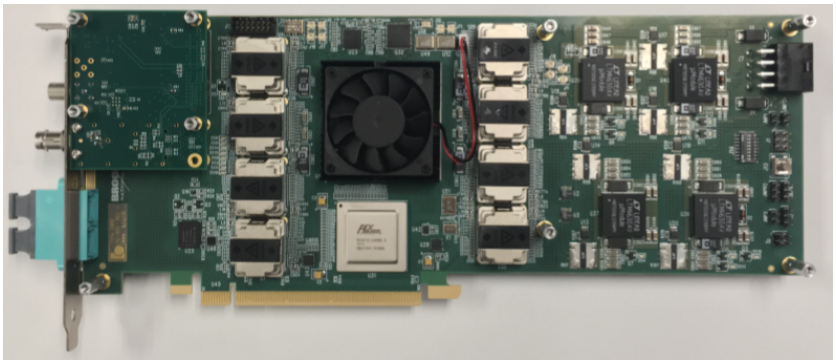


図 3.7: FELIX Board。XILINX Kintex UltraScale FPGA (XCKU115-FLVF-1924) を搭載し、MiniPOD を用いて 48 本の双方向高速通信を実現している。[16]

3.3　Field Programmable Gate Array (FPGA)

FPGA は 1980 年代中旬に XILINX 社により世界で初めて製品化された。FPGA は回路の要素だけが実装されており、ユーザーがそれらを思い通りに繋げ合わせることにより様々な回路を再現することができる。近年の技術発展に伴って FPGA 内部の要素数が加速度的に増加し、高速動作する複雑な機能を FPGA に実装することが可能になった。主に XILINX 社と Intel 社の ATLERA が FPGA を製造している。本研究は INTEL 社製 Arria 10 GX を用いたので、それに焦点を当てて説明する。

3.3.1　回路合成方法

多くの FPGA のデバイス規模はロジックエレメント (XILINX 社製の FPGA はロジックセル) と呼ばれる最小単位の搭載数が指標になる。FPGA はユーザー側が任意の回路を作成できるが、書き換え可能な箇所はロジックエレメント (LE) のデーブル情報と配線間のスイッチのみである。LE はルックアップテーブル (LUT) とフリップフロップ (FF) から構成されており一般的に広く FPGA に使われている LUT は 4 入力 1 出力となっている。LUT のテーブル情報は SRAM [*2]に書き込まれ、このテーブル情報

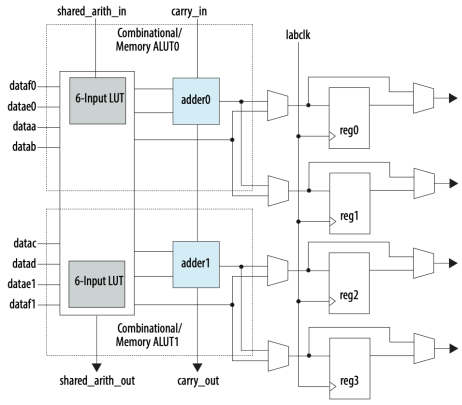
[*1] GBE Link は 4.8 Gbps の双方向通信を可能にする機構である。CERN により開発され、FELIX からの低レイテンシーでの TTC 信号の伝送を可能にしている [17]。

[*2] RAM には大きく分けて 2 種類存在する。SRAM は書き込んだデータが電源供給をやめない限り失われない。一方 DRAM は時間とともにデータが失われるため一定時間毎に再書き込みが必要である。

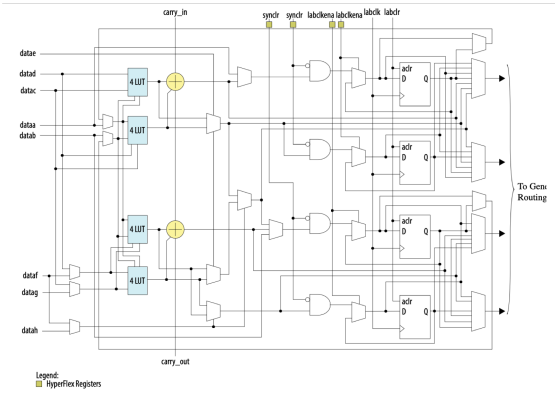
をユーザーが書き換えることにより様々な回路の合成を可能にしている。RAM のアドレスに対応する 4 入力に依存して値を読み出し、FF を用いることにより指定したクロックと同期したデータを作成している。それらの LE と配線間のスイッチを切り替えることにより大規模な回路合成を可能にしている。

3.3.2 Adaptive Logic Module (ALM)

従来の最小単位であった LE ではなく、それに加算器を加えた Adaptive Logic Module (ALM) と呼ばれるものが近年の Intel 社製の FPGA の論理リソースの最小単位である、Arria 10 GX に搭載されている ALM は最大 8 入力の LUT と 2 つの加算器, 4 つのレジスターで構成されている。LE に比べ、ALM は集積率が 2 倍以上になり演算機能や組み合わせ回路を効率的に実装できる。ALM の構造は図 3.8 のように FPGA の種類に依存し、例えば Stratix 10 では 4×2 入力の LUT を採用している。実際の FPGA には ALM 10 個をひとまとまりとして Logic Array Block (LAB) が一つの機能となる。本研究で用いた Arria 10 GX には ALM が 427,200 個 (LAB が 42,720 個) 搭載されている。



(a) Arria 10 の ALM[37]



(b) Stratix 10 の ALM[38]

図 3.8: どちらも Intel 社製の FPGA であるが ALM の配線図や内部の素子が異なる。Arria 10 の ALM 内の LUT は 8, 7, 6 入力など様々な入力に対応している。Stratix 10 は Phase-II Upgrade で用いられる。

3.3.3 Phase Locked Loop (PLL)

本研究で作成したファームウェアはいくつかの周波数のクロックを用いているが、それらすべてのクロックが検出器全体で用いられているクロックに同期していなければならない。そのため大元となるクロックから 図 3.9 に示す Phase Locked Loop (PLL) を用いて分周/逓倍させている。これによりファームウェア内でのクロックは検出器で用いるクロックと同期させることができ、かつ FPGA 内のクロック源[*3]では作り出せないクロックの作成も行うことができる。

[*3] 水晶の電圧効果を利用して高精度の発信を起こすことが可能な水晶振動子が一般的に用いられる。

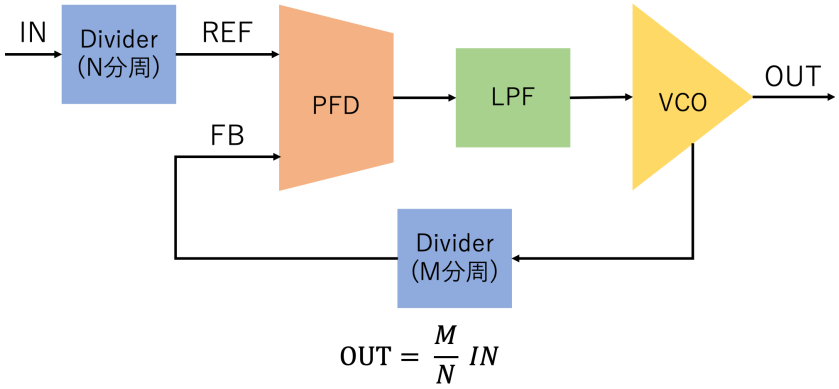


図 3.9: PLL は入力信号 (IN) と異なる周波数のクロック (OUT) を出力し、それらの同期を取るために用いられる。PFD が REF clock と FB clock の周波数を合わせる。VCO が適切な周波数に調節する。入力の clock は 1 つ目の Divder で分周され、出力のクロックが 2 つ目の Divder で逓倍されるのでそれらの周波数を合わせることにより様々な周波数を実現できる。ただし PLL が全ての任意の周波数を実現できるわけではない。

3.3.4 メモリ

Arria 10 には大規模開発用に M20K と呼ばれる 20480 memory bit[*4]までをカバーしたサイズの大きいメモリが 2,713 個搭載されている。数種類あるメモリうちの一つの Random Access Memory (RAM) は、クロックに同期した任意のタイミングで対応するアドレスを与えることにより値の書き込み、読み出しが可能なものである。それは最大で 2 つのクロックを同期させることが可能で 1 クロックのみに同期した RAM は 1-port RAM, 2 クロックに同期したものは 2-port RAM と呼ばれる。また Arria 10 は小規模メモリ用に LAB をメモリ (MLAB) として用いることも可能で 1 つで 640 memory bit までを扱うことができる。また任意のタイミングでの値の書き込みができない Read Only Memory (ROM) や図 3.10 に示す First In First Out (FIFO) や First In Last Out (FILO) なども論理合成に用いられている。

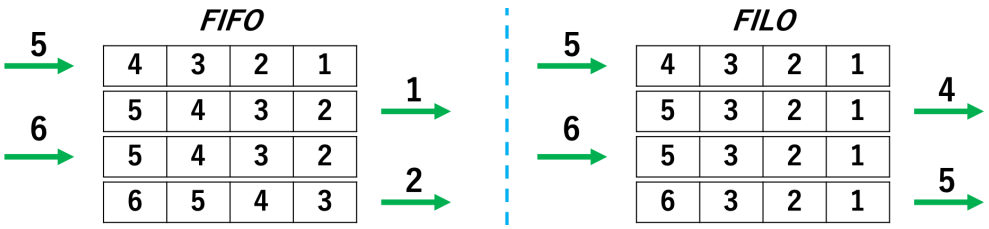


図 3.10: FIFO と FILO。どちらも depth 4 で FIFO は入力したデータを入力した順番に読み出す。FILO は初めに入力したデータを後から読み出す方式である。

[*4] アドレスが n bit, データ幅が m bit の場合 memory bit は $2^n \times m$ bit と計算される

3.3.5 Digital Signal Proccessor Block (DSP Block)

ハードウェア上で高速に乗算を行うためには Megafunction [*5]である DSP block を用いる。DSP block は乗算専用の Megafunction であるので効率的に乗算を行うことができる。Arria 10 に搭載されている DSP Block には 18 bit × 19 bit 演算用と 27 bit × 27 bit 演算用の 2 種類がある。本研究では 1 つの DSP block で 2 つの乗算を行うことのできる 18 bit × 19 bit 演算用のものを利用している。18 bit × 19 bit 演算用 DSP Block は図 3.11 と図 3.12 に示すように 2 種類の使い方が存在し、DSP block は Arria 10 内に一列に配置されているので、隣の DSP block の出力を受け取ることができ乗算と加算を効率よく行うことができるようデザインされている。

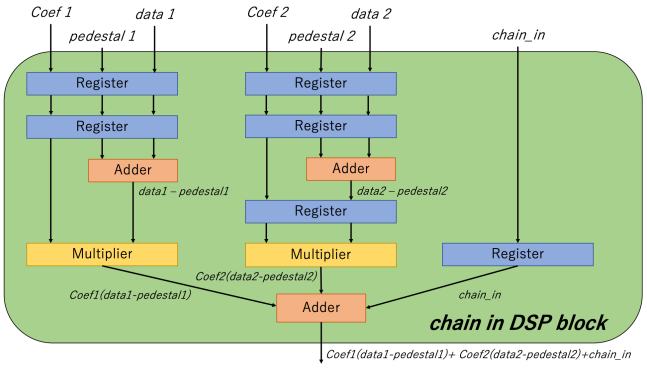


図 3.11: Chain in DSP Block の内部配線図。レジスター数がその経路のレイテンシーに対応する。入力のデータは全て 18 bit まで対応しており内部の加算器で最大 19 bit とされ乗算器で 18 bit × 19 bit の乗算を行うことが可能である。chain_in は 44 bit まで対応する。

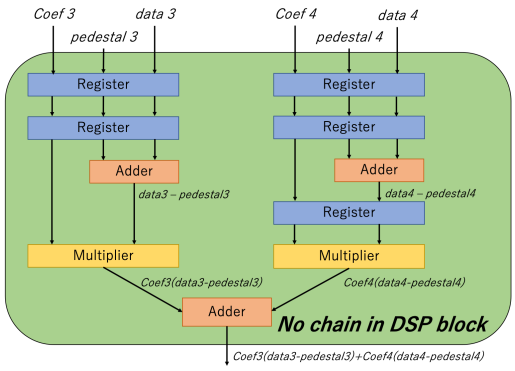


図 3.12: No chain in DSP Block の内部配線図。レジスター数がその経路のレイテンシーに対応する。

[*5] Intel 社が提供する FPGA にはもともと固有の機能を有したロジックのまとまりがいくつか存在する。それらは Megafunction と呼ばれ効率的に機能を用いることができる。

第 4 章

LATOME プロジェクト

4.1 LATOME ボード

RUN3 からの新しいトリガー読み出しのため、3.2.3 節で説明したように、LDPB が新たに導入される。LDPB 上に新たに搭載される AMC がフランスにある LAPP 研究所で開発された。図 4.1 に示すこの AMC は LAr Trigger Processing Mezzanine (LATOME Board) と呼ばれ、1 枚の LDPB に 4 枚搭載される。LATOME Board には 96 本のデータ送受信用の光ファイバーが接続され、またヒートシンク下には Intel 社製の FPGA Arria 10 GX (10AX115R4F40I4SGES)[28] が搭載される。

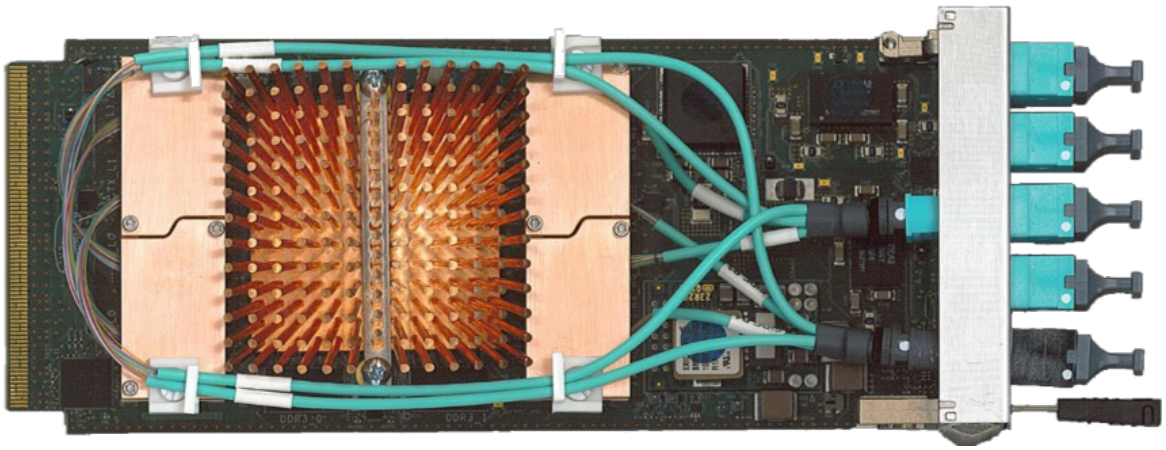


図 4.1: LATOME Board。サイズは横 156 mm 縦 73.5 mm である。ヒートシンク (写真中央の銅色) は剣山のような形であり表面積を大きくすることにより熱を逃し易くしている [23]。

4.2 LATOME ファームウェア

LATOME Board 内に搭載されている FPGA は、2020 年現在 Arria シリーズ中で最新であり、非常に高密度, 高速に通信を行うことができるのが特徴であり、浮動小数点数にも対応した乗算が可能である。その FPGA 内に実装される Firmware (LATOME Firmware) の開発, 検証を行ってきた。LTDB から

の ADC データは決まったレイテンシーで LATOME Firmware 内でエネルギーに変換され、さらにエネルギーの時間情報も再構成する。LTDB からの ADC データの並びは LTDB 内のハードウェアで決められている。また LATOME Firmware は FEX に 3 種類のエネルギークラスターを送信する。特に j FEX, g FEX は Trigger Tower の領域ごとにエネルギーを足し合わせた後送るという要請がある。そのため LATOME Firmware は Supercell の任意の並び替えを行えるようデザインされている。任意の並び替えを行うことが可能になることで Saturation した波形に対する処方も LATOME Firmware 内部に実装することが可能になる。詳しくは 5.3.2 章で説明する。その情報をもとに Level 1 trigger 発行が行われる。そのためトリガー発行に関し最も重要な Firmware である。また LATOME Board に搭載された素子との通信を通して FPGA で処理されたデータなどの送受信を行うことも可能であり非常に大規模な Firmware である。一つの LATOME で最大で 320 Supercell を一度に処理することが可能であり、各 LATOME Board は 200 Gbps で LTDB からの信号を受信し 450 Gbps で FEX に信号を送信する。以下 LATOME Firmware の構成要素を Module, Module 内部の各機能を Block と呼ぶ。各 Module と LATOME Firmware のレイテンシーは表 4.1 に纏めた。図 4.2 は LATOME Firmware の全体像を示す。以降は主にそれぞれの Module について解説する。

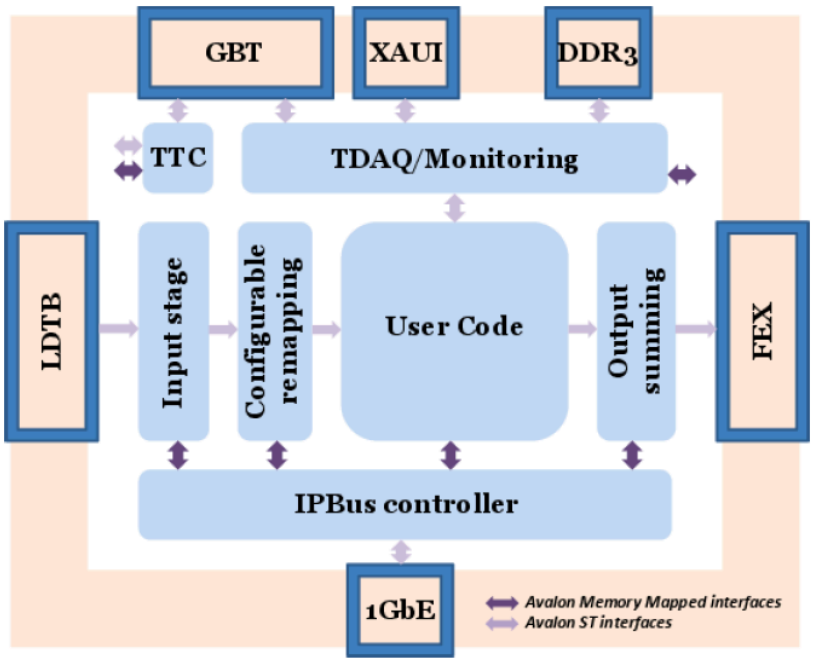


図 **4.2:** LATOME Firmware の全体像

4.2.1 Low Level Interface

Low Level Interface (LLI) の役割は図 4.3 に示すように様々である。

- Firmware 内で用いられるクロック、リセットの作成：FELIX からの TTC 信号は 160 MHz で同期されている。この 160 MHz のクロックを PLL を用いて Firmware 内部で用いる 320 MHz,

240 MHz, 280 MHz のクロックに分周する。PLL を用いることによる全てのクロックを 160 MHz に同期させて用いることができる。

- LTDB からの信号を μPOD を用いて受け取る：LTDB からの信号を 4 つの μPOD と 48 本の光ケーブルを用いて受信する。1 本あたり 5.12 Gbps, LATOME Board 全体で約 200 Gbps の受信速度を達成する。また LTDB からの ADC データをパラレル変換し、図 4.4 に示すよなフォーマットに変換する。LLI は ADC データを Istage に 48 stream × 16 bit のデータとして伝送する。
- LATOME Board で処理されたデータを μPOD を用いて FEX に送る：LATOME Firmware での処理結果を 4 つの μPOD と 48 本の光ケーブルを用いて伝送する。1 本あたり 11.2 Gbps, LATOME Board 全体で約 450 Gbps の受信速度を達成する。
- DDR3 などの LATOME Board 上の素子と Firmware のインターフェース：DDR3 は Micron[*1] 社製であり、データレートは最大で 3.2 Gbps である。
- ギガビットイーサネット (GbE), 10 ギガビットイーサネット (XAUI) を用いた Monitoring data の伝送：4.2.6 節で説明する IP bus, 4.2.7 節で説明する Monitoring block を用いて Firmware 内部の情報を読み出すインターフェース。
- MGT Link を用いた TDAQ への Monitoring data の転送：FELIX を介し、最大 9.6 Gbps で Monitoring data を TDAQ に伝送する。

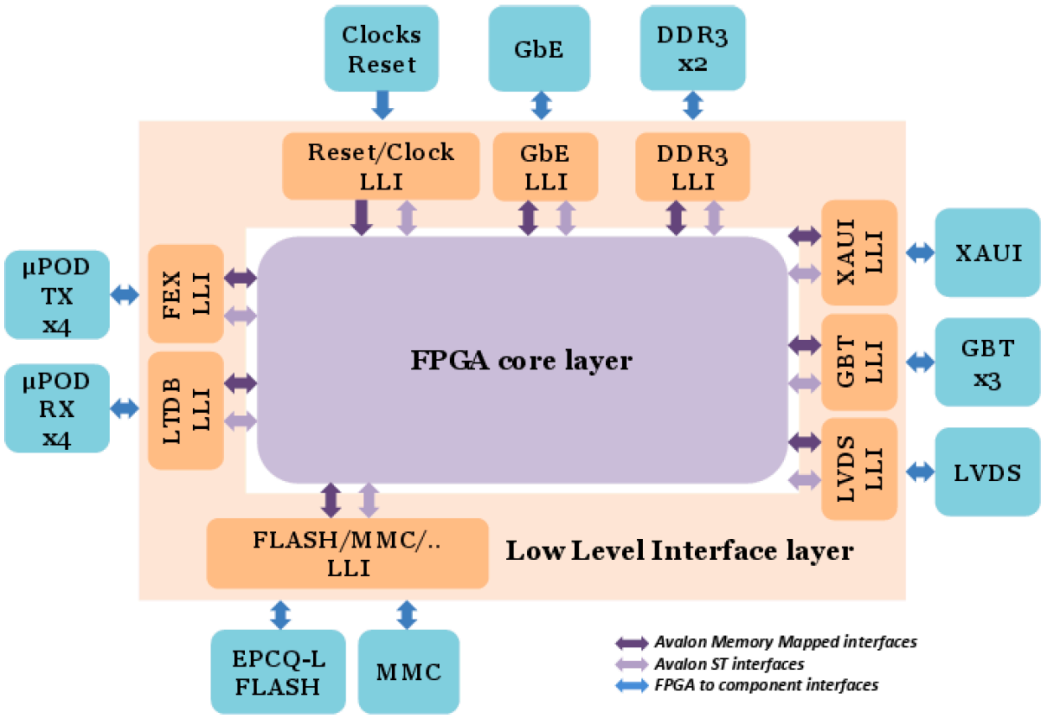


図 4.3: LLI の概要

[*1] https://www.micron.com/

4.2.2 Input Stage

検出器からの信号は LTDB から LLI を経由して FPGA core に入りまず初めに Input Stage (Istage) で処理される。LLI からの信号は 320 MHz に同期された 48 stream × 16 bit の信号として送られる。陽子衝突頻度が 40 MHz であるので 1 回の陽子衝突間に 8 (=320/40) Supercell 分のデータを扱うことができ 8 supercell × 16 bit = 128 bit を一つのフレームとして扱う (図 4.4)。128 bit の中にはデータの抜き出し方を示した 4 bit (T8-T11)、8 つの Supercell の ADC12 bit (D0-D11)、BCID の一部分 4 bit (T12-T15)、CRC [*2]用の 8 bit (T0-T7) と各 Supercell のうち使われていない 2 bit (D12, D13) で構成されている。

それぞれのデータフレームから抜き出されたデータ群は 48 stream × 12 bit の ADC、48 stream × 1 bit の valid signal、12 bit の BCID, 1 bit の start of packet (BCID の切り替わりのタイミングを示す) として Configurable Remap (Remap) に送られる (図 A.1)。

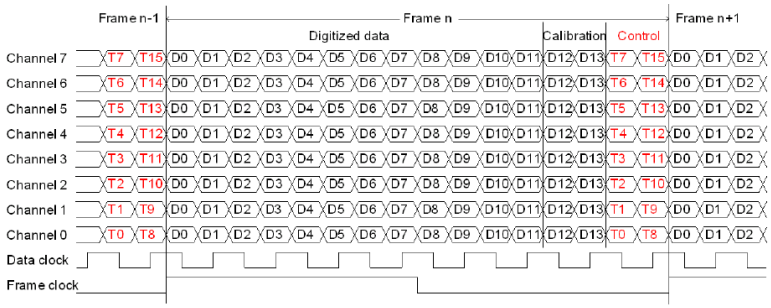


図 4.4: LLI からの Supercell 信号のフォーマット

4.2.3 Configurable Remap

Istage から送られたデータは 320 MHz で同期しているが、そのままでの周波数では後段の User Code が処理を行うことは難しい。そのため Configurable Remap (Remap) で User Code に適した動作周波数に落とす。元々 1 陽子衝突間に 1 stream で 8 Supercell を同時に扱っていたが周波数を落とすことにより、1 陽子衝突間に同時に扱える Supercell 数が減少し、stream 数が増加する。stream 数が増加することによりロジック数が増加する。そのため動作安定性とロジック数との兼ね合いから Remap では 240MHz に周波数を落とす。これに伴い 1 陽子衝突間に 1 stream あたり 6 Supercell までをシリアルに扱うことができ、stream 数が 62 に増加する (図 4.5)。そのため Supercell の並び替えを行う必要がある。並び替えを行うことにより Saturation した波形に対する同定精度を向上できることも見込まれる。Saturation した波形の取り扱いは 5.3.2 節と 5.4 節で説明する。並び替えは Supercell Mapping に従い LATOME Board に依存する。これは検出器のバレル領域, エンドキャップ領域, フォーワード領域で Supercell の配

[*2] データの有用性を確認するために行われる検査で、CRC の結果を valid として後段に送信する。

置の仕方が異なることなどに起因し、一つの LATOME Board で扱う Supercell の数 ($\leq$ 320) も異なる。

Remap は最終的に 62 stream × 12 bit の ADC、62 stream × 1 bit の valid signal, 12 bit の BCID, 1 bit の start of packet を 240 MHz のクロックに同期させて User Code に送る (図 A.2)。

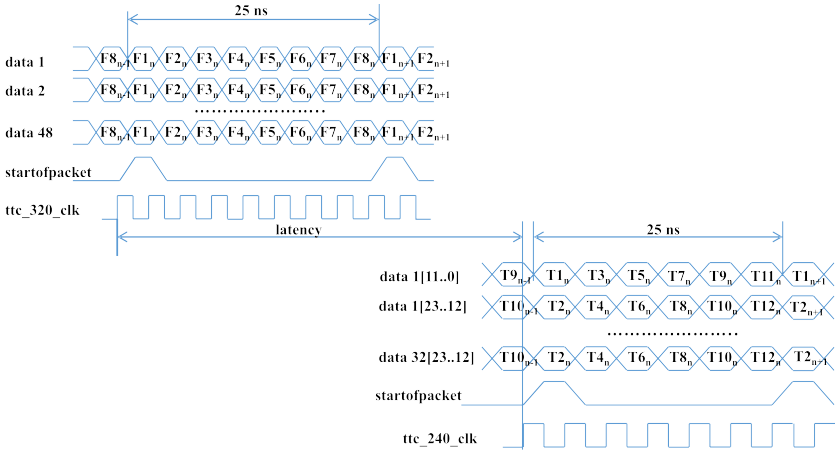


図 4.5: Remap における周波数の減少

4.2.4 User Code

LATOME Firmware の一番大きな目的は、検出器で落としたエネルギーを正確に見積もり、その時間情報を取り出すことにある。User Code はそれを実現している Module である。User Code は Remap から送られた ADC をもとに DSP block を用いてエネルギーを再構成する。この際 7 章で説明する LHC の陽子占有構造由来の影響 (Baseline Shift) を取り除く。また同時に Saturation した波形の同定も行なっている。大きなエネルギーを落とした場合、検出器からの信号は特徴的な波形になりその特徴を用いて同定を行う。User Code は 2 つの Filtering Algorithm を持ち、1 つは 3.1 章で説明したノイズを除去する機構、もう一つは 5.3 章で説明するエネルギーの時間情報を抜き出す機構である。

User Code は最終的に 62 stream × 18 bit の Energy、62 stream × 1 bit の valid、62 stream × 4 bit の quality、12 bit の BCID、1 bit の start of packet を 240 MHz に同期させて Output Summing (Osum) に送る (図 A.3)。このパートは私が設計, 実装, 検証を行った領域であるため次の節で詳しく述べる。

4.2.5 Output Summing

Output Summing (Osum) の主な目的は User Code で計算されたエネルギーを検出器の η - ϕ ごとの領域で足し合わせることである。Osum は FEX に送る 3 種類のデータを作成している。特に jFEX, gFEX は図 4.6 に示すように、それぞれ Trigger Tower, 2 Trigger Tower × 2 Trigger Tower の領域内のエネルギーを足し上げたものに対応する。それら 3 種類のデータを後段に送るためにヘッダーとフッターを付け加え、最終的に 48 stream × 32 bit の Energy、48 stream × 1 bit の valid signal を 280 MHz のクロックに同期させ LLI を経由させて FEX に伝送する (図 A.4)。

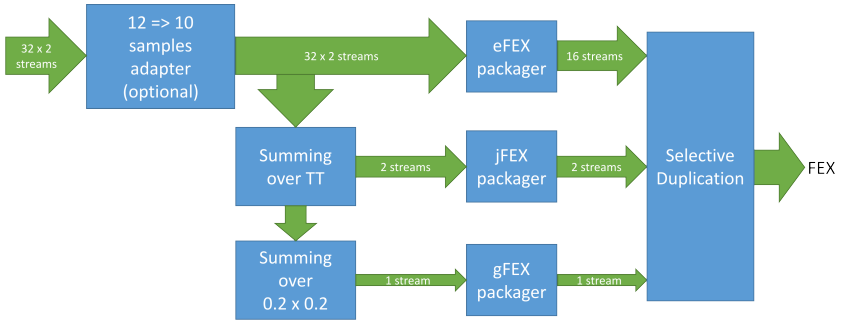


図 4.6: Osum でのデータの流れ。矢印の太さは大まかな stream の数を表しており各 FEX 毎に stream 数が異なる。

ブロック	理想値 [BC]	実際の値 [BC]
LLI	4.0	4.0
Istage	3.0	2.63
Remap	1.5	1.75
User Code	5.0	4.5
Osum	1.5	2.5
LATOME	15.0	15.38

表 4.1: LATOME Firmware のレイテンシーの実測値

4.2.6 IP bus controller

LATOME Firmware の内部のレジスター、メモリには、それぞれに固有のアドレスを有するものもあり、それらは IP bus controller (IP bus) で 1 Gbps の速さで読み出すことができる。IP bus を用いることにより、Firmware 内の設定を書き換えることが可能になり、コンパイルし直さずとも動作を変化させることが可能になる (Configurable Firmware)。User Code 内の Filtering Algorithm との関係は密接で、各 Supercell の係数, Pedestal の書き換えや Baseline Correction を計算に入れるか否かなどを設定することができる。また User Code 内には LATOME Firmware のメインパスとは関係のない Block がいくつか含まれており、それらの Block に直接アクセスすることにより LATOME Firmware の動作をモニタリングすることもできる。IP bus のアドレスは 32 bit でありそれぞれの stream、Module, Block ごとにアドレスが振られている。IP bus は GitHub を用いて配布されており IPbus Firmware[*3] と IPbus Software[*4] に分けて配布されている。IPbus Firmware は実際にハードウェアに実装する Module を提供し、ハードウェアにアクセスするためのライブラリ群が IPbus Software として配布されている。

[*3] https://github.com/ipbus/ipbus-firmware
[*4] https://github.com/ipbus/ipbus-software

4.2.7 TDAQ Readout, Monitoring block

LATOME Board は 3 つのデータ転送先がある。一つは FEX に 3 種類のエネルギー情報を伝送し、Level 1 Accept 発行に関わる。他 2 つは Level 1 Accept に応じて伝送されるものであり、1 つは TDAQ システムに送り、最後の一つが Monitoring 用の信号である。TDAQ システムへは GBT Link を用いて FELIX を通り最大 16 bit のエネルギーと ADC を 6.4 Gbps で伝送する。Monitoring data は'Raw ADC', 'ADC – Pedestal', 'Transverse Energy', 'Transverse Energy ID' の 4 種類から選択することができ、Multiplexer を用いてそれらの内いずれかを XAUI を用いて各 ATCA Shelf で最大 10 Gbps で専用の PC に伝送される。Monitoring data は User Datagram Protocol (UDP) による通信で FPGA からのデータを専用の PC に高速で保存し、いち早くデータ解析を行うことができる。Monitoring data については 5.8 章で詳しく説明する。

4.2.8 Pattern generator

図 4.2 には含まれないが Pattern generator は LATOME Firmware の検証を効率的に行う目的で開発された。ユーザーが特定のデータセットを効率的に LATOME Firmware に伝送することが可能である。Pattern generator には 2 種類あり LATOME Firmware 内に組み込むもの (Internal Pattern generator) と外部からデータを伝送するもの (External Pattern generator) がある。

第 5 章

User Code

LATOME Firmware は 10 の研究機関が共同で開発を行っており、東大グループは User Code を担当している。この章では User Code の各 Block の機能、実装方法などについて説明する。私はこの Firmware の作成を主導的に行った。図 5.1 に示す User Code は LATOME Firmware 内で最も重要な Module であり、ADC データからエネルギーとその時間情報を以下の手順で算出する。

1. Baseline Correction : 液体アルゴン特有の現象である Baseline Shift の影響を ADC データから取り除く (5.1 節)。
2. FIR Filter, Saturation Detection : 3.1 節で示したノイズを除去する機構を用いて、E_{T} candidate, $E_{\mathrm{T}}\tau$ candidate を算出する (5.2 節)
3. Selection Block : 算出された E_{T} candidate, $E_{\mathrm{T}}\tau$ candidate に 2 つの Criteria を課し E_{T} とその時間情報を抜き出す (5.3 節)。
4. Combine Block : Saturation した波形に対しての処方を持つ (5.4 節)。

また User Code はモニタリングに有用な情報を格納する機能もあり、IP bus 経由で直接データを読み出すことができる。それらは図 5.1 中の黄色の Block であり、5.6 節と 5.7 節で説明する。各 Block の計算時間は表 5.5 に纏めた。また各 Block の入出力信号の役割、詳細とリソース使用量については付録 B に纏めた。User Code は Configurable Firmware であり設定を変化させることにより再度コンパイルを行わなくても動作を変化させることができる。それらは表 5.1 にある global control signals と呼ばれ User Code Status として管理されている。

1 枚の LATOME Board で処理する Supercell は多くて 320 個であるが Remap からの信号を User Code は 62 stream × 6 Supercell で受け取る。そのため Firmware 上では常に 372 Supercell の情報として扱う。User Code はどの stream のどの Supercell にデータが含まれているかを認識せずに 62 stream パラレルに 6 Supercell ずつ処理を行う。6 つの ADC データはシリアルに並べられ図 5.2 のように 6 clock おきに同一の Supercell のデータを処理する機構である。

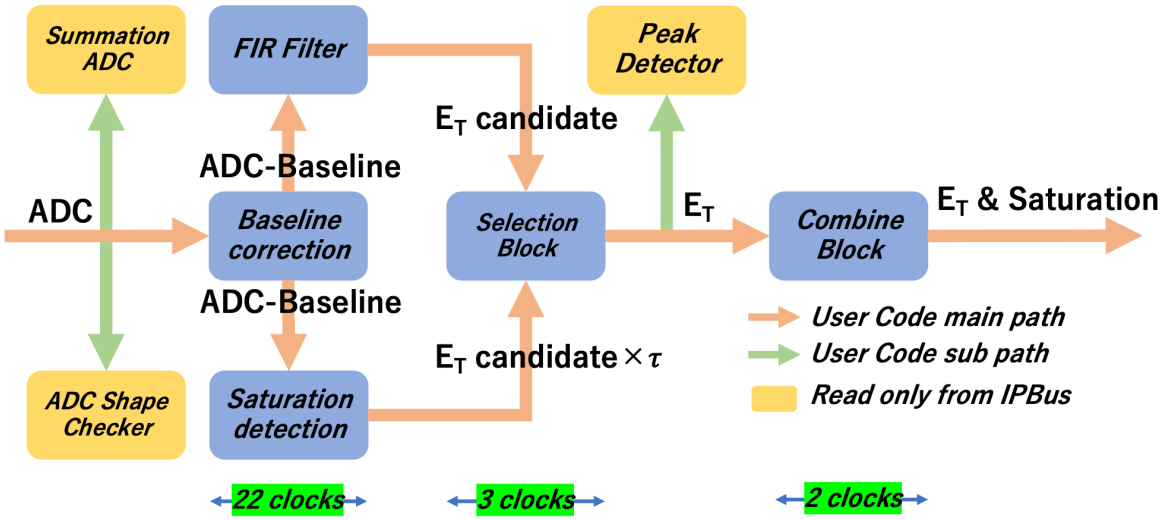


図 5.1: User Code の全体像。青い Block が User Code Main path, 黄色の Block が User Code Sub path にあたる。User Code sub path は後で説明する IP bus からのみデータを読み出すことができる。

global control signals	目的
hardpoint	FIR Filter, Saturation Detection の計算結果に整合性を持たせるか
selection	Tau Criteria を用いるか
saturation	Saturation Criteria を用いるか
combine	Combine Block を Filtering Algorithm に含めるか
baseline	Baseline Correction を Filtering Algorithm に含めるか

表 5.1: global control signals まとめ。これらの信号も IP bus を用いて設定可能である。これらは User Code 内部にレジスターとして情報が保存されそのレジスターの持つ値を確認することで User Code 内部の機能を変化させている。

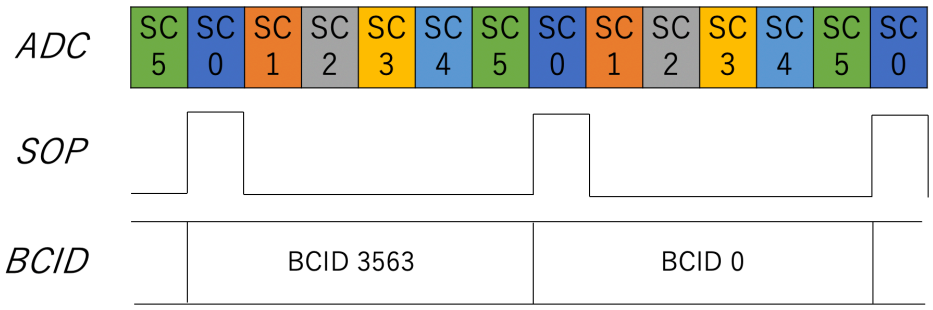


図 5.2: User Code に入る Remap からの信号。ADC データは 4.17 ns おきに、BCID は 25 ns おきに更新され SOP は BCID のはじめを表す。

5.1 Baseline Correction Block

LHC は全部で 3564 個のバンチで構成されており、それぞれのバンチには陽子を詰めることができ陽子衝突を起こし物理事象を記録することができる。全てのバンチに陽子が詰められている場合、ノイズ、パイルアップなどが均等に起こり 3.1 節で説明した Bipolar 波形の時間積分 0 の性質から、それらの影響を無視できる。その場合検出器からの信号は入射粒子が信号を残す領域以外は常に 0 であるはずである。しかし実際にはすべてのバンチに陽子が詰められているわけではなく、陽子は約 2700 バンチのみに詰められており、この特徴的な陽子占有構造をトレイン構造と呼ぶ (図 7.7)。トレイン構造によりノイズ、パイルアップの重なり合いが不十分な部分がでてきてしまい図 5.3 のようにベースラインがずれる。入射粒子の残した信号は、このずれたベースラインの上に乗り観測される。そのため正確なエネルギー計算ができなくなる。その影響をすべてのバンチ、Supercell に対し補正する Block が Baseline Correction である。陽子占有構造による検出器からのデータのズレは Baseline Shift と呼ばる。Baseline Correction は各バンチ、Supercell に対し式 5.1 のように 2^{10} 個の ADC データの平均を計算する。そうすることにより各バンチ、Supercell で Baseline Shift の特徴を捉えることができる。

$$Baseline_j^k = \frac{\sum_{i=0}^{2^{10}-1} ADC_j^k}{2^N} - Pedestal_j \text{[*1]} \tag{5.1}$$

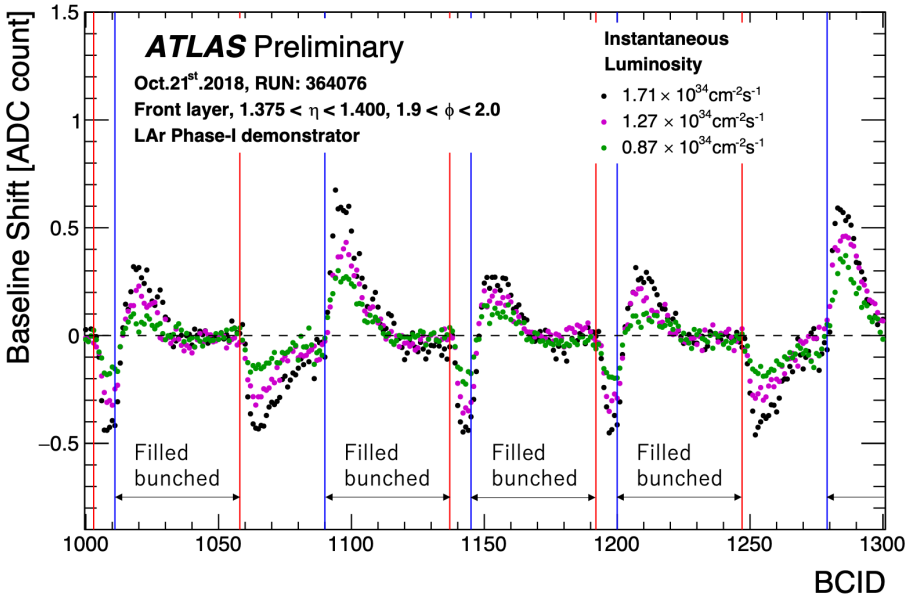


図 5.3: 各瞬間ルミノシティーに対する実際に RUN2 実験で得られた Baseline Shift の振る舞い。青、赤線はトレイン構造の始めと終わりを表しており、それらの間の黒矢印が陽子の詰められてバンチに対応する。詳しくは 7 章で説明する。

[*1] j:Supercell ID, k:Bunch Crossing ID, i:合計数

Baseline Shift は各バンチ、Supercell 毎に算出される値であり、さらに図 5.3 のように瞬間ルミノシティーの変化に応じてもその値が変化する。そのため値を常に更新し続ける必要がある。Baseline Shift は 2^{10} 個の ADC データの平均であるため計算途中の値を記録しておく必要があり、また値の更新を行うまでは以前計算した値を保持する必要がるので、計算用 RAM と保存用 RAM の 2 つの大きな RAM が必要になる。そのため最も単純に Baseline Correction を実装すると図 5.4 のようになる。

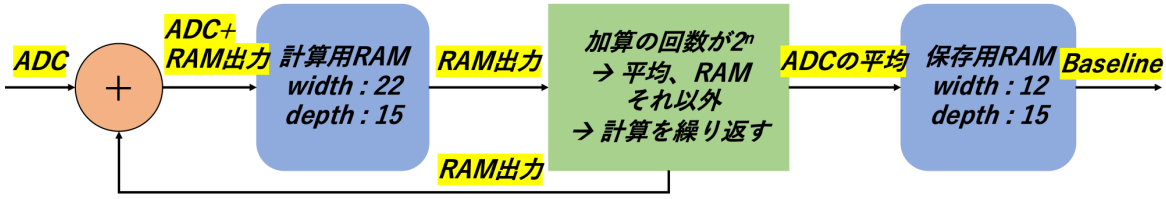


図 5.4: 最も単純な Baseline Correction の実装。ADC データを 2^{10} 回足し上げることから 1 つ目の RAM には 12+10 bit の width を用意する必要がある。1 度に全てのバンチ (12 bit)、Supercell (3 bit) に対し Baseline Shift を算出するので 15 bit のアドレスを用意する必要がある。

図 5.4 を実装した場合のそれぞれの RAM で使う 1 stream あたりのリソース量を表 5.2 に纏めた。

	Width	Depth	memory bit	M20K の個数	計算時間
計算用 RAM	22 (=12+10)	15	720896	44	9.12×10^{-2} 秒
保存用 RAM	12	15	393216	24	-

表 5.2: 最も単純に Baseline Correction を実装した場合のリソース量

これを 62 stream 全てに実装すると必要な M20K 数が 4216 個となり Arria 10 の搭載数 (2713 個) を大幅に超える。そのため M20K の使用数を大幅に減少させる機構を考案した。

Baseline Shift を更新しなければならない理由は瞬間ルミノシティーの変化に対応するためであった。瞬間ルミノシティーの変化は 1 つの Physics RUN 中で図 7.3 のように緩やかに変化するので、短い時間幅ではほとんど一定と考えられる。そのため Baseline Shift は 15 秒毎に全てを更新すれば良いという要請があった。図 5.4 での機構を実装した場合の Baseline Shift の更新時間は

$$\Delta T = 3564 \times 25 \text{ ns} \times 2^{10} \simeq 9.12 \times 10^{-2} \text{ sec} \tag{5.2}$$

となり十分すぎる性能である。計算用 RAM の depth は 1 度に計算する BCID と Supercell の数に依存するのでどちらかを減らすことにより必要な M20K の個数を減らすことができる。そのため 1 度に 2^5 個の BCID のみの Baseline Shift を算出することにした。これにより計算用 RAM で必要な memory bit 数を表 5.2 に比べ $1/2^7$ に削減した。計算時間については

$$\Delta T = \text{ceil}\left(\frac{3564}{2^5}\right) \times 3564 \times 25 \text{ ns} \times 2^{10} \simeq 10.3 \text{ sec} \tag{5.3}$$

となり、要求値を満たす。次に保存用 RAM については、Baseline Shift はノイズ、パイルアップの影響によるものであるのでその影響は ADC データの大きさにくらべ十分に小さいと考えられる。ADC デー

タは 12 bit の信号として扱われるので 0 ～ 4095 までの範囲を取りうるが、Baseline Shift は 32 ADC count までで表現できることを RUN2 実データを用いた解析を行い確認し、保存用 RAM の width を減らし実装した。以上の機構を実装後、各 RAM で使うリソース量を表 5.3 に纏めた。

	Width	Depth	memory bit	M20K の個数	計算時間
計算用 RAM	22 (=12+10)	8	5632	1	10.3 秒
保存用 RAM	9	15	294912	18	

表 5.3: 実際の Baseline Correction 実装に用いられるリソース

ADC データは Pedestal が付加されたものなのでその影響を差し引くために図 5.5 のように ADC データを 2^{10} 回足した後、10 bit shift させた Pedestal を差し引いている。Pedestal は小数点以下 3 bit を持つので Baseline Shift もその精度を落とさないように小数点以下 3 bit 用意した。また図 5.3 より Baseline Shift は負になる可能性があるので符号 bit も用意した。その結果、保存用 RAM の width を 9 bit とした。実際には計算用 RAM には MLAB を用いており、Baseline Correction は全体で M20K を 1116 個、MLAB を 992 個を使って実装されている。

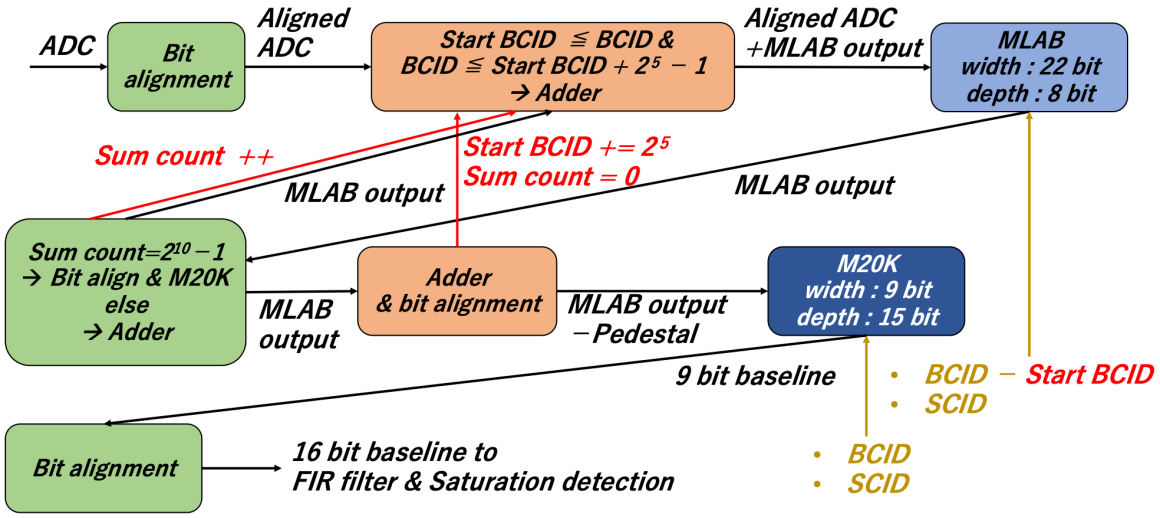


図 5.5: Baseline Correction の設計図。ADC データの流れを黒、内部の制御信号を赤、Remap からの信号を黄色で表示している。2^{10} 回加算を行うので、bit alignment を行い 12 bit の ADC データを 22 bit にする。1 度に計算する BCID の範囲は 2^5 個であり、6 Supercell を 1 stream 内で同時に扱うのでアドレスは 8 bit となる。2^{10} 回の加算後、10 bit shift された Pedestal を差し引き、M20K に 9 bit として保存する。この時 Baseline Shift は 32 ADC count 未満で表現できると仮定している。詳細は 7 章で与える。FIR Filter, Saturation Detection へは 16 bit として送るので、M20K から読み出した値を再び bit align する。一度に計算するバンチ数が 2^5 BCID であり、全部で 3564 BCID であるので。この機構を 112 回繰り返すことにより全ての Baseline Shift を計算する。1 stream あたり MLAB, M20K はそれぞれ 5632 memory bit, 294912 memory bit 使用する。

保存用 RAM からは 240 MHz で常に Baseline Shift が読み出され、FIR Filter, Saturation Detection

で用いられる。これは ADC データからエネルギーを再構成するために用いられるので非常に重要なパスである。RAM は書き込みと読み出しが同時に起こると予期せぬ値を出力する。保存用 RAM は読み出しと書き込みを Remap からの信号に基づいて行うため同時に起こることが予想される。そのため読み出しと書き込みのタイミングを 1 バンチ分 (6 clock) ずらしてそれを回避した (図 5.6)。

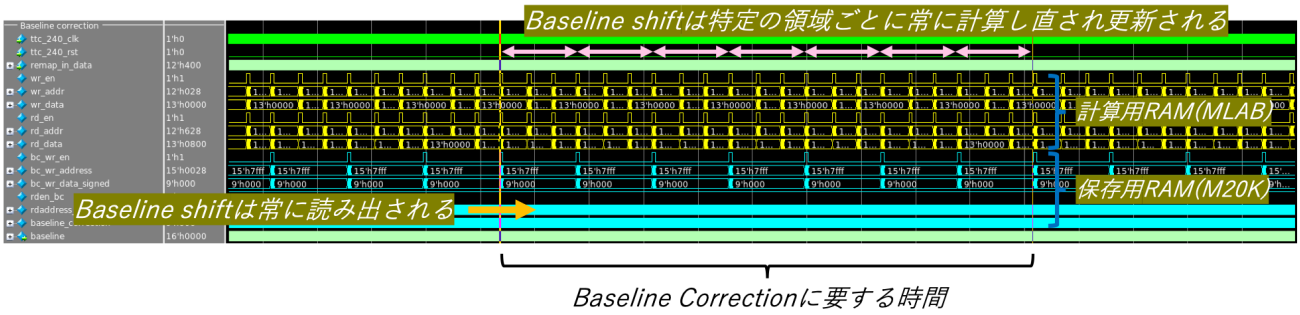


図 5.6: 計算回数を 2^1 回、同時に計算する BCID の範囲を 2^9 回と設定した場合の Baseline Correction の wave window。計算用 RAM (黄) は特定の範囲の BCID の時のみアクセスされる。保存用 RAM (青) は FIR Filter, Saturation Detection で用いられるので 240 MHz clock に同期して値が読み出され続ける。Baseline Shift の計算が完了後保存用 RAM に書き込まれるタイミングと保存用 RAM から値が読み出されるタイミングが一致しないようにデザインされている。

5.2 FIR Filter Block, Saturation Detection Block

3.1 節で示したように RUN3 からも ADC データ 4 点用いたエネルギー (E_{T}) とエネルギーと位相の積 ($E_{\mathrm{T}}\tau$) の再構成が行われる。この際 Pedestal と Baseline Shift の影響を取り除いた ADC データを用いてそれぞれ式 (5.4)、式 (5.5) のように再構成する。

$$E_{tj}^{i}\ \text{candidate} = \sum_{k=0}^{3} a_k \left(ADC_j^{i+k} - Pedestal_j - Baseline_j^{i+k} \right) \text{*2} \tag{5.4}$$

$$E_t\tau_j^{i}\ \text{candidate} = \sum_{k=0}^{3} b_k \left(ADC_j^{i+k} - Pedestal_j - Baseline_j^{i+k} \right) \tag{5.5}$$

このとき global control baseline を用いて、図 5.7 のように、Baseline Shift の影響までも考慮して計算するかを変更することができる。係数 a_i と b_i は Bipolar 波形の始めの 4 点を用いた際に正確に E_{T}, $E_{\mathrm{T}}\tau$ を算出するようデザインされているが、FIR Filter, Saturation Detection にはどの 4 点が正しいものかの情報は与えられず図 5.8 のように常に 4 つの ADC データと係数の積を出力している。そのため FIR Filter の出力はエネルギーの候補 (E_{T} candidate), Saturation Detection の出力はエネルギーと位相の積の候補 ($E_{\mathrm{T}}\tau$ candidate) であり、その中のどれかが正しいものである。ADC データは図 5.2 で示したように 6 Supercell が連続して並べられており、それらを効率良く低レイテンシーで処理することができる機構にデザインされている。

*2 i:BCID, j:Supercell ID, k:ADC データのサンプリング番号

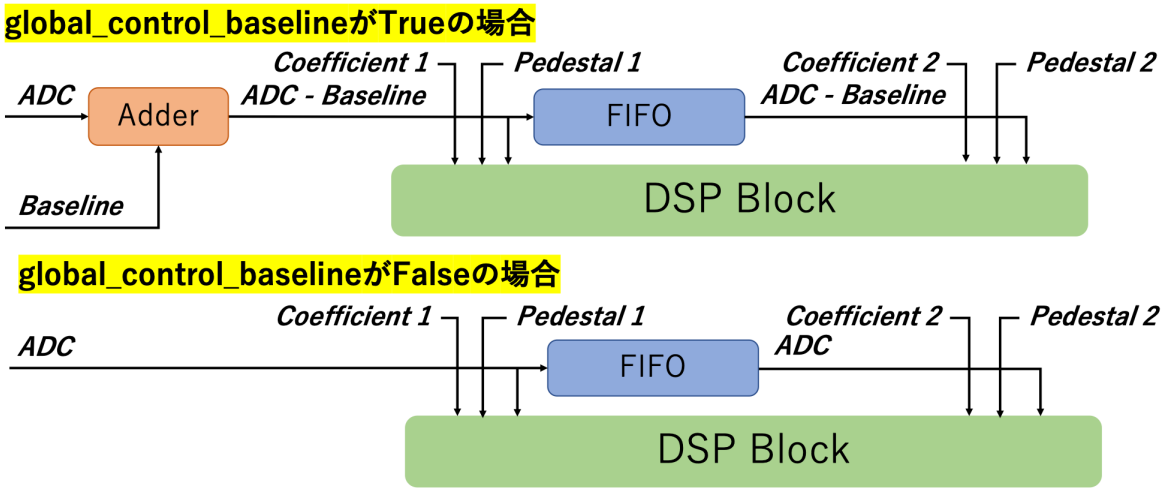


図 5.7: FIR Filter, Saturation Detection における Baseline Shift の取り扱い。global control baseline が True の場合は ADC データから Baseline Shift の影響が取り除かれたものが DSP Block の入力となる。ADC データは 12 bit であるのに対し Baseline Shift は符号付、小数点以下 3 bit を持つので $ADC - Baseline\ Shift$ は 16 bit として扱われる。False の場合は ADC データが 16 bit にアラインされた後 DSP Block の入力となる。DSP Block の入力の bit 数は Firmware のコンパイル時に固定されるので True/False の双方の場合で 16 bit にすることにより Baseline Shift の取り扱いを Configurable にしている。

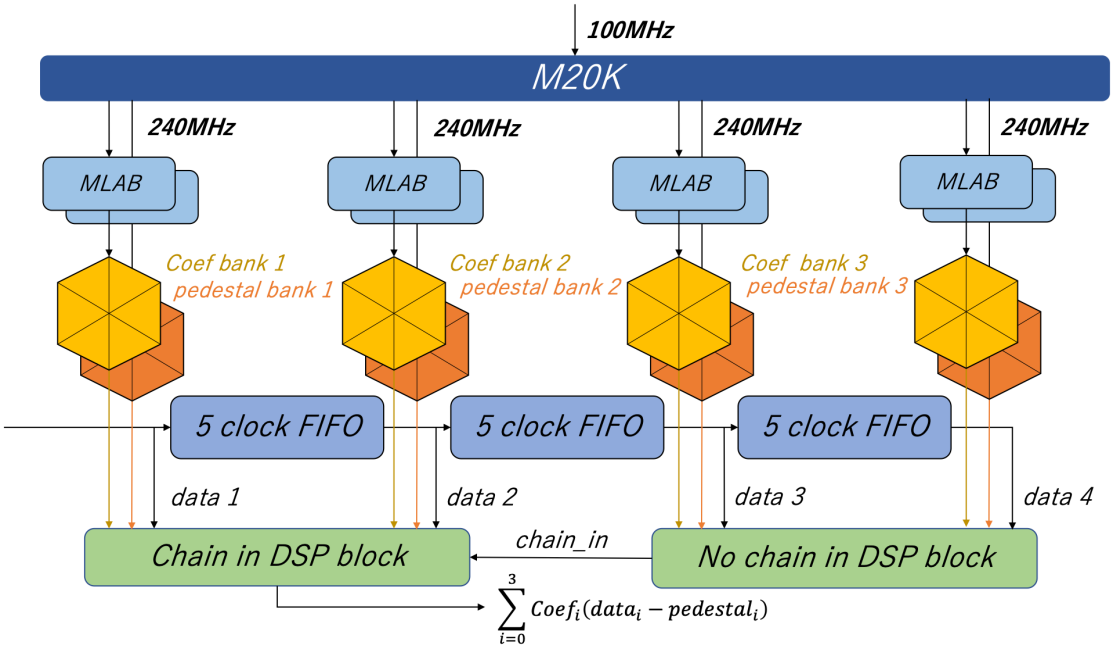
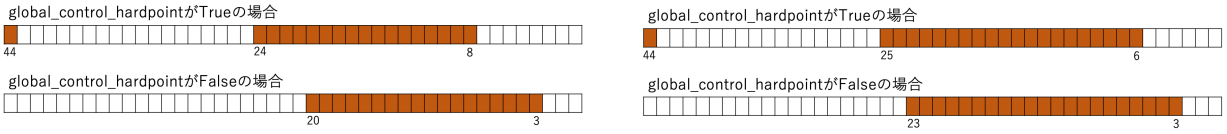


図 5.8: DSP Block を用いた 4 点 FIR 計算。FIR 計算に用いる係数、Pedestal の Circular buffer(黄, オレンジ)は 240 MHz に同期して回転し入力の ADC データに対応する係数・Pedestal が常に用いられる。1 つの 18 bit × 19 bit 乗算用 DSP block は図 3.11 と図 3.12 に示すように 2 つのデータ入力口を持つ。しかしそれぞれの乗算経路は 1 clock のレイテンシーの差がある。ADC データは 6 Supercell がシリアルに並べられているので 5 clock FIFO を用いることにより、ぞれぞれの乗算経路のタイミングを合わせる。これにより FIR Filter, Saturation Detection で費やされるレイテンシーを最小にする。

乗算に用いる係数と Pedestal は Circular buffer を用いており常に ADC データの Supercell に対応したものが使われる。さらにそれらは IP bus を用いて値の変更が可能である。DSP Block の出力は 44 bit だが E_{T} candidate は 18 bit, $E_{\mathrm{T}}\tau$ candidate は 21 bit で後段に送るので bit の丸めを行う必要がある。bit の丸めは global control hardpoint に依存し、それが True のときは図 5.9 のように LSB が 12.5 MeV に対応するように丸られ $|E_{\mathrm{T}}| < 1.6$ TeV 程度の範囲までをカバーする。False の場合は符号を考慮しない bit の丸めを行う。これは元々 DSP Block の動作を確認する目的で用いられており、丸めの 3 bit は係数の小数点以下 bit 数に対応する。



(a) FIR Filter **(b)** Saturation Detection

図 5.9: FIR Filter, Saturation Detection における bit の丸め。DSP output は符号付の 44 bit であり global control hardpoint が False の場合は符号を考えない bit の丸めになる。True の場合は LSB が 12.5 MeV に対応するよう bit の丸めを行う。

乗算は 3.3.5 節で記述した DSP Block を用いて行う。FIR Filter, Saturation Detection のレイテンシーは 22 clock (1 clock = 4.17 ns) でありそのうちの 18 clock は 4 つの ADC データをサンプルするのに費やされ、残りの 4 clock は DSP Block による乗算と DSP Block の出力のビットの丸めのために使われる (図 5.10)。

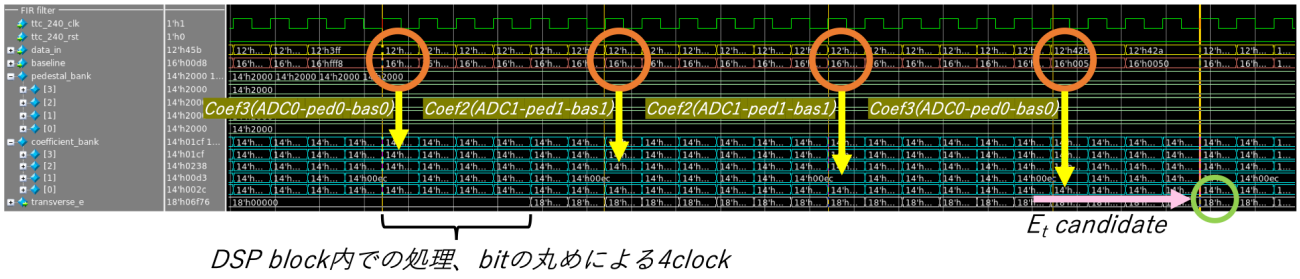


図 5.10: FIR Filter の Wave window。各オレンジの丸内の 3 つの signal が ADC データと Pedestal, Baseline Shift の値に対応し、矢印で示された係数と掛け合わされる。4 点それぞれの積が最終的に合計されて、bit の丸めにより E_{T} candidate となる。1 つ目のカーソルから最後のカーソルが 22 clock に対応し、その間で ADC データが E_{T} candidate に変換される。Saturation Detection も同様の機構で $E_{\mathrm{T}}\tau$ candidate を算出する。

5.3 Selection Block

FIR Filter, Saturation Detection で計算された E_{T} candidate, $E_{\mathrm{T}}\tau$ candidate から正しい E_{T} を算出するために Selection Block は 2 つの Criteria (Tau Criteria, Saturation Criteria) を用いる。図 5.15

で示すように Selection Block のレイテンシーは 3 clock である。

5.3.1 Tau Criteria

LHC のバンチ間隔は 25 ns であったのでいかなる Bipolar 波形を用いた場合でも正しい 4 点を用いて計算された τ は必ず-12.5 ns $\sim$ 12.5 ns の範囲に収まる。また τ を増加させる要因としてパイルアップなどのノイズによる影響が考えられる。これはエネルギーが大きい場合はほとんど無視することができ、小さいエネルギーを再構成する際に重要になる。つまり小さいエネルギーでは大きいエネルギーよりも条件を緩くすると良いことがわかる。しかしより効率的に正しい E_{T} のみを取得するにはより狭い範囲の τ に対応する E_{T} のみを考えるのが良い。そのため Simulation により入力のエネルギーに対する τ の広がりが調査された (図 5.11)。

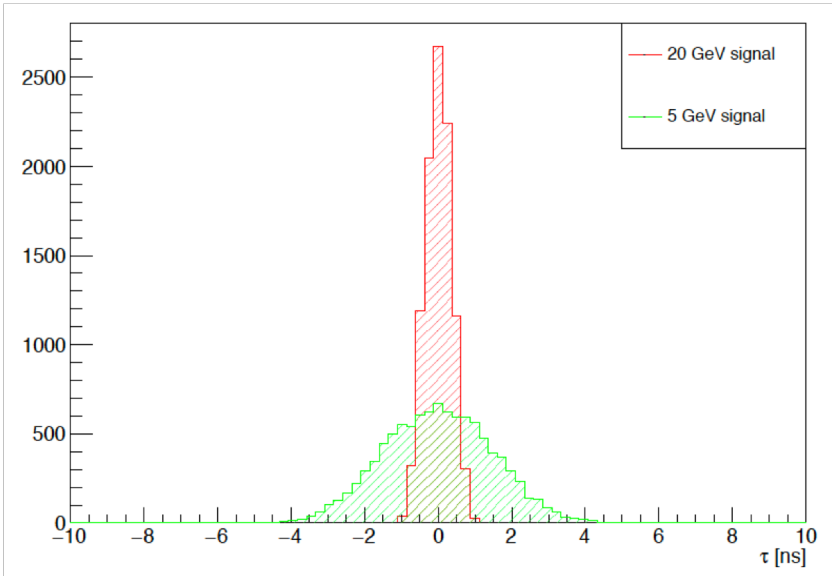


図 5.11: 5 GeV(緑), 20 GeV(赤) のエネルギーを入射した場合の τ の分散。平均相互作用数 80 程度に対応する状況で得られた結果。エネルギーが大きくなるにつれ τ の分散は小さくなる傾向がある。

Simulation の結果から正しい 4 点を用いて計算された E_{T} はバンチ間隔よりも小さい τ を持つ。Firmware で実装する場合 τ を直接取り出すには E_{T} と $E_{\mathrm{T}}\tau$ の割り算を行う必要があるが Firmware 上での割り算はレイテンシーが大幅に増加しかつリソースを余分に使用する[*3]。そのため bit shift による除算を Selection Block で採用した。bit shift による除算は 2 の冪乗による除算に対応する $|\tau| < 2, 4, 8, 16, \cdots$ のいずれかを用いることになり Tau Criteria は Firmware 内で以下のようにデザイ

[*3] Intel 社が提供するコンパイラである Quartus を用いた Firmware の合成の際、除算には lpm_divide と呼ばれる Mega-function が自動で生成される、1 つの lpm_divide は 892 ロジックを消費する。lpm_divide のレイテンシーは乗算の bit に依存する。レイテンシーは除算結果の余りを残すか小数点以下まで計算するかでも変わり、小数点以下まで計算する場合余りを残す場合に比べおおよそ 2 倍のレイテンシーが必要になる。

ンされている。

$$\begin{cases} |E_{\mathrm{T}}\tau| < 2E_{\mathrm{T}} : 10\ \mathrm{GeV} \leq E_{\mathrm{T}} \\ |E_{\mathrm{T}}\tau| < 8E_{\mathrm{T}} : -1\ \mathrm{GeV} \leq E_{\mathrm{T}} < 10\ \mathrm{GeV} \end{cases} \tag{5.6}$$

この条件を E_{T} candidate と $E_{\mathrm{T}}\tau$ candidate に課し要件を満たしたものが FIR Filter 計算により算出された実際に粒子が検出器に落とした E_{T} に対応する。Tau Criteria を用いることで図 5.12 のように Bipolar 波形から 1 点のエネルギーを抜き出すことができる。

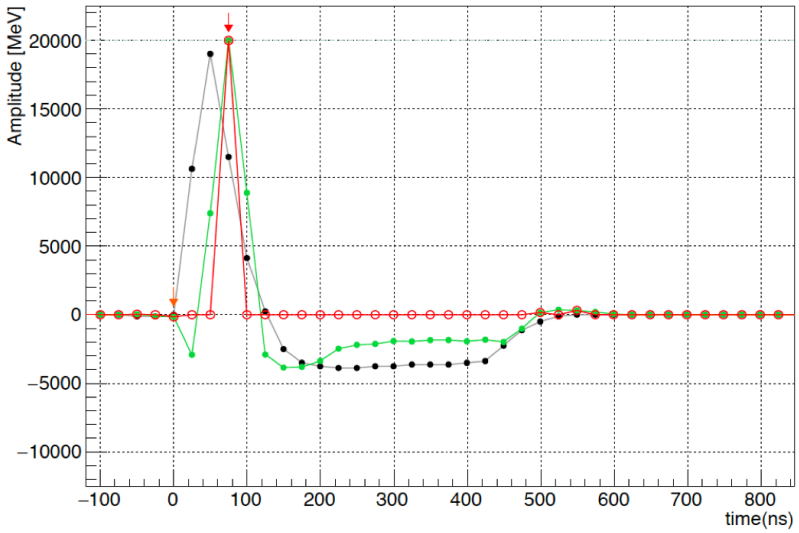
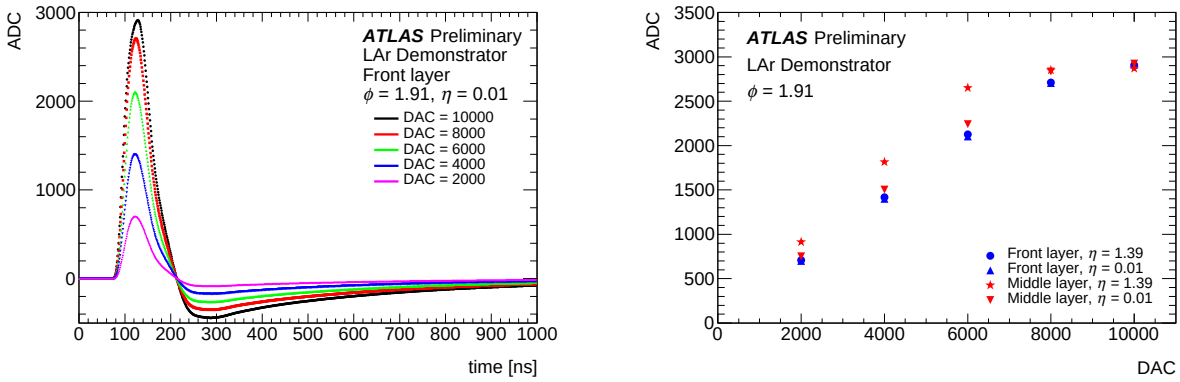


図 5.12: τ criteria を課した後の Bipolar 波形。黒が ADC データ, 緑が E_{T} candidate, 赤が τ criteria を抜けた E_{T} に対応する。FIR Filter 計算により入力の ADC データから正しい 1 点のみを抽出できている。

5.3.2 Saturation Criteria

Selection Block は Saturation した波形に対する処方も持つ。検出器からの信号はアンプで増幅され、最大で 3.3 V の信号になり、その後 LTDB でデジタイズされる。LTDB では 1 mV を 1 ADC count としデジタイズするためデジタイズによる Saturation は起こらない。一方検出器からの信号がアンプの最大入力電圧以上になると正常に増幅されず Saturation を起こす。Saturation した波形では Bipolar 波形波形が図 5.13a のように歪む。FIR Filter, Saturation Detection で用いた係数 a_i, b_i は理想的な波形から算出したものであり、計算には Bipolar 波形を仮定している。そのため大きなエネルギーが Supercell に落ちた場合 E_{T}, $E_{\mathrm{T}}\tau$ が正しく再構成されない。



(a) 各 DAC に対応する ADC データ　　**(b)** DAC に対する ADC データの線形性

図 5.13: DAC はエネルギーに対応する値であり、それが大きくなるにつれて ADC データも徐々に大きくなるべきだが Saturation が起こり波形が歪む [24]。

しかし Saturation を起こすほどの高エネルギーの信号は新粒子起源の可能性もあるため非常に重要な信号である。そのため全く異なる方法を用いて Saturation した事象を選別する。Saturation した波形は E_{T}, $E_{\mathrm{T}}\tau$ が正確に算出されないが図 5.14 のように特徴的な振る舞いをする。

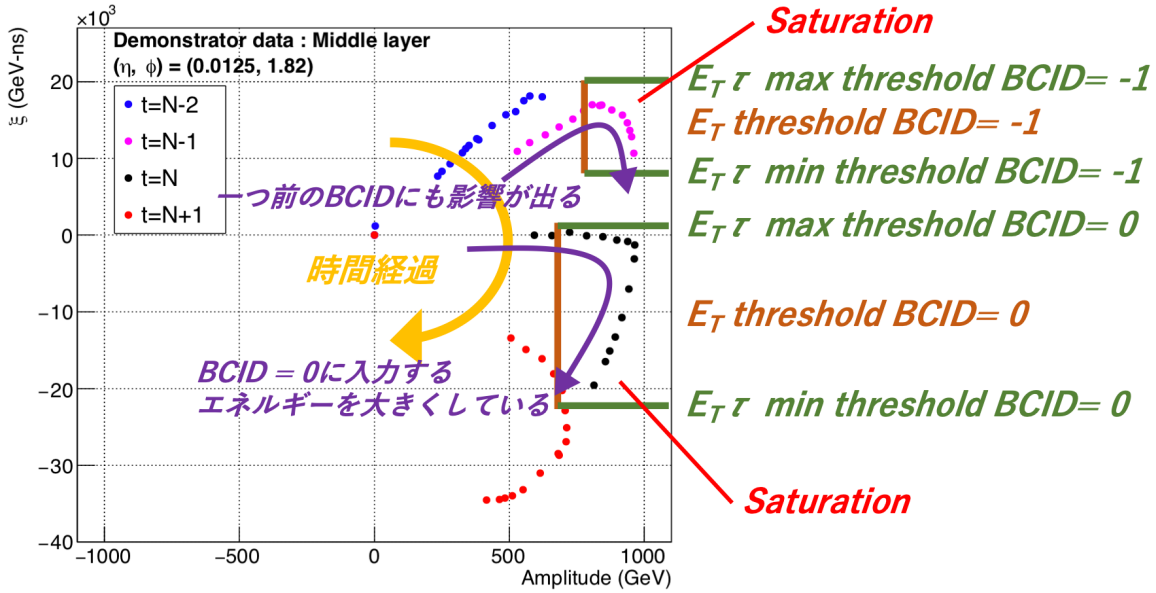


図 5.14: E_{T} と $E_{\mathrm{T}}\tau$ の相関図。ξ は $E_{\mathrm{T}}\tau$ に対応する。Saturation が起こった BCID とその一つ前の BCID を確認することで Saturation pulse の同定を行う。一つ後の BCID にも Saturation の影響が現れるが、一つ後の BCID を確認するには 6 clock のレイテンシーが追加で必要であるため実装されていない。それぞれの E_{T}, $E_{\mathrm{T}}\tau$ の閾値は Supercell 毎に IP bus を用いて変更することが可能であり各 Supercell に適した変数を用いることができる。初期設定では BCID= 0 の E_{T} の閾値が 800 GeV, BCID= -1 が 750 GeV に設定されている。

これは ADC データが頭打ちになり波形が歪むことに起因している。そのため図 5.14 で示す領域に入

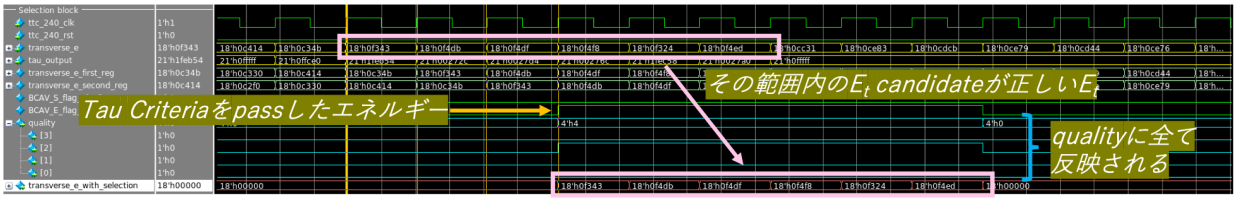
る E_{T}, $E_{\mathrm{T}}\tau$ を持つ BCID が Saturation した波形に対応すると言える。こうして Saturation した波形は E_{T} は正確に求められないが、その BCID を同定することが可能になる。Saturation した波形のエネルギーに対する処方は Combine Block に引き継がれる。

Tau Criteria は E_{T} の再構成とバンチの同定、Saturation Criteria は Saturation した波形のバンチを同定するものであった。そのためそれぞれの Criteria はバンチ同定に成功したか否かを示す信号 $BCAV_E, BCAV_S$ をもつ。それらを用いて Selection Block がバンチを同定できたかを

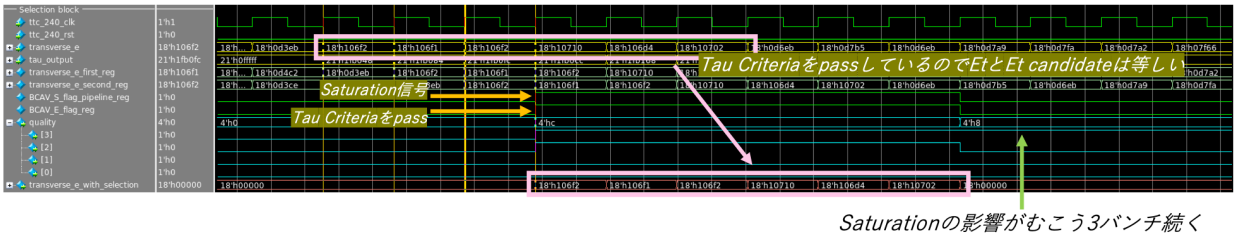
$$BCAV = BCAV_E \mid BCAV_S$$

と表す。どちらかの Criteria がバンチを同定できた場合 BCAV は 1 となる。Tau Criteria と Saturation Criteria はそれぞれ global control selection, global control saturation により Filtering Algorithm に含めるかを設定できる (表 5.1)。Selection Block は Tau Criteria と Saturation Criteria に関する情報を'quality' にまとめて Combine Block に送る (図 5.1)。

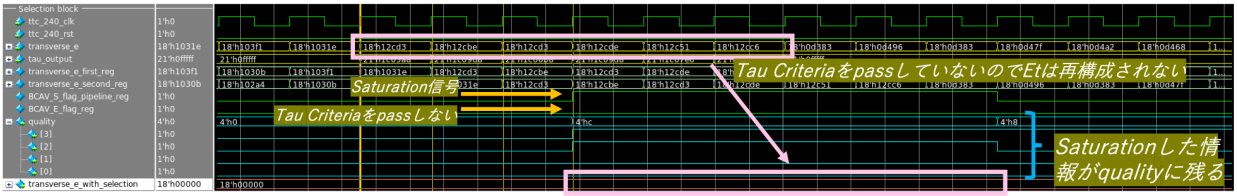
$$\mathrm{quality} = \{\mathrm{saturation\_flag}, BCAV, 0, 0\}$$



(a) τ criteria を満たした場合



(b) τ ctiteria を満たしかつ Saturation した場合



(c) τ criteria を満たさず Saturation した場合

図 5.15: Selection Block の wave window。Tau Criteria を満たした場合はその E_{T} candidate が E_{T} として割り当てられる。一方 Saturation Criteria を満たした場合の E_{T} candidate は粒子の落としたエネルギーに対応した値ではないので Selection Block では E_{T} に値は割り当てられない。Saturation に対する処方は Combine Block に引き継がれる。

5.4 Combine Block

Saturation した波形でもその特徴を捉えることにより Selection Block で BCID を同定することが可能であった。しかしそのままではたまたまその領域に入ってしまった E_{T}, $E_{\mathrm{T}\tau}$ の組み合わせまでも Saturation Criteria を抜けてしまう。そのため Combine Block では Saturation した Supercell の周りの Supercell を調査し、果たして本当に Saturation が起こったのかを図 5.17 に示すように 2 clock のレイテンシーで確認する。Saturation するほどの大きなエネルギーが落とされた場合、図 5.16 のように周りの Supercell にもある程度大きなエネルギーを持つものがいるはずである、また前後の Layer にも大きなエネルギーを落としているはずであるので Trigger Tower の領域を調査する。Saturation を起こした Supercell の周りに 1 GeV 以上のエネルギーを落とした Supercell がある場合は Selection Block で説明した BCAV に 1 を割り当て直し、Saturation の閾値に対応するエネルギーをその Supercell のエネルギーとして割り当てる。Combine Block は global control combine により Filtering Algorithm に含めるかを設定できる (表 5.1)。

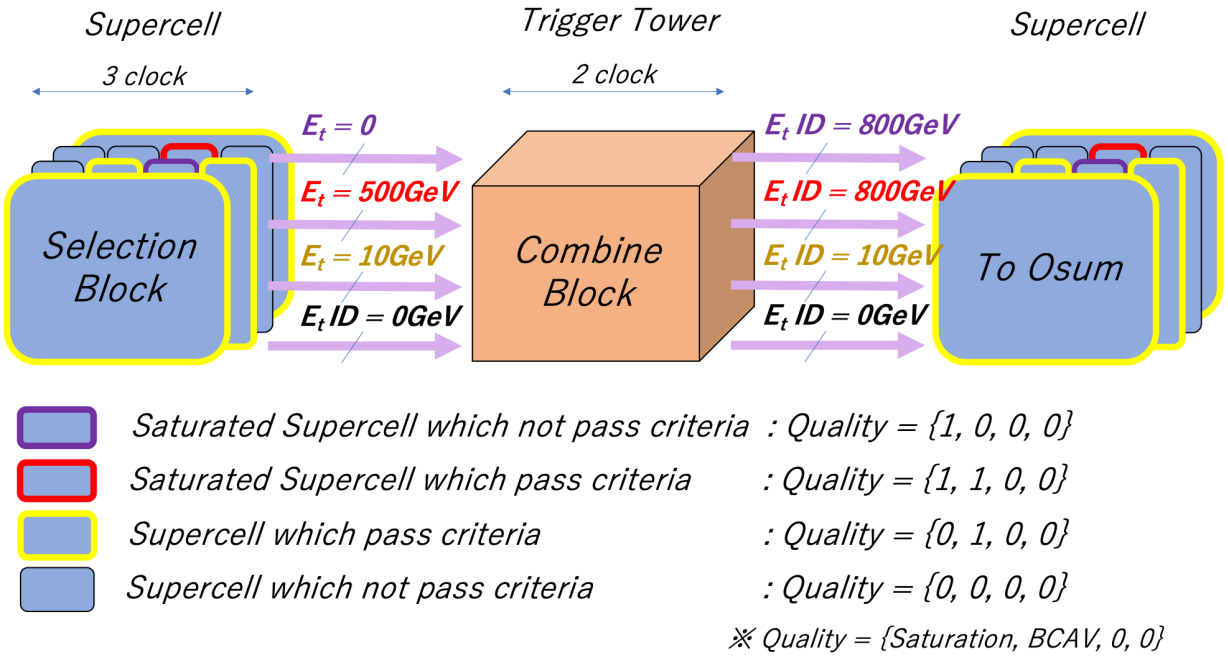
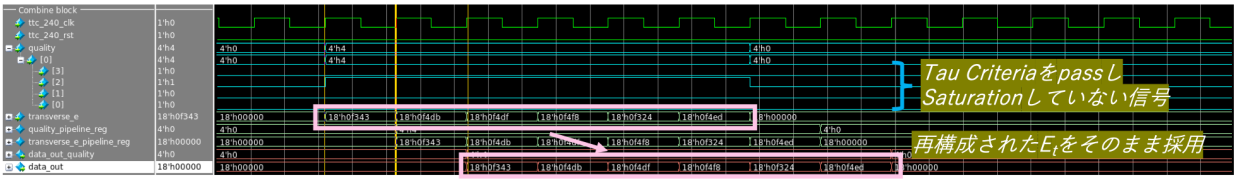
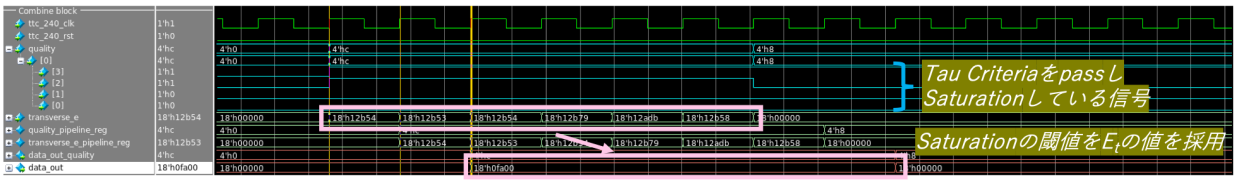


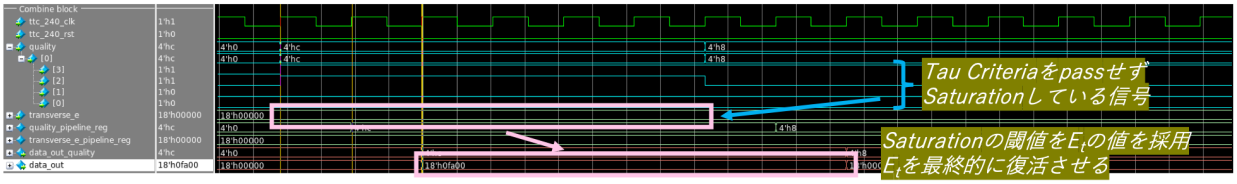
図 5.16: Combine Block の動作。各 Supercell の Quality を元に Saturation した波形に対し大きなエネルギーを割り当てる。また周りの Supercell に対しても Saturation の影響を考慮して大きなエネルギーを割り当てる。



(a) τ ctiteria を満たし Saturation していない場合 (図 5.15a に対応)



(b) τ ctiteria を満たし Saturation している場合 (図 5.15b に対応)



(c) τ criteria を満たさず Saturation している場合 (図 5.15c に対応)

図 5.17: Combine Block の wave window。Quality を元にそれぞれの Supercell に対し適切なエネルギーを割り当てる。Saturation した波形の場合、FIR Filter によりエネルギーを再構成できないのでエネルギーの値としては正確ではないが Saturation を起こすエネルギーの閾値を割り当てる。

5.5 IP bus controller

LATOME Firmware 内部のレジスター, メモリにはそれぞれに固有のアドレスを持つものがある (図 5.18)。それらのレジスターのアドレスにアクセスすることによりそのレジスターが持つデータを読み出し, 書き込みすることができる。それを実装しているのが IP bus である。アドレスは 32 bit であり読み出されるデータも 32 bit である。IP bus は 1 Gbps でデータのやりとりを行うことができる。IP bus 経由でアクセスできる User Code 内のデータは表 5.4 に示す。

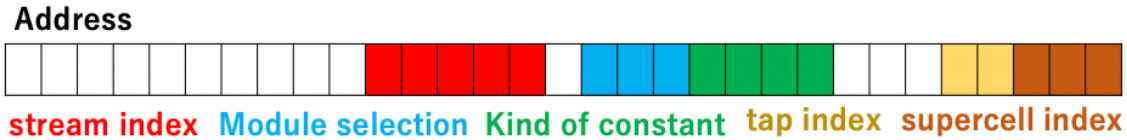


図 5.18: User Code における FIR Filter, Saturation Detection の IP bus の Address。白い部分は使われていない。tap index は DSP Block の入力口に対応する。

Block	固有のアドレスを持つレジスター/メモリ
User Code	User CodeStatus
FIR Filter	係数、Pedestal
Saturation Detection	係数、Pedestal
Selection Block	Saturationconstants
Combine Block	Saturationconstants
Summation ADC	ADC データの和
ADC Shape Checker	6 点 ADC データと Bunch Crossing
Peak Detector	4 点 ADC データと Bunch Crossing

表 5.4: User Code 内のレジスター, メモリと IP bus のつながり

5.6 Summation ADC Block

User Code の最も簡単な機能として LHC 1 周分の ADC データの和を計算し保存するものがある。User Code は ADC データを元にエネルギーを再構成する。そのため安定した ADC データの取得が重要である。LTDB からの ADC データは Istage で抜き出され、Remap で並び替えが行われる。それらの手順が正しく行われており、ADC データが安定しているかを確認するために作成した。図 5.19 に示した機構は全てのスーパーセルに対し実装されており、IP bus 経由でクリア信号を送るまでその値を保持し続ける。また M20K から抜き出した値を LHC のバンチ数である 3564 で割ることにより実際に得られたデータから Pedestal を再計算することができる。Clear 信号が送られると M20K 内に記録されたデータと各 Supercell に対応するレジスターが 0 に戻され、BCID が 0 になるまで待機する。再び BCID が 0 になると自動的に計算を再開する。

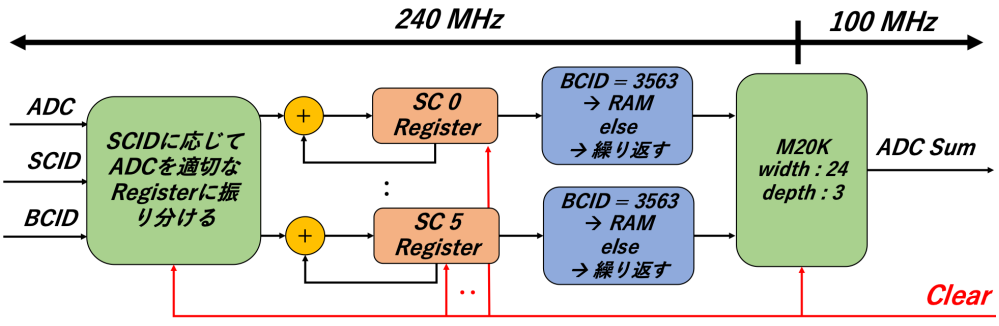


図 5.19: Summation ADC の設計図。計算は毎回 BCID が 0 の時点から始まる。BCID が 3564 になると ADC データの和が M20K に保存される。

5.7 Peak Detector Block, ADC Shape Checker Block

LATOME Firmware は Remap で Supercell 信号を Supercell Mapping [*4]に応じて並び変える。それらの並び替えが正確に行われているかを確認するツールとして Peak Detector, ADC Shape Checker を開発した。Peak Detector はエネルギー (E_{T}) を用いて、ADC Shape Checker は ADC データの振る舞いを確認することで LAr パルスを探し出す。

Peak Detector は User Code のメインパスで算出された Tau Criteria を満たしたエネルギー (Selection Block を通った E_{T}) を元に 5 GeV 以上のピークを探し出し、各 Supercell に対し大きなエネルギーを落とした BCID とそのエネルギー算出に用いた 4 つの ADC データを記録する (図 5.20)。ADC データと BCID は FIFO を用いて FIR Filter と Selection Block のレイテンシーの合計である 25 clock 遅らされる。これにより E_{T} と同じタイミングの ADC データ、BCID を記録することができる。ADC データは 6 clock おきに同一の Supercell に対応する値を持つので (図 5.2)、4 点の ADC データ全てを記録するには 18 clock 必要である。そのためその間に別の Supercell に 5 GeV 以上のパルスが来た場合は Pulse Flag を立てないようにデザインされている。

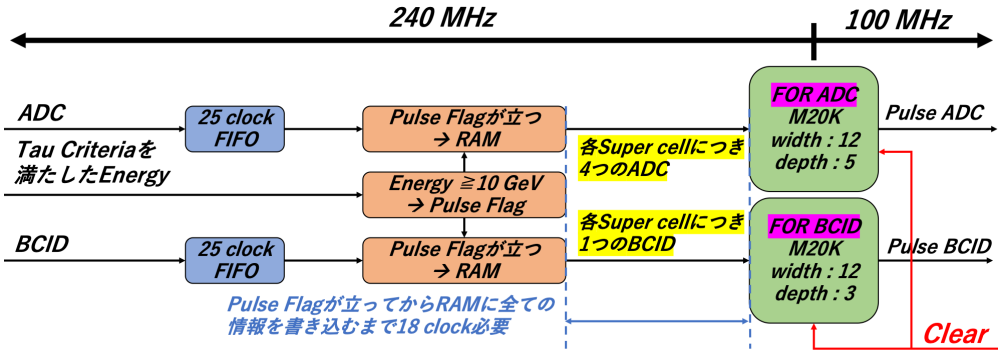


図 5.20: Peak Detector の設計図

Peak Detector は再構成されたエネルギーを用いて LAr パルスを探し出すため正確にパルスのみを探し出すことができる。しかし Peak Detector は FIR Filter での DSP Block を用いた乗算が正常に動作することを要求するのでパルスを見つけるまでのステップが多く開発段階での利用が難しい。そのため ADC データのみを観察することで LAr パルスを見つけ出す ADC Shape Checker を開発した (図 5.21)。LAr パルスは波長が 600 ns 程度と一定で Bipolar 波形は図 5.22 に示すよう大きなエネルギーを落とした場合毎回似た形の波形を残すので、式 5.7 の条件を ADC データに課すことで LAr パルスを見つけ出す。

[*4] 4.2.3 で説明したように、LTDB からの信号は Remap で各 LATOME Board に依存した Supercell の並び替えを行う。

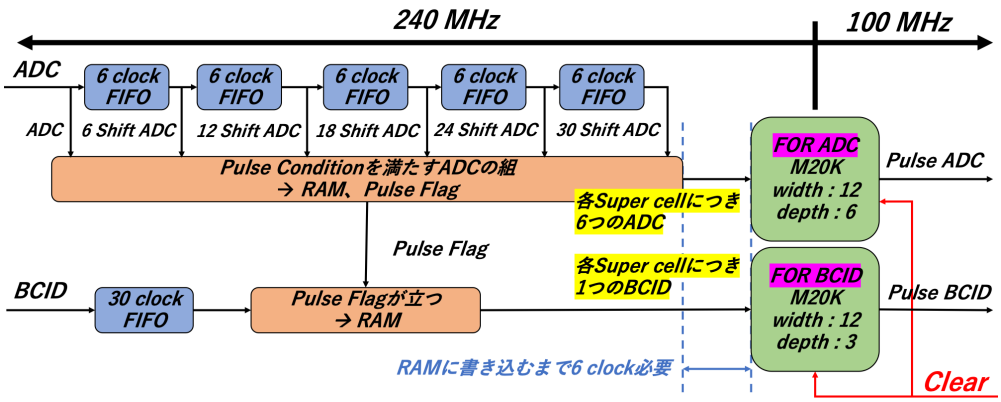


図 **5.21:** ADC Shape Checker の設計図

$$\begin{cases} \text{30 clock shift ADC} < \text{24 clock shift ADC} \\ \text{24 clock shift ADC} < \text{18 clock shift ADC} \\ \text{12 clock shift ADC} < \text{6 clock shift ADC} \\ \text{6 clock shift ADC} < \text{ADC} \\ \text{Gap} < \text{18clock shift ADC} - \text{30 clock shift ADC} \end{cases} \tag{5.7}$$

式 5.7 中の Gap は IP bus を用いて任意の値に設定可能である。そのため波高の大きな LAr パルスのみを記録することが可能である。ADC Shape Checker は各 Supercell あたり 6 個の ADC データを記録することから LAr パルスを見つけてから処理を完了するまで 6 clock 必要である。そのためその間は Pulse Flag を立てないようにデザインされている。

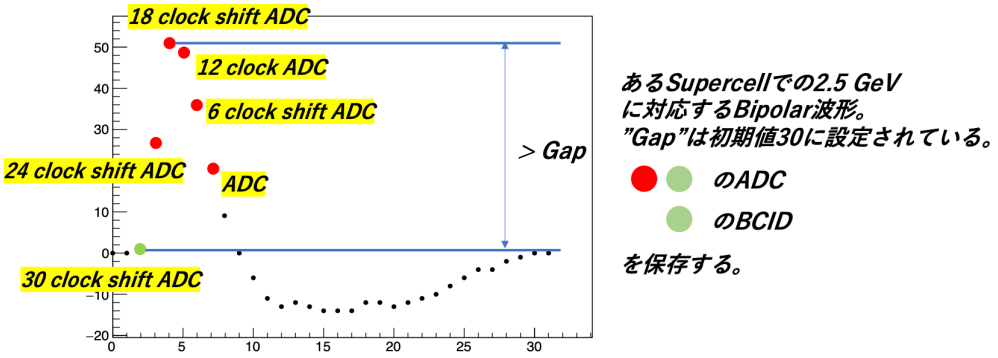


図 **5.22:** ADC Shape Checker による Pulse peak の条件。Bipolar 波形の横軸は BCID, 縦軸は ADC データから Pedestal を差し引いた値である。この Bipolar 波形はある Supercell の 2.5 GeV の信号に対応する。Gap の値は IP bus により書き換え可能であり大きな pulse のみを選んで記録することも可能である。

Peak Detector, ADC Shape Checker はどちらも IP bus を用いてのみ ADC データ、BCID の読み出しが可能である。それらのデータは Clear 信号が送られるまで保持されるため同一の Supercell に複数の LAr パルスが来た場合は、初めに記録された情報のみが保存され、後からきた LAr パルスの情報は保存されない。

5.8 Monitoring Block

User Code はモニタリングデータとして以下の信号を送る。

- User Code の入力である ADC データとそれの valid 信号
- ADC データ-Pedestal とそれの valid 信号
- エネルギーとそれの valid 信号 (Selection Block の出力に対応する)
- Saturation まで考慮したエネルギーとそれの valid 信号 (Combine Block の出力に対応する)

これらのデータは常 Monitoring Module に送られており User Code の各 Block のレイテンシーも考慮されている。Level 1 Accept 信号が発行された後 Monitoring Module では Multiplexer を用いていずれかのデータとそれに対応する valid 信号が 32 Bunch Crossing 分送られる。

Module	計算時間 [clock]	メインパス [Yes/No]	レイテンシーに寄与するか [Yes/No]
FIR Filter	22	Yes	Yes
Saturation Detection	22	Yes	Yes
Selection Block	3	Yes	Yes
Combine Block	2	Yes	Yes
Baseline Correction	$\mathrm{ceil}(\frac{3564}{2^5}) \times 2^{10} \times 3564 \times 6 + 1$	Yes	No
IP bus	-	No	No
Summation ADC	3564×6	No	No
Peak Detector	$\leq$ 3564×6	No	No
ADC Shape Checker	$\leq$ 3564×6	No	No
Monitoring Block	-	No	No
User Code	27	Yes	Yes

表 5.5: User Code の各 Module のレイテンシー。Baseline Correction は初期状態で 0 を返すようにデザインされておりまた保存用 RAM に計算結果を記録そこから常に読み出しており計算用 RAM とは独立である。そのため Main path ではあるがレイテンシーへの寄与は無い。

第 6 章

Firmware の検証

LATOME Firmware は Level 1 trigger において重要度が高く、また様々な機能が含まれていることから大規模な Firmware になる。そのため各機能が設計通りに働いていることの確認作業が重要である。この章は Simulation を用いた User Code の検証と実機での LATOME Firmware の検証について述べる。

6.1 Simulation を用いた検証

Simulation には Mentor Graphics 社[*1]が販売する Questa Sim のバージョン 10.6 c を用いた。System Verilog, VHDL など多くのハードウェア言語に対応した Simulator である。Firmware の検証には Software で作成した Model との動作比較をすることにより行う。Firmware 側には全く変更を加えず適切な入力を用意し、それに対する出力を Software で予想し比較することにより検証を行う。各 Block に対し一連の機構 (図 6.1) を作成した。全ての Simulation の機構は Firmware と Software 間で不一致があった時点で強制終了するようにデザインされており、正常終了した場合は全ての Firmware の出力が Software と一致していることを意味する。

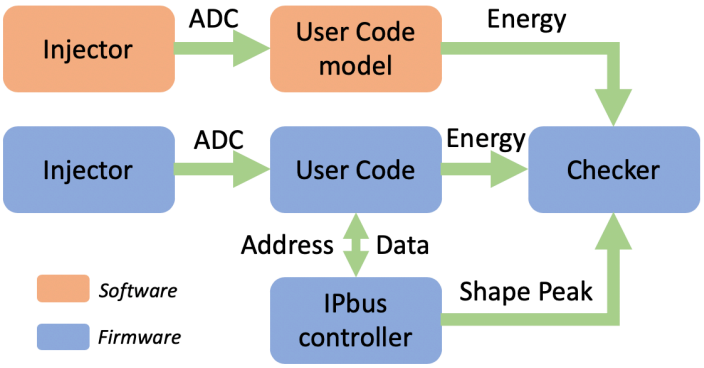


図 6.1: Simulation による User Code の検証。Software は C++, python を用いて、Firmware は System Verilog, VHDL を用いて作成されている。

[*1] https://www.mentor.com

Software で予測した出力は Memory Initialization File (MIF) として保存され、それを用いて Checker 内の ROM を初期化する。Checker は Firmware で作成されており User Code のレイテンシーも考慮に入れた検証を行うことができる。Simulation による User Code の検証の詳しい流れは付録 C に纏めた。作成した Simulation ツールを用いることにより User Code 内の全ての Block のロジックについて任意のデータセットを用いて検証することができる。

6.1.1 Universal VHDL Verification Methodology

Universal VHDL Verification Methodology (UVVM) は Bitvis AS[*2]が管理しているオープンソースである。VHDL を用いた TestBench を作成する際の強力なツール, ライブラリ群が UVVM であり 2016 年に初めて提供された。またユーザー側もこれらのツールを改良, 作成することが可能であり UVVM は GitHub を用いて配布されている[*3]。Firmware の開発は可読性、再利用可能性、拡張性が重要であり Simulation を用いた検証では UVVM を利用することによりこれらを実現した。

6.1.2 A Software Framework for ATLAS Readout Electronics Upgrade Simulation

A Software Framework for ATLAS Readout Electronics Upgrade Simulation (AREUS)[25] は Phase-I Upgrade, Phase-Ⅱ Upgrade を想定して作成された液体アルゴン検出器全体の Simulation ツールである。Geant4 により生成された検出器全体の情報を入力とし各読み出しに対し個別に Simulation を行う。液体アルゴン検出器の場合パイルアップの影響まで加味した Simulation を行うことができる。Simulation による User Code の検証では AREUS で生成された ADC データを入力とした、これにより現実に近い状況での検証を可能にした。

6.1.3 User Code main path の検証

Filtering Algorithm

User Code の Main path は Remap からの ADC データを 27 clocks の一定のレイテンシーで、Osum にエネルギーを、Monitoring Block に ADC データやエネルギーを送る。User Code は FIR Filter, Saturation Detection, Selection Block それぞれの計算結果を直接確認することはできない[*4]。そのため一連のアルゴリズムを一度に検証する必要がある。Software により Osum と Monitoring Block への出力を Supercell, Bunch Crossing ごとに予測し、bit by bit で比較する (図 6.2)。LHC1 周以上の Firmware からの出力が Software で予測した値と一致した。入力のデータは LHC1 周分であるので入力データ全てに対し Firmware が Software で予測した値を出力した。

[*2] https://bitvis.no
[*3] https://github.com/UVVM/UVVM
[*4] 厳密には Monitoring data を用いれば間接的に確認することは可能であり実機での検証 (6.2 節) ではこれを用いた。

図 6.2: Filtering Algorithm の Simulation による検証。ピンク、青、赤、黄色がそれぞれ ADC データ、*ADC − Pedestal*、Selection Block の出力、Combine Block の出力に対応する。

Combine block

Combine Block の検証のためには Saturation した波形を用いる必要がある (図 6.3)。Combine Block の検証に用いたデータセットは 100 BCID おきに Saturation しない波形、Saturation するが Tau Criteria を通る波形、Saturation し Tau Criteria を通らない波形が連続したものであり Combine Block の動作を全て検証することができる。そのデータセットにおいて LHC1 周分の Firmware の出力が Software と完全に一致することを確認した。

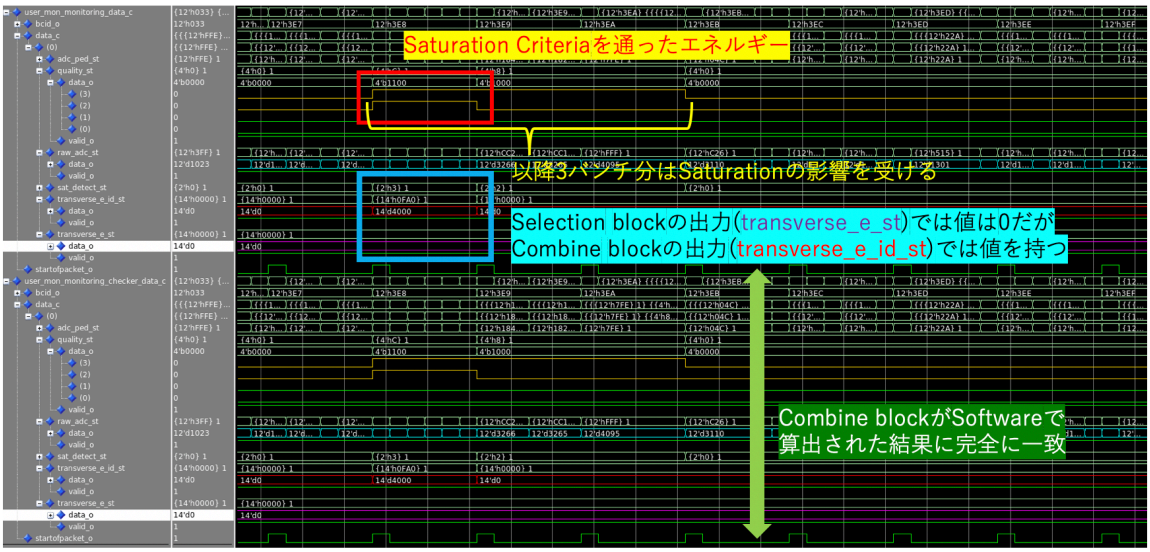


図 6.3: Combine Block の Simulation による検証。ピンク、青、赤、黄色がそれぞれ ADC データ、*ADC − Pedestal*、Selection Block の出力、Combine Block の出力に対応する。

Baseline Correction

Baseline Correction の検証のためには ADC データの入力方法を工夫する必要がある。普段通りに ADC データを入力してしまうと Baseline Correction の計算結果がその ADC データの値と一致してしまい User Code の出力が式 6.1 のように全て 0 になってしまう。その場合、正しく計算されているのか何かしらのバグなのかの判定が難しくなる。

$$\begin{aligned} E_{tj}^{i} &= \sum_{k=0}^{3} a_{3-k} \left(ADC_{j}^{i+k} - Pedestal_{j} - Baseline_{j}^{i+k} \right) \\ &= \sum_{k=0}^{3} a_{3-k} \left(ADC_{j}^{i+k} - Pedestal_{j} - \left(\frac{\sum_{l=0}^{2^N-1} ADC_{j}^{i+k+l\times L}}{2^N} - Pedestal_{j} \right) \right) \\ &= \sum_{k=0}^{3} a_{3-k} \left(ADC_{j}^{i+k} - Pedestal_{j} - ADC_{j}^{i+k} + Pedestal_{j} \right) \\ &= 0 \end{aligned} \tag{6.1}$$

ここで L は LHC1 周に相当する Bunch Crossing 数とする。Baseline Shift はそれぞれの Supercell, Bunch Crossing に対し計算されるものなので、2^N 回同じ ADC データを足し上げその平均をとるので最終的に計算結果がすべて 0 になってしまう。Baseline Correction も同様その出力を直接確認することができないので適切な入力を用意し User Code の出力を確認する。User Code の出力には Filtering Algorithm が自ずと必要になり Filtering Algorithm には Bipolar 波形を入力しなければ出力は全て 0 になってしまう。そのため全く新しい ADC データの入力方法を考案した。以降の説明のため新しい用語 2 つを定義する

- Before → Baseline Correction が全ての Supercell, Bunch Crossing の Baseline Shift を計算し終わる前。この間の Baseline Shift は常に 0 である。
- After → Baseline Correction が全ての Supercell, Bunch Crossing の Baseline Shift を計算し終えた後。この間の Baseline Shift は Before で計算された値である。

Baseline Correction が正しく動作しているかを確認するためには Baseline Shift が正常に計算されてから確認する必要がある。つまり After では

$$\begin{aligned} ADC_{j}^{i} - Pedestal_{j} - Baseline_{j}^{i} &= Bipolar_{j}^{i} \\ Baseline_{j}^{i} &= -Bipolar_{j}^{i} + ADC_{j}^{i} - Pedestal_{j} \\ Baseline_{j}^{i} &= -Bipolar_{j}^{i} \end{aligned}$$

を満たす必要がある。ここで簡単の為に ADC_{j}^{i} と $Pedestal_{j}$ が等しいと仮定した。After でこの条件を満たすには Before では

$$\begin{aligned} Baseline_{j}^{i} &= \frac{\sum_{k=0}^{2^N-1} ADC_{j}^{i}}{2^N} - Pedestal_{j} \\ &= ADC_{j}^{i} - Pedestal_{j} \\ &= -Bipolar_{j}^{i} \\ ADC_{j}^{i} &= Pedestal_{j} - Bipolar_{j}^{i} \end{aligned}$$

を ADC_{j}^{i} として入力すると良い。以上のことから

- Before $\rightarrow ADC_j^i = Pedestal_j - Bipolar_j^i$
- After $\rightarrow ADC_j^i = Pedestal_j$

と ADC データを入力することにより最終的に Bipolar 波形を残すことができ、Filtering Algorithm を検証する時に用いた予測値を流用させることができる (表 6.1)。

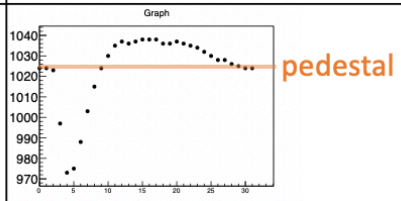
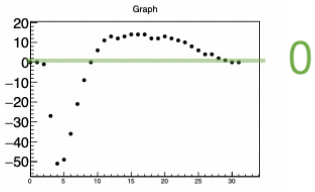
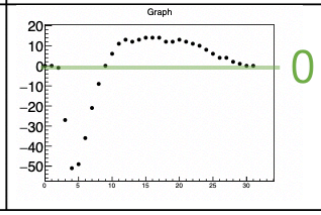
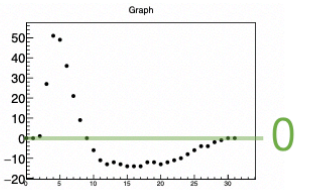
	Before	After
ADC		pedestalと同じ値
Baseline	Baselineの計算が完了する前は0を返す設計	
ADC – ped –Bas		

表 6.1: Baseline Correction の Simulation による検証方法。Before と After の切り替えは BCID 信号を用いて行われるため User Code 自体に変更を加えず検証を行える。

Filtering Algorithm での検証と異なり ADC データの入力の仕方を変えているので Monitoring data の内'ADC' と'ADC-Pedestal' は Filtering Algorithm で用いた予測値とは異なる値を出力する。そのためそのほかの信号が予測値と一致するかを確認した (図 6.4)。その結果 LHC1 周以上の時間幅で Firmware の出力が Software と一致することを確認した。

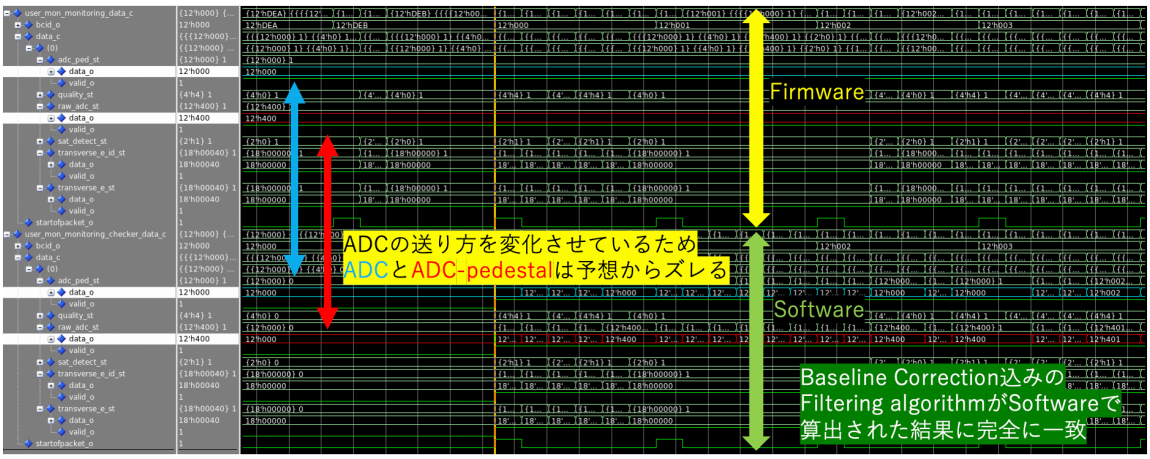


図 6.4: Baseline Correction の Simulation による検証。ADC データの送り方を変化させているため、ADC データ (青) と $ADC - Pedestal$ (赤) は予測値とは異なる値を出力する。しかし Selection Block, Combine Block の出力は Filtering Algorithm の検証で予測した値と一致する。これにより Baseline Correction で正しく計算されていることが確認された。

6.1.4　User Code それ以外の検証

Main path 以外の User Code の Module は、IP bus により Module に直接アクセスすることができる。そのため各 Module の処理結果を IP bus で読み出しそれらと予測値を比較する TestBench を作成した。

Peak Detector

Peak Detector は 1 つの液体アルゴン pulse のみを保存することができるので single pulse のデータセットを用いて検証を行った (図 6.5)。Firmware の出力である 4 つの ADC データ, 1 つの BCID は全て Software の予測値と一致することを確認した。

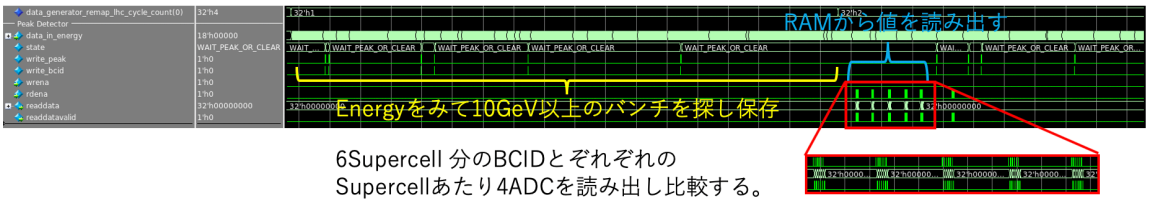


図 6.5: Peak Detector の Simulation による検証。LHC 1 周待ち、各 Supercell に pulse を見つけさせる。その後 IP bus を用いて読み出され、予測値と比較する。

ADC Shape Checker

ADC Shape Checker は 1 つの Peak Detector 同様液体アルゴン pulse を探しだしその情報を保存することからデータセット内に一つの液体アルゴン pulse をもつ single pulse のデータセットを用いて検証を行った (図 6.6)。Firmware の出力である 6 つの ADC データ, 1 つの BCID は全て Software の予測値と一致することを確認した。ADC Shape Checker は異なる Supercell に連続して pulse が来ると片方の pulse しか ADC データ, BCID を格納できない。Software ではそれまで考慮に入れて Firmware の動作を予測した

図 6.6: ADC Shape Checker の Simulation による検証。LHC 1 周待ち、各 Supercell に pulse を見つけさせる。その後 IP bus を用いて読み出され、予測値と比較する。

Summation ADC

Summation ADC は LHC1 周分の ADC データを足し上げるだけの機能であるのでいかなるデータセットを用いても検証が可能である (図 6.7)。検証は Step function, Single pulse, Saturation pulse のデータセットを用いて行い、いかなるデータセットでも Firmware の出力が予測値と一致することを確認した。

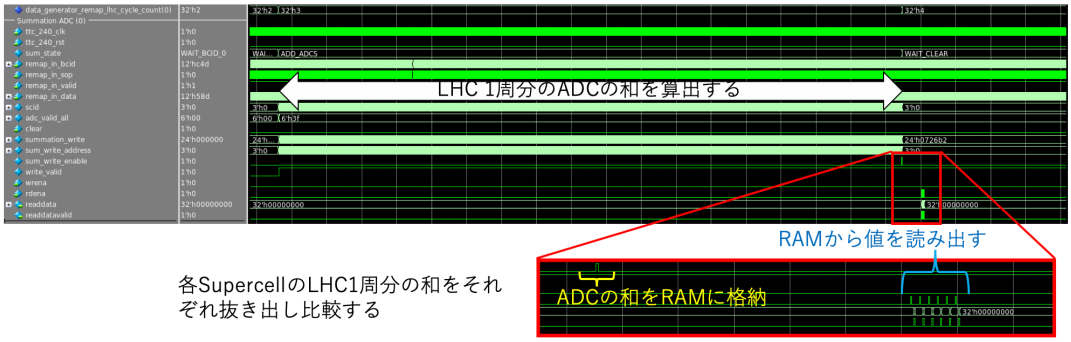


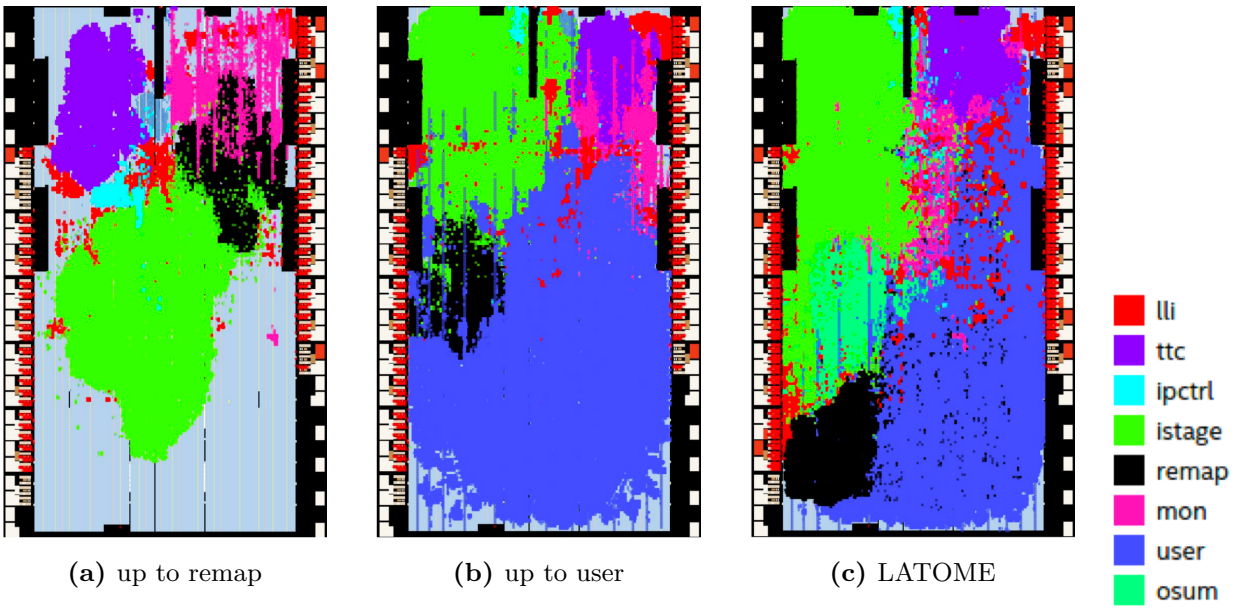
図 6.7: Summation ADC の Simulation による検証。計算された値は RAM に格納される (黄)。その後に IP bus を用いて読み出され、予測値と比較される。

6.1.5　Simulation を用いた検証のまとめ

以上で述べた機構はすべて私が主導的に作成したものであり、実際に Firmware 開発に用いられている。Simulation を用いた User Code の検証は、すべての Block に対し正常に動作し各 Block のロジックを検証することが可能となった。これらの機構はすべて自動で Firmware の出力と Software での予測値を比較するものである。そのため Firmware に変更を加えた後、この Simulation を走らせることでロジックの変更の是非を確認することができる。また入力のデータセットは AREUS で生成したものなので、実際に RUN3 環境に近い状況での Simulation による検証が可能になった。Simulation での検証は各 Block のロジック、レイテンシーを確認するものであり実機でのレジスター間の信号の遅延は再現されていない。そのため次のステップとして、動作安定性を確認するために実機での検証を行なった。

6.2　実機を用いた検証

Simulation 上では User Code が設計通りに機能していることが確認された。しかし Simulation での検証ではレジスター間の信号の遅延やその FPGA の用いられる環境などは考慮に入れられず、ロジックの確認のみを行う。そのため実際の Firmware の安定性を検証するために次のステップとして実機での検証が必要である。Firmware のコンパイルは Intel 社が販売している Quartus Pro Edition バージョン 19.3[*5] を用いた。LATOME Firmware は様々な機能が含まれることから非常に大規模な Firmware であるので実機での動作安定性の検証は重要なタスクである。



(a) up to remap　**(b)** up to user　**(c)** LATOME

図 6.8: LATOME Firmware の Arria 10 の占有領域。M20K や DSP block は Arria 10 内で 1 列に配置されている。そのため User Code 入りの Firmware では 1 列にロジックが使用されていることが分かる。

[*5] https://www.intel.co.jp/content/www/jp/ja/software/programmable/quartus-prime/download.html

図 6.8 は Arria 10 内の使用されているロジックの領域を示しておりそれぞれの Module ごとに色分けされている。これからもわかるよう User Code は Arria 10 のおおよそ半分の領域 (ロジック) を使用している。またより詳細な情報は表 6.2 に纏めた。

		up to remap	up to user	LATOME Firmware	Arria 10
ALM		127854 (30%)	259338 (61%)	297547 (70%)	427200
LAB		21419 (50%)	38666 (91%)	40879 (96%)	42720
Block memory bit		8396848 (15%)	29208232 (53%)	29208232 (53%)	55562240
M20K		526 (19%)	2439 (90%)	2439 (90%)	2713
DSP block		0 (0%)	248 (16%)	248 (16%)	1518
320 MHz clock domain	Slack[*6]	-0.260 ns	-0.149 ns	-0.879 ns	-
	TNS[*7]	-0.362 ns	-0.197 ns	-2999.55 3ns	-
240 MHz clock domain	Slack	-0.052 ns	-0.439 ns	-2.600 ns	-
	TNS	-0.068 ns	-135.747 ns	-9605.830 ns	-
280 MHz clock domain	Slack	-	-	-0.514 ns	-
	TNS	-	-	-6.011 ns	-

表 6.2: 各 Firmware の Arria 10 の占有率とクロック

このように User Code 入りの Firmware は実装段階でのロジックの占有率がおおよそ 90% と高く、User Code の Main clock domain である 240 MHz のクロックが Worst Case [*8]で Negative である。また Quartus によるコンパイルでは初期段階で乱数を用いているのでコンパイルごとに結果が違い問題の再現性がなく、クロックの Negative slack を持つロジック間のパスはコンパイル依存する。また Firmware は FPGA のおかれる環境にも依存し温度変化などによっても動作が多少変化する。これらの問題は Simulation では確認することのできない問題であり予測ができない。そのため実機での検証はその時の Firmware の安定性や環境など不確定な部分が多い。このような状況下では適切な環境のテストベンチで可能な限りロジックの少ない (検証すべきロジックまでを含んだ) Firmware を用いるべきである。そのため図 6.8b に示す User Code までを含んだ Firmware を用いた検証を行った。User Code は Osum の有無にかかわらず動作は制限されない。CERN 内にあるテストベンチでは 5 つの LATOME Board が利用可能であり、1 つは LTDB と繋がっており残り 4 つの LATOME は 2 つずつ互いに繋げられている (図 6.9)。

[*6] Slack は Firmware 内部のレジスター間のデータの伝達がクロックの要求値に対し間に合わない場合 negative の値になる。周波数 320 MHz のクロックの場合 3.125 ns の間にデータを送り先のレジスターに伝達しなければならないがそれが最悪の場合 0.879 ns 遅れることを表す。ここでは全てのレジスター間の中で最低の Slack を持つものを表示している。

[*7] TNS は Total Negative slack の略で、Negative Slack の合計であり Firmware の安定性を表す一つの指標になる。TNS が 0 はすべてのレジスター間でクロックに違反することなくデータの伝達を行えることを意味する。0 から離れるにつれて Firmware の動作安定性を保証できなくなる。

[*8] FPGA のタイミング解析には Quartus の Timing Analyzer を用いて行った。Timing Analyzer 内では FPGA の製造のばらつき、電圧、温度の条件を変化させて解析を行うことができる。FPGA の動作は電圧が高く低温の場合に高速に動作する。表 6.2 の結果は 100 度の環境下で 900 mV の電圧を用いて動作させた場合を想定している。

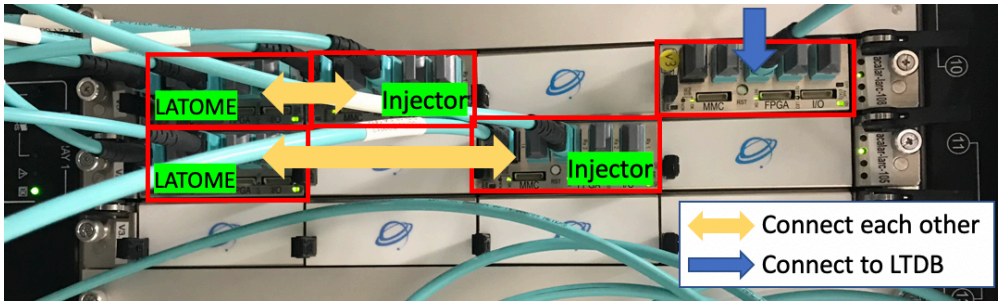


図 6.9: CERN の Point1 内にある EMF に設置されたテストベンチ。全ての LATOME Board は 96 本の受信用、送信用光ファイバーがつなげられており RUN3 と同じ状況を再現している。

User Code の実機の検証では主に 2 つの LATOME Board を用いた。片方は Injector Firmware を搭載して Pattern generator を用いて指定したデータセットをもう片方の LATOME Board に送る。もう一方は LATOME Firmware を搭載して Injector Firmware からのデータを受け取り User Code の各機能の検証を行う。それぞれの LATOME Board は実際に RUN3 から用いられるものであり、96 本の受信, 送信用光ファイバーを搭載して RUN3 での環境と同じ条件での検証を行うことができる。実機での User Code の検証の詳しい流れは付録 D に纏めた。

6.2.1　User Code main path の検証

User Code は LATOME Firmware の内部の Block であるので Simulation のように User Code の出力を直接調べることはできない。そのため Monitoring data を用いて間接的に実機での検証を行った。User Code は Monitoring data として'Raw ADC', 'ADC - Pedestal', 'Transverse Energy', 'Transverse Energy ID' の 4 種類を送っている。Monitoring data は Level 1 Accept に応じて連続した 32 Bunch Crossing 分のデータが専用の PC (mon PC) に記録される。そのため全ての LHC cycle (3564 BCID) をカバーするのには少なくとも 112 回の Level 1 Accept を TTC から送る必要がある。また Monitoring data はいくつかのバージョンが存在 (表 6.3) し、バイナリデータとして保存されそれぞれ図 6.10 に示す特別な Header, Trailer が付加される。今回は Monitoring data version v2 を用いそれに対応した Decoder も作成した。

Version	description
v2	ATLAS Header を付加させない
v3	ATLAS Header を付加させたもの
v4	ROD, ROB header 無しの ATLAS Header を付加させたもの

表 6.3: Monitoring data の Version。ATLAS Header は offline でのデータ解析のために用いられる。ROD (Read Out Driver) と ROB (Read Out Buffer) header は ROD での処理のために用いられる。今回は Monitoring data の取得が目的であるので ATLAS Header 無しのバージョンを用いた。

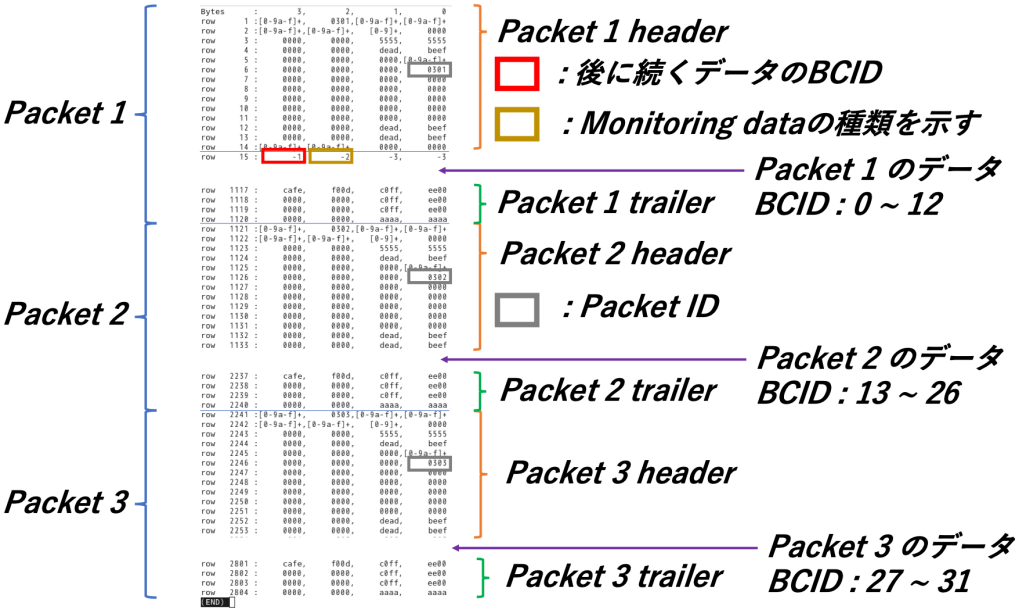


図 6.10: Monitoring data のフォーマット。各 Monitoring data は 3 つのパケットに分割されて記録される。それぞれのパケットには別途 Header と Trailer が付加され、Monitoring data の種類やパケットの ID などの情報を持つ。

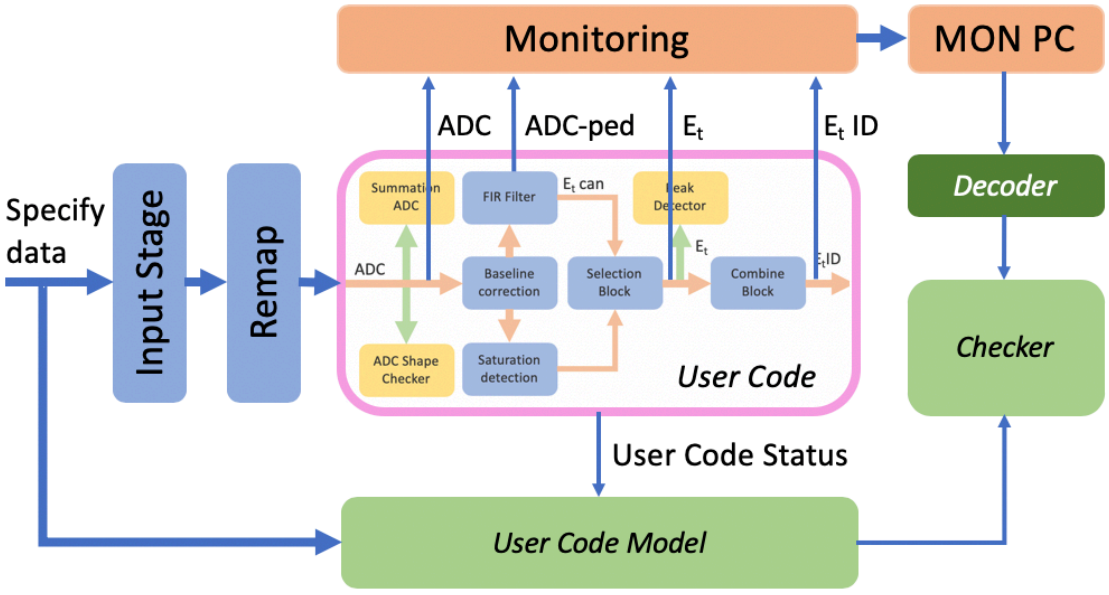
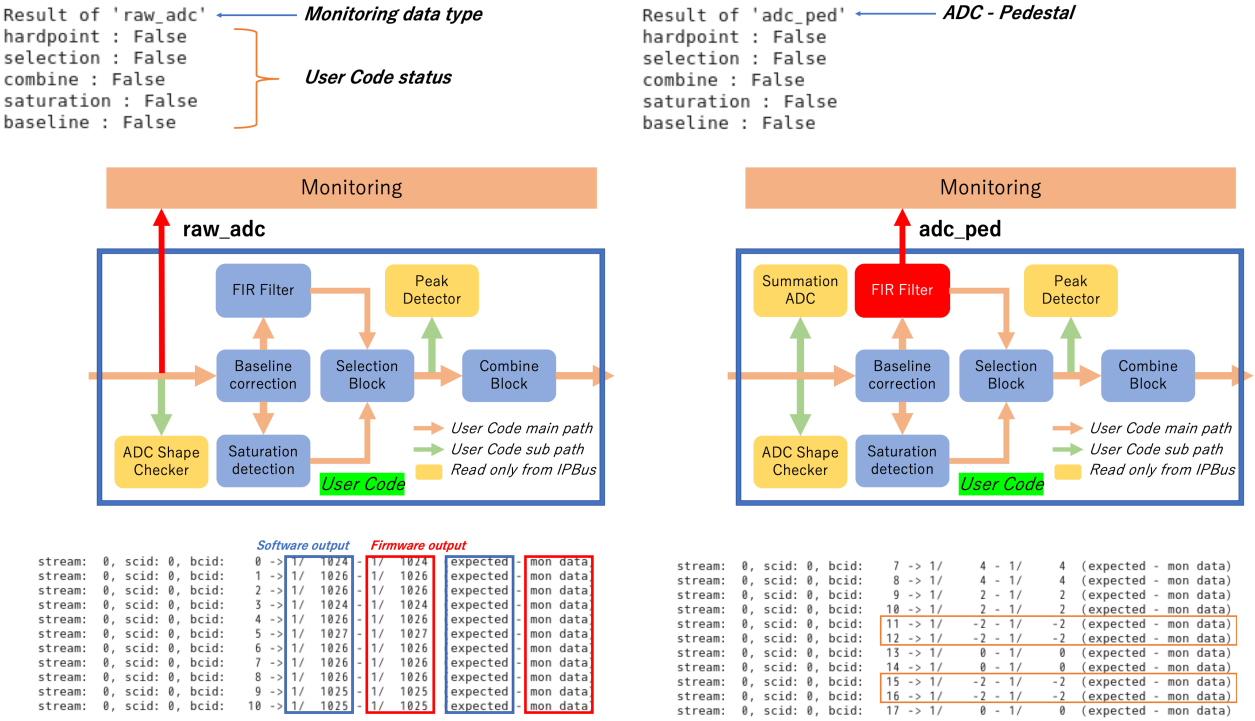


図 6.11: 実機での User Codemain path の検証の概要。Simulation 同様に User Code には変化を加えず、適切な入力を用意し出力を予測する。Decoder, checker, Model の作成を新たに行った。

Injector Firmware から送られてくる信号は実際に RUN3 で検出器から送られてくるフォーマットであり、Istage, Remap を通り各 Supercell データが適切に並び替えられることを想定している。そのため

User Code の検証には安定した Istage, Remap が必要不可欠である。User Code は ADC データをそのまま Monitoring data として送る機能を持つのでそれを用いて正確に並び替えが行われ、ADC データを適切に抽出できているかを確認する。また Remap からの ADC データは Pedestal が付加されているおり FIR Filter, Saturation Detection での FIR 計算ではそれを差し引く。Pedestal を引いた ADC データは負になる可能性があり Firmware 内で正確に符号付演算が行われているかを確認する必要がある。



(a) ADC データ　　**(b)** *ADC − Pedestal*

図 6.12: ADC と ADC-Pedestal の検証。(a) により User Code の出力が完全に Model と一致することを確認した。(b) により ADC データから適切に Pedestal を差し引き、符号付きの演算が行われることを確認した。

図 6.12 の結果は single pulse のデータセットを用いて得られた結果である。User Code Model が予測した値と全ての Supercell, BCID で一致していることが確認できた。Checker 内部では 320 Supercell,3564 Bunch Crossing 全て (320 × 3564 = 1140480 のデータ数、図 6.12 ではそのうちの 11 個のみを表示している。) に対し bit by bit で Firmware の出力と Model の出力が等しいか確認している。また step function[*9]を用いて検証も行い正しく予想通りの結果が得られた。このことから Istage により適切に ADC データ, BCID, Supercell 情報が抜き出され Remap による並び替えも正しく行われていることが確認できた。

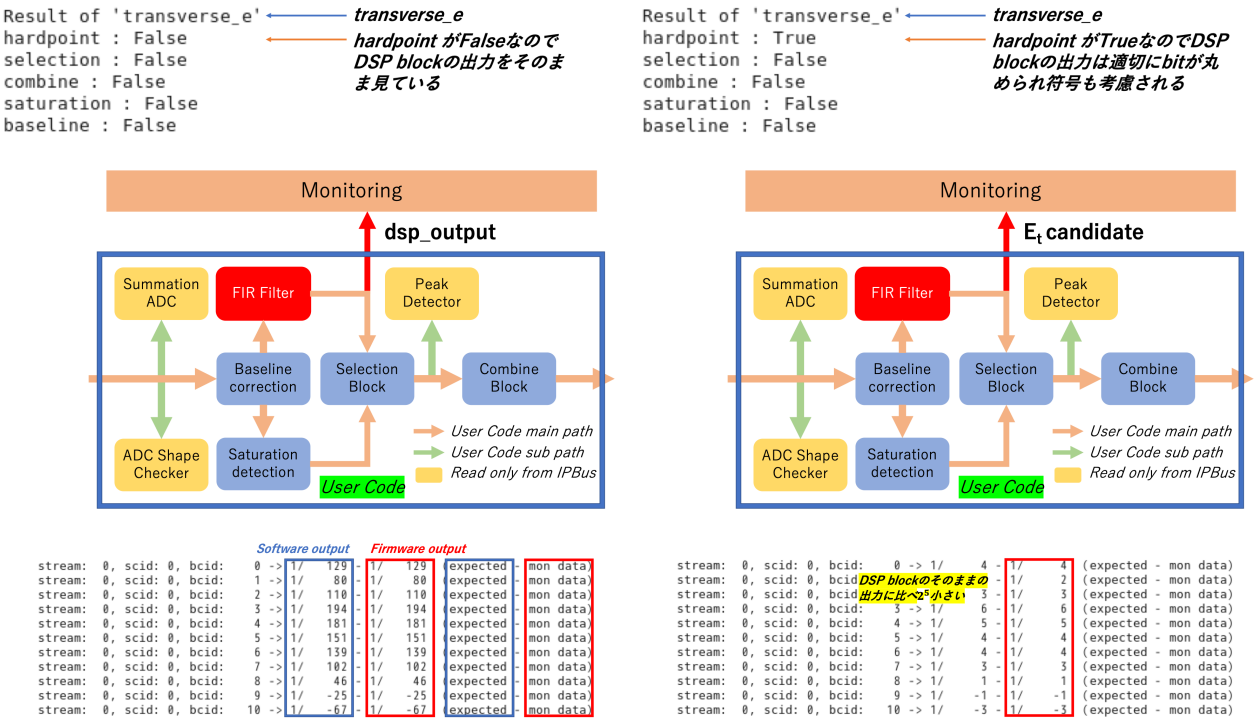
User Code は global control signals により内部の機能を変更することができる global control signals は IP bus 経由で設定することが可能であり Firmware をコンパイルし直さなくても動作を変更することが可能であった。そのため同じ Firmware を用いていくつもの検証を行うことができる。User Code の

[*9] BCID の増加に伴い ADC データが 1 ずつ大きくなる

Main path 検証では図 6.11 に示すように Monitoring data の種類と User Code Status を適宜変更させることにより各 Block の検証を行った。

FIR filter

FIR Filter の検証には DSP block が正しく動作しておりまた bit の丸めが正しく行われていることが重要である。それを確認するためにいくつかの User Code Statue 下で Transverse Energy を記録し User Code Model の予想値を比較した (図 6.13)。



(a) DSP output

(b) E_{T} candidate

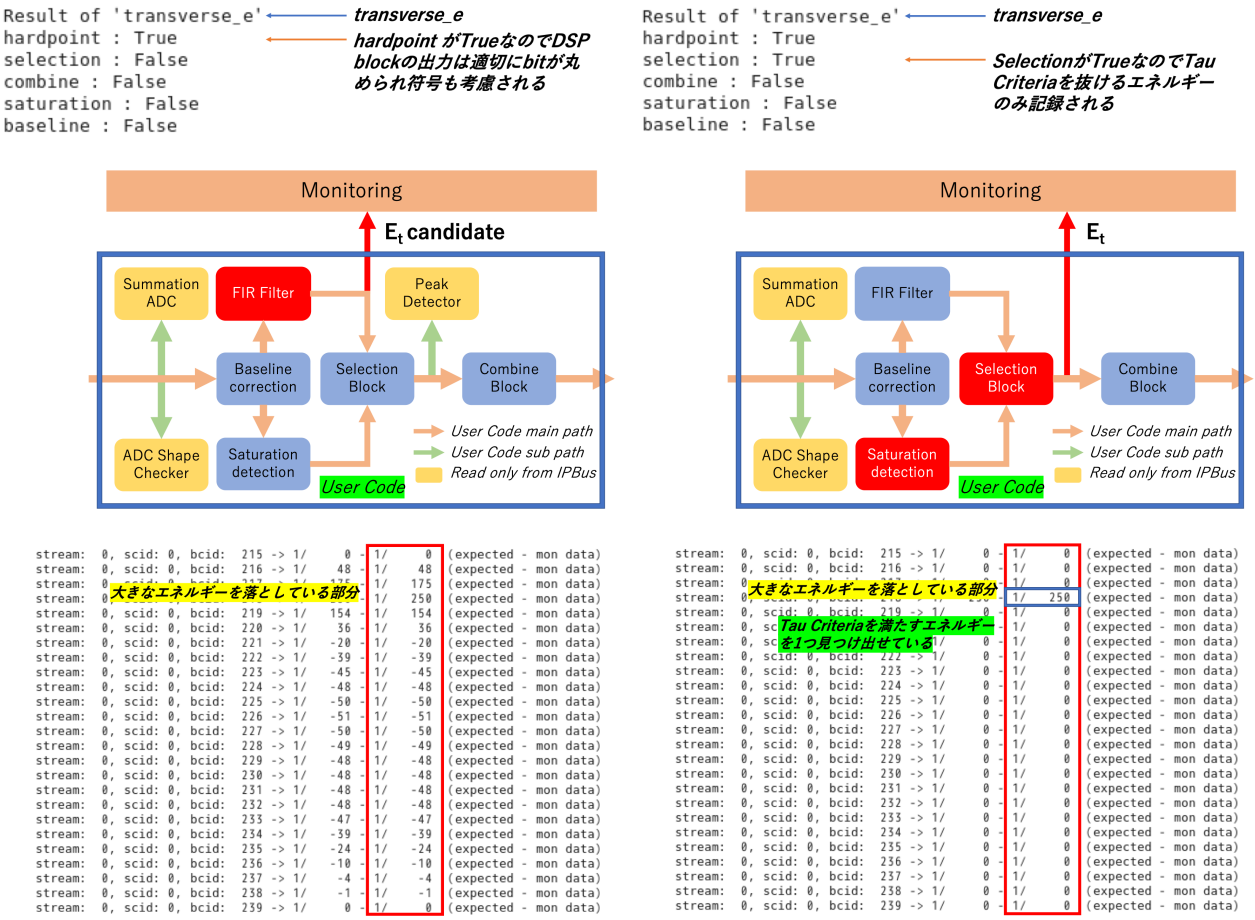
図 6.13: FIR Filter の検証。(a) は global control hardpoint を False にすることにより DSP block の乗算が適切に行われているかを確認できる。(b) は True の場合であり、適切な bit align を実行している。どちらの場合の User Code の動作と Model が一致することを確認した。

図 6.13 より FIR Filter 内部で DSP Block が適切に乗算を行えていることが確認できる。また DSP Block からの出力の bit の丸めも予想通り行われており E_{T} candidate を正しく計算出来ている。FIR Filter の検証は先ほど同様に step function を用いても行われており同様の結果が得られた。

Saturation block, Selection block

User Code は直接 Saturation Detection の出力を確認する手立てはなく Monitoring data にも Saturation Detection の出力は送らない。そのため Selection Block が正常に動作するかを確認することにより間接的に Saturation Detection の検証を行う。Tau Criteria は E_{T} candidate と $E_{\mathrm{T}}\tau$ candidate を用いて最終的な E_{T} を算出するので Selection Block の出力にあたる E_{T} が予測値と完全に一致する場合間

接的に Saturation Detection の検証を行うことができる。そのため Saturation Detection と Selection Block を同時に検証する。



(a) E_{T} candidate **(b)** E_{T}

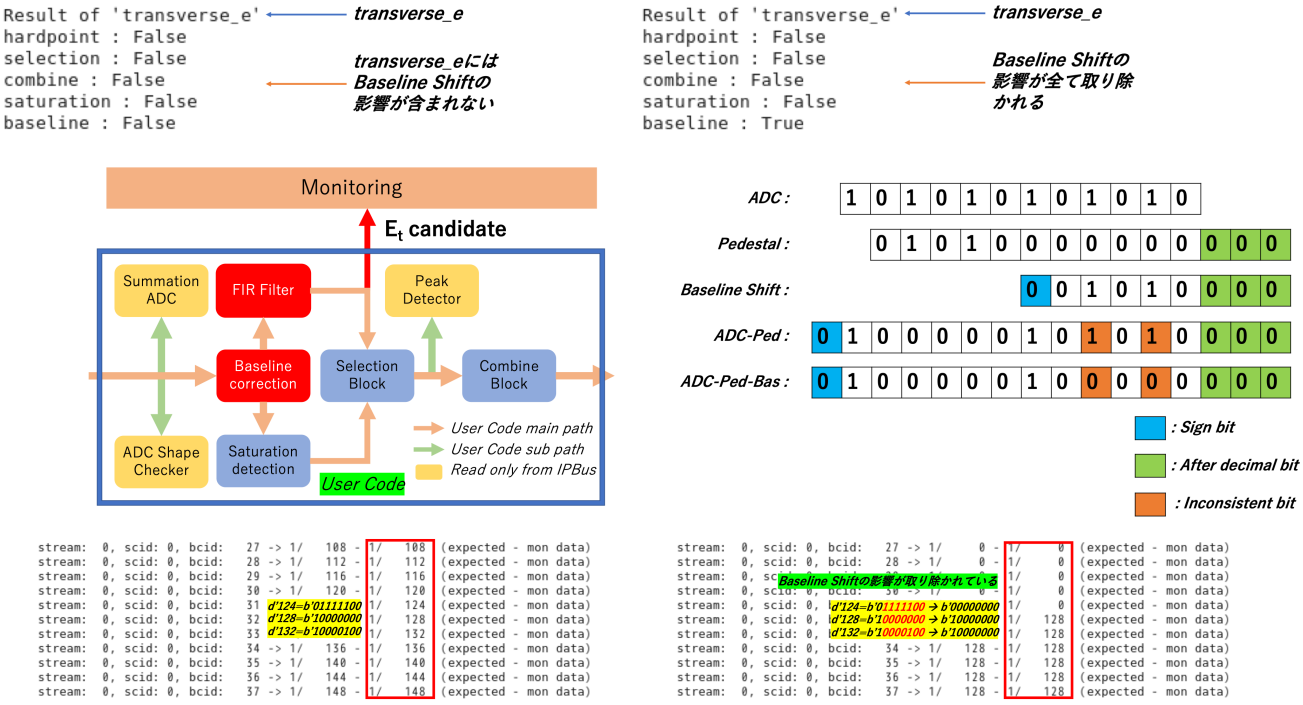
図 6.14: Saturation Detection, Selection Block の検証。(a) は global control selection を False にすることで FIR Filter の出力を直接保存する。(b) は True とすることで正しい E_{T} のみを記録する。だた一つのバンチのみでエネルギーを抜き出していることが確認できる。

図 6.14 より Selection Block が Model が予測した BCID で正しくエネルギーを出力していることがわかる。これは Saturation Detection が正常に $E_{\mathrm{T}}\tau$ candidate を算出していることをも表している。Selection Block, Saturation Detection の検証のためには Tau Criteria を抜ける液体アルゴン信号が必要であるので single pulse データを用いてのみ検証を行った。Firmware からの出力は全ての Supercell, BCID で Software と一致することを確認した。

Baseline Correction

Simulation による Baseline Correction の検証でも見たように単一のデータセットを用いてしまうと出力が全て 0 になる性質があった。なぜなら Simulation での検証では Baseline Shift 単体で Bipolar 波形を再現できる程度の bit 数を持つ必要があったため 16 bit 用意する必要があったからである。しかし

実際の LATOME Firmware 内部では Baseline Shift は符号付で 9 bit しかない[*10]ので ADC データと Pedestal の大きさによっては必ずしも $ADC - Pedestal - Baseline\ Shift$ が 0 になるとは限らない。そのためこれを利用して FIR Filter の係数を $a_i = (1, 0, 0, 0)$ と設定して E_{T} candidate を読み出すことにより Baseline Correction の検証を行った (図 6.15)。



(a) Baseline disable　　**(b)** Baseline enable

図 6.15: Baseline Correction の検証。(a) は global control baseline を False とすることで Baseline Correction による補正無しでの FIR Filter の出力を記録する。(b) では True とすることで符号 bit を除いた 8 bit 以下はすべて 0 になる。この性質から FIR Filter の出力も変化する。どちらの場合も Model の予測した値を User Code が出力することを確認した。

図 6.15 から Baseline Shift の影響が適切に取り除かれていることが確認できる。Baseline Correction は single pulse データセットを用いても検証が行い、どちらのデータセットを用いても Software での予測値と一致した値を Firmware が出力することを確認した。

6.2.2　それ以外の User Code の検証

User Code の main path 以外の Block は IP bus を用いて直接内部のレジスター, メモリの値を参照することができる。そのため Simulation 同様に Software で算出した値と Firmware の出力を比較する図 6.16 に示すようなテストベンチを作成し検証を行った。

[*10] Arria 10 のリソースの観点から 9 bit 以下出ないと実装できない

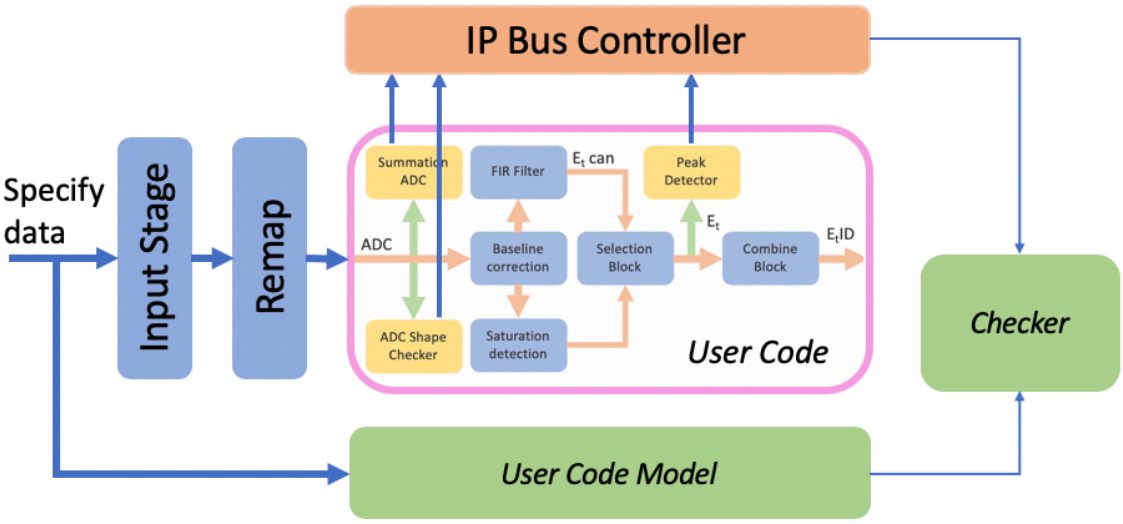


図 6.16: 実機での User Codesub path の検証の概要

現在の LATOME Firmware には Peak Detector はふくまれていない、その代わりに ADC Shape Checker がインプリメントされておりそれに準じた Firmware を用いた検証を行った。

Summation ADC

Summation ADC は Simulation 同様いかなるデータセットに対しも機能する。そのため single pulse と step function[*11]のデータセットを用いて検証を行った。

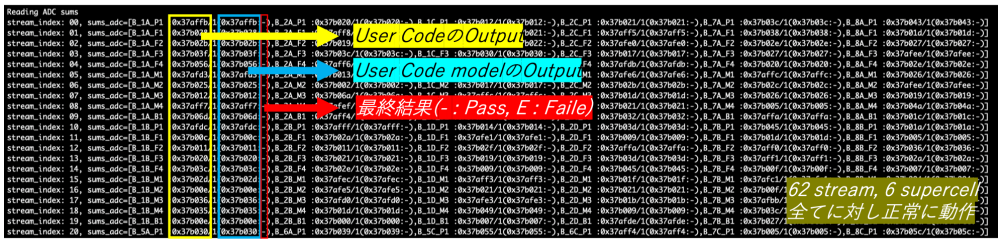


図 6.17: 実機での Summation ADC の検証結果。全 372 個の Supercell に対し値の確認を行い、全てで予測値と等しい値を出力することを確認した。

User Code Main path での ADC データを直接抜き出し確認したのと同様に LHC 1 周全てで予測と同じ ADC データを User Code が獲得できていることが確認できた。

[*11] BCID の値に合わせて ADC データが 1 ずつ大きくなる

ADC shape checker

ADC Shape Checker は 1 つの Pulse を見つけ出すことから、Simulation 同様に single pulse のデータセットを用いて検証を行った。

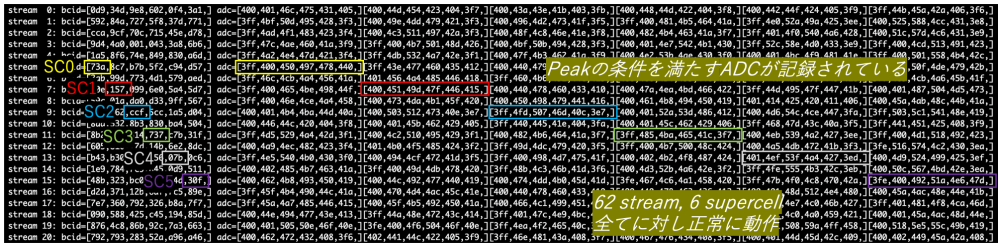


図 6.18: 実機での ADC Shape Checker の検証結果。各 Supercell で Peak を構成する 6 つの ADC データと BCID を確認した。そのため Checker 内部では 2604 (=372×7) 回の値の比較を行う。

ADC Shape Checker が正しく動作していることが確認できたので本来の目的である Supercell Mapping を確認する機構を作成中である。

6.2.3 RUN3 実験に向けた Firmware の検証

RUN3 実験では最大で 24 時間連続で LATOME Firmware を用いる可能性がある。そのため 2.01×10^{13} $(= 24 \times 60 \times 60 \text{ sec} \times 240 \text{ MHz})$ 回のクロック動作に耐える User Code を提供する必要がある。User Code 内の各 Block でエラーが起こり予測値とは異なる値を出力する可能性を p、試行回数を n とすると、エラーの起こる期待値は $\lambda = np$ と書け、ポワソン分布 $X_\lambda \sim Po(\lambda)$ に従うと考えられる。ここで $Y_n \sim \chi^2(n)$ である集団 Y_n を考えると $P(X_\lambda \leq k) = P(Y_n \geq 2\lambda)$[*12]を満たす。ただし n=2(k+1) が成り立つとする。そのためポワソン母集団 $Po(\mu)$ に従う $1-\alpha$ 信頼区間は

$$\frac{\chi^2_{2t}\left(1-\frac{\alpha}{2}\right)}{2n} \leq \lambda \leq \frac{\chi^2_{2(t+1)}\left(\frac{\alpha}{2}\right)}{2n}$$

で表される、ここで t を母集団から得られる実現値とした。個々の Block に対して信頼区間を算出する際、例えば Selection Block の検証には FIR Filter が安定動作していることが必要条件である。そのため検証を行ったデータセットの数と Block 間の関係に応じて各 Block を検証した回数も変化する。その関係を表 6.4 に纏めた。

[*12] 証明は付録 F で与える

Module	用いたデータセット	必要条件	検証回数
安定した ADC	single pulse, step function		10
FIR Filter	single pulse, step function	安定した ADC	3
Saturation Detection, Selection Block	single pulse	安定した ADC, FIR Filter	1
Baseline Correction	single pulse, step function	安定した ADC	2
Summation ADC	single pulse, step function	安定した ADC	2
ADC Shape Checker	single pulse	安定した ADC	1

表 6.4: User Code の各 Block に対する検証のまとめ

今回はすべての検証結果に対し User Code の出力にエラーが確認されなかったので式 (6.2) の t は 0 になり、95% 信頼区間を考える場合片側 5% となる領域を考えることにした。

$$\frac{\chi^2_{2t}\left(1-\frac{\alpha}{2}\right)}{2n} \quad \xrightarrow{t=0} \quad 0$$
$$\frac{\chi^2_{2(t+1)}\left(\frac{\alpha}{2}\right)}{2n} \xrightarrow{t=0,\frac{\alpha}{2}=0.05} \frac{\chi^2_2\left(0.05\right)}{2n} = \frac{5.99}{2n}$$

各 Supercell あたり LHC 1 周に対応する 3564 個のデータを比較することにより各 Block の検証を行ったので一回の検証が $n = 3564$ に対応する。そのため各 Block の信頼区間は表 6.5 と纏められる。

Block	n	$\lambda_{90\%}$	$\lambda_{95\%}$	$\lambda_{99\%}$
安定した ADC	3564×10	$\leq 6.47 \times 10^{-5}$	$\leq 8.40 \times 10^{-5}$	$\leq 1.29 \times 10^{-4}$
FIR Filter	3564×3	$\leq 2.16 \times 10^{-4}$	$\leq 2.80 \times 10^{-4}$	$\leq 4.31 \times 10^{-4}$
Saturation Detection, Selection Block	3564×1	$\leq 6.47 \times 10^{-4}$	$\leq 8.40 \times 10^{-4}$	$\leq 1.29 \times 10^{-3}$
Baseline Correction	3564×2	$\leq 3.23 \times 10^{-4}$	$\leq 4.20 \times 10^{-4}$	$\leq 6.46 \times 10^{-4}$
Summation ADC	3564×2	$\leq 3.23 \times 10^{-4}$	$\leq 4.20 \times 10^{-4}$	$\leq 6.46 \times 10^{-4}$
ADC Shape Checker	3564×1	$\leq 6.47 \times 10^{-4}$	$\leq 8.40 \times 10^{-4}$	$\leq 1.29 \times 10^{-3}$

表 6.5: 各 Block におけるエラー回数の信頼区間

達成すべきエラーの信頼区間は各 Block が 2.01×10^{13} 回のクロック動作に耐える必要があるので、$\lambda < 4.98 \times 10^{-14}$ でる。そのために必要な試験回数は少なくとも

$$\frac{\chi^2_{2t+2}\left(\frac{\alpha}{2}\right)}{2n} < 4.98 \times 10^{-14} \xrightarrow{t=0,\frac{\alpha}{2}=0.05} n > 6.01 \times 10^{13}$$

となる。Monitoring data は 10 Gbps でデータを記録することができる、各 Monitoring data はおおよそ 2.0×10^5 bit で一つを構成している (図 6.3)。そのため最速で 5.0×10^4 Hz で保存することができる。一つの Monitoring data には 32 Bunch Crossing 分のデータが含まれており、全ての Supercell に対し 1.6×10^6 Hz でデータを記録できることから、目標の試行回数を実現するには 6.01×10^{13} / $6 \times 1.6 \times 10^6 = 6.26 \times 10^6$ 秒 (= 72 日) Monitoring data を記録し続ける必要がある。IP bus は 1 Gbps でデータを読み出すことができるが、それらの計算結果 LHC 1 周の間に計算された結果であり最大で 1.12×10^4 Hz

の頻度で読み出し結果を確認することができる。そのため 5.34×10^9 秒 IP bus を用いてデータを読み出す必要がある。

Monitoring data を用いた User Code Main path の検証に限って言えば信頼区間を 95% に設定し 72 日間データを記録し続ける必要があるが現実的ではない、そのため異なる方法も考案した。

現在 Injector Firmware はデータを LATOME Firmware に伝送する機能しか持たせていないがそれに Checker Firmware を搭載することが考えられる。仮に User Code の出力を 3564 バンチ全てに対し Checker Firmware 内で 240 MHz で比較することが可能であれば最速で約 25 分で達成することができる計算になる。[*13] この機構を用いた検証は Arria 10 が搭載された評価ボードを用いて行った。この評価ボードに搭載されている FPGA(10AX115S2F45I1SG) は LATOME Board に搭載されているもの (10AX115R4F40I4SGES) と型番が異なるが同じ Arria 10 シリーズであり ALM の構造も同じなので Firmware の検証を行うことが可能である。詳細は付録 E に纏めた。

実機を用いた検証により Monitoring data を用いで User Code の安定性を確認する手法を構築した。User Code は LATOME Firmware の中でも ADC データをエネルギーに変換する Module でありその重要度は Level 1 trigger 全体を通しても非常に高い。Monitoring data は全ての Supercell と BCID に対し値を取得できるので、それぞれの計算結果を確認しながら検証できるという利点がある一方、十分な検証に多くの時間がかかる欠点がある。Checker Firmware を用いた検証では 240 MHz の頻度で出力を比較するため高速に検証を行うことができるが計算結果を確認できない欠点がある。そのため User Code の安定性を実機を用いて完全に証明するには Monitoring data, IP bus 読み出しを用いたものと Checker Firmware を用いたものを組み合わせることにより達成する。

6.2.4 実機を用いた検証のまとめ

以上で述べた機構はすべて私が主導的に作成したものであり、実機を用いた詳細な検証を可能にした。Simulation を用いた検証では出力を確認する Checker は Firmware で作成し、全てのロジックとレイテンシーを検証した。一方実機の検証では Checker は Software を用いて作成し、Firmware の出力を個々に調査することを可能にした。これにより User Code のすべての Block に対し RUN3 と同等の環境下で適切に動作することを確認した。

[*13] 試験回数は各 Supercell に対し 6.01×10^{13} 回以上、User Code に入る信号は 240 MHz で動作するが 6 つの Supercell を同時に扱うことから各 Supercell は 40 MHz で試験することに対応する。そのため $\frac{6.01 \times 10^{13}}{4.0 \times 10^{10}} = 1.50 \times 10^3$ 秒となり、おおよそ 25 分程度である。

第 7 章

Demonstrator

液体アルゴン検出器からの信号は三角波でありそれを Bipolar 波形に変換して LATOME Firmware で処理をする。Bipolar 波形の重要な特徴として Pedestal を差し引いた後の波形の時間積分が 0 になるという性質があった。液体アルゴン検出器からの波形の長さは 600 ns 程度とバンチ間隔の 25 ns に比べかなり長く、それらが重なり合うことによりプラスの部分とマイナスの部分がうまく打ち消しあうようにデザインされている。そのため LHC 全てのバンチに均等に陽子が詰められてさえいればそれぞれのバンチから来るパイルアップ, ノイズの影響はほとんど等しいものなのでそれらを無視した信号の読み出しを行うことができる[*1]。しかし実際には LHC 3564 バンチあるうちの 2700 バンチ程度にのみ陽子が詰められておりその構造はトレイン構造と呼ばれている。この状況ではパイルアップやノイズは完全に打ち消されない。限られたバンチに理想的な Bipolar 波形 (図 7.1) が等しく影響すると仮定すると図 7.2 のような複雑な波形が得られる。このような Pedestal からのズレは Baseline Shift と呼ばれ、液体アルゴン特有の現象である。すでに 5.1 節で述べたように Firmware 内部ではこれらを補正する機構が実装されている。

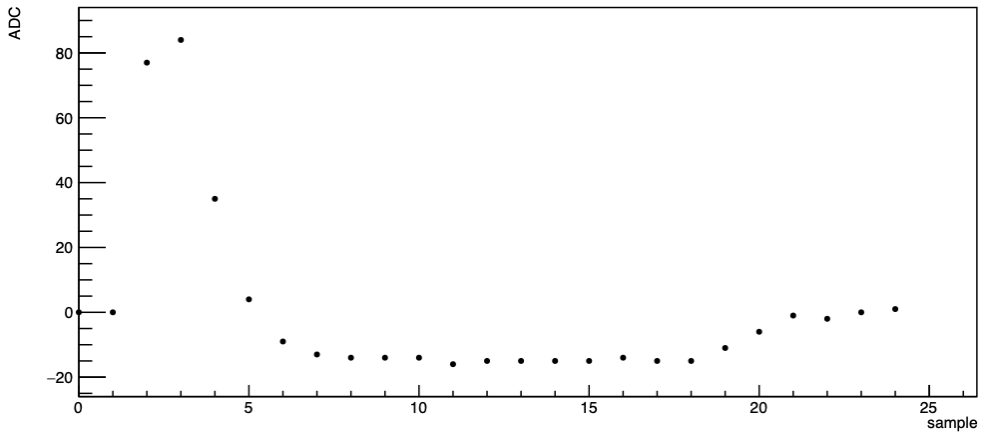


図 7.1: 理想的な Bipolar 波形。2.5 GeV のエネルギーに対応する

[*1] ノイズ, パイルアップもその影響の大きさに依存した Bipolar 波形を残すのでそれらが全て足し合わされると全てのバンチに対しおおよそ同程度の振る舞いになると考えられる。

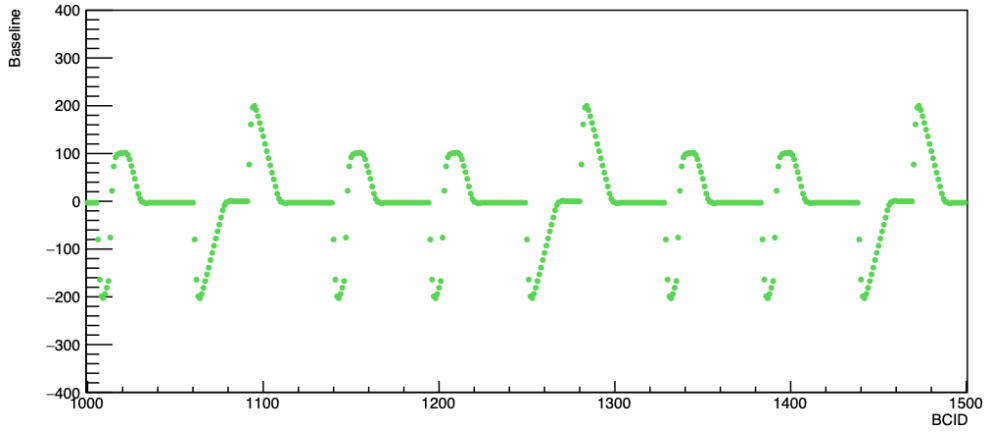
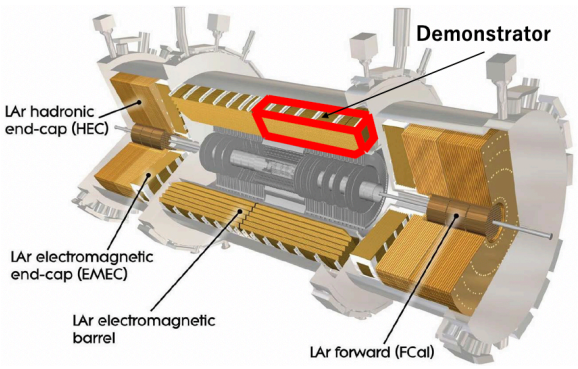


図 7.2: Simulation で得られた Baseline Shift。用いたトレイン構造は以下で見るように RUN2 実験で実際に得られたものである。

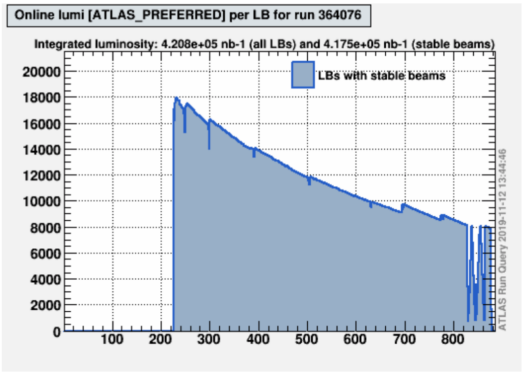
図 7.2 に示す結果は、実際に RUN2 で用いられたトレイン構造を用いてノイズなどが各バンチに 2.5 GeV の信号を残すと仮定した場合に得られた結果である。Baseline Shift はトレイン構造と大きく関係しており、またその形状は各 Supercell の Bipolar 波形の形状に依存している。

Baseline Shift は ADC の平均として算出され ADC は最大で 4096 まで値を取る可能性があることから、計算上は Baseline Shift も同様に 4096 ADC count までカバーするといかなる場合にも対応できる。しかし FPGA のリソースの観点からできる限り少ない memory bit で実装するべきであり、LATOME Firmware では符号付、1/8 ADC count の精度で 32 ADC count までカバーしており 9 bit で実装することにした。そのため Baseline Shift が RUN3 の環境下で 32 ADC count 以下で計算できることを保証する必要がある。

図 7.3 に示す Domonstrator は RUN2 時に $\Delta\phi \times \Delta\eta = 0.4 \times 1.4$ の領域に設置されていた検出器でそれを用いて得られたデータを使って RUN3 からの新しいトリガー読み出し機構の開発を行っている。RUN2 で用いられていたトリガー機構とは干渉せず RUN3 からの Supercell によるデータの読み出しを



(a) Domonstrator の領域



(b) 瞬間ルミノシティーの変化

図 7.3: RUN2 における Domonstrator と瞬間ルミノシティー

行っていた。以降は Domonstrator で得られた RUN2 実データを用いて Baseline Shift の検証に関する内容である。RUN2 での重心系エネルギーは 13 TeV、最大瞬間ルミノシティーは 2.0×10^{34} $\mathrm{cm}^2\mathrm{s}^{-1}$ 程度であり RUN3 と非常に近い状況で得られたデータを用いた解析を行うことができる。

7.1 データ処理方法

Domonstrator データは LATOME Firmware 内で処理された ADC を Monitoring data として読み出し保存したものである。Monitoring data は Level 1 Accept の発行に応じて 32 Bunch Crossing 分のデータを保存するものであった。今回用いたデータは 2018 年 10 月 21 日 (RUN ID : 364076) に実際に Physics RUN で得られたデータであり、Random trigger (表 7.1) を用いて取得された。

Trigger Type	特徴
ZeroBias	いかなる BCID にも Level 1 Accept をランダムに発行する
Random	陽子が詰められている BCID の内ランダムに Level 1 Accept を発行する

表 7.1: Trigger Type

Domonstrator データは 1 分ごとのデータにまとめられ 1 分あたりおおよそ 5000 個の Monitoring data で構成されている。1 つの Monitoring data は 32 BCID を一度に記録していることから 1 分あたり 160000 Bunch Crossing 分のデータで構成されている、そのため 1 BCID あたり 160000/3564 $\simeq$ 45 個のデータを持つことになる。Baseline Shift は各 Supercell, Bunch Crossing に依存するのでそれぞれに対し ADC の平均を算出し Baseline Shift とした。1 分あたりのデータを元に Baseline Shift を計算すると ADC の標準偏差が 1 count 程度だとすると平均のエラーは

$$\sigma\left(Baseline\ Shift\right) = \frac{\sigma\left(ADC\right)}{\sqrt{45}} \simeq 0.15$$

となり Baseline Shift を調査するには大きすぎる (図 7.4)。

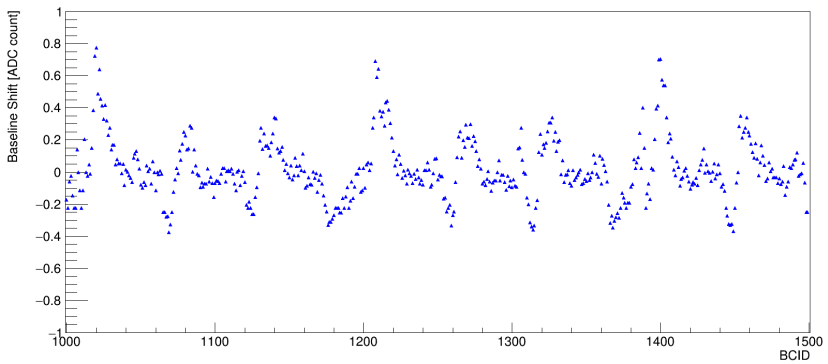


図 7.4: 1 分間分の Domonstrator データによる Baseline Shift。分散が大きく綺麗に Baseline Shift の影響を確認できない。

そのため 11 分間分のデータをひとまとまりにし、1 Bunch Crossing あたり平均 500 個の ADC の平均を Baseline Shift とした。瞬間ルミノシティーはその間に大きく変化することはなく安定していることを確認しており Baseline Shift はその間安定しているはずである。こうすることにより Baseline Shift の標準偏差は

$$\sigma\left(Baseline\ Shift\right) = \frac{\sigma\left(ADC\right)}{\sqrt{500}} \simeq 0.045$$

程度となり十分 Baseline Shift の検証を行える (図 7.5)。

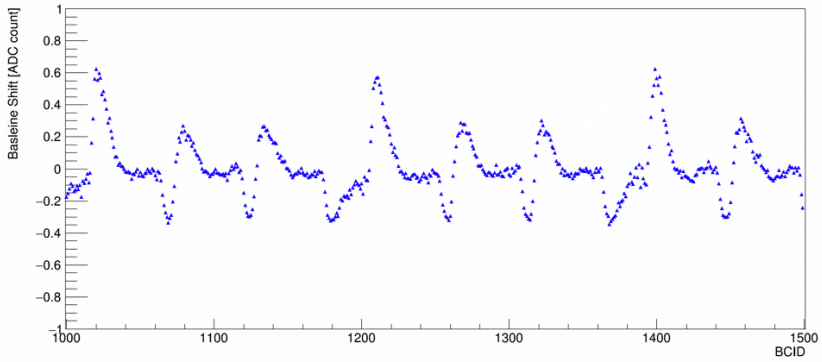


図 7.5: 11 分間分の Domonstrator データによる Baseline Shift。図 7.4 に比べ詳細に Baseline Shift を確認できる。

LHC-ATLAS 実験で用いられる BCID は検出器全体で同一のものを用いるべきであるが LATOME Firmware を開発中に同時並行してデータを記録していたことからそれらは一致していない。そのため何らかの方法でそれらを合わせる必要がある。LHC のトレイン構造はその RUN に依存しており一意に決まっている訳ではなく RUN Query[*2]にまとめられている。しかし全ての RUN で Abort Gap と呼ばれる長い Bunch Crossing 間 (200 BCID 程度) に陽子が詰められていない箇所が存在する。その領域ではパイルアップなどが起こらないことから Baseline Shift はほとんど起こらないと考えられる。また Random Trigger を用いて取得されたデータであるので Abort Gap 間では Level 1 Accept が発行されないのでサンプル数が他と比べ少なくなる (図 7.6)。なのでこれを利用し Domonstrator データと RUN Query の BCID を合わせた。

[*2] https://atlas-runquery.cern.ch

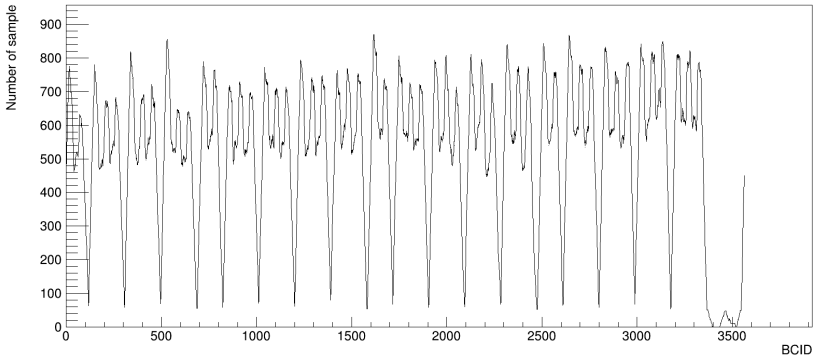


図 7.6: 各 BCID における ADC のサンプル数。BCID 0 - 3400 ではある程度の規則でサンプル数が増減しているが BCID 3400 - 3500 程度では他に比べサンプル数が著しく少ない。

Baseline Shift はトレイン構造 (図 7.7) の影響によるものでありトレインの始めと終わりが大きく Shift するものであると予想される。Bipolar 波形はおおよそ 600 ns の波長であり Bunch Crossing 間隔は 25 ns であるので 24 BCID 分の波長であるのでその範囲内の Baseline Shift が大きく振れるはずである (図 7.8)。

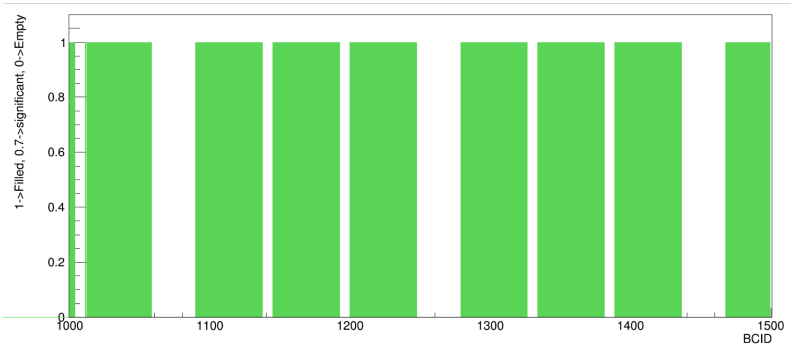


図 7.7: LHC のトレイン構造。緑色の BCID にのみ陽子が詰められる

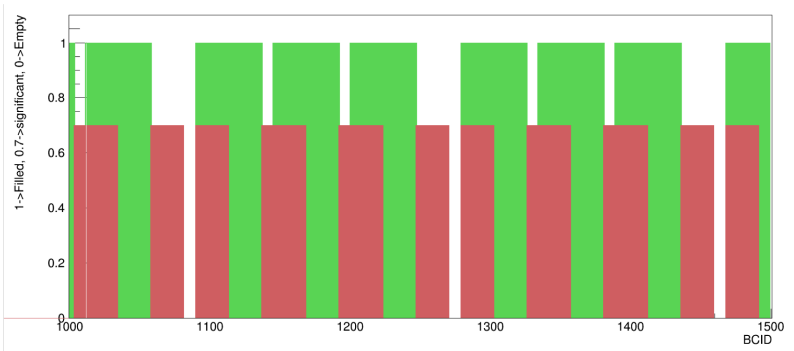


図 7.8: LHC のトレイン構造と Baseline Shift において重要な BCID、赤の BCID は Baseline Shift を考える上で重要な部分に対応する重要な BCID = トレイン構造の初めの 24 BCID | トレイン構造終わり後の 24 BCID を条件としている。

そのため RUN Query から取得したトレイン構造の情報を Baseline Shift と組み合わせ実際に Baseline Shift の振り幅が大きな部分がトレインの始めと終わりにあたるのかを確認した。つまり BCID のズレの補正後 Baseline Shift の振れ幅が大きな部分がトレインの始めと終わりに対応する BCID であれば良い (図 7.9)。

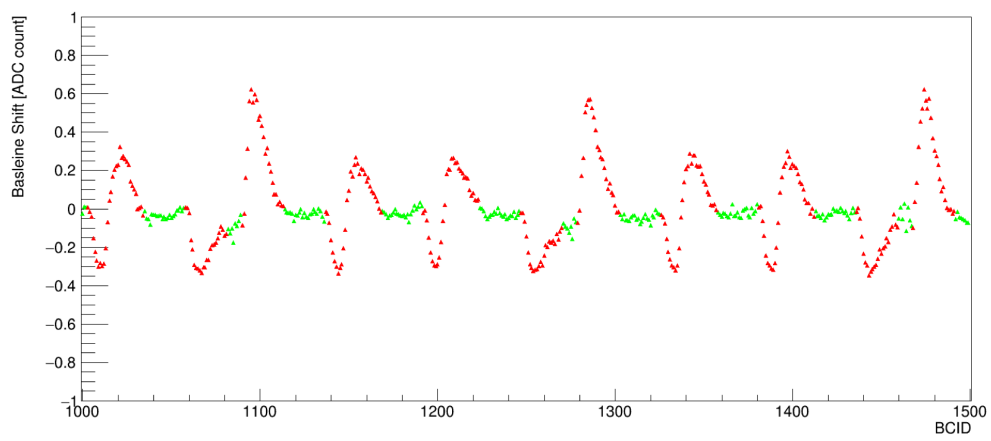


図 7.9: Domonstrator データの BCID の補正。赤点がトレイン構造の始めと終わりに対応し、その部分が特に大きな振幅を持つことが確認できる。

こうして Domonstrator で得られた ADC を元に正しい Baseline Shift を得ることができ、Baseline Shift は予想通りトレイン構造の始めと終わりで大きく振れることが確認できた。Simulation では 2.5 GeV の信号を仮定して得られた結果で Baseline Shift の最大振幅はおおよそ 200 ADC count であった。実際に RUN2 データを用いた解析で得られた結果は約 0.6 ADC count であるので各バンチはノイズ, パイルアップの影響として

$$\frac{0.6}{200} \times 2.5 \text{ GeV} = 75 \text{ MeV} \tag{7.1}$$

程度の影響を持つと予想できる[*3]。

7.2　ルミノシティー依存性

Baseline Shift はパイルアップやノイズなどの影響により Pedestal からのズレが生じるものであった、パイルアップは瞬間ルミノシティーの減少に伴いその影響は小さくなると考えられるので、Baseline Shift もそれに依存して小さくなると考えられる。そのため Domonstrator データ 11 分間分のデータをひとまとまりにしたものを 8 セット用意し、6 分目の瞬間ルミノシティーをそのデータセットの代表値として、Baseline Shift の瞬間ルミノシティーに対する変化を確認した。ルミノシティー依存性のためには、Baseline Shift の絶対値の和をバンチ数で割り、振れ幅の平均を用いて比較するやり方と最も大きな振れ幅を持つバンチを比較する方法があった。前者の方は単純に絶対値を足し上げるだけであり容易では

[*3] Simulation で用いた Bipolar 波形と実際の波形は AREUS で生成された 2.5 GeV の信号であり、実際に検出器からの信号とは異なる。また Bipolar 波形は各 Supercell に依存の形であり、FIR Filter で用いる係数も異なることから一概には言えないが、おおよそ 75 GeV 程度の影響を持つと予想できる。

あるが、ノイズの影響を直接受けてしまい純粋な Baseline Shift の影響を確認し難い。そのため後者を採用した (図 7.10)。RUN Query の BCID と Domonstrator データの BCID を合わせた理由は主にこのためである。

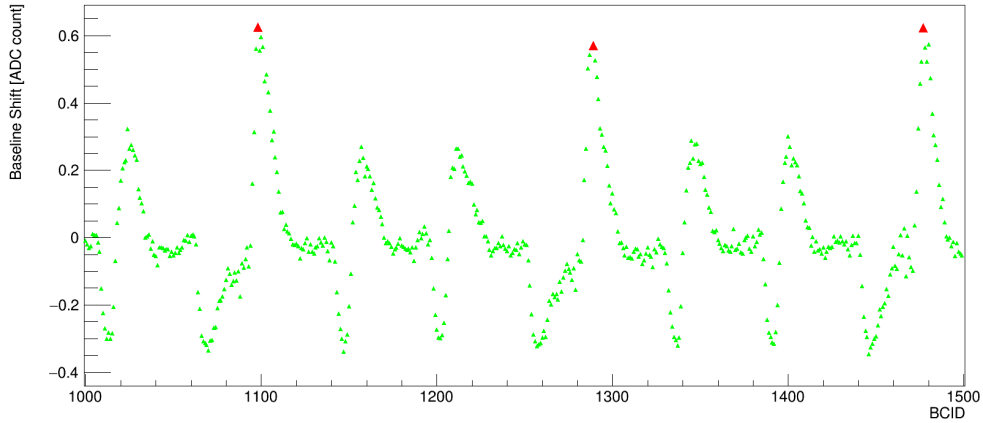


図 7.10: ルミノシティー依存性の調査の際に用いた点。プラスの方向に振れ幅の大きい赤点をその瞬間ルミノシティーの代表点とし、それらの平均を算出する。

大きな振れ幅を持つ Baseline Shift は今回用いた Physics RUN の中に 18 点存在し、それらの平均をそれぞれの 11 分間のデータセットごとに算出した。こうすることによりそれぞれのピークにたまたま乗ってしまったノイズの影響を小さくすることができる。それらを図 7.11 にプロットしまとめた。

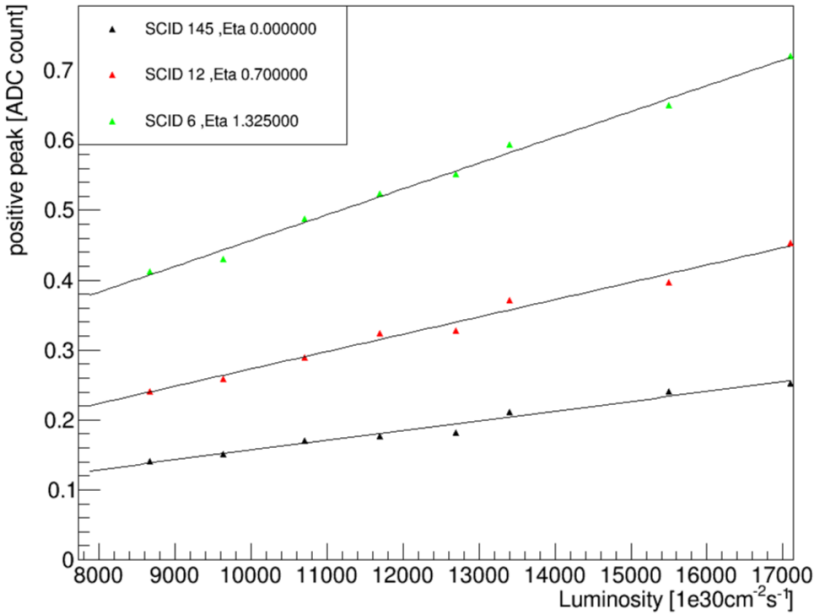


図 7.11: Middle layer における Supercell の Baseline Shift の瞬間ルミノシティー依存性、いかなる η の Supercell に対し瞬間ルミノシティーに対し良い線形性を持つ。

以上の結果から Baseline Shift は瞬間ルミノシティーに対し良い線形性をもつことが確認された。瞬間ルミノシティーは平均相互作用数に対し良い線形性を持ち、パイルアップやノイズの影響は平均相互作用に対し線形であるので、それらの重ね合わせである Baseline Shift が瞬間ルミノシティーに対し線形であるのは整合性の取れた結果である。Bipolar 波形の波高はエネルギーに比例し、RUN3 実験では 13.5(14) TeV での実験が行われる。また最大瞬間ルミノシティーは 2.0×10^{34} $\mathrm{cm}^{-2}\mathrm{s}^{-1}$ 程度であるので、今回得られた結果から RUN3 環境下でも Baseline Shift は最大 1 ADC count 程度であることが確認された。

7.3 η 依存性

Domonstrator は $0 \leq \eta \leq 1.4$ の範囲に設置されていたので Barrel 領域での Baseline Shift の η 依存性も確認することができる。η 依存性に対してもルミノシティー依存性同様 Baseline Shift の振れ幅の大きい BCID にのみ注目し確認した。

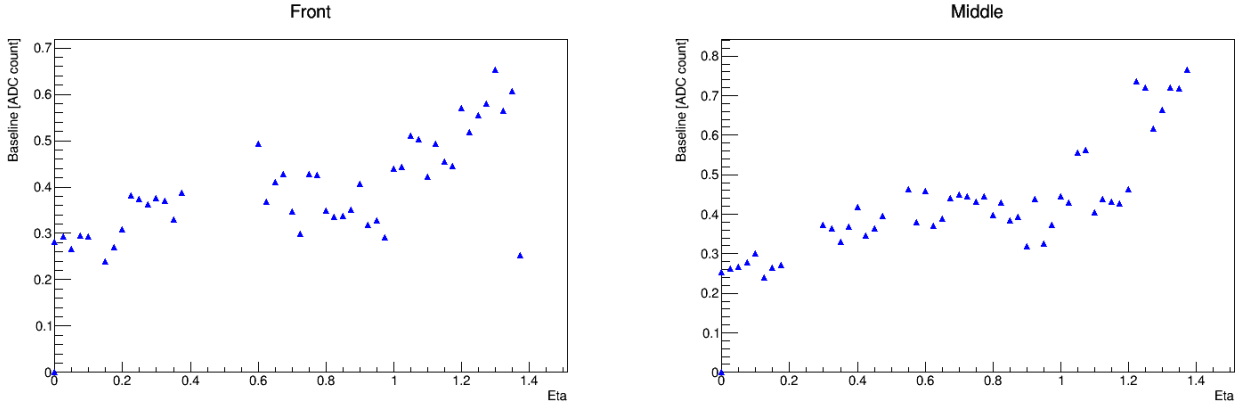


図 7.12: Baseline Shift の η 依存性、Supercell は 4 層構造 (Presampler, Front, Middle, Back) をしているが Presampler と Back layer は $\eta = 0.1$ おきに設置されている。その 2 層のデータは取得時安定しておらず η 依存性を確認するために Front layer, Back layer の 2 層のみ注目した。

図 7.12 から Barrel 領域では η が大きくなるにつれて Baseline Shift の振れ幅も大きくなる特徴があることが確認された。次に各 Supercell に対して ADC で算出した Baseline Shift をエネルギーに換算する。その変換因子は [36] から算出した。それらの変換因子を η に対しプロットしたものが図 7.13 である。

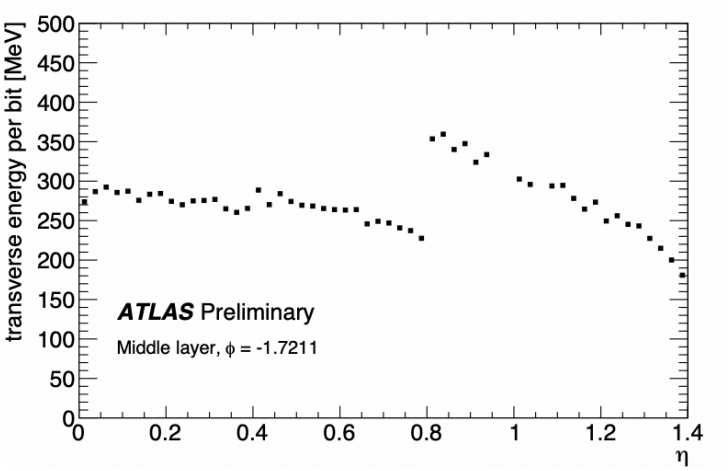


図 7.13: Middle layer における 1 ADC count に対するエネルギー [36]。

各 Supercell に対し ADC ベースで算出された Baseline Shift に変換因子を掛け合わせることにより各 Supercell が持つ Baseline Shift によるエネルギーのズレの最大値を算出することができる (図 7.14)。

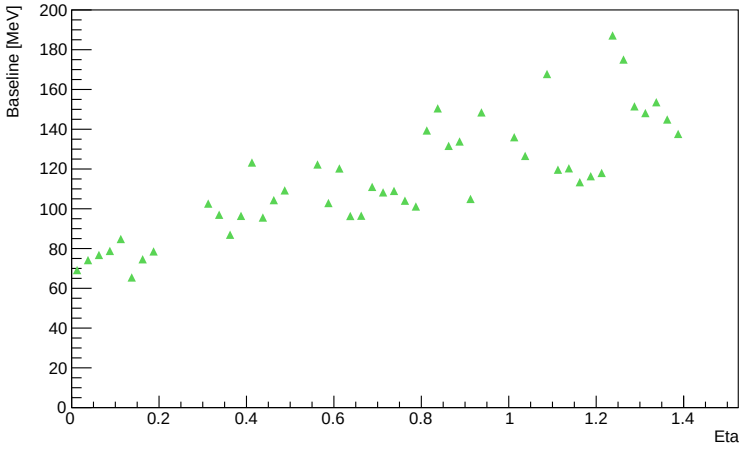


図 7.14: エネルギーに換算された Baseline Shift

エネルギー換算された Baseline Shift は式 7.1 の予測値と近い値であり、η の絶対値が大きくなるにつれて大きくなる傾向ある。LHC リングに対し時計回り、反時計回りに加速された陽子は同じエネルギーをもち、それらが検出器中央で衝突を起こす。そのためビームの進行方向に多くのノイズを残す。そのため η の増加に伴い、エネルギー換算された Baseline Shift も増加する。特に $|\eta| < 4.9$ までをカバーする FCal での Baseline Shift は Barrel 領域に比べ非常に大きな影響を持つと考えられる。Baseline Shift は Supercell によって変化する Bipolar 波形の形に依存するので、その振る舞いは実際にデータを取得することで調査する必要があり、今後の課題である。

第 8 章

結論と今後に向けて

8.1 User Code の開発・検証

RUN3 からはトリガー読み出し機構が刷新され RUN2 の Trigger Tower に比べ 10 倍細かい Supercell での読み出しを行う。Supercell でのトリガー読み出しを導入することにより、エネルギー分布の情報をトリガーに用いることができ、効率的に興味ある事象のみを選択することができる利点がある。しかし Supercell 読み出しに更新することにより検出器全体では 25 Tbps にもなるデータ量を常に処理しなければならない。Supercell の信号はデジタイズされた後 LATOME Firmware 内で ADC データからエネルギーを算出される。User Code は LATOME Firmware 内で各 Supercell に対しエネルギーを算出する Block であり東大グループが作成を担当している。Level 1 trigger はエネルギー情報を用いてトリガーを発行するので非常に重要な Firmware である。液体アルゴン検出器はその特徴から入射粒子の落とすエネルギーが Baseline Shift の影響により User Code でエネルギーを正確に算出できない。そのためそれらをすべて補正する機構を作成した。この機構は大量のリソースを用いるため実装が非常に難しかった。しかし FPGA に実装可能なリソース量にするためにより少ない memory bit 数での実装を考案し全ての Supercell, BCID に対し補正を可能にした。これにより User Code の Main path に関する Firmware の最終版を完成させた。また LATOME Firmware 内部でのデータをモニタリングする機構を作成し、IP bus を用いてそれらを読み出すことを可能にした。これにより LATOME Firmware 全体のモニタリングシステムを強化した。

User Code は Level 1 Accept 発行に関与した Firmware であるため決まったレイテンシーでの処理が要求される。また LATOME Firmware は大規模な Firmware であり全ての Supercell データを処理することから安定した動作を保証する必要がある。そのため Simulation による検証と実際の LATOME Board を用いた RUN3 環境下と同じ状況での実機での検証を行い、User Code の各機能と動作安定性を確認した。Simulation での検証を用いることにより User Code 内部の Block に実装されたロジック、レイテンシーを様々なデータセットを用いて確認することが可能になった。これは Firmware の開発に実際に用いられており、安定した Firmware の提供のために用いられている。また実機での検証を行うことにより LATOME Board を用いて Monitoring data, IP bus 読み出しを利用した User Code の動作確認を行なった。これにより各 Block での処理による出力を一つづつ比較することにより動作を確認することが可能となった。User Code の動作安定性はさらに Checker Firmware を利用した機構を用いて、それ

らを組み合わせることにより達成する。User Code はエネルギー再構成のために 2 つの異なるアルゴリズムを持ち、それぞれの検証のためには多くのデータセットを必要とする。今回用いたデータセットは数種類であり、より詳細な検証を行うことは今後の課題である。また Saturation した波形を用いた実機での検証は行われておらず、これも今後の課題である。本研究により User Code Firmware の最終版を完成させ、それらの詳細な検証を自動で行うことを可能にした。今後は Firmware 全体の動作安定性の検証のために統合試験を行い RUN3 実験までに安定した Firmware を提供する。

8.2　Demontrator data を用いた Baseline Shift の調査

Firmware 内では Baseline Shift を 32 ADC count 以内で表せることができると仮定し、それに符号 bit と 1/8 ADC count の精度を付け加え 9 bit で計算すると決めた。それにより Arria 10 内に実装可能なリソース量に削減することが可能になった。そのため、その仮定が正しいことを確認する必要があった。Domonstrator データは RUN2 実験で実際に取得されたデータであり重心系エネルギー 13 TeV、瞬間最大ルミノシティー $2.0 \times 10^{34}\mathrm{cm}^{-2}\mathrm{s}^{-1}$ 程度と RUN3 環境に非常に近い状況で得られたデータである。それらを用いて Baseline Shift を算出した結果 Supercell の検出器内での位置にも依存するが Barrel 領域では最大でもおおよそ 1 ADC count 程度であることが確認された。RUN3 では重心系エネルギー 14(13.5) TeV, 最大瞬間ルミノシティーが RUN2 後半同様 2.0×10^{34} $\mathrm{cm}^{-2}\mathrm{s}^{-1}$ 程度であるので現在の Firmware で十分対応することができることを確認した。Domonstrator データは液体アルゴン検出器の Barrel 領域からのデータであるので、その領域ではさらに Baseline Shift の bit 数を減らし更なる Firmware の安定動作を目指すことも考えている。エネルギー換算された Baseline Shift は η が大きくなるにつれて同様に大きくなる傾向ある。特に FCal は η は 4.9 までをカバーしており Baseline Shift の影響が非常に大きいと考えられる。それらの領域で Baseline Shift を 32 ADC count までで表現できることを示すには実際にデータを取得し、振る舞いを確認する必要があり、これは今後の課題である。

謝辞

本研究を進めるにあたり多くの方々の助言や助けを得ることができました。指導教官である田中純一教授、江成祐二助教には毎週の研修室ミーティングの際に的確な助言を賜りました。また毎回の学会発表や液体アルゴングループ内部の研究進捗会などのスライドの添削、発表練習なども休日にも関わらず付き合って頂き自信を持って発表することができました。同研究室の先輩にあた山崎友寛氏・陽易霖氏・井口竜之介氏・宇野健太氏・松澤暢生氏には CERN での研究に関することから CERN での生活面でもお世話になりました。特に井口竜之介氏には Bipolar 波形の詳しい性質や Optimal filtering について詳しく教えて頂きました。宇野健太氏には System Verilog, VHDL の書き方から Simulator, Compiler の使い方、User Code Firmware 内での Optimal filtering の実装など Firmware の基礎を全て教えて頂きました。松澤暢生氏には ROOT の使い方から Monitoring data の扱い方、データの構造まで教えて頂きました。

LATOME Firmware の開発にあたりプロジェクトリーダーである George Aad と Nicolas Chevillot にも多くの助言を頂きました。特に Nicolas Chevillot とは毎日のように議論し Firmware だけでなく UVVM などの強力なツールの扱い方、LATOME Board の扱い方など多岐にわたり助言を賜りました。Nordin Aranzabal Barrio には Monitoring data 取得の方法や実際のデータのデコード方法について事細かに教えて頂き、Alexis Vallier には qualification task の Supervisor ということもあり研究の進め方などについて多くの助言を頂きました。

研究室の同期である舘野君とは液体アルゴン検出器についての議論から学会の宿の手配まで多くのことについてお世話になりました。同じく ICEPP の同期である大矢君、梶原君、豊田君、茂木君、野内君、山田君とは大学の授業や学会などで一緒になることが多くよく一緒にご飯などに行きました。また ICEPP 秘書の塩田雅子様、竹本美智子様、加瀬由美様、金子苑子様には CERN での出張の手続きなどをはじめ多くの事務作業を行なって頂き研究に集中することができました。最後に地元から遠く離れても常にサポートしてくれて、どんなことに対しても背中を押し続けてくれた家族に感謝します。多くの方々のおかげで修士 2 年間を充実したものにすることができました、ありがとうございました。

参考文献

[1] TEXample.net, Standard model of physics, http://www.texample.net/tikz/examples/model-physics/

[2] The ATLAS Collaboration, The ATLAS Experiment at the CERN Large Hadron Collider, JINST 3 (2008) S08003

[3] ATLAS 実験のホームページ, https://atlas.cern

[4] 高エネルギー加速器研究機構, ATLAS Detector, https://www2.kek.jp/ipns/ja/release/atlas-press/

[5] L.Fiorini, C.Sänche, Measurement of Higgs boson properties in the diphoton decay channel and a search for di-Higgs production in the $\gamma\gamma$, bb final state with the ATLAS detector, ERN-THESIS-2018-269

[6] The ATLAS Collaboration, Study of the material of the ATLAS inner detector for Run 2 of the LHC, 2013 JINST 12 P12009

[7] CERN PhotoLab, Computer Generated image of the ATLAS calorimeter, https://cds.cern.ch/record/1095927

[8] B.Peralva, Calibration and Performance of the ATLAS Tile Calorimeter, arXiv:1305.0550v1

[9] ATLAS Collaboration, ATLAS LAr Calorimeter Phase-I upgrade TDR, CERN-LHCC-2013-017

[10] ATLAS Collaboration, ATLAS TDAQ System Phase-I upgrade TDR, CERN-LHCC-2013-018

[11] ATLAS Collaboration, Performance of the electronic readout of the ATLASliquid argon calorimeters, 2010 JINST 5 P09003

[12] K.Chen, H.Chen, M.Citterio, H.Deschamps, A.Grabs, S.Latorre, M.Lazzaroni, H.Lui, P.Schwemling, S.Simion, Design and Evaluation of LAr Trigger Digitizer Board in ATLAS Phase-I Upgrade, arXiv:18060.08046v2 22 Jun 2018

[13] K.Johns, Consideration of LAr LDPB for the MM trigger processor, https://slideplayer.com/slide/10361854/

[14] R.Carbone, Upgrade of the ATLAS Calorimeters for Higher LHC Luminosities, PoS (ICHEP2016) 236

[15] W.Qian, Design and test perfomance of ATLAS Feature Extractor trigger boards for Phase-1 Upgrade, ATL-DAQ-PROC-2016-024

[16] W.Wu, FELIX:the New Detector interface for the ATLAS Experiment, arXiv:1806.10667v1 22

Jun 2018

[17] P.Moreira, J.Christiansen, K.Wyllie, GBT PROJECT THE GBTX LINK INTERFACE ASIC V1.7 DRAFT, https://cms-docdb.cern.ch/cgi-bin/PublicDocDB/RetrieveFile?docid=3857&version=3&filename=gbtxSpecsV1.7.pdf

[18] HEP Software Foundation Collaboration, A Roadmap for HEP Software and Computing R&D for the 2020s, http://cds.cern.ch/record/2298968/files/10.1007_s41781-018-0018-8.pdf

[19] 井口竜之介, ATLAS LAr カロリメーターアップグレード：エネルギー再構成のためにフィルタリングアルゴリズム開発, http://www.icepp.s.u-tokyo.ac.jp/info/sympo/22/slides/iceppiguchi.pdf

[20] Y.Enari, The Phase-1 Trigger Readout Electronics Upgrade of the ATLAS Liquid Argon Calorimeter, https://iopscience.iop.org/article/10.1088/1742-6596/1162/1/012041/pdf

[21] C.W.Fabjan, F.Gianotti, Calorimetry for Particle Physics, CERN-EP/2003-075

[22] Analog Devices, FUNDAMENTAL PHASE LOCKED LOOP ARCHITECTURE, https://www.analog.com/media/en/training-seminars/tutorials/MT-086.pdf

[23] A.Camplani, Phase-I Trigger Readout Electronics Upgrade for the ATLAS Liquid-Argin Calorimeters, https://indico.ihep.ac.cn/event/6387/session/7/contribution/89/material/slides/0.pdf

[24] ATLAS Collaboration, ATLAS Experiment - Public Results, https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LArCaloPublicResultsUpgrade

[25] N.Madysa, A Software Framework for ATLAS Readout Electronics Upgrade Simulation, ATLAS-LARG-PROC-2018-008

[26] ATLAS Collaboration, ATLAS RUN Queries, https://atlas-runquery.cern.ch

[27] ATLAS Collaboration, ATLAS Experiment public results, Public Liquid-Argon Calorimeter Plots on Upgrade, https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LArCaloPublicResultsUpgrade

[28] Intel Corporation, Arria 10 GX FPGA Development Kit, https://www.intel.com/content/www/us/en/programmable/products/boards_and_kits/dev-kits/altera/kit-a10-gx-fpga.html

[29] R.Frazier, G.Iles, M.Magrans de Abril, D.Newbold, A.Rose, D.Sankey, T.Williams, The IPbus Protocol, http://ohm.bu.edu/~chill90/ipbus/ipbus_protocol_v2_0.pdf

[30] T.Williams, IPbus: A flexible Ethernet-based control system for xTCA hardware, https://indico.cern.ch/event/299180/contributions/1659676/attachments/563129/775787/IPbus_TWEPP_20140924_v1.pdf

[31] bitvis AS のウェブサイト、https://bitvis.no/dev-tools/uvvm/

[32] Alexis Vallier, The Phase-I Trigger Readout Electronics Upgrade of the ATLAS Liquid Argon Calorimeters, https://indico.cern.ch/event/818783/contributions/3598487/

attachments/1952649/3242185/avallier_CHEF2019_ATLAS-LAr-Phase1.pdf

[33] A.Haas, ATLAS Pileupand Overly Simulation, https://indico.cern.ch/event/279530/contributions/634995/attachments/511924/706533/Haas_ATLAS_PileupOverlay_3-19-2014.pdf

[34] P.Falke, LAr Phase-1 upgrade demonstrator and resonance searches in the dilepton final state at the ATLAS experiment, https://indico.in2p3.fr/event/14813/contributions/57767/attachments/45350/56427/LArPhase1Demonstrator_PeterFalke.pdf

[35] G.Aad, ATLAS LAr Calorimeter Trigger Electronics Phase-1 Upgrade, https://cds.cern.ch/record/2287802/files/ATL-LARG-SLIDE-2017-901.pdf

[36] G.Tateno, LHC-ATLAS Phase-1 upgrade: Calibration and simulation of new trigger readout system of the Liquid Argon calorimeter, https://indico.cern.ch/event/818783/contributions/3598501/attachments/1951578/3239985/ATL-LARG-SLIDE-2019-862.pdf

[37] ALTERA corporation, Arria 10 FPGA Development Kit User Guide, https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_a10-fpga-prod-devkit.pdf

[38] Intel corporation, インテル Stratix 10 デバイスのロジック・アレイ・ブロックおよびアダプティブ・ロジック・モジュール・ユーザーガイド, https://www.intel.co.jp/content/www/jp/ja/programmable/documentation/wtw1441782332101.html

[39] ALTERA corporation, Design Debuggin Using the Signal Tap II logic Analyzer, https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/signal.pdf

[40] 久保川達也、国友直人、「統計学」、東京大学出版会

[41] 唐渡広志。カイ 2 乗分布表、www3.u-toyama.ac.jp/kkarato/2016/statistics/handout/chisqdist.pdf

[42] 宇野健太、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、https://www.icepp.s.u-tokyo.ac.jp/download/master/m2016_uno.pdf

[43] 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、http://www.icepp.s.u-tokyo.ac.jp/download/master/m2014_hisajima.pdf

付録 A

LATOME Firmware interface

LATOME Firmware は LTDB からデジタイズされた Supercell の信号を受け取り、FEX に Supercell のエネルギーを伝送するのであった。ここでは各 Module の信号の IO とそれらの役割最後に各 Module のリソースについて纏める。

A.1 Input Stage

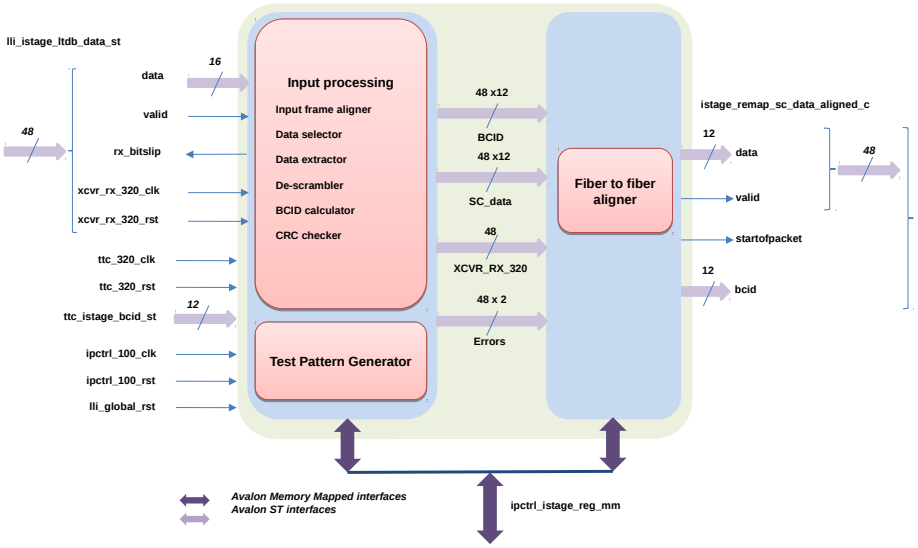


図 **A.1:** Istage インターフェース

Module	signal	width	I/O	description
TTC	ttc_320_clk	1	I	320 MHz clock domain
	ttc_320_rst	1	I	320 MHz clock domain synchronous reset
	ttc_istage_bcid_st	12	I	Bunch Crossing の ID を示す
LLI	data	16	I	デジタイザー (LTDB) からの Supercell 信号
	valid	1	I	Supercell 信号に対応する valid 信号
	rx_bitslip	1	O	Supercell 信号を正しくフォーマットするために用いる
	xcvr_rx_320_clk	1	I	320 MHz clock domain
	xcvr_rx_320_ret	1	I	320 MHz clock domain synchronous reset
Remap	data	12	O	Remap に送られる Supercell のデータ
	valid	1	O	Supercell データに対応する valid
	startofpacket	1	O	BCID の始まりを示す
	bcid	12	O	Bunch Crossing の ID を示す

表 **A.1:** Istage の IO

A.2 Configuration Remap

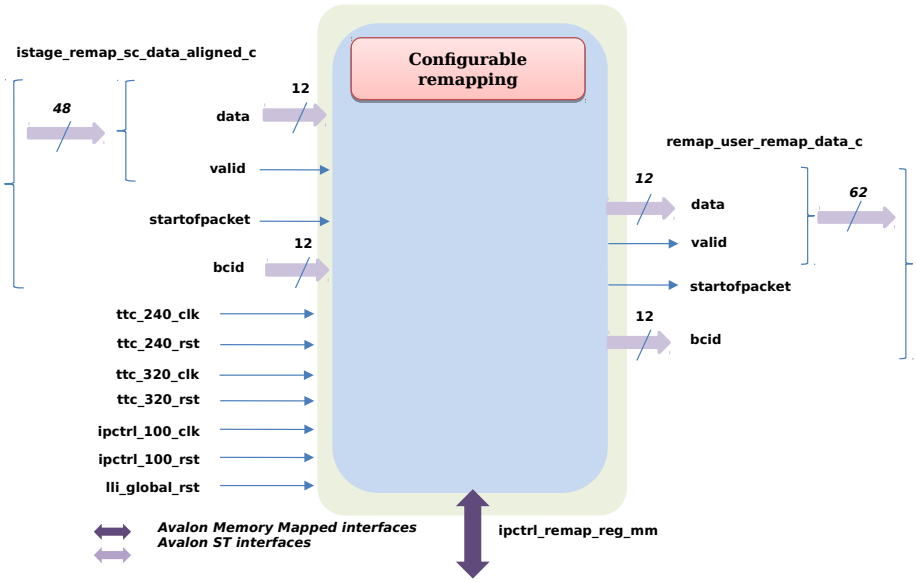


図 **A.2:** Remap インターフェース

Module	signal	width	I/O	description
TTC	ttc_320_clk	1	I	320 MHz clock domain
	ttc_320_rst	1	I	320 MHz clock domain synchronous reset
	ttc_240_clk	1	I	240 MHz clock domain
	ttc_240_rst	1	I	240 MHz clock domain synchronous reset
Istage	data	12	I	Supercell データ
	valid	1	I	Supercell データに対応する valid
	startofpacket	1	I	BCID の始まりを示す
	bcid	12	I	Bunch Crossing の ID を示す
User Code	data	12	O	Remap された Supercell データ
	valid	1	O	Remap さらた Supercell データに対応する valid
	startofpacket	1	O	BCID の始まりを示す
	bcid	12	O	Bunch Crossing の ID を示す

表 **A.2:** Remap の IO

A.3 User Code

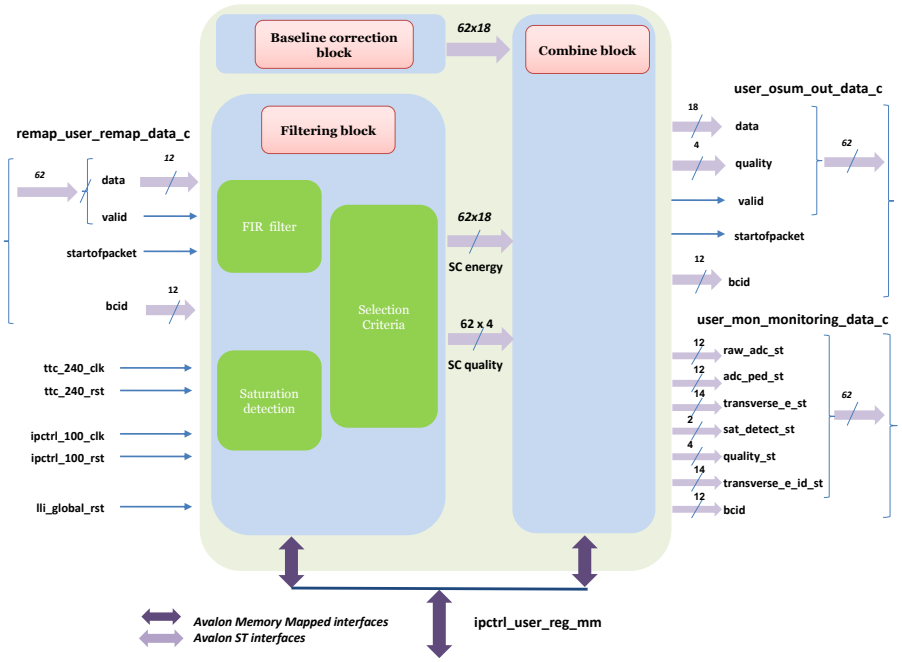


図 **A.3:** User Code インターフェース

Module	signal	width	I/O	description
TTC	ttc_240_clk	1	I	240 MHz clock domain
	ttc_240_rst	1	I	240 MHz clock domain synchronous reset
Remap	data	12	I	Remap された Supercell データ
	valid	1	I	Remap さらた Supercell データに対応する valid
	startofpacket	1	I	BCID の始まりを示す
	bcid	12	I	Bunch Crossing の ID を示す
Osum	data	18	O	Supercell のエネルギー
	quality	4	O	Supercell のエネルギーに対応する quality
	valid	1	O	Supercell のエネルギーに対応する valid
	startofpacket	1	O	BCID の始まりを示す
	bcid	12	O	Bunch Crossing の ID を示す
Monitoring Block	raw_adc	12	O	Supercell の ADC
	adc_ped	12	O	ADC から Pedestal を差し引いたもの
	transverse_e	14	O	Tau Criteria を抜けたエネルギー
	transverse_e_id	14	O	Saturation まで考慮に入れたエネルギー
	quality	4	O	Supercell の Quality
	sat_detect	2	O	Saturation, BCAV 情報
	bcid	12	O	Bunch Crossing の ID を示す

表 **A.3:** User Code の IO

A.4 Output Summing

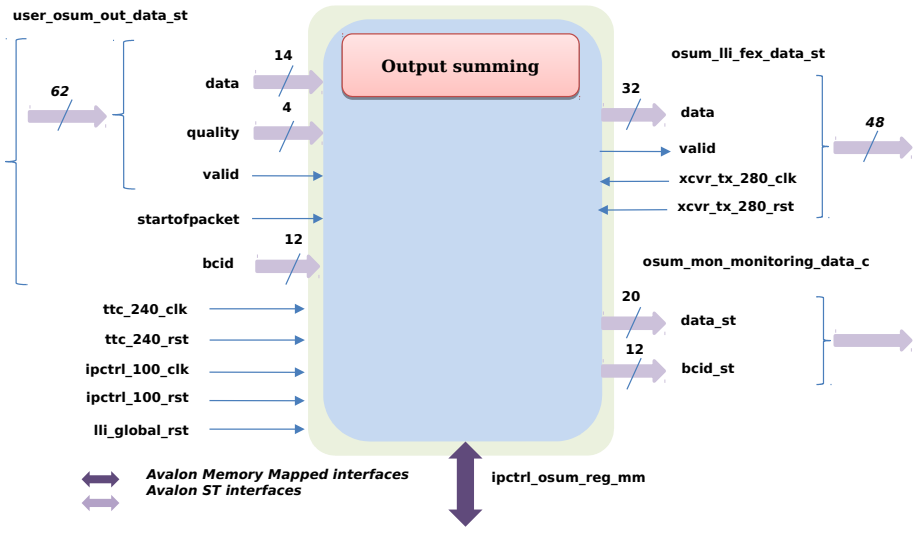


図 **A.4:** Osum インターフェース

Module	signal	width	I/O	description
TTC	ttc_240_clk	1	I	240 MHz clock domain
	ttc_240_rst	1	I	240 MHz clock domain synchronous reset
User Code	data	18	I	Supercell のエネルギー
	quality	4	I	Supercell のエネルギーに対応する quality
	valid	1	I	Supercell のエネルギーに対応する valid
	startofpacket	1	I	BCID の始まりを示す
	bcid	12	I	Bunch Crossing の ID を示す
LLI	data	32	O	FEX の伝送する LATOME Firmware の出力
	valid	1	O	対応する valid
	xcvr_tx_280_clk	1	I	320 MHz clock domain
	xcvr_tx_280_rst	1	I	320 MHz clock domain synchronous reset
Monitoring Block	data_st	20	O	任意の FEX の伝送するデータ
	bcid_st	12	O	Bunch Crossing の ID を示す

表 A.4: Osum の IO

A.5 各 LATOME Firmware 内の Module のリソース使用量

Module	ALM	Block mwmory bit	M20K	DSP block
LLI	15157	61024	0	0
TTC	13355	2080	7	0
Istage	63812	4992	48	0
Remap	19234	0	0	0
User Code	138703	18518656	1860	248
Osum	54885	0	0	0
IP bus	1989	558848	35	0
Monitoring Block	9670	10061240	541	0
LATOME Firmware	318544	29106840	2501	0
Arria 10	427200	55562240	2713	1518

表 A.5: LATOME Firmware の各 Module のリソース

付録 B

User Code 内の Block

User Code の各 Block は Remap からの Supercell の信号と BCID 情報、TTC(LLI を経由する) からのクロック, リセット信号を受け取る。さらに IP bus を用いて内部のレジスター, メモリにアクセス可能である。それらは概ね共通した信号であり表 B.1 に纏められる。ここでは各 Block における信号の IO とそれらの役割最後に各 Block のリソース使用量について纏める。

Module	signal	width	I/O	description
TTC	ttc_240_clk	1	I	240 MHz clock domain
	ttc_240_rst	1	I	240 MHz clock domain synchronous reset
Remap	remap_in_data	12	I	ADC data
	remap_in_valid	1	I	ADC data valid
	remap_in_bcid	12	I	BCID information
	remap_in_sop	1	I	Start of packet
IP bus	ipctrl_100_clk	1	I	100 MHz clock domain
	ipctrl_100_rst	1	I	100 MHz clock domain synchronous reset
	writedata	32	I	Write data
	wrena	1	I	Write enable
	readdat	32	O	Read data
	rdena	1	I	Read enable
	readdatavalid	1	O	Read data valid
	address	32	I	IP busaddress Bus

表 B.1: User Code の TTC, Remap, IP bus 関連の信号

B.1 FIR filter

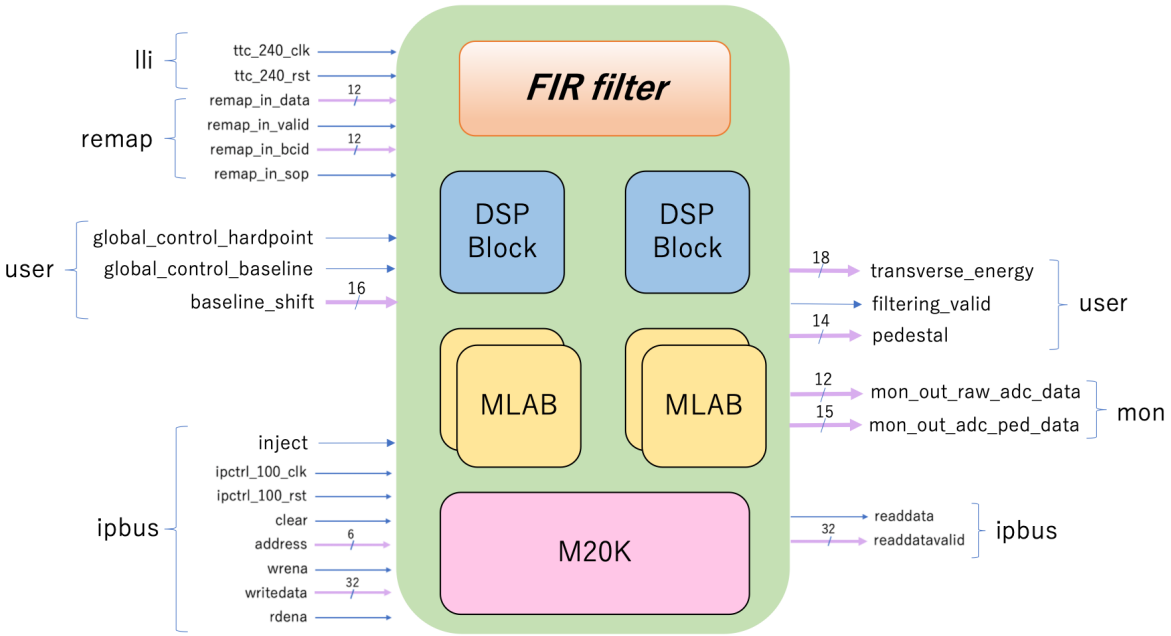


図 B.1: FIR Filter

Module	signal	width	I/O	description
User Code	global_control_hardpoint	1	I	DSP block の bit の丸めに関する global control
	global_control_baseline	1	I	Filtering Algorithm の Baseline Correction に対する global control
	baseline_shift	16	I	Baseline Shift
	transverse_energy	18	O	Energy candidate
	filtering_valid	1	O	Energy candidate の valid
Monitoring Block	mon_out_raw_adc_data	12	O	ADC
	mon_out_adc_ped_data	15	O	ADC - pedestal
IP bus	inject	1	I	Pedestal と計数 a_i を Circular buffer に書き込む

表 B.2: FIR Filter の IO

B.2 Saturation detectrion

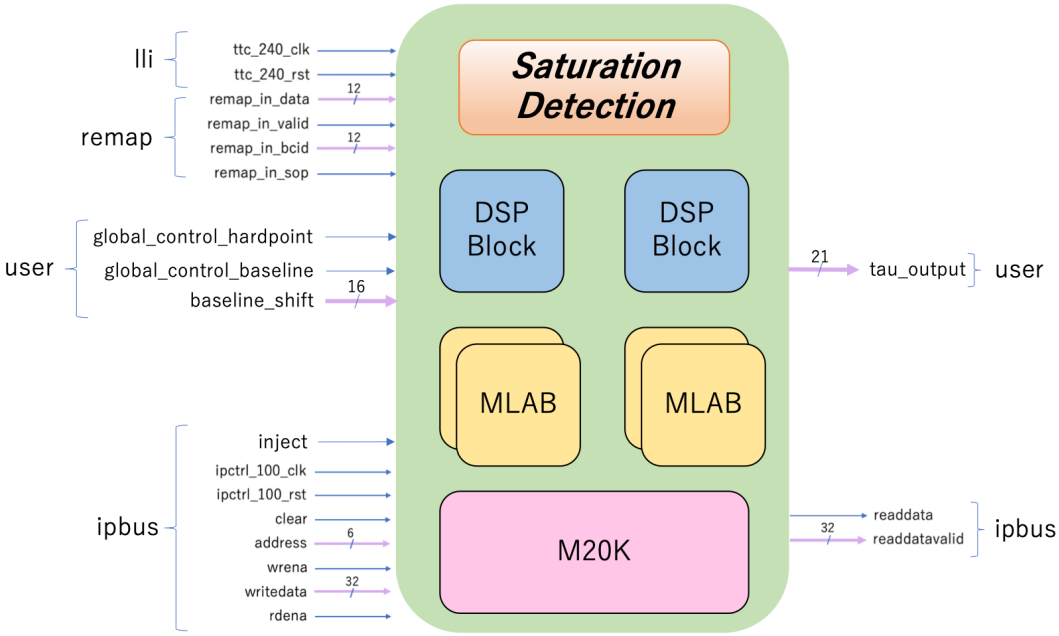


図 **B.2:** Saturation Detection

Module	signal	width	I/O	description
User Code	global_control_hardpoint	1	I	DSP block の bit の丸めに関する global control
	global_control_baseline	1	I	Filtering Algorithm の Baseline Correction に対する global control
	baseline_shift	16	I	Baseline Shift
	tau_output	21	O	Energy tau candidate
IP bus	inject	1	I	Pedestal と計数 a_i を Circular buffer に書き込む

表 **B.3:** Saturation Detection の IO

B.3 Selection block

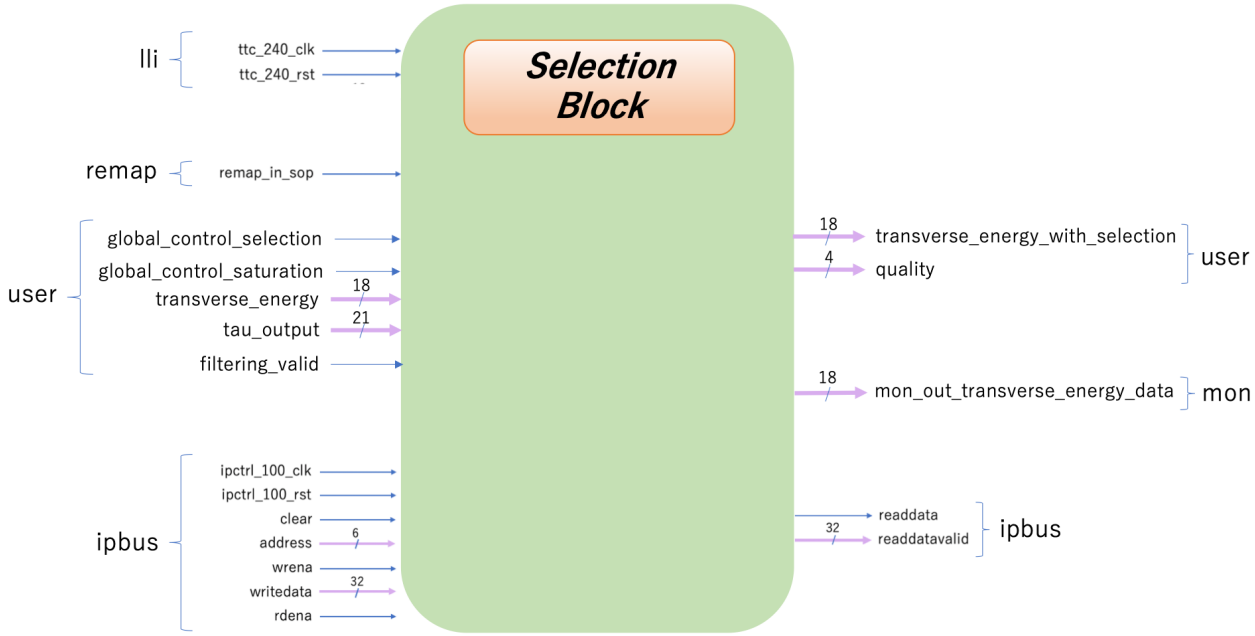


図 B.3: Selection Block

Module	signal	width	I/O	description
User Code	global_control_selection	1	I	Tau Criteria に関する global control
	global_control_saturation	1	I	Saturation Criteria に関する global control
	transverse_energy	18	I	Energy candidate
	tau_output	21	I	Energy tau candidate
	filtering_valid	1	I	Energy candidate の valid
	transverse_energy_with_selection	18	O	Tau Criteria により再構成された Energy
	quality	4	O	BCAV と Saturation の情報
Monitoring Block	mon_out_transverse_energy_data	18	O	Tau Criteria により再構成された Energy

表 B.4: Selection Block の IO

B.4 Combine block

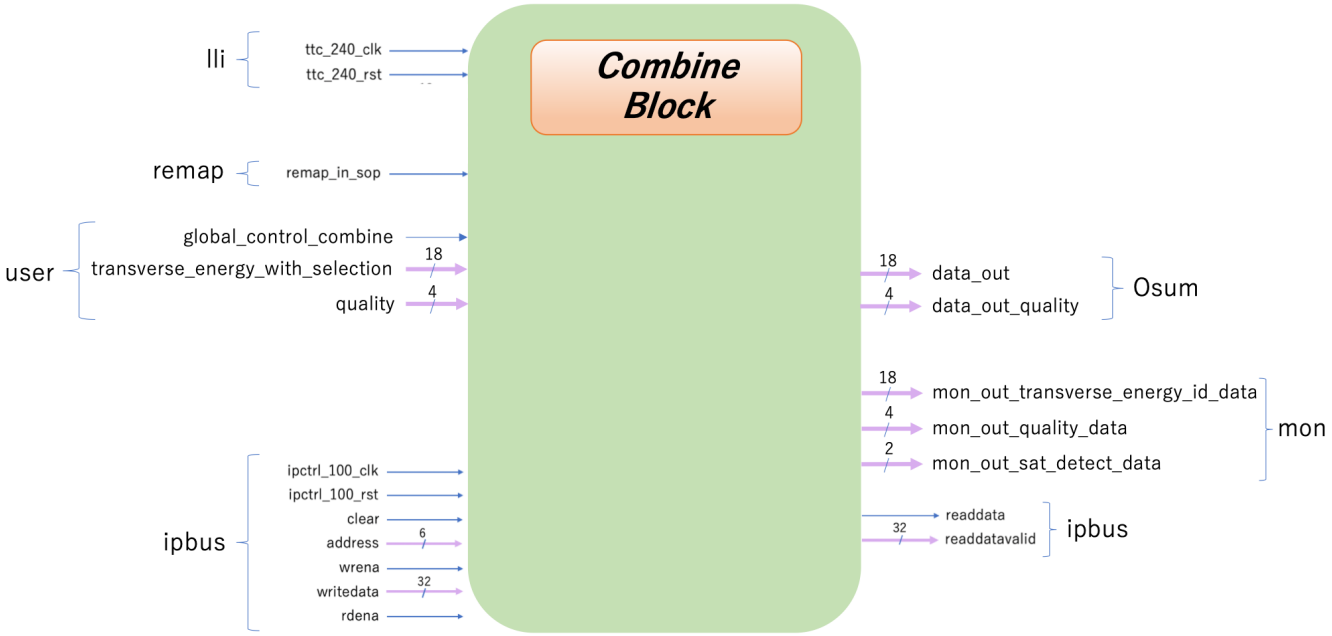


図 B.4: Combine Block

Module	signal	width	I/O	description
User Code	global_control_combine	1	I	Combine block 全体に関する global control
	transverse_energy_with_selection	18	I	Tau Criteria により再構成された Energy
	quality	4	I	BCAV と Saturation の情報
Osum	data_out	18	O	User Code の Energy の出力
	data_out_quality	4	O	User Code の Quality の出力
Monitoring Block	mon_out_transverse_energy_id_data	18	O	Saturation まで考慮にいれた Energy
	mon_out_quality_data	4	O	Saturation まで考慮にいれた Quality
	mon_out_sat_detcet_data	2	O	Quality に上 2 bit

表 B.5: Combine Block の IO

B.5 Baseline correction

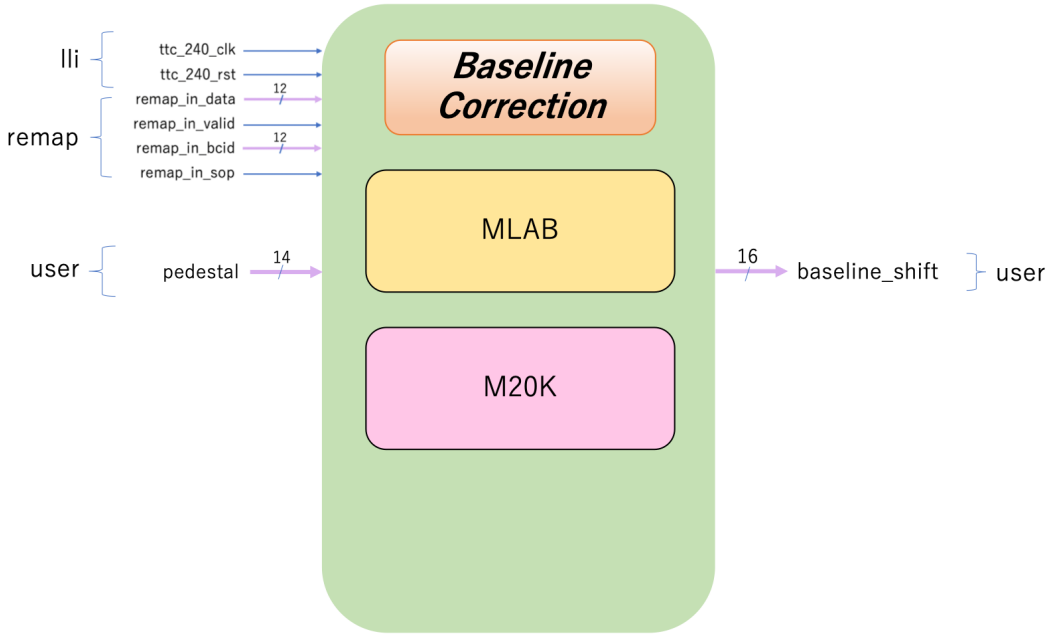


図 **B.5:** Baseline Correction

Module	signal	width	I/O	description
User Code	pedestal	14	I	各 Supercell に依存する Pedestal
	baseline_shift	16	O	Baseline Shift

表 **B.6:** Baseline Correction の IO

B.6 Summation ADC

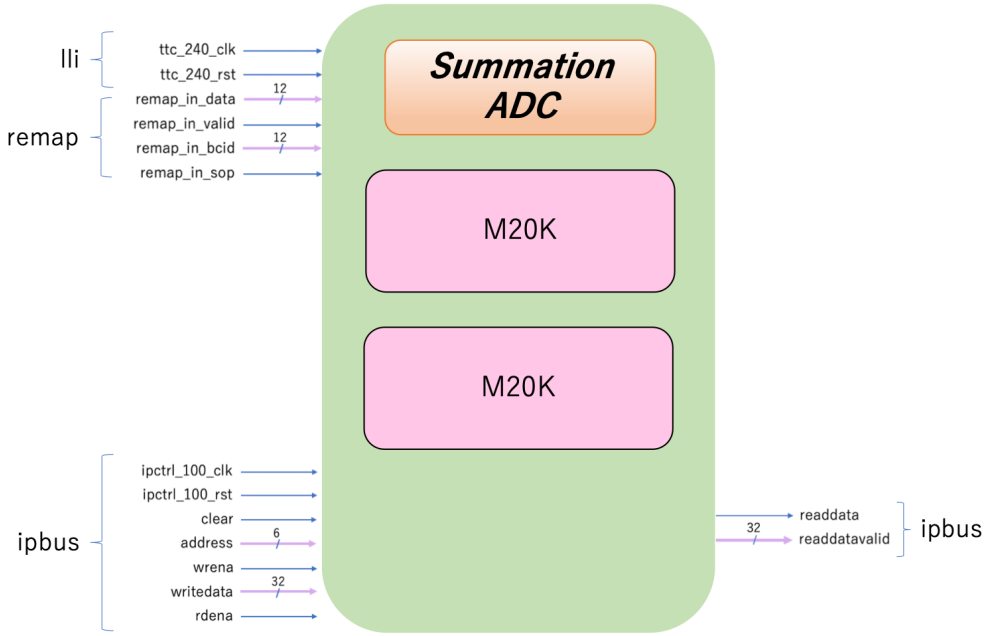


図 B.6: Summation ADC

B.7 Peak detector

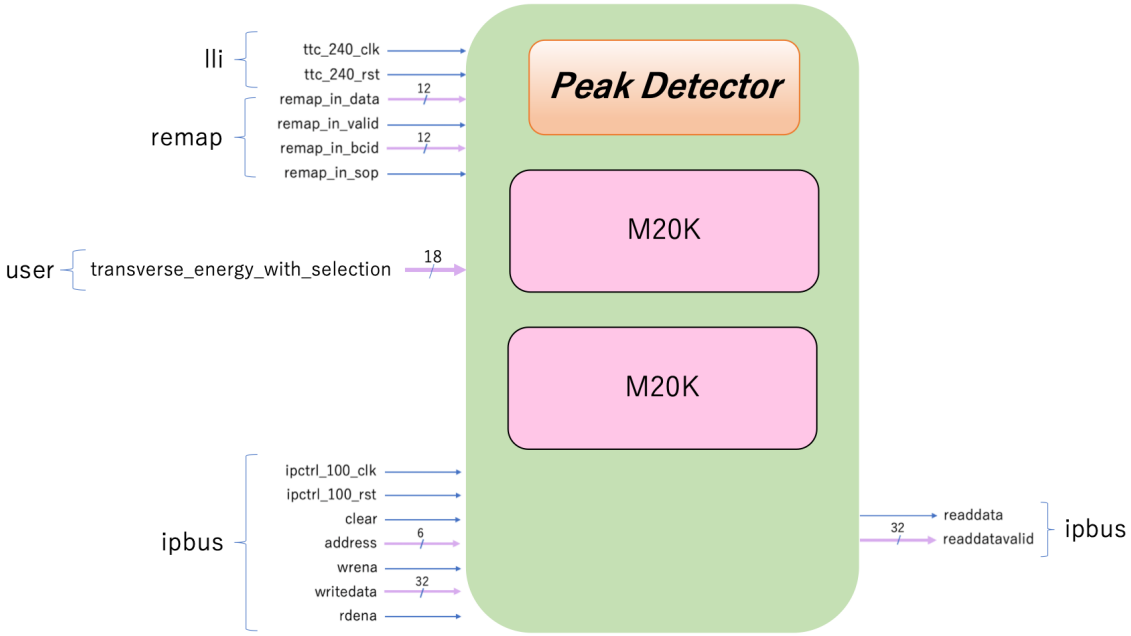


図 **B.7:** Peak Detector

Module	signal	width	I/O	description
User Code	transverse_energy_with_selection	18	I	Tau Criteria をぬけた Energy

表 **B.7:** Baseline Correction の IO

B.8 ADC shape checker

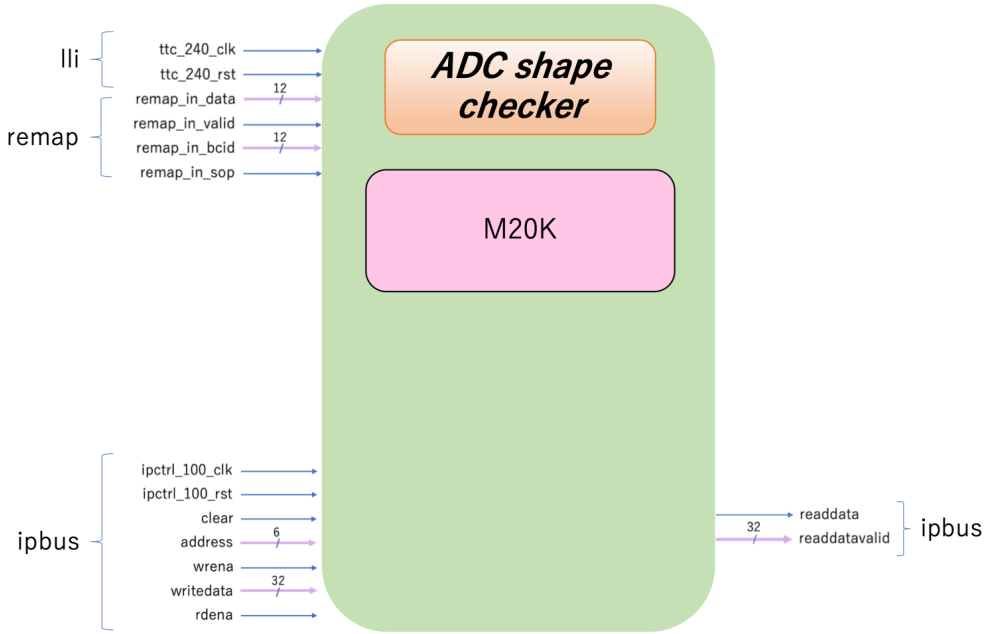


図 **B.8:** ADC Shape Checker

B.9 各 User Code 内の Block のリソース使用量

Block	#	ALM	Block memory bit	M20K	DSP block
User Code	-	138703	18518656	1860	248
FIR Filter	62	15203	59272	124	124
Saturation Detection	62	10817	55552	62	124
Selection Block	62	48981	0	0	0
Combine Block	6	6224	0	0	0
Baseline Correction	62	27052	18284544	1116	0
Summation ADC	62	7638	12400	62	0
ADC Shape Checker*1	62	18739	107136	496	0
LATOME Firmware	-	318544	29206840	2501	248

表 **B.8:** User Code の各 Block のリソース

付録 C

Simulation による検証の流れ

ここでは User Code の Simulation による検証の詳しい流れについて説明する。シーケンス図には統一モデリング言語 (UML) を用いている[*1]。

C.1 Summation ADC

1. データセットを指定する
2. User CodeModel が指定されたデータセットに基づいて予測値を計算
3. User CodeModel の予測値を MIF として保存
4. MIF を用いて Firmware で実装されている checker 内の RAM を初期化
5. User Code に同じデータセットを伝送する
6. 計算が完了するまで待つ (LHC 1 周)
7. Summation ADC の計算結果を IP bus を用いて読み出す
8. 対応する Supercell の値を checker 内の RAM から読み出し Summation ADC の計算結果と比較
9. clear 信号を送り一度リセットする
10. リセットがうまく動作しているか確認
11. 再び計算が完了するまで待つ
12. 再び checker 内の RAM と Summation ADC の出力を比較
13. Simulation の最終結果を checker が返す (PASS/FAILED)

[*1] https://liveuml.com/ を用いてシーケンス図を作成した

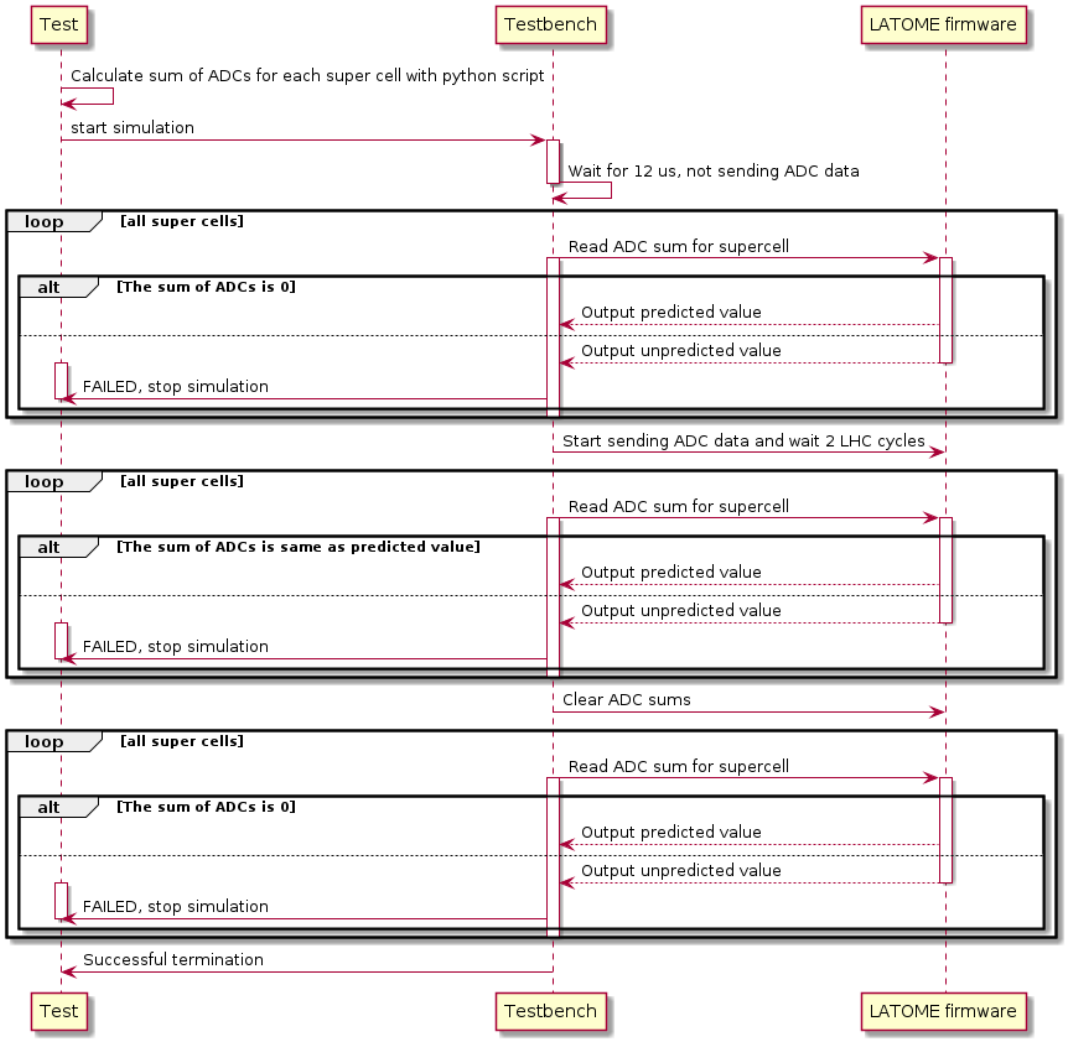


図 C.1: Summation ADC の simulation による検証

C.2 ADC Shape Checker

1. データセットを指定する
2. User CodeModel が指定されたデータセットに基づいて予測値を計算
3. User CodeModel の予測値を MIF として保存
4. MIF を用いて Firmware で実装されている checker 内の RAM を初期化
5. IP bus を用いて Gap Constant を書き換える
6. 計算が完了するまで待つ (LHC 1 周)
7. ADC Shape Checker の計算結果を IP bus を用いて読み出す
8. 対応する Supercell の ADC, BCID を checker RAM から読み出す
9. clear 信号を送り一度リセット
10. リセット動作確認

11. Gap constant を大きな値に設定
12. ADC Shape Checker の計算を待ち読み出す (Gap constant が大きいのでそれを満たすピークは見つからないはず)
13. Simulation の最終結果を checker が返す (PASS/FAILED)

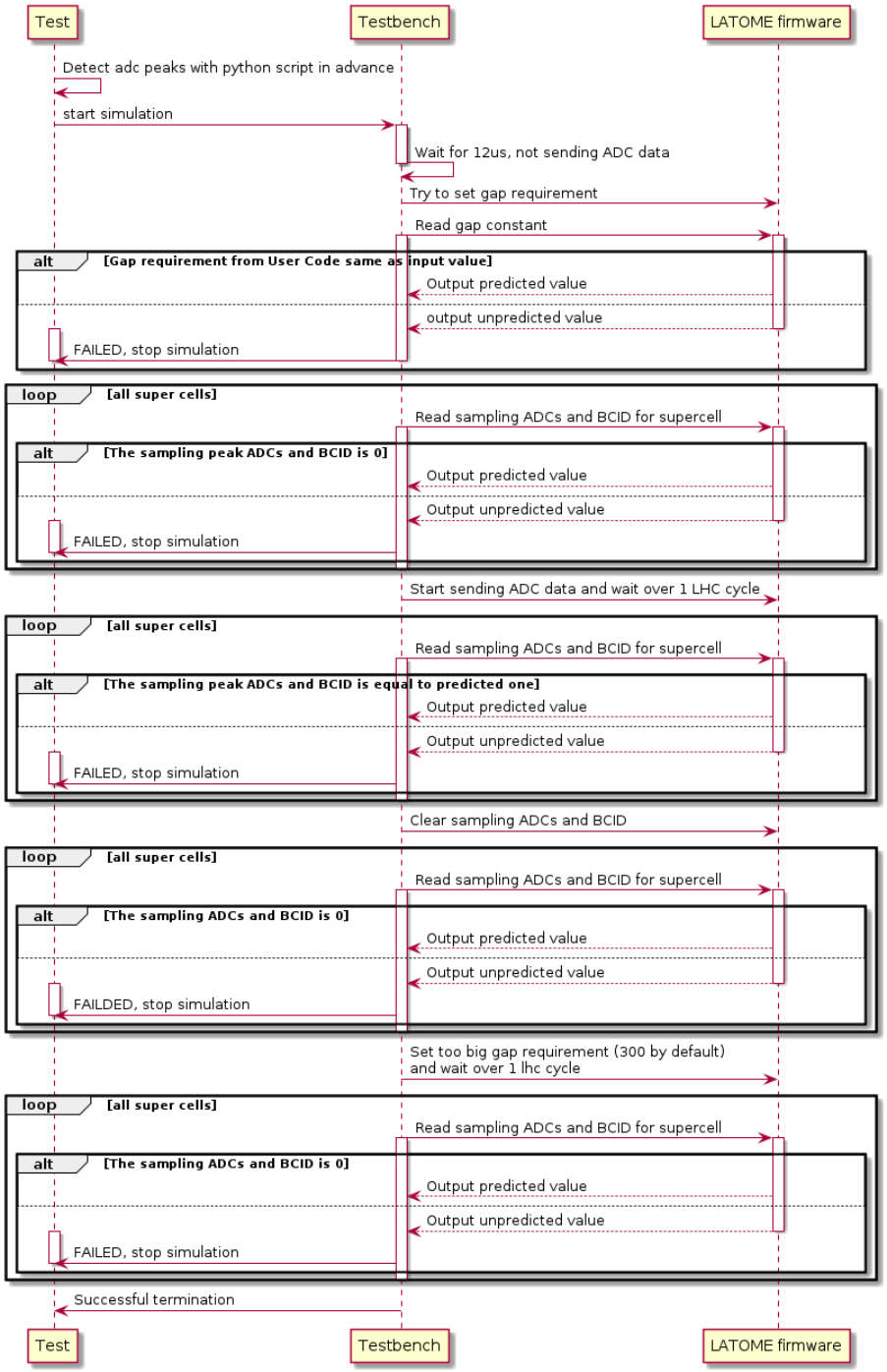


図 **C.2:** ADC Shape Checker の simulation による検証

C.3 Peak Detector

1. データセットを指定する
2. User CodeModel が指定されたデータセットに基づいて予測値を計算
3. User CodeModel の予測値を MIF として保存
4. MIF を用いて Firmware で実装されている checker 内の RAM を初期化
5. 計算が完了するまで待つ (LHC 1 周)
6. Peak Detector の計算結果を IP bus を用いて読み出す
7. 対応する Supercell の ADC, BCID を checker RAM から読み出す
8. clear 信号を送り一度リセット
9. リセット動作確認
10. Simulation の最終結果を checker が返す (PASS/FAILED)

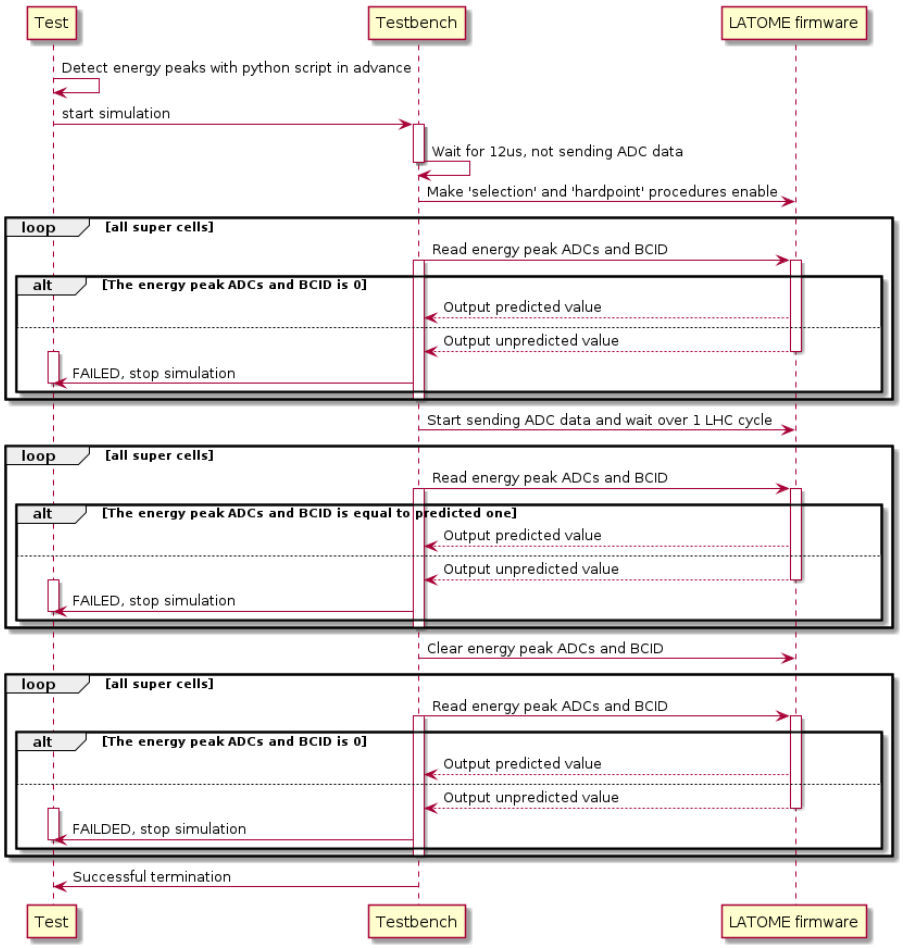


図 **C.3:** Peak Detctor の simulation による検証

C.4 Filtering Algorithm

1. データセットを指定する
2. User CodeModel が指定されたデータセットに基づいて予測値を計算
3. User CodeModel の予測値を MIF として保存
4. MIF を用いて Firmware で実装されている checker 内の RAM を初期化
5. 計数, Pedestal などの定数が書き換え可能かどうか IP bus を用いて確認する
6. User Code にデータを送る
7. LHC 1 周以上の長さで Osum と Monitoring Block への出力が bit by bit で予測値と一致するか確認
8. Simulation の最終結果を checker が返す (PASS/FAILED)

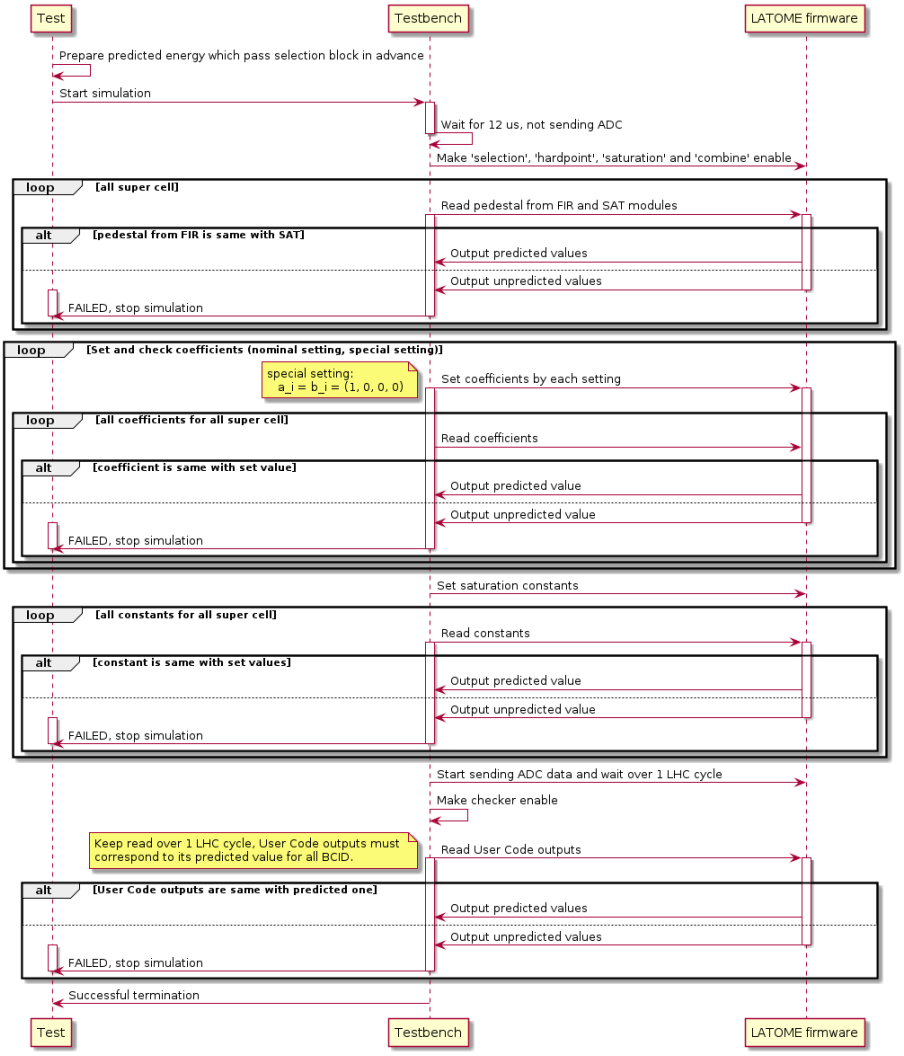


図 C.4: Filtering algorithm の simulation による検証

C.5 Baseline Correction

1. データセットを指定する
2. User CodeModel が指定されたデータセットに基づいて予測値を計算 (Filtering の検証で用いたものを流用させる)
3. User CodeModel の予測値を MIF として保存
4. MIF を用いて Firmware で実装されている checker 内の RAM を初期化
5. User Code にデータ 1[*2]を送る
6. Baseline Correction が計算を完了するまで待つ
7. User Code にデータ 2[*3]を送る
8. LHC 1 周以上の長さで Osum と Monitoring Block[*4] への出力が bit by bit で予測値と一致するか確認
9. Simulation の最終結果を checker が返す (PASS/FAILED)

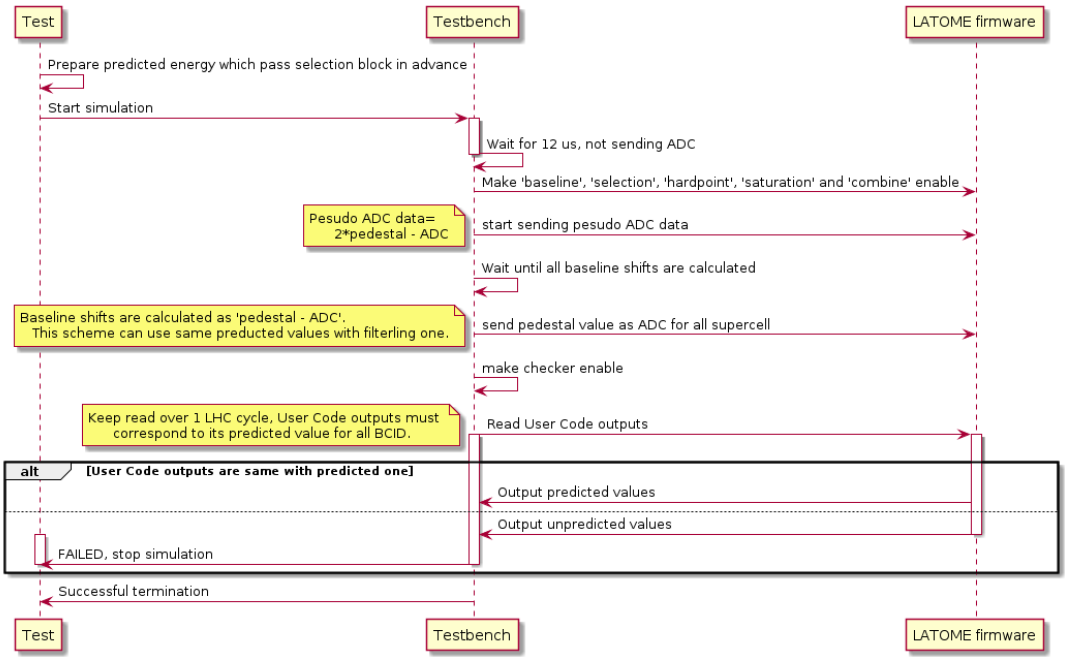


図 **C.5:** Baseline Corrction の simulation による検証

[*2] 入力 $ADC = Pedestal -$ 元々の ADC
[*3] 入力 $ADC = Pedestal$
[*4] ADC と $ADC - Pedestal$ の予測値は入力を変化させているので一致しないので除外する

付録 D

実機を用いた検証の流れ

Simulation 同様に実機での検証方法とその手順を纏める。検証には IP bus を用いる必要があり LATOME Board と接続された pc-emf-fw-03 上で行う必要がある。また LATOME Firmware からの Monitoring data は pcemf-mon-04 に保存される。

D.1 Summation ADC

1. データセットを指定する
2. データセットを Inject LATOME に書き込む
3. Injectot LATOME と LATOME Firmware の Configuration
4. clear 信号を送り一度リセット
5. User CodeModel が Summation ADC の出力を予想する
6. User Code から Firmware の計算結果を IP bus を用いて読み出す
7. 全ての Supercell に対し予測値が Summation ADC の出力と等しいか確認
8. 最終結果を返す (PASS/FAILED)

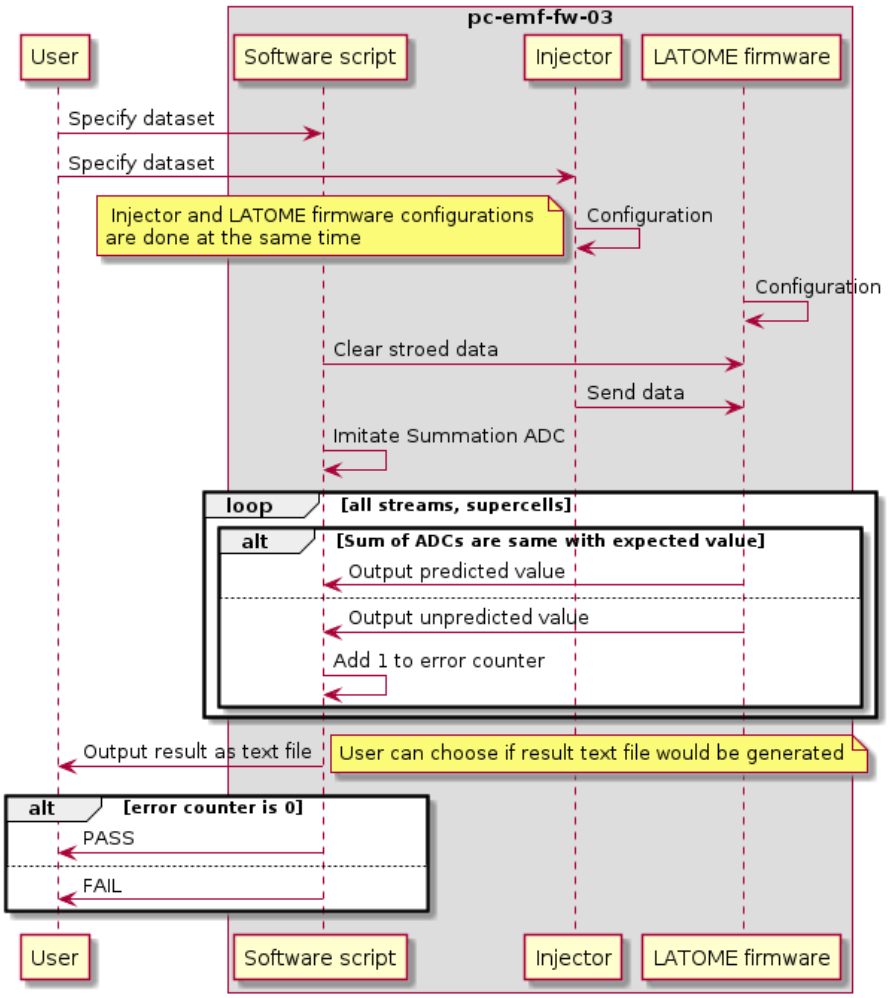


図 **D.1:** Summation ADC の実機での検証

D.2 ADC Shape Checker

1. データセットを指定する
2. データセットを Inject LATOME に書き込む
3. Injectot LATOME と LATOME Firmware の Configuration
4. clear 信号を送り一度リセット
5. ADC Shape Checker から Gap Constant を読み出す
6. User CodeModel が ADC Shape Checker の出力を読み出した Gap Constant に基づき予想する
7. User Code から Firmware の計算結果を IP bus を用いて読み出す
8. 全ての Supercell の ADC, BCID に対し予測値が ADC Shape Checker の出力と等しいか確認
9. 最終結果を返す (PASS/FAILED)

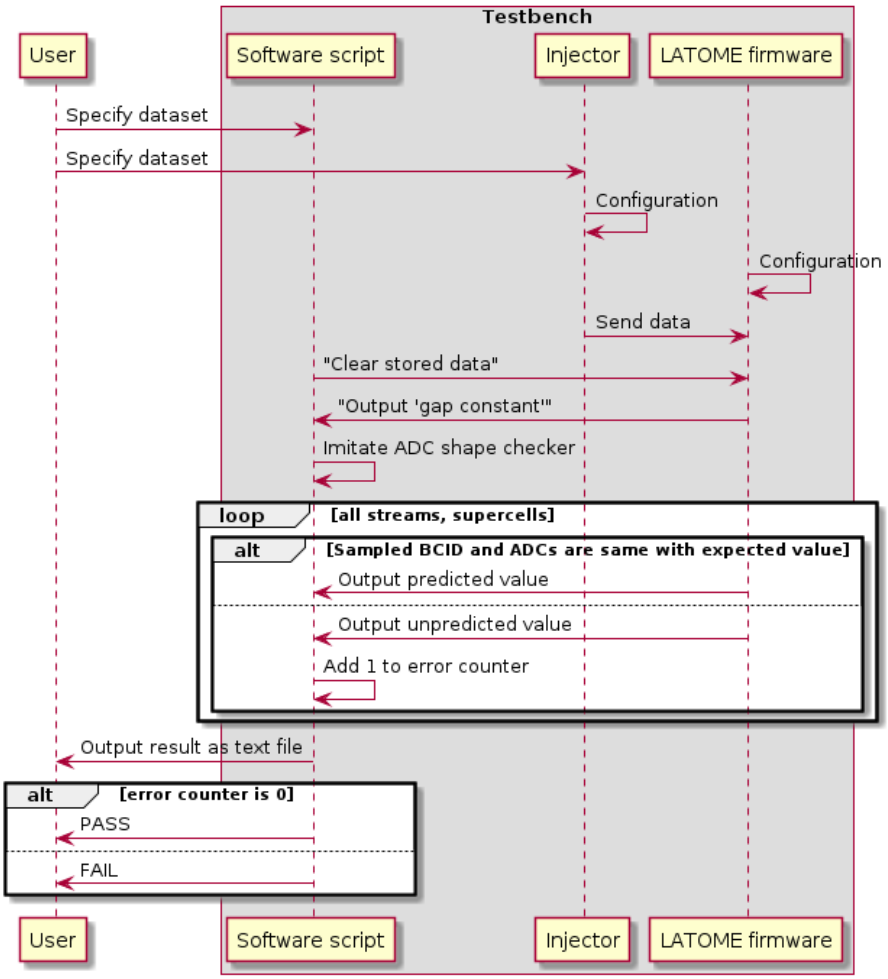


図 **D.2:** ADC Shape Checker の実機での検証

D.3 Peak Detector

1. データセットを指定する
2. データセットを Inject LATOME に書き込む
3. Injectot LATOME と LATOME Firmware の Configuration
4. clear 信号を送り一度リセット
5. User CodeModel が Peak Detector の出力を予想する
6. User Code から Firmware の計算結果を IP bus を用いて読み出す
7. 全ての Supercell の ADC, BCID に対し予測値が Peak Detector の出力と等しいか確認
8. 最終結果を返す (PASS/FAILED)

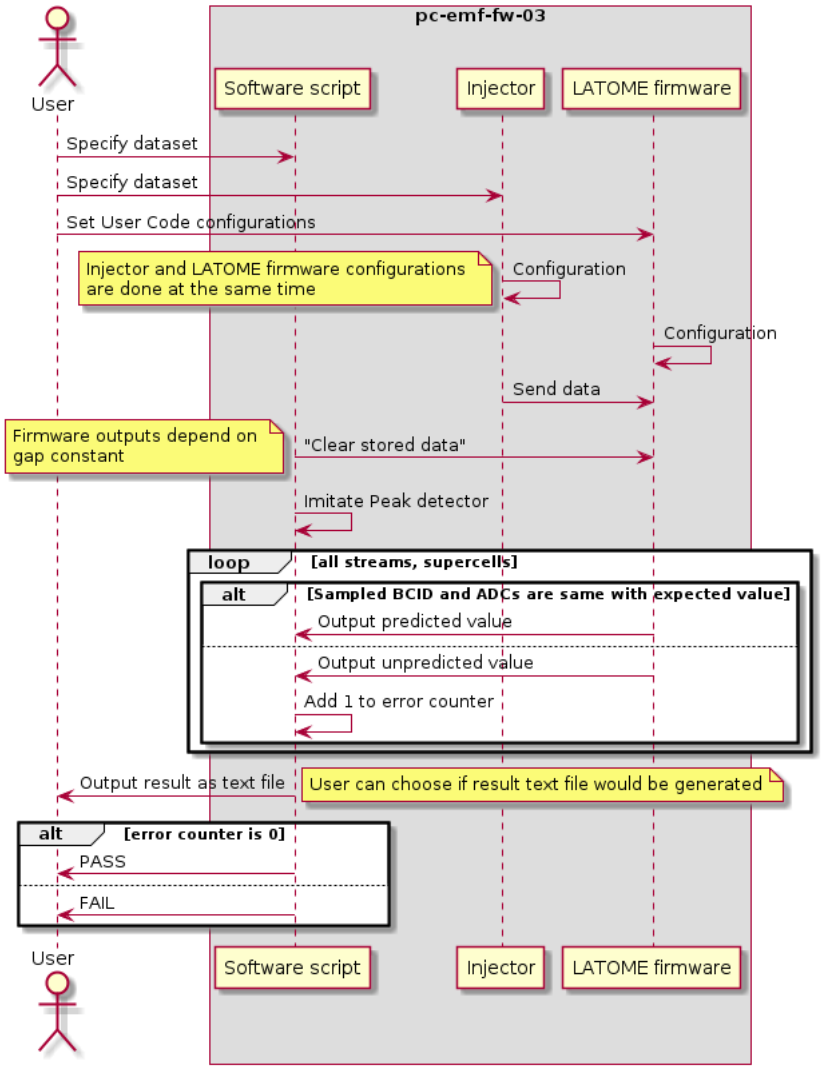


図 **D.3:** Peak Detcetor の実機での検証

D.4 Filtering Algorithm

1. データセットを指定する
2. データセットを Inject LATOME に書き込む
3. Injectot LATOME と LATOME Firmware の Configuration
4. User Code から Status・係数・Pedestal を読み出す
5. User CodeModel が読み出した Status に応じ User Code の Monitoring data を予測
6. UDP Server と UDP port の設定
7. Level 1 Accept を TTC から送り Monitoring data を伝送する[*1]

[*1] 各 Level 1 Accept につき 32BCID 分のデータが保存される LHC 1 周分を網羅するため少なくとも 112 回 Level 1 Accept

8. バイナリデータである Monitoring data をデコードする
9. 全ての Supercell,BCID に対して予測値と User Code の出力が bit by bit で等しいか比較
10. 最終結果を返す (PASS/FAILED)

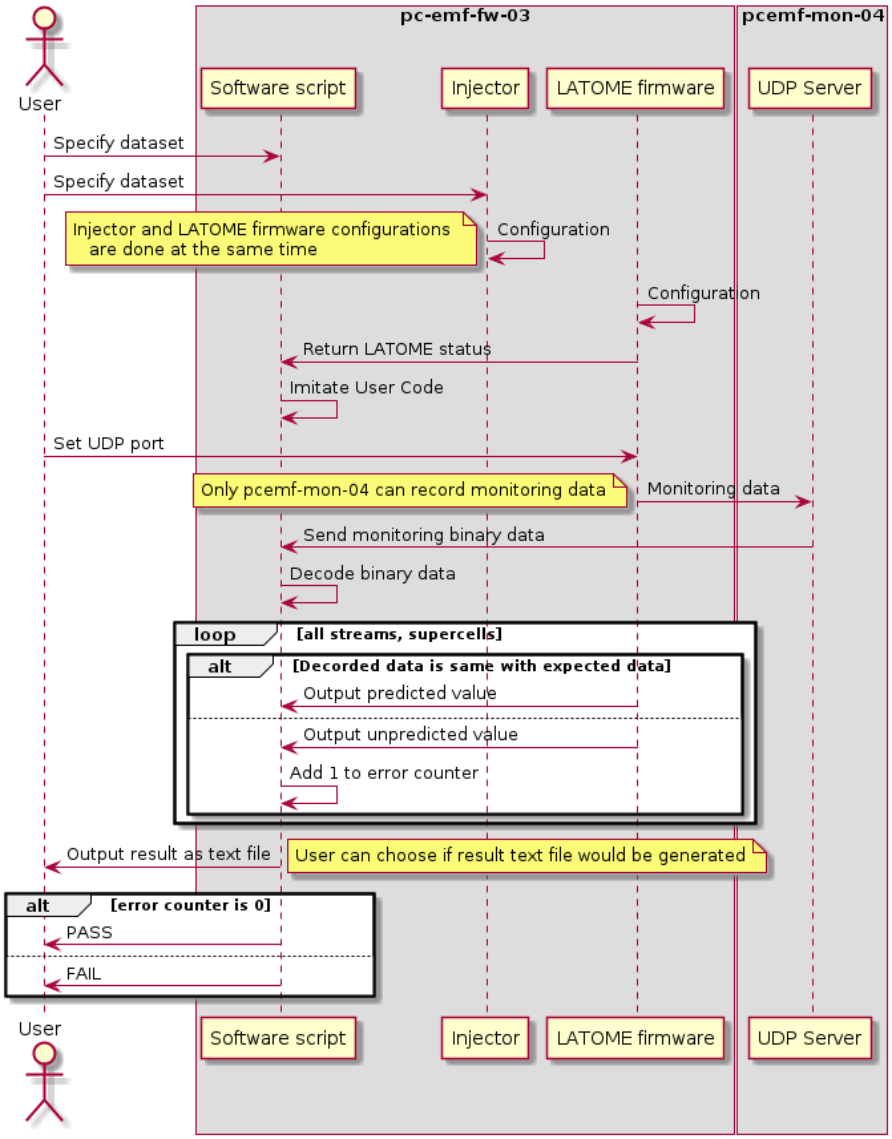


図 **D.4:** Filtering algorithm の実機での検証

を送る実際にはその倍の 224 回送っている

付録 E

評価ボードを用いた User Code の検証

User Code は LATOME Board を用いて Monitoring data の出力を予測値と比較することにより行われた。しかしこの機構では LHC cycle 1 周に対し 1 度の頻度でしか Monitoring data を取得することができない。LATOME Board は 24 時間連続で動作させることが想定されるのでその間にエラーが起こらないことを確認したい。この場合 Monitoring data を用いた検証では莫大な時間データを記録し続けなければならなかった。詳細は 6.2.3 節で記述されている。そのため Firmware 内部に Checker を用意して User Code の出力をそのまま Firmware 内部で予測値と比較する検証も行った。この機構は既に行われていたが Baseline Correction に対しては実装されていなかった [42]。ここでは User Code Main path に対する評価ボード (図 E.1) での検証について述べる。

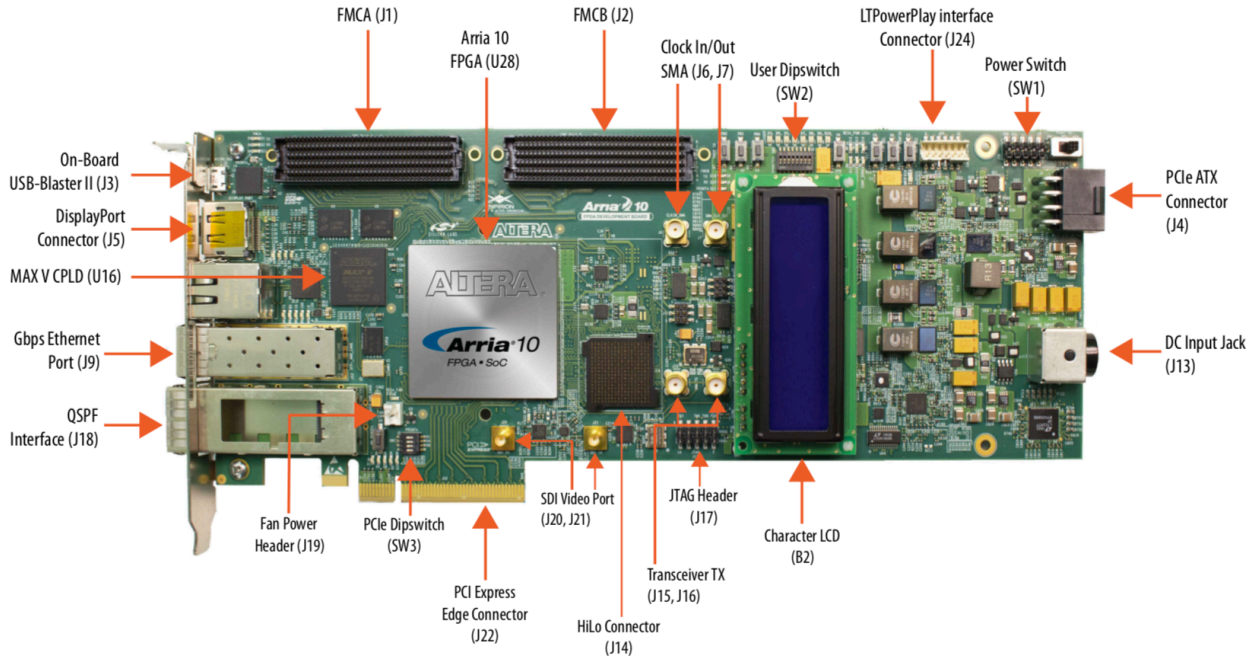
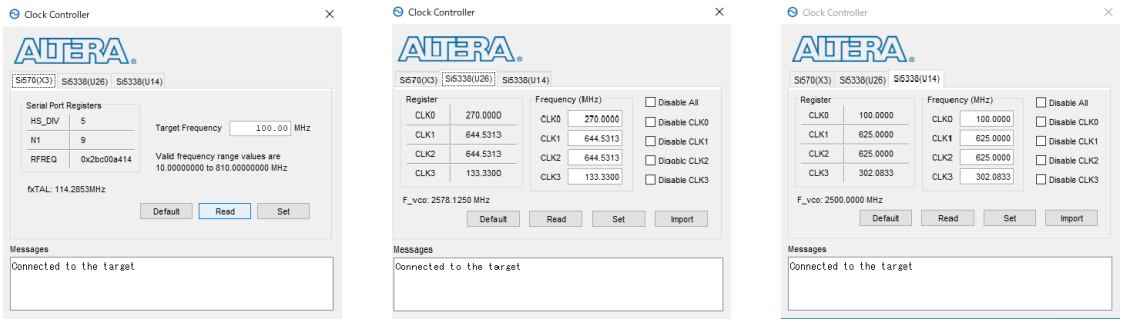


図 E.1: Arria 10 搭載の評価ボード。写真では Arria 10 が抜き出しであるが実際にはファンが FPGA, CPLD 上に搭載されている。

この評価ボードには任意の周波数を FPGA core に送る機能を持つ。これらのクロックは Intel 社提供の Board Test System を用いて制御することが可能であり、合計 8 つのクロックを FPGA core 内で用いることができる。今回は図 E.2 の Si5338(U26) と Si5338(U14) のうち 6 個のクロックを用いた。



(a) Si570(X3) **(b)** Si5338(U26) **(c)** Si5338(U14)

図 E.2: Clock control system で利用可能な 3 つのオンボードチップ。X3, U26, U14 は評価ボードのピンの番号に対応する。最大で小数点以下 8 桁までの精度で周波数を設定することができる。

評価ボードはこれらの他にも 50 MHz, 125 MHz などのオンボードクロックを用いることも可能でありまた SMA ケーブルや 2 つの FMC ポートを用いて外部から信号を入力することも可能である (図 E.3)。

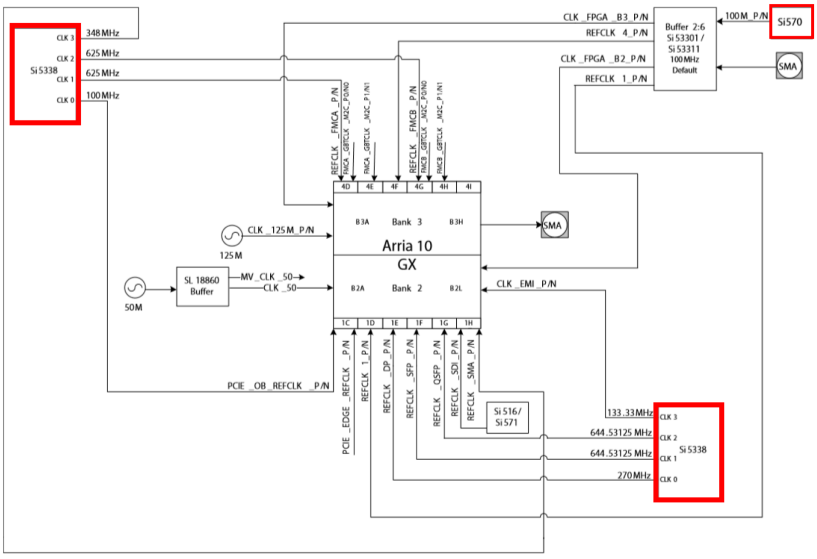


図 E.3: Arria 10 内部のオンボードクロック。赤枠で囲われた部分が Clock control system に対応する。50 MHz, 125 MHz のクロックは水晶振動子を用いて一定の周波数を作成する。

User Code の各 Block の機能を視覚的に確認するために Signal Tap ロジック・アナライザ [39] を用いた。Signal Tap は指定されたノードの情報を RAM に記録し、それを読み出すことでそのノードの値の変化を視覚的に確認することができる。アドレスの bit 数に対応して記録するデータ数を変化させることができる。Signal Tap は指定された信号数、信号の bit 数、アドレス数に依存した RAM を自動で生成す

る。そのため User Code で用いるロジック, RAM に加えてデータ記録用の RAM を用意する形になる。仮に RUN3 実験で 1 枚の LATOME Board によって扱われる 372 Supercell に対応した 62 stream で検証を行った場合、評価ボード内の M20K の枚数が足りなくなる。そのため 1 stream のみでの検証を行った。こうすることにより評価ボードに実装可能な容量で User Code のロジックを確認することができる。検証に用いた Firmware には LLI と User Code のみが実装されてものを用いた。

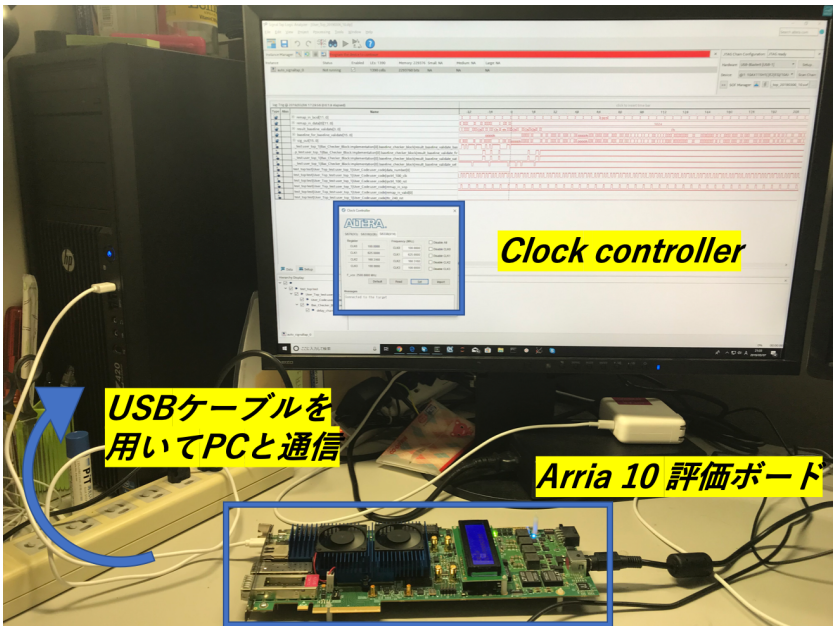


図 E.4: Arria 10 評価ボードを用いた検証のセットアップ

Baseline Correction までを含めた User Code Main path 全体の検証では 6.1.3 章での方法を用いた。User Code の各 Block からの出力を直接抜き出し、それを Checker が予想した値と比較する (図 E.5)。Checker Firmware も評価ボードに同時に実装されているので 240 MHz の頻度で常に出力を比較することが可能である。

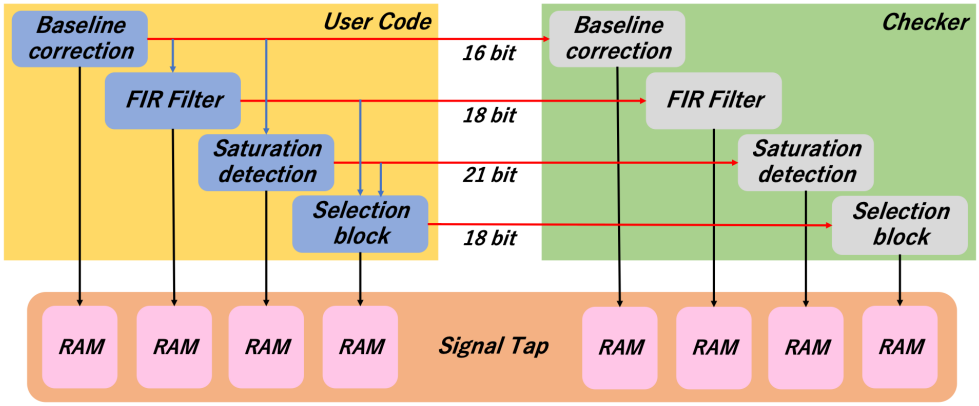
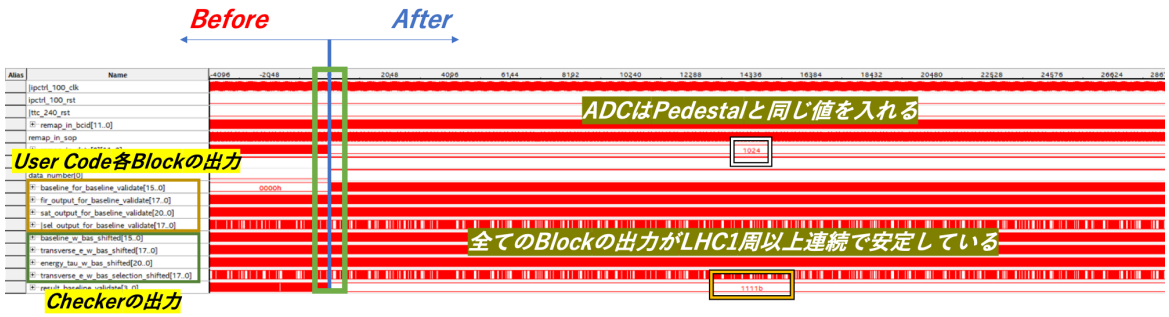


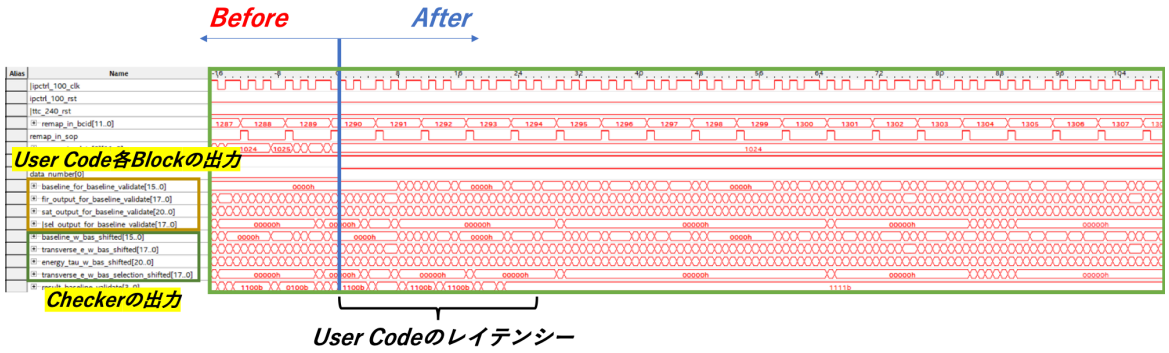
図 E.5: Arria 10 内に実装される Firmware の構造。Checker 内にはそれぞれの Block 用に ROM が設置されておりそこから対応する値を読み出し比較する。

Checker はそれぞれの Block の出力を予測値と比較し、各 Block 毎にその結果を 1 bit の信号として出力する。予測値と一致していた場合'1'、そうでない場合は'0' となる。それら 4 つを合わせた 4 bit の信号が result_baseline_validate として最終的に Checker から出力され Signal Tap で観測される。

$$\text{result\_baseline\_validate} = \{\text{FIR},\ \text{Saturation},\ \text{Selection},\ \text{Baseline の出力結果}\} \tag{E.1}$$



(a) LHC 1 周以上の時間幅の結果



(b) 検証を開始し始めた部分の結果

図 E.6: Arria 10 を用いた User Code の検証結果。(a) は LHC 1 周以上の時間幅で各 Block の出力が Checker 内で一致したことを確認できる。(b) はおおよそ 100 clock 間での User Code の出力の比較を示す。User Code のレイテンシーまで考慮に入れて検証を行った。

図 E.6 のように User Code の各 Block の機能を検証することができた。この試験は 8 時間以上連続で行われ、それぞれの Block の出力は予測値と完全に一致しており動作安定性を確認した。

付録 F

ポワソン分布に従う母集団の信頼区間推定

ポワソン分布に従う集団 $X_\lambda \sim Po(\lambda)$ とカイ二乗分布に従う集団 $Y_n \sim \chi^2(n)$ を用意する。この時 $k \in \mathbb{N}, n = 2(k+1)$ とする。ポワソン分布確率関数 $P(X=i) = \frac{\lambda^i e^{-\lambda}}{i!}$ から

$$P(X_\lambda \leq k) = \sum_{i=0}^{k} \frac{\lambda^i e^{-\lambda}}{i!}$$

一方自由度 k のカイ二乗分布の確率密度関数は $f(x;k) = \frac{1}{2^{\frac{k}{2}}\Gamma\left(\frac{k}{2}\right)} e^{-\frac{x}{2}} x^{\frac{k}{2}-1}$ であるので累積分布関数 $F(x;k)$ は

$$\begin{aligned}
F(x;k) &= \int_0^x \frac{e^{-\frac{y}{2}} y^{\frac{k}{2}-1}}{2^{\frac{k}{2}}\Gamma\left(\frac{k}{2}\right)} dy \\
&= \frac{1}{2^{\frac{k}{2}}\Gamma\left(\frac{k}{2}\right)} \int_0^x e^{-\frac{y}{2}} y^{\frac{k}{2}-1} dy \\
&= \frac{1}{2^{\frac{k}{2}}\Gamma\left(\frac{k}{2}\right)} \int_0^{\frac{x}{2}} e^{-t} (2t)^{\frac{k}{2}-1} 2dt \\
&= \frac{1}{\Gamma\left(\frac{k}{2}\right)} \int_0^{\frac{x}{2}} e^{-t} t^{\frac{k}{2}-1} dt \\
&= \frac{\gamma\left(\frac{k}{2}, \frac{x}{2}\right)}{\Gamma\left(\frac{k}{2}\right)}
\end{aligned}$$

$\gamma\left(\frac{k}{2}, \frac{x}{2}\right)$ は第一種不完全カンマ関数であり以下を満たす。

$$\begin{aligned}
\gamma(a,x) &= \int_0^x t^{a-1} e^{-t} dt \\
&= \left[-t^{a-1} e^{-t}\right]_0^x + (a-1) \int_0^x t^{a-2} e^{-t} dt \\
&= \gamma(a-1,x) - x^{a-1} e^{-x}
\end{aligned}$$

ここでカイ二乗分布 Y_n がポワソン分布 X_λ の強度の 2 倍以上になる確率を考える

$$\begin{aligned}
P\left(Y_n \geq 2\lambda\right) &= 1 - P\left(Y_n < 2\lambda\right) \\
&= 1 - F(2\lambda, n) \\
&= 1 - F(2\lambda, 2(k+1)) \\
&= 1 - \frac{\gamma\left(k+1, \lambda\right)}{\Gamma\left(k+1\right)} \\
&= 1 + \frac{\lambda^k e^{-\lambda}}{k!} + \frac{\lambda^{k-1} e^{\lambda}}{(k-1)!} \cdots - \left(1 - \frac{\lambda^0 e^{-\lambda}}{0!}\right) \\
&= \sum_{i=0}^{k} \frac{\lambda^i e^{-\lambda}}{i!}
\end{aligned}$$

つまり $P(X_\lambda \leq k) = P(Y_n \geq 2\lambda)$ を満たす。

ポワソン母分布から実現値 t が得られた場合、信頼度 $1-\alpha$ で λ の信頼区間を求める。下限、上限はそれぞれ λ_l, λ_u と表すと

$$P(X_{\lambda_l} \geq t) = \frac{\alpha}{2} \Leftrightarrow P(X_{\lambda_l} \leq t-1) = 1 - \frac{\alpha}{2}$$

$$P(X_{\lambda_u} \leq t) = \frac{\alpha}{2}$$

を満たす。ここで $t = x_1 + x_2 + \cdots + x_n$ とし、各 x_i が独立に期待値 λ のポワソン分布に従うとすると、ポワソン分布の再生性から $nX_\lambda \sim Po(n\lambda)$ と書ける。t は nX_λ からの実現値なので

$$P(nX_{\lambda_l} \leq t-1) = 1 - \frac{\alpha}{2}$$

$$P(nX_{\lambda_u} \leq t) = \frac{\alpha}{2}$$

先ほどのポワソン分布とカイ二乗分布の関係を用いると

$$P(nX_{\lambda_l} \leq t-1) = P(Y_{2t} \geq 2n\lambda_l) = 1 - \frac{\alpha}{2} \Leftrightarrow \lambda_l = \frac{\chi^2_{2t}\left(1 - \frac{\alpha}{2}\right)}{2n}$$

$$P(nX_{\lambda_u} \leq t) = P(Y_{2t+2} \geq 2n\lambda_u) = \frac{\alpha}{2} \Leftrightarrow \lambda_u = \frac{\chi^2_{2t+2}\left(\frac{\alpha}{2}\right)}{2n}$$

よってポワソン分布に従う母集団から試行回数 n 回のうち実現値が t であった場合の信頼区間は信頼度 $1-\alpha$ とすれば

$$\frac{\chi^2_{2t}\left(1 - \frac{\alpha}{2}\right)}{2n} \leq \lambda \leq \frac{\chi^2_{2t+2}\left(\frac{\alpha}{2}\right)}{2n}$$

と表すことができる。ここで自由度 n のカイ二乗分布の上側 $100\alpha\%$ となる点を $\chi^2_n(\alpha)$ と置いた。