

平成 26 年度修士論文

ATLAS 実験における FTK 受信モジュールの
ハードウェアエミュレーションを用いた性能評価

早稲田大学
先進理工学研究科 物理及応用物理学専攻
寄田研究室

白神賢

2015 年 2 月 7 日

概 要

欧州原子核研究機構 (CERN) では、大型ハドロン衝突型加速器 (LHC) を用いた新粒子・新現象の探索お行っている。ATLAS 検出器は LHC に設置されている汎用型検出器であり、2012 年には 8TeV での稼働を終え、現在はシャットダウン期間として、2015 年からの再稼働に向けて重心系エネルギーやルミノシティを増強するための改良が行われている。今後は、ヒッグス粒子の性質測定や電弱対称性の破れの測定の検証、さらには、標準理論を超える物理現象の探索を目指す。

陽子・陽子衝突では、クォークやグルーオン同士の散乱により反応が複雑になるため、トリガーが重要になる。ATLAS では、LVL1 Trigger、High Level Trigger(HLT) の 2 段階でトリガーを行っている。今後のビーム輝度増強に伴う高輝度環境下では、衝突粒子数増加への対策や第 3 世代粒子の同定等には飛跡情報が不可欠であるが、現行のトリガーでは HLT の処理時間を割いて飛跡情報を算出しており、今後、処理能力を圧迫して行くと考えられるため、改善は急務である。

FTK は ATLAS 増強計画の 1 つとして、LVL1 Trigger と HLT の間に挿入されるハードウェアシステムである。シリコン検出器における LVL1 Trigger を通過した事象のうち、横運動量が 1GeV 以上の全飛跡情報を HLT へと供給する。ハードウェア処理によって短時間で飛跡を再構成できる。HLT では FTK の飛跡情報を用いることで HLT の処理時間を使用して飛跡再構成する必要がない分、より洗練されたアルゴリズムを採用することが可能となる。

FTK システム最上流に位置する受信モジュール Input Mezzanine(IM) は、シリコン検出器から 100kHz のイベントレートで Hit 情報を受信し、クラスター化と FTK データ形式への変換を行った後、後段のシステムへと送信する。これらの処理は FPGA にファームウェアによって実装される。2014 年 3 月に最終版プロトタイプを作製し、早稲田および CERN にてハードウェアテストを始めとした各種テストを行った。2015 年 2 月から IM80 台の量産を開始し、量産された全 IM は早稲田にてクオリティーコントロール試験を行った後に CERN へと郵送し、順次挿入していく。

2015 年からのインストールに向けて、今後のファームウェア開発を見据えた IM ハードウェアエミュレーションの構築を行った。さらに、IM 最終版プロトタイプを使用した IM アウトプットの詳細な検証を行った。ハードウェアエミュレーションを構築するにあたって、主に下記の昨日を実装した。

・FTK データ形式への変換

ビーム輝度増強に伴い、シリコン検出器におけるヒット占有率が増加するため、Run number によって Read Out Driver(ROD) に接続するシリコンモジュールの変更を行う。FTK ではこれに対応するため、Run number に関係なく一貫して FTK 共通のモジュール情報を定義しており、IM において変換を行う。・SCT クラスタリング

IM が受け取る SCT の Hit 情報は、(1)Module 情報、(2)Hit が存在した読み出しチップおよびストリップ情報、(3)Hit の Time Over Threshold (ToT) 情報である。これらを FTK は 32bit 幅のデータで受け取る。FTK ではこれらの情報をより圧縮するため、隣接する Hit をクラス

ターとして扱い、(1) クラスターの中央値、(2) クラスターの大きさのみを後段に渡す。これにより、情報量を平均して 2/3 程度に減らすことができる。

これらをはじめとした IM の各機能をハードウェアエミュレーションに実装するとともに、エミュレーション前/後のデータを、IM が扱う 32bit 幅のデータ形式へと変換した。さらに、ハードウェアエミュレーションから作製した IM インプットデータを、IM 最終版プロトタイプの入力データとして用いて、実機の IM アウトプットデータを取得し、これとハードウェアエミュレーションから作製した IM アウトプットデータとの比較を行った。

目次

第 1 章 序論	6
第 2 章 LHC-ATLAS 実験	8
2.1 LHC	8
2.2 ATLAS 検出器	11
2.2.1 検出器全体像	11
2.2.2 ATLAS 座標系	12
2.2.3 内部飛跡検出器	12
2.2.4 ソレノイド磁石	16
2.2.5 カロリメータ	17
2.2.6 トロイダル磁石	17
2.2.7 ミューオン検出器	17
2.3 ATLAS トリガーシステム	19
2.3.1 トリガー全体像	19
2.3.2 LVL1 トリガー	19
2.3.3 High Level Trigger	20
第 3 章 FTK システム	21
3.1 FTK システム全体像	21
3.2 FTK 導入の効果	23
3.3 動作原理	23
3.3.1 Road Finder	24
3.3.2 Track Fitter	24
3.4 FTK のハードウェア構成	25
3.4.1 FTK ハードウェア全体像	25
3.4.2 Dual Output HOLA	26
3.4.3 Input Mezzanine	27
3.4.4 Data Formatter	27
3.4.5 Processor Unit AUX	30
3.4.6 SSB	34
3.4.7 FLIC	35

第 4 章	FTK 受信モジュールの開発構築およびテスト	38
4.1	Input Mezzanine 概要	38
4.2	IM への要請	39
4.3	IM の開発	39
4.3.1	ハードウェア	39
4.3.2	ファームウェア	48
第 5 章	ハードウェアエミュレーションによる IM の性能評価	51
5.1	ハードウェアエミュレーションの構築	51
5.1.1	データ形式の変換	51
5.1.2	クラスタリング	55
5.1.3	モジュール情報の変換	59
5.2	テストスタンド	60
5.3	テスト結果	60
第 6 章	まとめと展望	63
第 7 章	謝辞	65

図 目 次

2.1	LHC の概念図	8
2.2	瞬間最高ルミノシティの推移	9
2.3	ATLAS で取得された積分ルミノシティの推移	10
2.4	ATLAS の前段加速の概要	10
2.5	ATLAS 検出器の断面図	11
2.6	ATLAS 検出器の粒子の検出概念	12
2.7	内部飛跡検出器の概念図	13
2.8	内部飛跡検出器の断面図	13
2.9	Pixel 検出器の断面図	14
2.10	pixel モジュール	15
2.11	SCT バレルモジュール	16
2.12	SCT エンドキャップモジュール	16
2.13	SCT エンドキャップのディスクごとのポジションとモジュール数	17
2.14	カロリメータの全体図	18
2.15	ミューオン検出器の概念図	18
2.16	トリガーシステムの概要	20
3.1	ATLAS トリガーシステムと FTK	22
3.2	FTK の概念図	22
3.3	FTK とオフラインの分解能の比較	23
3.4	衝突点の個数ごとの FTK の飛跡再構成率	24
3.5	パターン認識の概念図	25
3.6	FTK の開発機関	26
3.7	HOLA	26
3.8	Dual Output HOLA	27
3.9	FTK 並列処理における ϕ 方向の分割	28
3.10	FTK 並列処理における η 方向の分割	28
3.11	DF における FTK $\eta - \phi$ tower 間のデータシェアリング回数	29
3.12	DF における FTK $\eta - \phi$ tower 間の接続の概念図	29
3.13	DF のブロックダイアグラム	30
3.14	Data Formatter	31
3.15	AUX のブロックダイアグラム	32
3.16	AM ボードと AUX	34
3.17	LAMB のデータフロー	34

3.18	AMchip の動作原理	35
3.19	SSB の機能	36
3.20	FLIC のブロックダイアグラム	36
3.21	FLIC	37
4.1	IM version 3.3	38
4.2	IM board design	40
4.3	Spartan 6 FPGA(FGG484LX150T) のピンとその機能	43
4.4	Spartan 6 FPGA(FGG484LX150T) のバンク構造	44
4.5	IM version 1.0	44
4.6	IM version 3.1	45
4.7	IM version 3.2	45
4.8	IM version 3.3	46
4.9	X 線を用いた FMC の撮影 (1)	46
4.10	X 線を用いた FMC の撮影 (2)	46
4.11	実体顕微鏡を用いた FMC の撮影 (1)	47
4.12	実体顕微鏡を用いた FMC の撮影 (2)	47
4.13	IM version 4.0	47
4.14	IM FW のブロック図	49
4.15	IM FW の FIFO の概念図	50
4.16	クラスタリングファームウェアのブロックダイアグラム	50
5.1	ハードウェアエミュレーションの概念図	52
5.2	Clustering IM Output	55
5.3	SCT Clustering の例	56
5.4	Pixel module の FE の読み出し順序	57
5.5	Moving Window を用いた Pixel クラスタリングの例	58
5.6	Pixel クラスタリングにおける Centroid Calculation の例	59
5.7	FTK データ形式への変換の概念図	59
5.8	ハードウェアエミュレーションのハードウェアテストスタンド	60
5.9	各データフォーマットにおける 1event 中の Hit word 数のヒストグラム	61
5.10	クラスターサイズのヒストグラム	61

表 目 次

2.1	LHC のデザインパラメータ	9
3.1	69Pileup 時の WH event における AUX が受信する各層ごとの Hit 数	33
3.2	AMchip のプロトタイプ毎の各パラメーター	35
4.1	シリコン検出器 ROD の Module 数とファイバーの数	39
4.2	IM の電源供給	41
4.3	IM PCB layer と厚さ	42
4.4	Quality Control 試験の主な項目	48
4.5	Spartan6(XC6SLX150T) と Artix7(XC7A200T) のリソース数	49
4.6	開発中の Pixel Clustering の各リソースの占有率	50
5.1	ROD Header の内訳	52
5.2	ROD SCT Data Format	53
5.3	ROD Pixel Data Format	54
5.4	ROD Trailer の内訳	54
5.5	Clustering 後の data format	55
5.6	600event における 1event 中の平均 Hit word 数	60

第1章 序論

スイス・フランス国境に位置する欧州原子核研究機構（CERN）では、大型ハドロン衝突型加速器（LHC）を用いて陽子同士を衝突させ、生成される粒子の中から新粒子・新現象を探索する実験が行われている。ATLAS 検出器は、LHC の衝突点の一つに設置されている汎用型検出器であり、2012 年には 8TeV で稼働を終え、現在はシャットダウン期間として、2015 年からの再稼働に向けて重心系エネルギーやルミノシティを増強するための改良が行われている。今後は、ヒッグス粒子の性質測定や電弱対称性の破れの検証、さらには標準理論を超える物理現象の探索を目指す。

陽子・陽子衝突であるハドロンコライダーでは、構成要素のクォークやグルーオン同士の散乱によって反応が複雑になる。従って、興味のある物理事象を効率よく取得するトリガーシステムの役割が重要になる。ATLAS 実験では、Level 1 Trigger、High Level Trigger (HLT) の 2 段階のトリガーシステムが採用されている。今後のビーム増強に伴う高輝度環境下では、衝突粒子数増加への対策や第 3 世代粒子の同定等には飛跡情報が不可欠だが、現在のトリガーシステムでは、HLT の処理時間の多くを使用して飛跡再構成を行っているため、今後、トリガーの処理能力を圧迫していくと考えられている。今後のさらなる物理探索に向けてトリガーシステムの改善は必要不可欠であり、高速飛跡トリガー (Fast TracKer:FTK) の開発が薦められている。

FTK は ATLAS の増強計画の一つとして、Level 1 Trigger と HLT の間に挿入されるハードウェアシステムである。ATLAS 内部飛跡検出器のシリコン検出器における Level 1 Trigger を通過した事象のうち、横運動量が 1GeV 以上の全粒子の飛跡情報を HLT へと供給する。ハードウェア処理を行うことで、短時間での飛跡再構成が可能となっている。HLT では FTK の飛跡情報を用いることで HLT の処理時間を使用して飛跡再構成する必要がない分、より洗練されたアルゴリズムを採用することが可能となる。

FTK システム最上流に位置する受信モジュール Input Mezzanine (IM) は、シリコン検出器から 100kHz のイベントレートでヒット情報を受信し、クラスター化と FTK データ形式への変換を行った後、後段のシステムへとデータを送信する。これらの処理は IM に搭載された FPGA 上に Firmware によって実装される。IM は早稲田大学が主担当で開発を行っている。2014 年 3 月に最終版プロトタイプを作製し、早稲田および CERN にてハードウェアテストを始めとした各種テストを行った。2015 年 3 月から IM80 台の量産を開始する予定で、量産された全 IM は早稲田にてクオリティーコントロール試験を行った後に CERN へと郵送し、順次インストールを行う予定である。

IM の開発に携わる中、とくに、今後の Firmware 開発や 2015 年からのインストールに向けて、IM ハードウェアエミュレーションの構築を行い、IM 最終版プロトタイプを用いた IM アウトプットの詳細な検証を行った。具体的には、IM の機能をハードウェアエミュレーションに

実装すると共に、エミュレーション前/後のデータを、IM が扱う 32bit 幅のデータ形式へ変換し、さらに、ハードウェアエミュレーションから作製した IM インプットデータを、最終版プロトタイプの IM インプットデータとして用いて実機の IM アウトプットデータを取得し、これとハードウェアエミュレーションから作製した IM アウトプットデータとの詳細な比較を行った。

第2章 LHC-ATLAS実験

私は ALTAS 実験の中でも 2015 年から一部領域で実装が予定されている高速飛跡トリガー (Fast TracKer:FTK) の研究・開発に携わっている。本章では、前提知識となる LHC および ATLAS 実験、ATLAS 実験の今後について概説する。

2.1 LHC

Large Hadron Collider(LHC) は欧州合同原子核研究機構 (CERN) にある、世界最大の陽子・陽子衝突型円形加速器である。スイスジュネーブ郊外の地下 100m に設置され、周長は 27km におよび、スイスとフランスの国境をまたいで建設されている。2000 年に運転終了した電子・陽電子衝突型シンクロトロン Large Electron Positron Collider(LEP) で使用していたトンネル内に設置されている。陽子は電子に比べて質量が約 1800 倍と大きく、加速により生じるシンクロトロン放射による運動エネルギー損失が小さい。従って、超伝導加速空洞と偏向用超伝導電磁石の採用により高エネルギー領域まで加速が可能となっている。

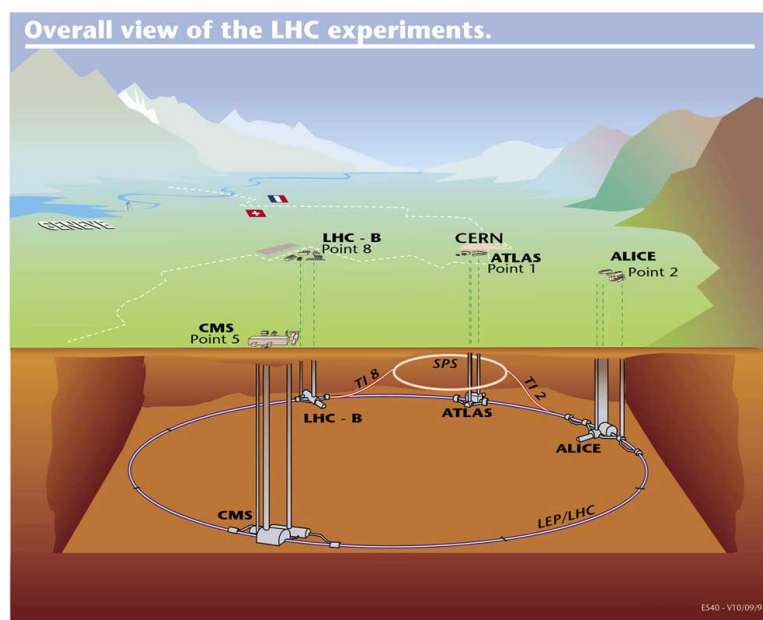


図 2.1: LHC の概念図

主リング長	27km
重心系エネルギー	14TeV
瞬間最大ルミノシティ	$3 \times 10^{34} \text{cm}^{-2} \text{s}^{-1}$
陽子衝突頻度	40MHz
バンチ数	2808 個
バンチあたりの陽子数	1.15×10^{11} 個
バンチ衝突あたりの陽子の衝突回数	23 回

表 2.1: LHC のデザインパラメータ

Fig.2.1 に LHC の概念図を、Table.2.1 に LHC の主なデザインパラメータを示す。重心系エネルギーとルミノシティはまだデザイン値に達しておらず、3 回のアップグレードを経て改善していく方針である。2009 年に稼働を開始し、2010 年から 2013 年までの稼働 (Run1) においては、重心系エネルギー最大 8[TeV]、瞬間最大ルミノシティ $0.7 \times 10^{34} [\text{cm}^{-2} \text{s}^{-1}]$ 、バンチ衝突頻度 20[MHz] で運用されていた。2012 年の運転では、LHC の積分ルミノシティ 23.3fb^{-1} に対して、 21.7fb^{-1} の高い効率でのデータ取得に成功した。その後、2013 年 2 月から 2015 年までの期間は、Run2 に向けた加速器および検出器のアップグレードのためシャットダウン期間となっている。2015 年から再稼働予定の Run2 では、重心系エネルギー 13[TeV]、瞬間ルミノシティ $1 \times 10^{34} [\text{cm}^{-2} \text{s}^{-1}]$ 、バンチ衝突頻度 20[MHz] での運用が予定され、その後順次、バンチ衝突頻度 40[MHz] まで増加させ、瞬間ルミノシティを増加させて行く予定である。最終的に瞬間ルミノシティは $2 \times 10^{34} [\text{cm}^{-2} \text{s}^{-1}]$ まで増加させる予定となっている。Fig.2.2 に 2010 年から 2012 年の期間に ATLAS で取得した瞬間最大ルミノシティの推移を、Fig.2.3 に 2012 年での ATLAS で取得された積分ルミノシティの推移を示す。

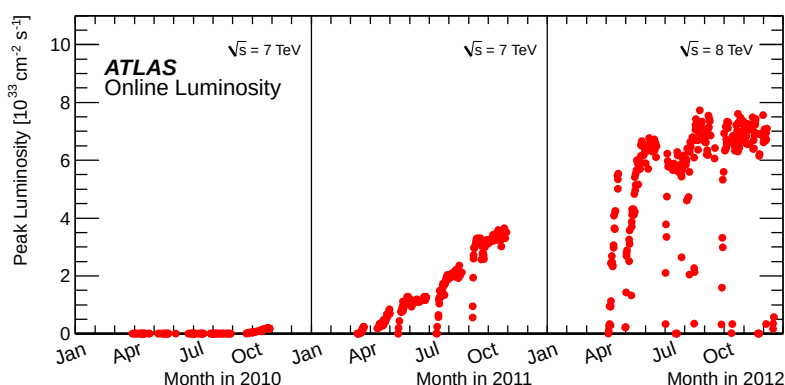


図 2.2: 瞬間最高ルミノシティの推移

LHC では、複数の前段加速器を設け段階的にエネルギーを増加させていく。まず、陽子イオン源から出力された陽子イオンは線形加速器の Linac2 で加速され、Proton Synchrotron(PS) に入射する前に、PS Booster と呼ばれるシンクロトロンで 1.4[GeV] まで引き上げられる。そ

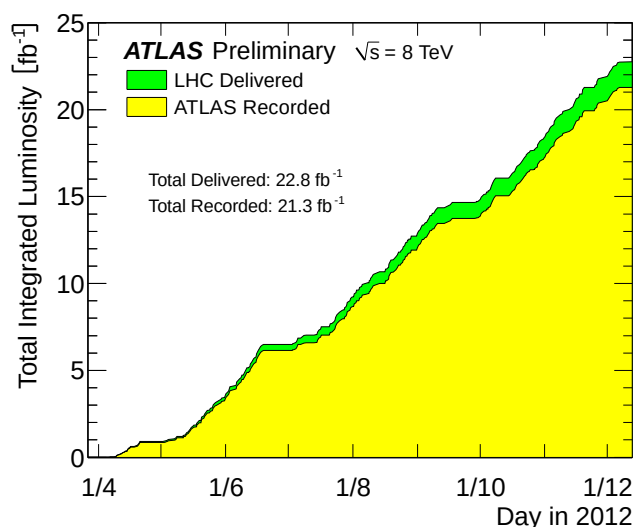


図 2.3: ATLAS で取得された積分ルミノシティの推移

の後、PS、Super Proton Synchrotron (SPS) により、エネルギーを 450[GeV] まで増加させ、その後、LHC へと入射する。概要を Fig.2.4 に示す。

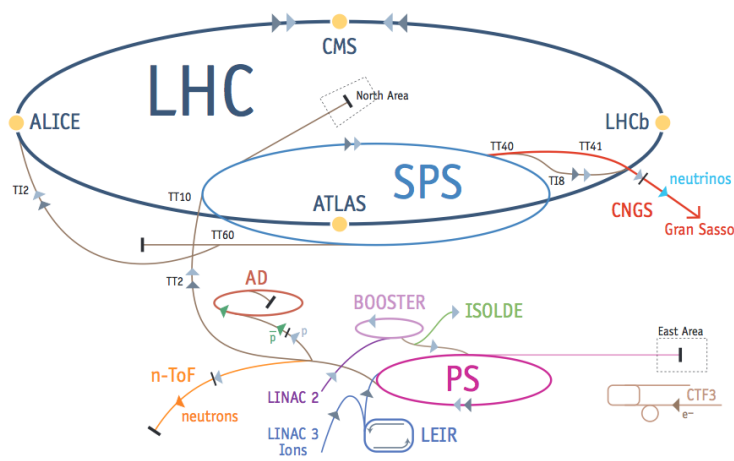


図 2.4: ATLAS の前段加速の概要

LHC には 4 カ所のビーム衝突点があり、それぞれに異なる検出器が設置されている。A Toroidal LHC ApparatuS(ATLAS) と Compact Muon Solenoid(CMS) はいずれも汎用大型検出器となっており、標準模型の精密測定およびヒッグス粒子や標準模型を超える物理探索を目的としている。A Large Ion Collider Experiment(ALICE) では重イオン同士を衝突させ、宇宙初期のグルーオンとハドロンが入り交じった状態を再現し、Quark Gluon Plasma(QGP) の精製を証明し、この新しい物理状態の性質解明を目指している。Large Hadron Collider beauty(LHCb)

では b クォークの物理に焦点を当て、 B メソンにおける CP 対称性の破れを観測する。

2.2 ATLAS 検出器

ATLAS 実験は LHC の衝突点の 1 つに置かれた ATLAS 検出器を用いて行われる実験である。ATLAS 実験には、3000 人を超える研究者が参加しており、日本からも多数の研究者や学生が実験に参加している。ATLAS 検出器は全方位型の複合型汎用検出器で、陽子衝突点に対して、前方と後方が対象になるように設計されている。全長 44[m]、直径 25[m]、総重量は 7000[t] に及ぶ。内層から、内部飛跡検出器、ソレノイド磁石、電磁カロリメータ、ハドロンカロリメータ、トロイダル磁石、ミューオン検出器で構成されている。

2.2.1 検出器全体像

Fig.2.5 に ATLAS 検出器の全体像を示す。飛跡検出器は後述するソレノイド磁石で曲げられた荷電粒子を検出し、飛跡の再構成を行い運動量を測定する。電磁カロリメータは、電磁シャワーにより電子および光子のエネルギーを測定する。ハドロンシャワーを起こさせる事で、電磁カロリメータ同様にジェットのエネルギー測定を行う。ミューオンは荷電粒子であるが、制動輻射の断面積が小さく、強い相互作用も起こさないため、最外部に設置されたミューオン検出器によって運動量と位置を測定される。粒子の検出概念を Fig.2.6 に示す。ATLAS 検出器はマグネットに特徴があり、一般的なソレノイド磁石に加えてトロイダル磁石を採用し、ミューオン検出器での運動量測定を行っている。これらを総合して様々な衝突反応に対応し、超前方を除いた全立体角をカバーしている。次節以降で、それぞれについて詳細に説明する。

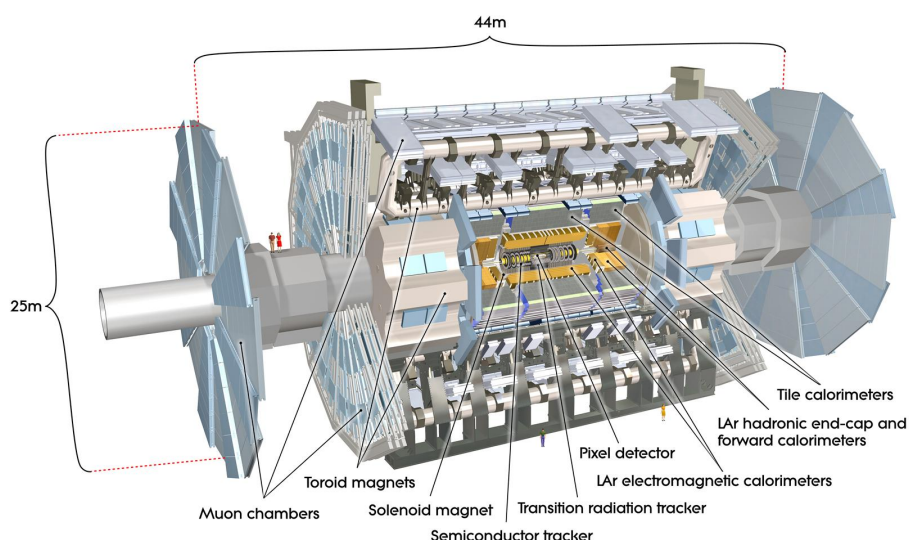


図 2.5: ATLAS 検出器の断面図

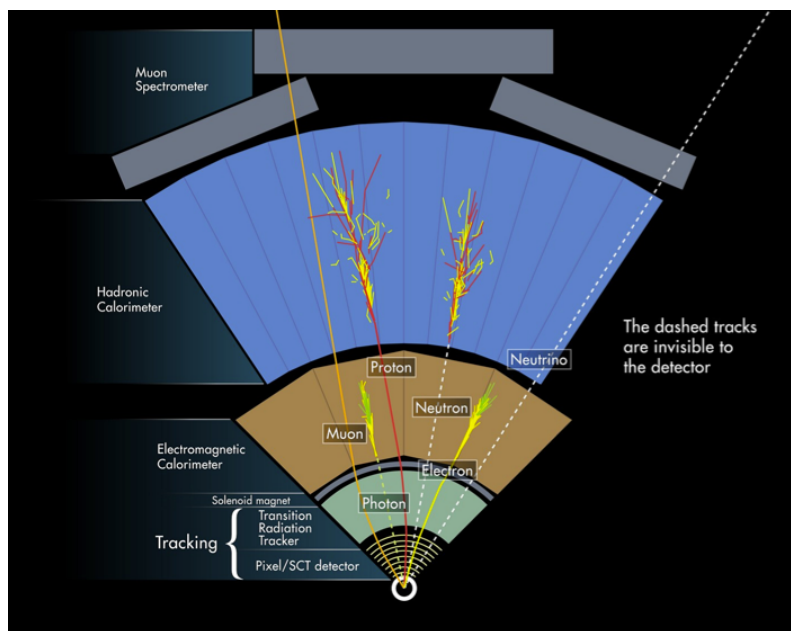


図 2.6: ATLAS 検出器の粒子の検出概念

2.2.2 ATLAS 座標系

ATLAS では主に xyz 座標系または、 $r\phi z$ 座標系を用いる。ビーム軸に沿って主リング半時計回りを正として z 軸を取り、水平方向、主リングの中心に向かって x 軸、鉛直上向きに y 軸を右手系で取る。 $r\phi z$ 座標系では、ビーム軸に垂直な方向を r 方向とし、ビーム軸周りを ϕ 方向 ($-\pi < \phi < \pi$) となるような円筒座標系を用いる。また、原点から引いた線とビーム軸とのなす角を θ ($0 < \theta < \pi$) とし、擬ラピディティ η を、

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

と定義して θ の代わりに用いる。ATLAS では粒子の方向や検出器の位置を ϕ と η を用いて表す。

2.2.3 内部飛跡検出器

内部飛跡検出器は ATLAS 検出器の再内層に位置し、全長 6.2m、直径 2.1m の円筒形をしている。ソレノイド磁石によりビームパイプ方向に 2T の磁場がかけられているため、荷電粒子は ϕ 方向に飛跡が曲げられ、その曲率から運動量・電荷の符号を求めることができる。内側から IBL(Insertable B Layer)、Pixel 検出器、SCT(Semi Conductor Tracker)、TRT(Transition Radiation Tracker) となっている。 $|\eta| < 1.0$ のバレル部と、 $1.0 < |\eta| < 1.9$ のエンドキャップ部に分けられ、それぞれでモジュールの形状やポジションが大きく異なる、Fig.2.7 に内部飛跡検出器の全体像を、Fig.2.8 に内部飛跡検出器の断面を示す。

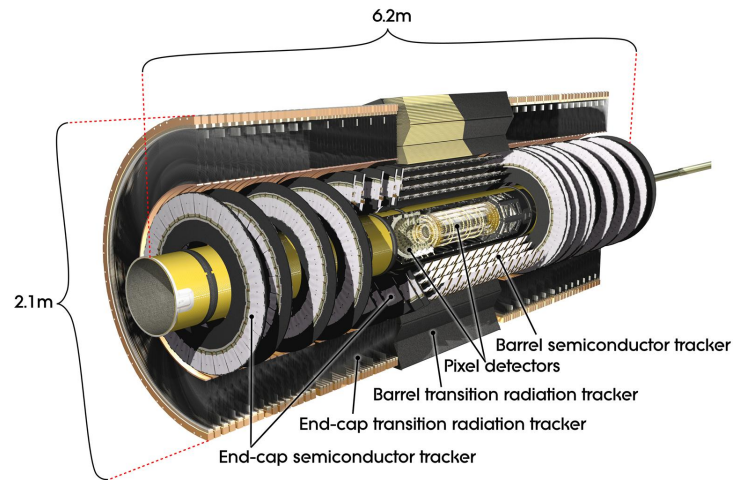


図 2.7: 内部飛跡検出器の概念図

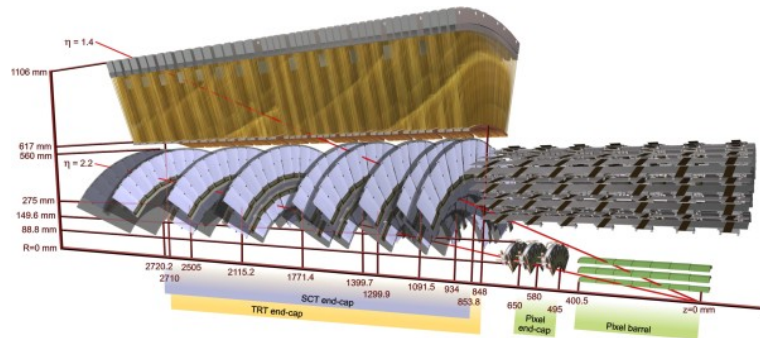


図 2.8: 内部飛跡検出器の断面図

シリコン検出器

ATLAS 検出器最内層である内部飛跡検出器の内層には、飛跡再構成に必要な精度が要求されるため、位置分解能に優れるシリコン検出器を採用する。内層から、Run2 よりインストールされる IBL、Pixel 検出器、SCT の 3 種類がある。

Pixel 検出器

Pixel 検出器は IBL 検出器の外層に配置され、 $|\eta| < 2.5$ の領域をカバーする。バレル部 3 層、エンドキャップ部 3×2 層設置されている。Fig.2.9 に Pixel 検出器の概観図を示す。

バレル部はシリンダー型をしており、ビームパイプに平行になるようにモジュールが配置される。それぞれの層はビーム軸から、半径 50mm、89mm、123mm に配置され、 z 方向に 800mm の長さを持つ。各モジュールは z 方向に 13 枚連なったセクターと呼ばれる固まりで固定され、1 層目の ϕ 方向にはセクターが 22 枚、2 層目には 38 枚、3 層めには 52 枚配置される。隣同士

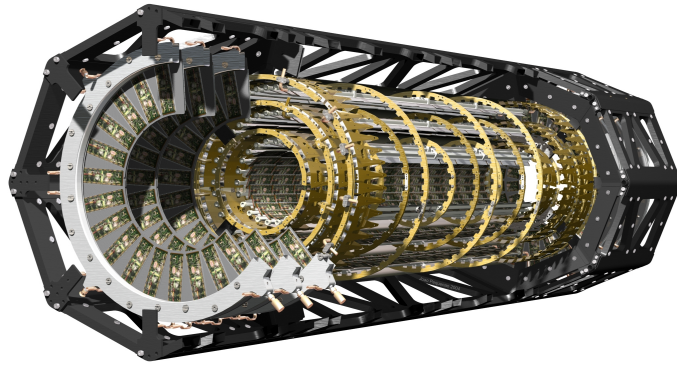


図 2.9: Pixel 検出器の断面図

のセクターはそれぞれ 22 度傾いて固定されている。エンドキャップ部はビーム軸を中心とした円盤状になっており、ビームパイプに垂直になるようにモジュールが配置される。バレルの両端に z 方向 495mm、580mm、650mm、にそれぞれ 3 層ずつ配置される。モジュールが ϕ 方向に表裏 24 枚ずつで 1 周するように固定されており、表裏で互い違いになるように固定することで不感領域がないようにしている。pixel モジュールを Fig.2.10 に示す。1 つのモジュールには $50\mu\text{m} \times 400\mu\text{m}$ の pixel センサーが 47 個配置され、バレル部とエンドキャップ部で 8040 万の読み出しチャンネルが存在する。 $R-\phi$ 平面で $10\mu\text{m}$ 、 z 方向で $115\mu\text{m}$ の非常に高い位置分解能を持っており、飛跡を正確に再構成することができ、主に衝突点や崩壊点の測定に使用される。一つの pixel 内では、逆バイアス電圧によって空乏層が作られ、荷電粒子が通った際に生じる電子正孔対を p 型半導体に収集することで読み出す。

SCT

SCT は Pixel 検出器の外側に位置し、バレル部 4 層、エンドキャップ部 9×2 層からなる。 $|\eta| < 2.5$ の領域をカバーする。pixel 検出器と同様にバレル部ではシリンダー型、ビームパイプに平行になるようにモジュールが置かれている。バレル部のそれぞれの層は、1 層目から半径 300mm、374mm、447mm、520mm に配置され、 z 方向の長さは 1600mm となっている。それぞれのモジュールは z 方向に 12 枚連なって固定され、1 層目には ϕ 方向に 32 枚、2 層目には 40 枚、3 層目には 48 枚、4 層目には 56 枚配置される。1/2 層目は隣同士 11 度、3/4 層目は隣同士 11.25 度傾いている。4 層全てで 2112 モジュール存在する。SCT バレルのモジュールを Fig.2.11 に示す。

SCT バレルモジュールは 4 枚のシリコンセンサーが表裏に 2 枚ずつ、それぞれ 40mrad スライドさせて配置することで、2 次元読み出しが可能となっている。1 枚のシリコンセンサーには、768 本のシリコンストリップセンサーが $80\mu\text{m}$ で平行に配置され、片面 6 個のチップによって読み出される。SCT エンドモジュールの概観図を Fig.2.12 に示す。

SCT エンドキャップモジュールはインナーモジュール、ミドルモジュール、アウターモジュールの 3 種類が存在する。それぞれ、SCT バレルモジュール同様、表裏 2 枚のシリコンセンサー

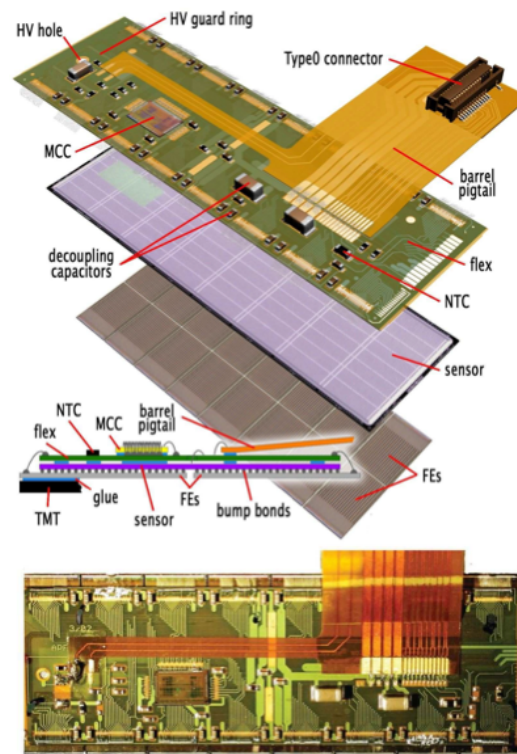


図 2.10: pixel モジュール

が 40mrad ズラして配置され、これによって2次元読み出しが可能となっている。1枚のシリコンセンサーには768本のシリコンストリップが存在し、6個のチップによって読み出される。ストリップ間隔とストリップの長さはインナーモジュール、ミドルモジュール、アウターモジュールによって異なる。エンドキャップ部ではpixelエンドキャップモジュール同様に、モジュールがビーム軸に垂直に、円盤(ディスク)状に配置される。ディスクは、 -854mm から 2720mm 、 854mm から 2720mm の範囲に渡って 9×2 層設置されている。ディスクごとに固定されるモジュール数は異なる。Fig.2.13にSCTエンドキャップ部のディスクごとのポジションとモジュール数を示す。

SCTモジュールでは逆バイアス電圧によって空乏層が作られ、荷電粒子が通った際に生じる電子正孔対をp型半導体に収集することで読み出される。 $R-\phi$ 平面で $17\mu\text{m}$ 、 z 方向で $580\mu\text{m}$ の位置分解能を持っている。

TRT

遷移放射飛跡検出器TRTはSCT検出器の外側、内部飛跡検出器の再外層に配置され、 $|\eta| < 2.0$ の領域をカバーする。TRTは半径 4mm のドリフトストローチューブ検出器を多層に並べた構造となっており、ドリフトチューブは z 方向に設置されているため、 $R-\phi$ 方向のみ位置分解能を持つ。各チューブの位置分解能は $130\mu\text{m}$ となっており、ポリエチレン層を通過する際の遷

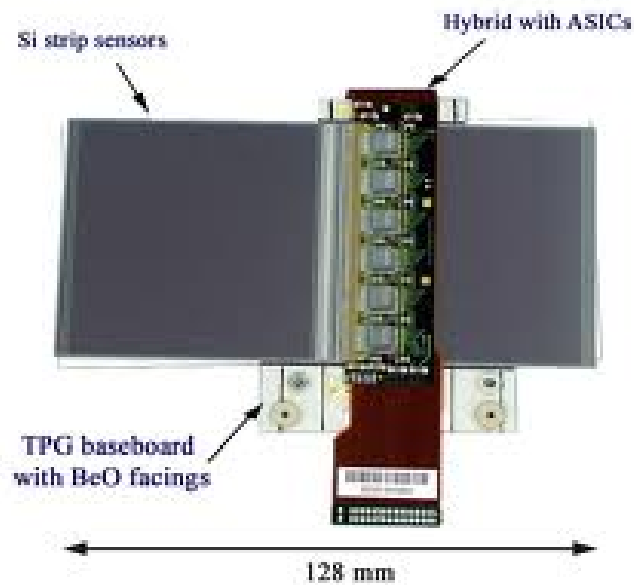


図 2.11: SCT バレルモジュール

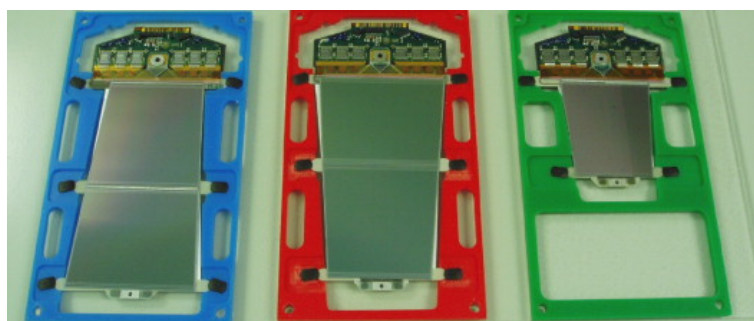


図 2.12: SCT エンドキャップモジュール

移行放射を用いて粒子識別が可能となっている。移行放射とは、電荷を持つ粒子が異なる誘電率を持つ物質へ入射する際に、光子が放出される現象である。放出される光子のエネルギースペクトルは専属的で、そのエネルギーピークは粒子のローレンツ因子 γ に比例する。これにより、入射粒子が荷電粒子の場合、電子かどうか識別することが可能となっている。

2.2.4 ソレノイド磁石

内部飛跡検出器の外側、およびカロリメータの内側のバレル部に設置される。内径 2.46m、外形 2.56m、 z 方向に 5.8m の長さを持つ円筒形をしている。内部飛跡検出器に 2T の磁場をビーム軸方向に作り出し、その磁場によって ϕ 方向に飛跡を曲げる。この曲率から運動量を測定することができる。

2.2.5 カロリメータ

電磁カロリメータ、ハドロンカロリメータからなり、 $|\eta| < 4.9$ の領域をカバーする。 η の範囲によって放射線量や対象とする物理が異なるため、異なる検出器を組み合わせている。カロリメータは主に電子と光子の識別や、ジェットの識別、粒子のエネルギー測定に用いる。

電磁カロリメータは $|\eta| < 3.2$ の領域に設置され、主に電子と光子のエネルギー測定に使用される。バレル部はソレノイド磁石の外側、エンドキャップ部は内部飛跡検出器の外側に設置される。吸収体に鉛、サンプリング層に液体アルゴンを使用し、交互に重ねたアコーディオン構造となっており、隠すは細分化されて位置分解能を高めている。電磁シャワーのエネルギー量等により、粒子の種類識別とエネルギー測定を行う。電子／光子とジェットの識別は電磁仮メータの外側に設置されたハドロンカロリメータへのエネルギーの漏れだしによって行う。また、電子と光子の識別は内部飛跡検出器の飛跡の有無によって行われる。

ハドロンカロリメータは、電磁カロリメータの外側に設置される。バレル部では、鉄の吸収体とプラスチックシンチレータからなり、これらの層を交互に並べたタイルカロリメータを使用している。エンドキャップ部では銅の吸収体と液体アルゴンシンチレータを交互に並べた液体アルゴンカロリメータが用いられ、前後方部 ($|\eta| > 4.9$) では銅とタングステンの吸収体と液体アルゴンからなる。放射線によるダメージが大きいエンドキャップ部と前後方部では放射線耐性が大きく交換のしやすい液体アルゴンが使用されている。主にハドロンと同定、エネルギー測定、ジェットの再構成を行う。カロリメータの全体図を Fig.2.14 に示す。

2.2.6 トロイダル磁石

トロイダル磁石はカロリメータの外側に設置され、バレル部に 0.5T、エンドキャップ部に 1T の磁場をかける。8つのコイルがビーム軸に対して8回対象になるように配置されている。 ϕ 方向に最大で約 8T の磁場をかけることで、ミューオンの運動量測定に用いられる。

2.2.7 ミューオン検出器

ミューオン検出器はトロイダル磁石の外側、ATLAS 検出器の再外層に位置する。ミューオンは物質透過能力が非常に強く、寿命も $2.2\mu\text{s}$ と長いため、ミューオン検出器は再外層に配置されている。トロイダル磁石により、バレル部では 0.5T、エンドキャップ部では 1T の磁場がかけられているため、飛跡は R-z 平面で曲げられる。これにより、ミューオン検出器は横運動量測定が可能となっている。ミューオン検出器は位置の精密測定を行う Monitored Drift Tube(MDR)、

Disk	1	2	3	4	5	6	7	8	9
z (mm)	853.8	934.0	1091.5	1299.9	1399.7	1771.4	2115.2	2505.0	2720.2
Outer	52								
Middle	40								None
Inner	None	40					None		

図 2.13: SCT エンドキャップのディスクごとのポジションとモジュール数

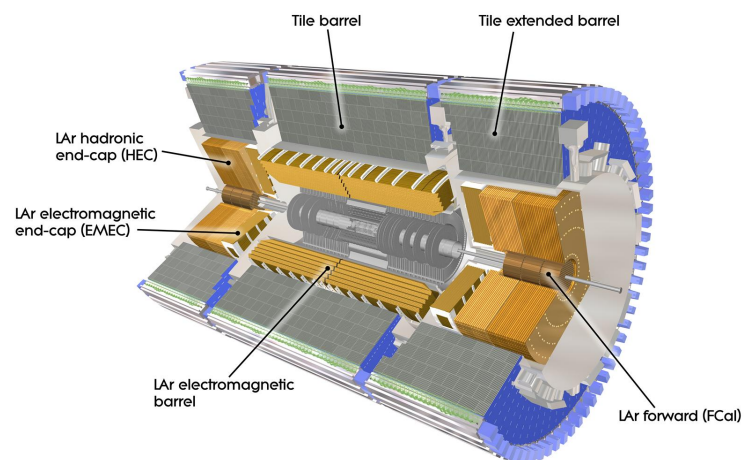


図 2.14: カロリメータの全体図

Cathod Strip Chamber(CSC) とトリガー発行を行う Resistive Plate Chamber(RPC)、Thin Gap Chamber(TGC) の4種対で構成される。TGC、RPC はトリガー発行のために設置された検出器であり、MDT に比べて位置分解能は小さいが、読み出し速度は速くなっている。ミューオンの質量は電子の役 200 倍と多きいため、強い相互作用を起こさず、制動輻射も起こさない。物質の透過力が高いためミューオン検出器は再外層に設置されている。ミューオン検出器の全体像を Fig.2.15 に示す。

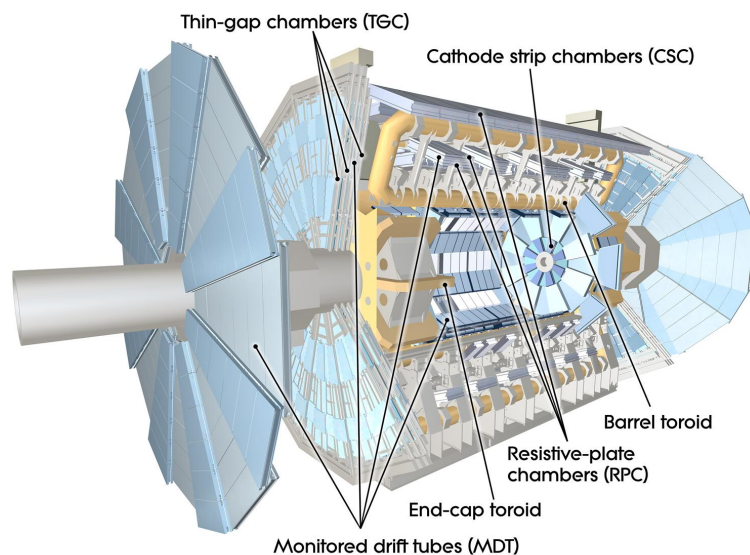


図 2.15: ミューオン検出器の概念図

2.3 ATLAS トリガーシステム

高エネルギー実験、特に非常に稀な事象を探索する必要がある ATLAS 実験にとって、トリガーシステムとデータ収集システム (Trigger and Data Acquisition:TDAQ) の性能は非常に重要である。14[TeV] の重心系エネルギーで陽子・陽子衝突を行った場合、Higgs 粒子は 10 億回に 1 回程度でしか生成されない。従って、いかに膨大なバックグラウンドの中から効率よく興味のある事象を選択するかが重要である。本節では ATLAS 実験の TDAQ について説明する。

2.3.1 トリガー全体像

LHC では 40MHz の頻度でバンチ衝突が起き、1 回のバンチ交差あたり平均 23 回の陽子・陽子衝突が起きると予想されている。従って、ATLAS 検出器の衝突点で起こる陽子・陽子衝突は 1GHz にもなると考えられ、1 事象あたりのデータ量は約 1.5MByte と見積もられているため、1 秒間あたりのデータ量は約 1.5PByte にもなる。これらの全情報を記録するのは現在の技術では困難であり、またこれらの事象の大多数は QCD 事象などの物理的に意味のない事象となっているため、ATLAS 実験ではトリガーシステムを用いてリアルタイムで事象の選別、保存を行っている。最終的なイベントレートはデータストレージの書き込み速度の制約から、200Hz まで落とす。ATLAS 実験ではトリガーシステムを、LVL1 トリガー、HLT(High Level Trigger) の二段階に分けており、LVL1 トリガーでは 75kHz に、HLT で 200Hz まで順次イベントレートを下げている。最終的なデータは CERN の計算機センターのストレージに保存される。LVL1 トリガーは 2.5[μ s] という厳しいレイテンシが要請されるため、ハードウェアでの処理が行われ、HLT ではソフトウェア処理を行っている。Fig.2.16 トリガーシステムの概要図を示す。

2.3.2 LVL1 トリガー

Lvl1 トリガーは、40MHz の膨大な事象を 75kHz(100kHz までアップグレード可能) まで落とす。ハードウェアによる処理を行っているため高速なデータ処理が可能となっている。処理時間の大きくかかる内部飛跡検出器のからの情報は使用せずに、ミューオンの P_T 情報および電子、光子、ジェット等のエネルギー情報を用いて行われる。 P_T 情報は TGC、RPC によって提供され、エネルギー情報は電磁カロリメータ、ハドロンカロリメータから精度を落としたものが提供される。これらの検出器と、情報を統合してトリガー判定を行う Central Trigger Processor(CTP)、さらにトリガーを分配する Timing Trigger and Control distribution system(TTC) で LVL1 トリガーシステムが構成されている。イベントレートは 25ns だがパイプライン処理により 128 イベント分保持出来るため、 $25\text{ns} \times 128 = 32\mu\text{s}$ の間、全情報を保持することが出来る。LVL1 トリガーでは 2.5 μs の間に処理することが決められている。LVL1 Accept(L1A) は TTC によりフロントエンドの LVL1 Buffer に届けられ、その後、Read Out Driver(ROD) に送信される。ROD までを各検出器システムが担当し、ROD はデータを ATLAS 共通のフォーマット (ROD フォーマット) に変換し、後段の Read Out System(ROS) に送信する。ROS は複数の Read Out Buffer(ROB) によって構成され LVL2 の処理の間、データを保持している。また、このデータフローとは別に Region Of Interest(ROI) と呼ばれる、LVL1 トリガーで得られた粒子の存在領域を $\eta\phi$ で表した情報が HLT に送られる。

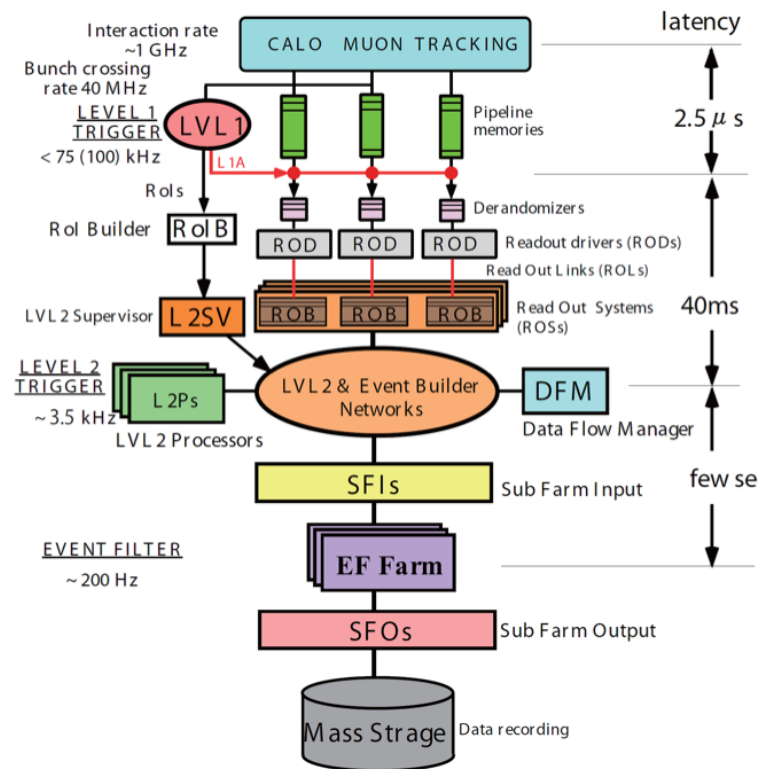


図 2.16: トリガーシステムの概要

2.3.3 High Level Trigger

HLT は、ソフトウェアを用いたトリガーである。LVL1 トリガーによって与えられる Region Of Interest(ROI) を用いて、注目する検出器の領域を絞り込む事で、短い演算時間の中でも精度よく選別を行っている。LVL1 で用いられなかった MDT、CSC、ID 等の情報にアクセスでき、LVL1 より高精度なトリガー判別を行うことができる。ただし、40[ms] のレイテンシ制約があるため、Region Of Interest(ROI) 情報を基に適切なデータを Read OutSystem(ROS) から取り出している。

第3章 FTK システム

高速飛跡トリガーシステム (Fast TracKer:FTK) は ATLAS のアップグレード計画の一つで、LVL1 と HLT の間に挿入されるハードウェアシステムである。内部飛跡検出器から、LVL1 トリガーを通過した全事象のヒット情報を受け取り、 P_T が 1GeV 以上の飛跡を再構成し HLT へと出力する。2015 年から始まる Run2 での瞬間最大ルミノシティは $3 \times 10^{34} \text{cm}^{-2}\text{s}^{-1}$ に到達する予定で、このような高ルミノシティ下では QCD 事象が爆発的に増加し、現状のトリガーシステムでは処理能力に困難が生じる考えられ、より洗練されたトリガーシステムが必要不可欠である。この問題を解決するために、FTK の開発が行われている。FTK を実装することにより、HLT では、トリガーの初期の段階から FTK で再構成された飛跡情報にアクセスすることができ、HLT の処理時間を使用して飛跡を再構成する必要がある分、より洗練されたトリガーアルゴリズムを採用することができ、より質の高いトリガー判定を行うことが可能になる。FTK は限られた時間の中で全内部飛跡検出器情報を処理しなければならないため、FTK は検出器ごとに分割して並列処理を行っている。 ϕ 方向に 22.5 度ずつの 16 分割、 η 方向に 4 分割し、計 64 の並列処理を行っている。 η 方向は衝突点の z 方向の幅を考慮し、 ϕ 方向には、 P_T が低い粒子の曲率を考慮して 10 度のオーバーラップを持たせている。分割された各々の領域を $\eta - \phi$ tower と呼ぶ。本章では FTK の動作原理とハードウェアの各機能の詳細を説明する。

3.1 FTK システム全体像

FTK は LVL1 トリガーと HLT の間に挿入され、内部飛跡検出器の 100M チャンネルを扱う。FTK はこれらのヒット情報を $100\mu\text{s}$ 以内で処理し、 P_T が 1GeV 以上の飛跡のトラックパラメータ (飛跡の Impact parameter である d_0 、飛跡方向を表す ϕ/η 、 xy 方向の運動量 P_T 、 z 方向の Primary Vertex である z_0) を再構成し、HLT の PC ファームに事象あたり約 2.5kByte の大きさで出力する。現行のトリガーシステムで使用しているデータフローを遮らないように、本流からコピーされたデータを受け取る。これにより、万が一 FTK で飛跡が再構成できずにエラーと処理されても、本流のデータフローは影響を受けないため、現行通り HLT でトリガー判定ができる。Fig.3.1 に ATLAS トリガーシステムと FTK の概要を示す。

現行の HLT で採用されているトラッキングをはじめ、飛跡再構成の処理には非常に時間がかかる。飛跡検出器からのヒット情報は、粒子の通過点の情報にすぎないため、膨大なヒットから飛跡を再構成するのは困難である。とくに、膨大なヒットの中からヒットを組み合わせて飛跡を算出するため、ヒットの組み合わせは、検出器の層が増えるごとに乗じて増加するため、非常に莫大な数の計算が必要になる。さらに、ビーム輝度増強に伴い、ヒットの占有率が上がるにつれて、乗算でヒットの組み合わせが増加する。このような問題が飛跡再構成を難しくする要因となっている。そこで、FTK では、予め飛跡パターンを専用のチップに格納しておき、ヒッ

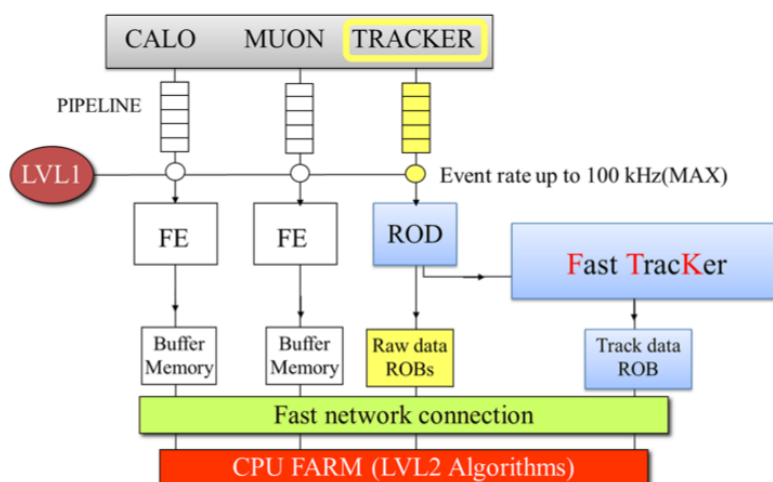


図 3.1: ATLAS トリガーシステムと FTK

トを受信するたびに、これを参照して一致する飛跡パターンを算出することでこれらの問題に対処している。さらにこれらの処理は並列化されたハードウェアで処理されるために、非常に高速に再構成することができる。

Fig.3.2 に FTK の概念図を示す。

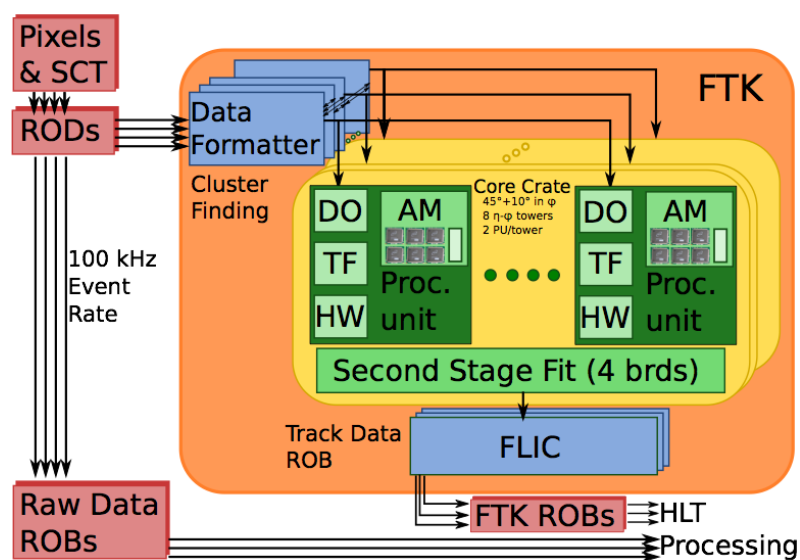


図 3.2: FTK の概念図

3.2 FTK 導入の効果

Fig.3.3 に示すように、FTK が出力する飛跡情報は HLT の始まる段階で使用できるにも関わらず、現行のオフラインでの飛跡情報と比較しても非常に質の高い飛跡を供給できることがわかる。また、Fig.3.4 にしめるように、衝突点の個数が大きくなっても FTK の飛跡再構成率は一定して高い値を示しており、今後のビーム輝度像に伴う高輝度環境下において、安定的に飛跡を供給できることがわかる。このような高い質の飛跡情報を HLT 以前で高速に再構成することで、現行の HLT のように飛跡情報を HLT で再構成しなくて済む分、その CPU パワーと処理時間を他のアルゴリズムに使用することができる。とくに FTK の飛跡情報を用いて、衝突点を再構成する事により、事象中の粒子の一つ一つの発生点を把握することができ、今後の高輝度環境下において予想される pile-up の影響を効率的に取り除くことができると考えられる。

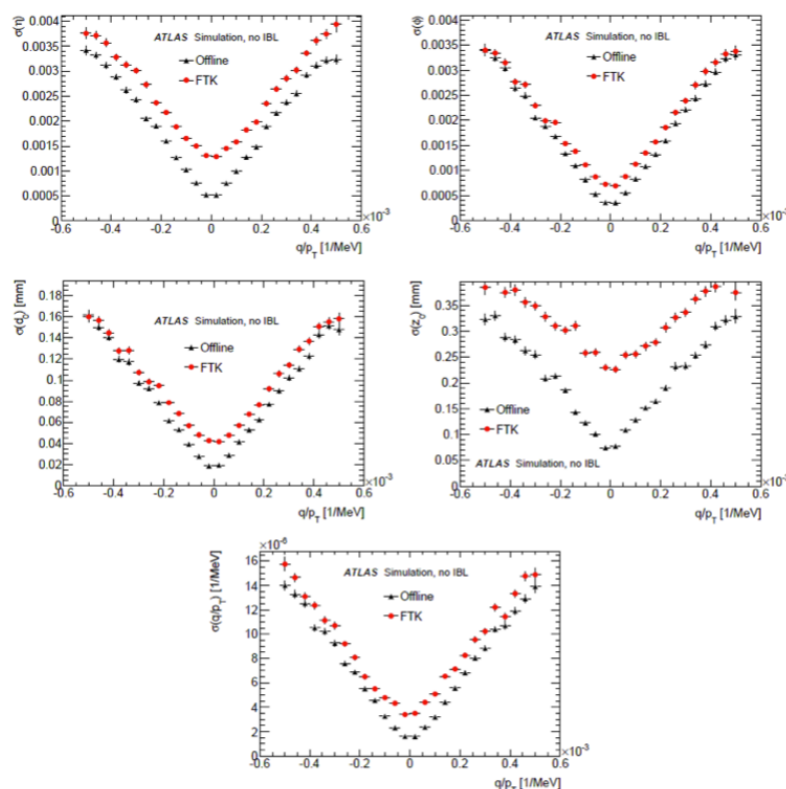


図 3.3: FTK とオフラインの分解能の比較

3.3 動作原理

FTK は並列処理によるパターン認識のアルゴリズムを採用することで高速に飛跡を再構成する。FTK の動作原理は主に、Road finder と Track Fitter の 2 段階に分けられる。

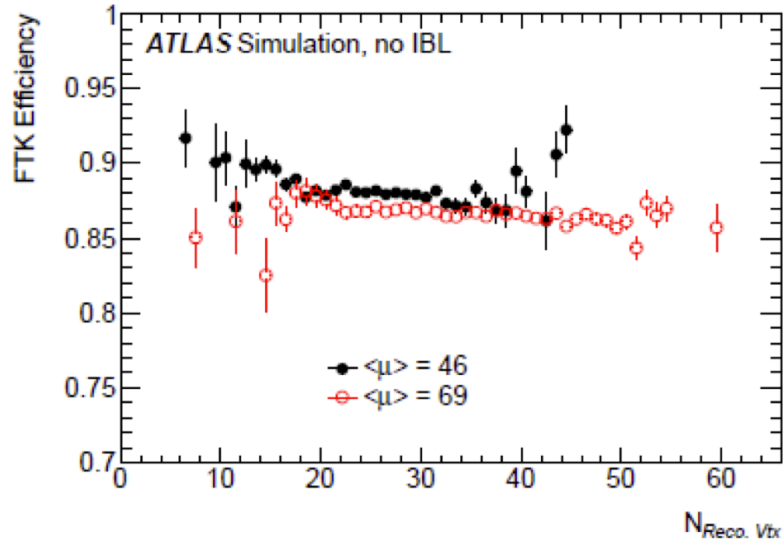


図 3.4: 衝突点の個数ごとの FTK の飛跡再構成率

ここでは Road Finder と Track Fitter についてのおおまかな動作原理を述べる。詳細な動作については以下の各ハードウェアにて述べる。

3.3.1 Road Finder

FTK では、resolution を落とした粗いヒット情報と、予め用意しておいた飛跡パターンとを比較することにより、Road と呼ばれる飛跡パターンを算出している。この飛跡パターンは大量の Mini-Asic chip に予め格納しておく。とくに格納する飛跡パターンは予めシミュレーションを用いて用意する。これらの処理は 64 の並列処理によって行われる。Fig.3.5 にパターン認識の概念図を示す。

3.3.2 Track Fitter

Track Fitter では、Road Finder により算出された Road から、hit 情報を用いて線形近似により飛跡のパラメータを求める。パラメータとは、R 方向の再近接距離 d_0 、z 方向の再近接距離 z_0 、横運動量である P_t 、 η 、 ϕ である。線形 1 次近似では、パラメータを p_i (i は上記の 5 つのパラメータを表す)、内部飛跡検出器の層を j とし、記憶されている hit 情報を x_{0j} 、観測データから得られる hit 情報を x_j とし 1 次近似の係数を C_{ij} とすると、

$$p_i \simeq \sum_{j=1}^{11} C_{ij}(x_j - x_{0j}) = \sum_{j=1}^{11} C_{ij}x_j - q_j \quad (3.1)$$



図 3.5: パターン認識の概念図

と 1 近似で表わせる。飛跡による定数項 C_{ij} と q_j を予め算出して Road と一緒に格納することにより Road 算出後にこの定数項と hit 情報とで飛跡のパラメータを再構成することができる。

このように飛跡のパラメータを算出した後に、Road 内で考える全パターン of χ^2 を計算し、精度の良い飛跡を飛跡として Hit Warrior へ渡す。Hit Warrior は受け取った情報に同一のものが含まれていたら削除をする。この後に SCT の残りの層を追加し、再び χ^2 のみを算出することで飛跡の質のチェックを行う。

$$\chi^2 = \sum_{i=1}^5 \left(\sum_{j=1}^{11} C_{ij} x_j + q_j \right)^2 \quad (3.2)$$

3.4 FTK のハードウェア構成

FTK は全てハードウェア上で処理を行っている。以下に各ボードの詳細を示す。

3.4.1 FTK ハードウェア全体像

FTK は様々な機関で開発が行われている。Fig.3.6 に各ボードの開発機関を示す。

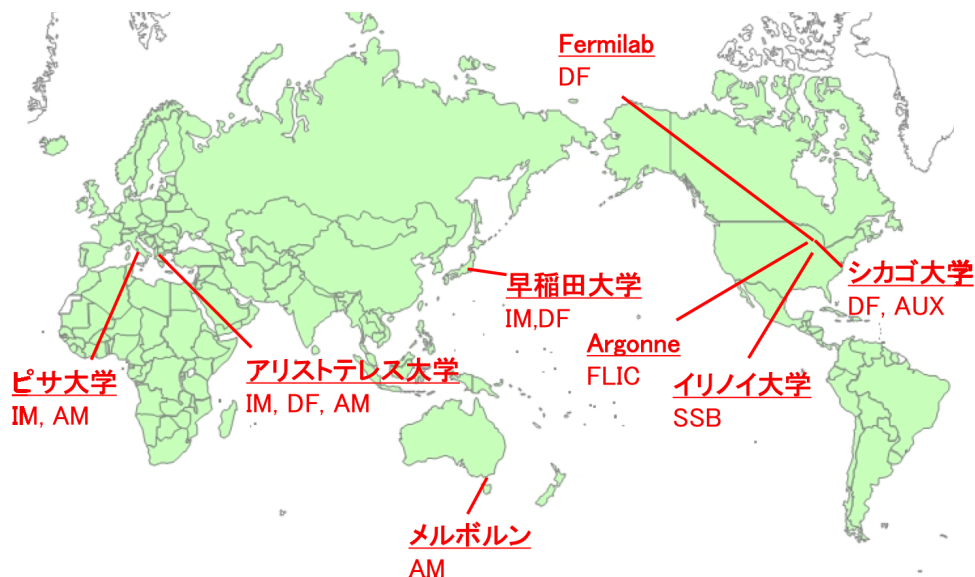


図 3.6: FTK の開発機関

3.4.2 Dual Output HOLA

シリコン検出器の Read Out Driver(ROD) には、Run1 では HOLA(High-speed Optical Link for ATLAS) が搭載されていたが、Run2 の開始とともに Dual Output HOLA が搭載される。

HOLA は、シリコン検出器の ROD から Level 1 トリガーを通過した事象のヒット情報を 32bit/40MHz のパラレルデータで受信し、SerDes(Serializer-Deserializer) により高速なシリアルデータへと変換して、S-Link で Read Out System(ROS) へと出力する。

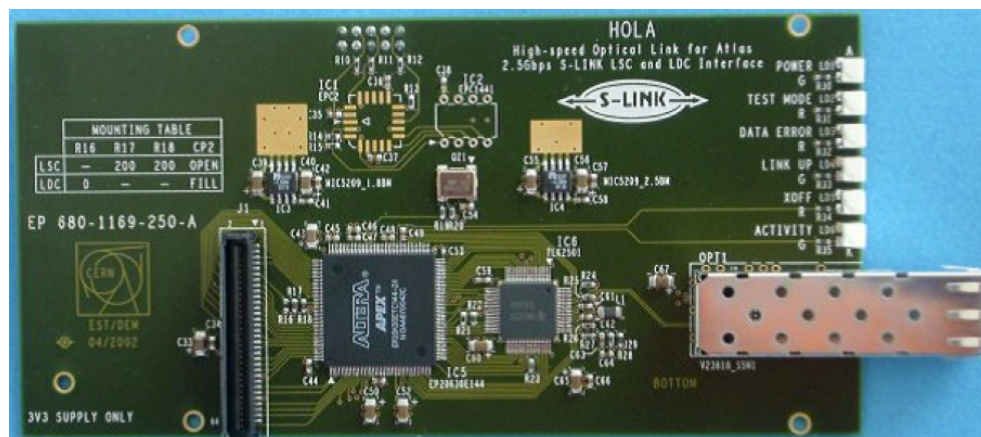


図 3.7: HOLA

Dual Output HOLA は、ROS 用の Output 加えて、FTK 用の Output を搭載している。FTK 用のチャンネルは XON/XOFF が実装されており、データを送り出すか止めるかの制御を行う

ようになっている。これによってFTK システムを ATLAS の Run の最中でも本流のデータフローに干渉せずに挿入を行うことが出来る。Dual Output HOLA がデータをコピーすることによって、例え FTK で何かしらのエラーが発生し、飛跡が再構成されなくてもトリガーフロー本流には影響を与えないようになっている。

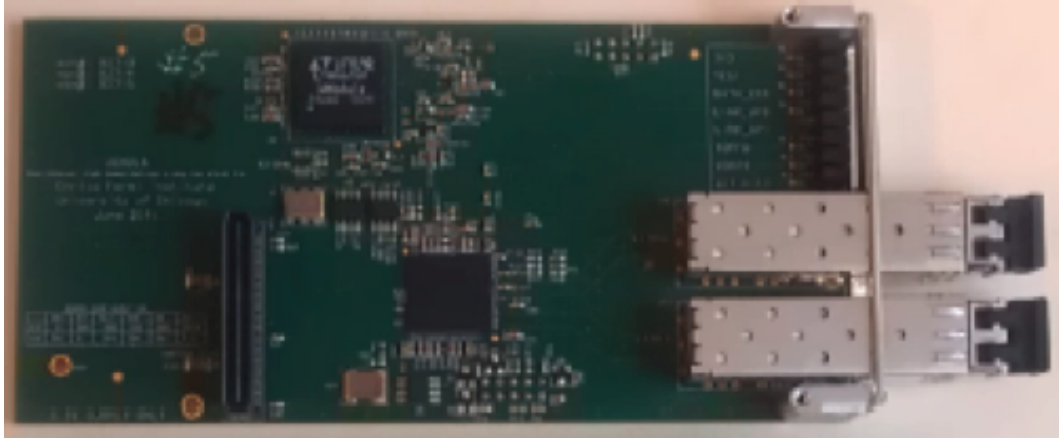


図 3.8: Dual Output HOLA

ROD で扱われるデータフォーマットの詳細は*章に後述する。ROD はファイバーケーブルによって内部飛跡検出器の各モジュールからヒット情報を受け取るが、Run ごとによって接続されるファイバーは異なる。これは Run ごとに内部飛跡検出器の各モジュールのヒットの Occupancy が変わるため、FTK はこの各 ROD に接続される内部飛跡検出器のモジュールを把握できるように IM にてテーブルを用意している。詳細は後述する。

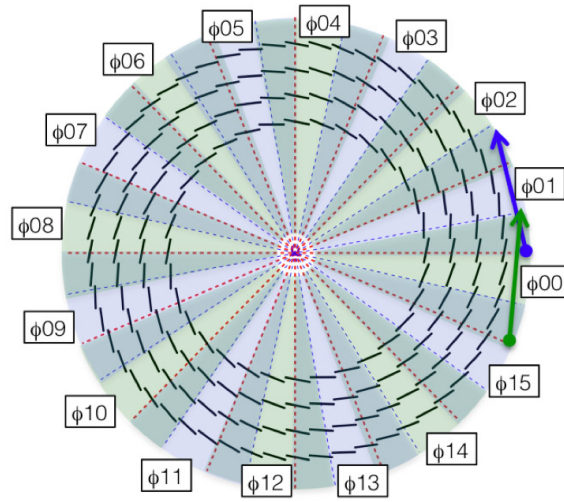
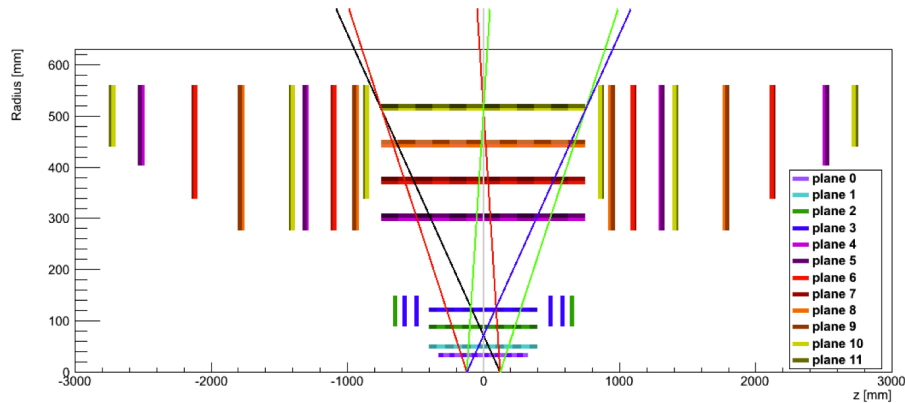
3.4.3 Input Mezzanine

FTK システム最上流に位置する Input Mezzanine Card は早稲田およびイタリア原子核研究所 (INFN) にて開発が行われている。データを受信し、IM に搭載された FPGA にてクラスタリングを行い、FTK 用のデータフォーマットへ変換を行った後、後続の Data Formatter へとデータを送信する。一つの IM には 4 つの SLink Fiber が搭載され、4 つの ROD Input を扱う。DF に 4 枚実装され、合計で最大 128 枚実装される。詳細は後述する。

3.4.4 Data Formatter

FTK は processor unit において、64 の領域 (64 FTK $\eta - \phi$ tower) で並列処理を行っている。 ϕ 方向の分割の概念図を Fig.3.9 に、 η 方向の分割の概念図は Fig.3.10 に示す。tower 間の境界をまたいで存在する飛跡による inefficiency を作らないように、tower はそれぞれ overlap を持っている。

ϕ 方向では 16 分割を行っている。 ϕ 方向では磁場による飛跡の曲率が存在するため、とくに、1GeV の飛跡が 10 度程度の曲がることから 10 度分の overlap region を設けている。

図 3.9: FTK 並列処理における ϕ 方向の分割図 3.10: FTK 並列処理における η 方向の分割

η 方向では4分割を行っている。 η 方向ではビーム衝突点が10cmに渡り、これにより region の境界をまたぐ飛跡が存在するため、10cm分の overlap region を設けている。

DF は IM から、シリコン検出器のクラスター化された Hit 情報を受け取り、overlap region 間の Hit のシェアリングや 64 FTK $\eta - \phi$ tower への分割を行った後、それぞれの Hit 情報を Processor Unit (PU) へと送信する。

4 shelves の ATCA に実装され、1 shelf の ATCA につき最大8枚の DF が実装される。FPGA について、

シリコン検出器の ROD からの Input となる IM の S-Link は DF でのデータシェアリングが最小となるように最適化される。Fig.3.11 に DF での 64 FTK $\eta - \phi$ tower 間でのデータシェアリングの回数を示す。赤い線で囲まれた領域が1つの ATCA shelves となっており、shelf 内でのデータシェアリングを表す。Fig.3.12 は FTK $\eta - \phi$ tower 間の接続の概念図を表

しており、緑の玉一つが1 FTK $\eta - \phi$ towerを表す。緑の玉2つが一つのFPGAに該当し、ピンク線が同じFPGA内でのコネクション、青線が同 shelf 内のコネクション、黄線が shelf 間でのコネクションを表す。

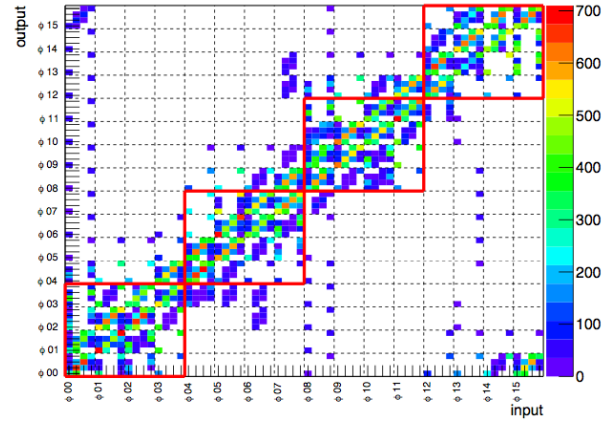


図 3.11: DF における FTK $\eta - \phi$ tower 間のデータシェアリング回数

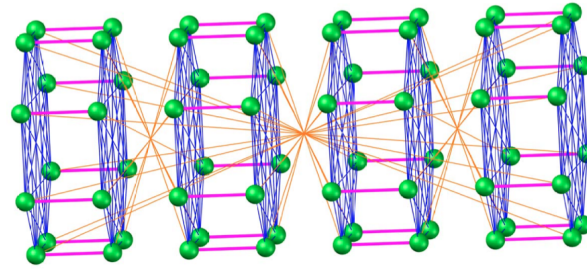


図 3.12: DF における FTK $\eta - \phi$ tower 間のコネクションの概念図

DF ハードウェアは Input および Output からくる要請を満たすようデザインされているが、Input S-Link の数の変更等にも対応できるよう拡張性を持ったデザインとなっている。前述したデータシェアリングの必要性から、Full-mesh の Advanced Telecom Computing System(ATCA) システムを採用している。それぞれのボードは full-mesh ATCA backplane によって他のボードと接続されている。

- 2 FTK $\eta - \phi$ tower / 1 FPGA
- 1 FPGA / 1DF
- 8 DF / 1 ATCA
- 32 FPGs, 32 DF, 4 ATCA shelves

Fig.3.13 に DF システムのブロックダイアグラムを示す。ただし、以前の version の DF を用いたダイアグラムとなっているため FPGA が 2 つ搭載されているが、この FPGA が 1 つになつ

のみで、基本的に構造は変わらない。黄線により、IM からシリコン検出器のクラスター化されたヒット情報を受け取る。オレンジ線がFPGA 内での FTK $\eta-\phi$ tower 間でのデータ共有、ピンク線が ATCA backplane へ接続された shelf 内でのデータ共有、緑線が Rear Transition Module(RTM) を介した shelves 間でのデータ共有を表す。各線を用いてデータ共有された後、水色線を用いて各 FTK $\eta-\phi$ tower へと分割したデータを後段の PU へと送信する。

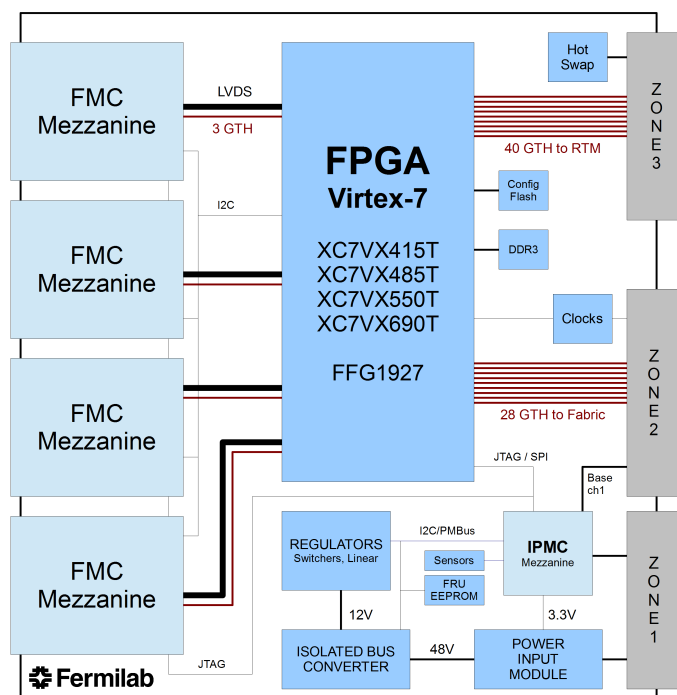


図 3.13: DF のブロックダイアグラム

3.4.5 Processor Unit AUX

パターン認識および最初のトラックフィット (1st stage track fitting) は Processor Unit (PU) にて行われる。一つの PU は、後述する Associative Memory (AM) と AUX によって構成される。

PU の個数など、

AUX

AUX は DF からシリコン検出器 8 層分の Hit 情報を受信する。この際に受信するのは、それぞれの PU が受け持つ FTK $\eta-\phi$ tower の Hit のみを受信する。Hit は後にパターンの持つ Hit 情報を即座に抜き出せるよう、Data Organizer(DO) にストアされ、AM に送信される。ま

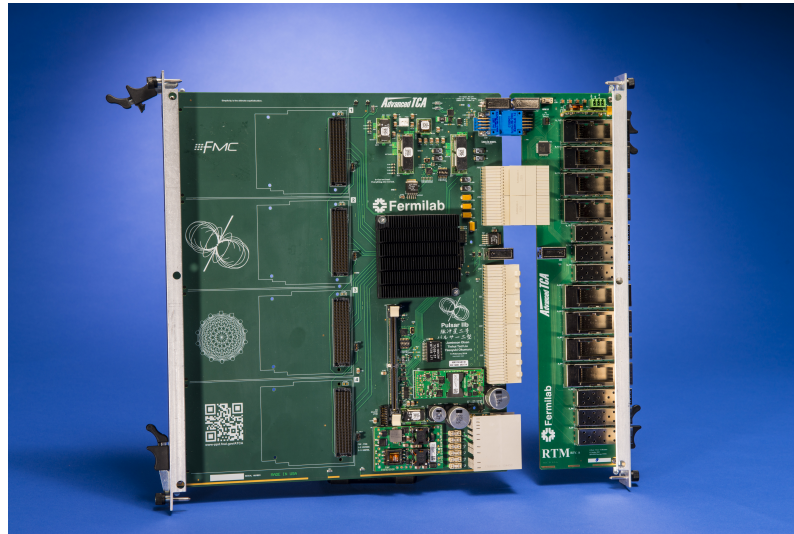


図 3.14: Data Formatter

た、受信した Hit 情報を、Super-Strip(SS) と呼ばれる単位に変換する。SS はシリコン検出器の Strip を数个束ねた単位で、resolution を落とすことになるが、Full resolution でパターン認識するのではなく、SS でパターン認識することで、パターン数を減らすことができる。ただし、SS を大きくすることにより AM に格納するパターン数を減らすことはできるが、一方で、Track fitting における Fit 数は層の数の乗数で増加する事になるため、AM に記憶できる飛跡パターン数と Fit にかかる時間を考慮し最適化を行って行く必要がある。

AM に Hit に対応する SS を送信した後、AM により Road number (Road:SS 単位で構成された飛跡パターン) を受信する。その後、受信した Road を構成する SS number に存在する Hit を DO より抜き出す。その後、これらの、Hit、Road number、Sector number を Track Fitter (TF) に送信する。

TF は Road 内に存在する全 Hit の組み合わせに対して、線形一次近似を用いてそれぞれの χ^2 を計算する。この際、8 層中 7 層で Hit があった場合、欠落した層に Hit があったとして (Majority fit)、 χ^2 が最小になるよう計算をする。全ての組み合わせで $\chi^2 > \chi^2_{cut}$ となった場合、TF から送信される。また、全 8 層 Hit を持ち、 $\chi^2_{cut} < \chi^2 < \chi^2_{high}$ となった場合、inefficiency の層があったとして、その Track 候補のうち、1 層ずつ Hit を欠落させて、Majority fit 同様、欠落した層の Hit を χ^2 が最小となるように再度計算する (Recovery fit)。もし refit した Track が χ^2 の閾値を満たした場合、TF から Hit Warrior へと送信する。Track Fitter から送信された Track は HW へと送られる。HW では、1 つの Road 内の 2 つ Track が同じ Hit を共有していた場合、 χ^2 の低い Track のみを残して他の Track は Fake と判断し消される。HW で消されずに Track と判断されるものは、Second Stage Borad(SSB) へと送信される。

AUX のブロックダイアグラムを Fig.3.15 に示す。DF から送信された Hit は QSFP+ コネクタを通して、AUX と送信される。2 つの QSFP+ コネクタで、1 つのシリコン検出器 1 層分をカバーする。まず、Input FPGA に Hit は入り、ここで受信した Hit が属する SS number へと変換され、SS number は VME クレーットの P3 コネクタを介して、AM へと送信される。ま

た、同時に SS number と Full-resolution の Hit は4つの FPGA へと送信される。これら4つの FPGA はそれぞれ DO と TF の機能を搭載されており、1つの FPGA につき、後述する LAMB の1つを担当する。そして、AM から P3 コネクタを介して、少なくとも7層以上で Hit があった Road number が返信され、これらの FPGA 内の DO 及び TF で Track を算出し、最終的に残った Track が SFP+コネクタを介して SSB へと送信される。

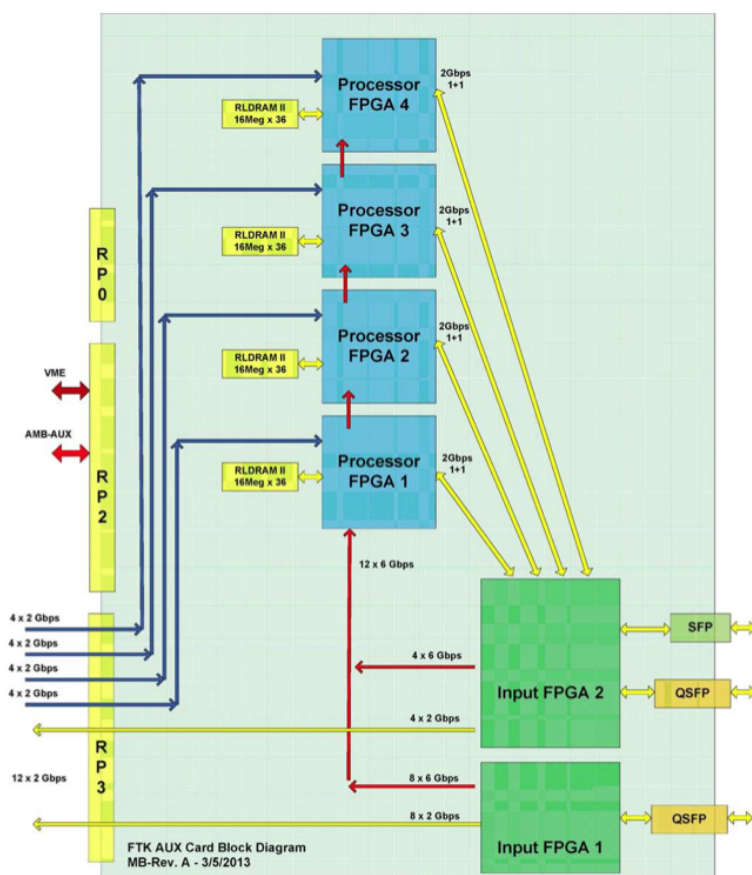


図 3.15: AUX のブロックダイアグラム

Table.3.1 に 69 pileup 時の WH event における AUX が受信する各層の平均 Hit 数を示す。QSFP+ファイバーの最大データレートは6Gb/sであり、1wordに32bit、100kHz Level 1 のトリガーレートで受信するため、1500 hits/event まで受信することができる。Table.3.1 から十分な余裕を持っていることがわかる。2Gb/s のシリアルリンクが Input FPGA から SS を AM へと送信するが、16bit word が SCT,pixel それぞれ1つのリンクによって送信されるため、このラインも十分に余裕を持っていることがわかる。また、AM からは1つの FTK $\eta - \phi$ tower につき 32 の 2Gb/s シリアルリンク (16/PU $\times$ 2PUs/tower) を介して AUX へと Road が送られる。それぞれの 2Gb/s シリアルリンクにつき 500 の 32 bit words/event を送信することができる。また、69 pileup 時には AUX につき 50 tracks/event となる推定で、10 word/track (8hits, Road number, Sector number) となるため、100kHz Level 1 トリガーレート下ではこれらは

50M 32bit word/s となり、6Gb/s SFP+ファイバーによる十分な余裕を持たせている。

Layer	barrel	endcap
IBL	-	-
Pixel 0	440	440
Pixel 1	380	360
Pixel 2	290	330
SCT 0	210	300
SCT 1	210	280
SCT 2	210	280
SCT 3	210	270
SCT 4	200	250

表 3.1: 69Pileup 時の WH event における AUX が受信する各層ごとの Hit 数

AM

AM は AUX から受信した SS number からパターン認識を行い Road を算出する。AM は 9U VME ボードであり、AM の Mezzanine ボードである Local AM Board(LAMB) と、LAMB に搭載された大量の AMchip と呼ばれる Mini-ASIC によって構成される。Fig.3.16 に AM ボードの概念図と AUX を示す。黄色にハイライトしているのが 1 つの LAMB で、赤線が 12 本の合計シリアルリンクを表し、P3 コネクタを介して AUX から SS を受信する。また、青線はそれぞれ 4 つのシリアルリンクを表し、合計 16 本のシリアルリンクによって、Road number を P3 コネクタを介して AUX へと返信する。それぞれのリンクは 2Gb/s で動作する。128AMchip/AM で 8M パターンが搭載される予定で、非常に大量の Road を収集し、AUX へと送信する必要があるため、このように非常に高速なラインを大量に用意している。

Fig.3.17 に LAMB のデータフローを示す。LAMB と AM ボードは LAMB の真ん中に位置する SMD コネクタ (Fig.3.16 における緑色の領域、または、Fig.3.17 における黄色の領域) を介してデータの送受信を行う。それぞれの LAMB は表裏にそれぞれ 16 枚の合計 32 枚の AMchip を搭載する予定である。

それぞれの SS は 8 層に対応した 8 本のバスによって LAMB に伝えられ、32 の AMchip に並列に送られる。各 chip は Contents Addressable Memory (CAM) 構造になっており、8 層の SS number を記憶している。Fig.3.18 に示すように、各層において SS が一致した際には Flip-Flop(FF、Fig.3.18 における赤四角) が鳴り、全層で FF が鳴った場合、パターンマッチしたとして、その Road number を AMB へと送る。

AMchip は段階的に開発を行っている。現在はプロダクと Version である AMchip06 の開発が進められている。プロトタイプ Version ごとのパラメータを Table.3.2 に示す。プロセスルールは AMchip04 から 65nm となり、AMchip04 では 8k パターンのみしか格納できなかったが、AMchip06 からは 128k パターン格納することができる。

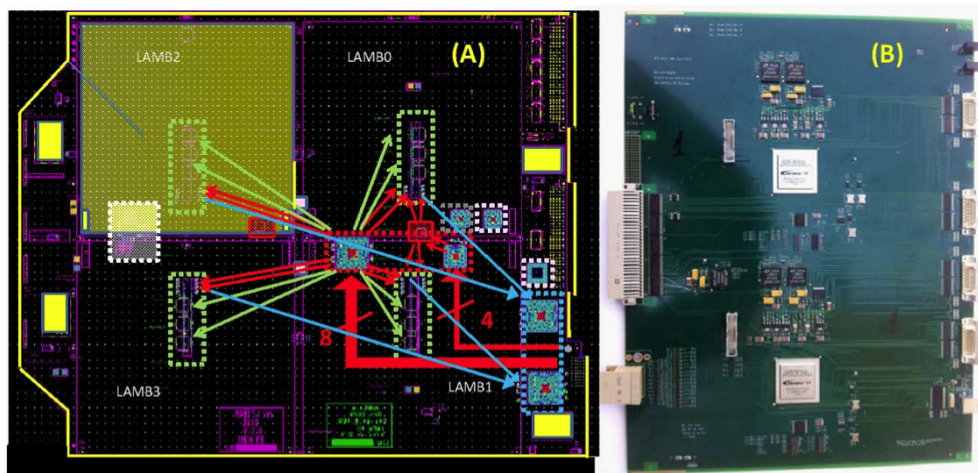


図 3.16: AM ボードと AUX

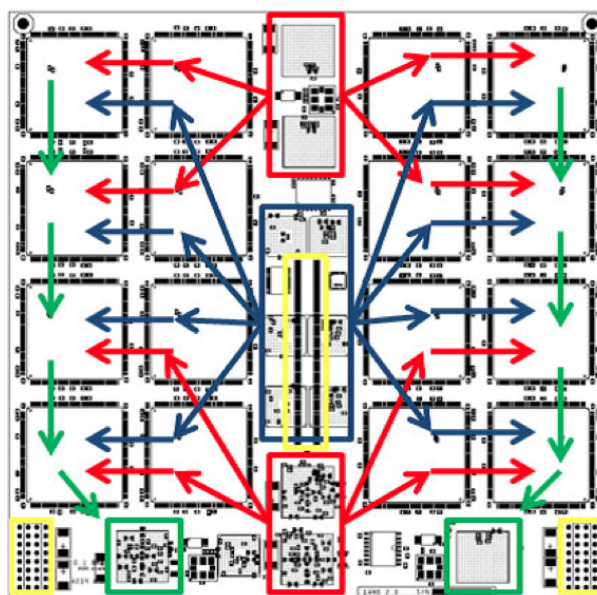


図 3.17: LAMB のデータフロー

3.4.6 SSB

Second Stage Board (SSB) では、AUX ボードから First stage の Track 情報を、DF からは First stage では使わなかった残りの層の Hit 情報を受信する。1つの SSBにつき、4つの AUX ボードの出力とそれに対応する 2 FTK η - ϕ tower 分の扱う。SSB では、DF から受信した残りの層の情報を加えることにより、フェイク Track の除去を行う。また、全 12 層の情報をを使用することで、最終的な飛跡パラメータの分解能向上を行う。その後、1つの Core Crate の全情報を合わせて、Rear Transition Module の光ファイバを介して、FTK to Level 2 Crate (FLIC)

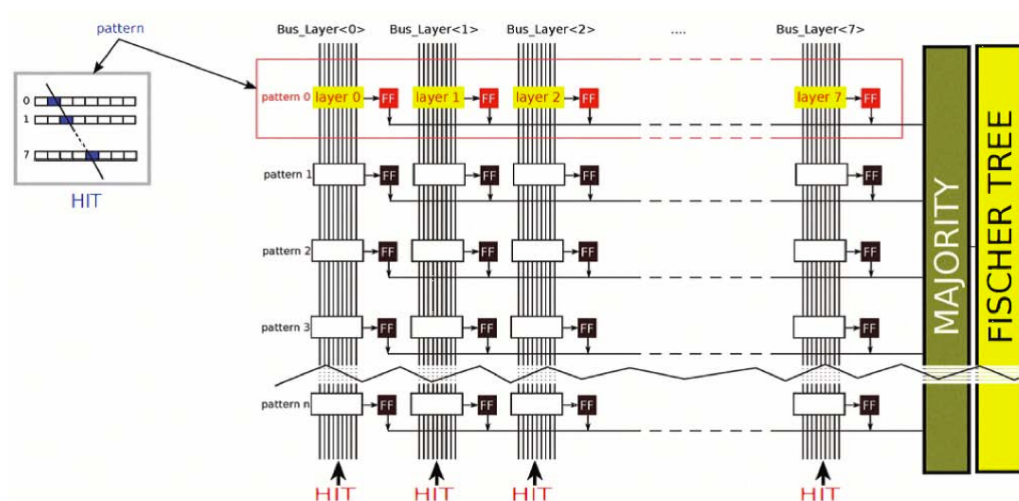


図 3.18: AMchip の動作原理

	AMchip03	AMchip04	AMchip06	Effect
Technology	180 nm	65 nm	65 nm	×8 pattern density
Clock freq.	50 MHz	100 MHz	100 MHz	faster, higher power consumption
Die size	10 × 10 mm ²	14 mm ²	12 × 12 mm ²	×1.5 increased area
Core voltage	1.8 V	1.2 V	1.2 V	lower power consumption
Core power	1.3 W	0.23 W	3.6 W	
Selective precharge	No	Yes	Yes	~ 80% power saving
Full custom	No	Yes	Yes	×2 pattern density
Layers	6 (or 12)	8	8	$\frac{3}{4}$ pattern density
Patterns/chip	5k	8k	128k	
Bits/layer	up to 18	up to 15	up to 16	
Ternary bits/layer	N/A	3 to 6	3 to 6	better S/N (see text)

表 3.2: AMchip のプロトタイプ毎の各パラメーター

へと送信する。これらの機能のを Fig.3.19 に示す。

3.4.7 FLIC

FLIC は FTK の最下流に位置し、各 FTK $\eta - \phi$ tower で再構成した全 Track をマージする。SSB から Track 情報を受信し、ATLAS ROD フォーマットへと変換した後、ROS へと Track 情報を送信する。Fig.3.20 に FLIC のブロックダイアグラムを示す。

Fig.3.21 に FLIC の実際のボードの概観を示す。FLIC は ATCA 規格で実装されるが、DF が Full-mesh のバックプレーンを採用したのに対し、Dual-star 構造のバックプレーンを採用している。Input Card の SFP により Core Crate よりデータを受信し、FPGA で処理を行った後に、ATCA Zone3 コネクタにより、RTM へとデータを送信し、最終的に RTM に搭載された

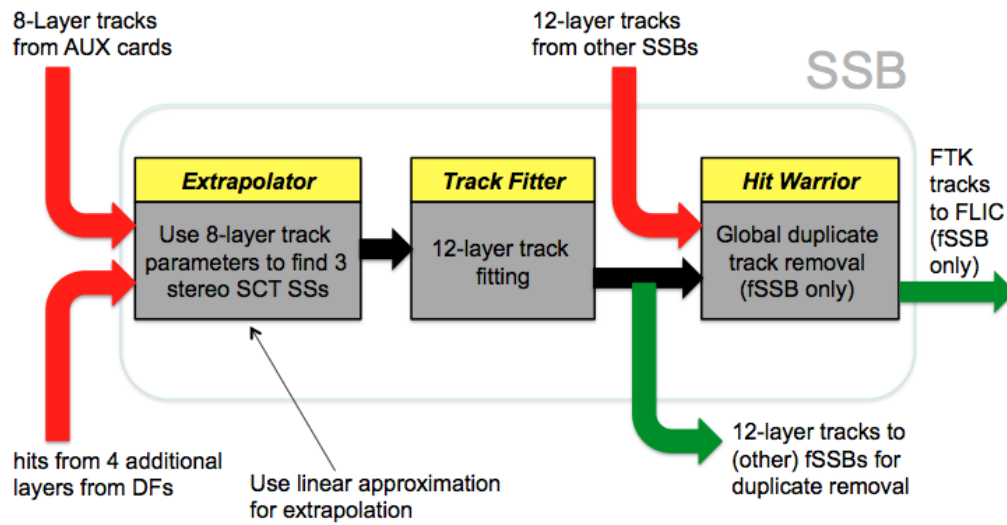


図 3.19: SSB の機能

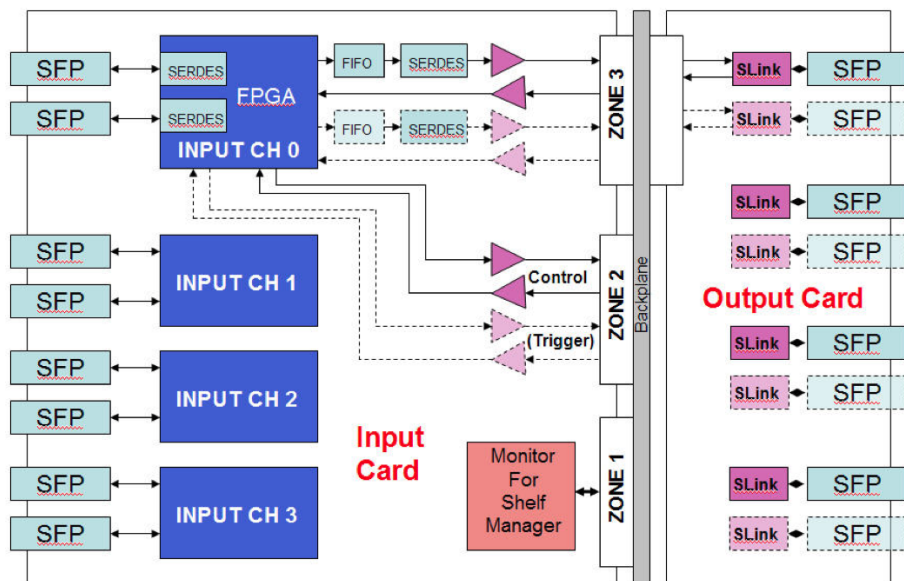


図 3.20: FLIC のブロックダイアグラム

SFP によって ROS へ全データは送信される。現在は未だ FLIC は開発段階で、FLIC に実装する処理は、ROD フォーマットへの変換のみであるが、今後 FLIC において、衝突点再構成やビームスポットの決定等の追加処理を行うことも可能となっている。特に ATCA 規格を用いていることから、FTK の全飛跡情報にアクセスできるため、事象全体での処理が可能となっている。

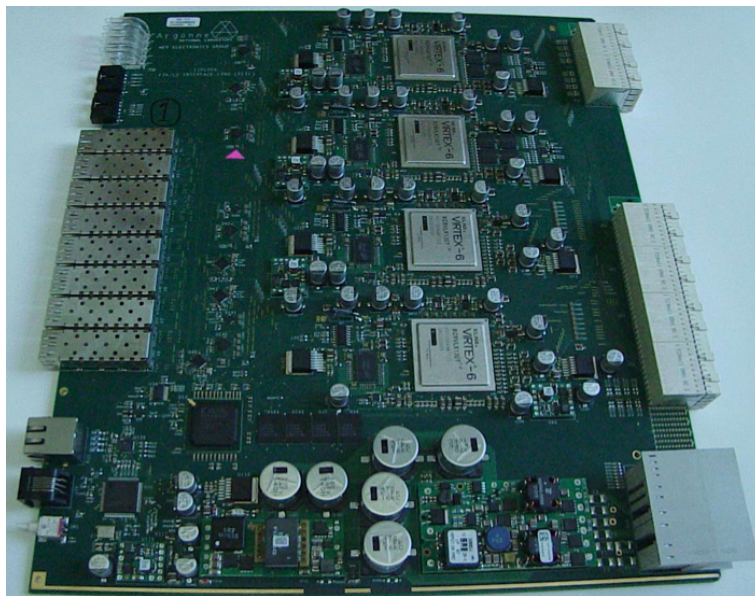


図 3.21: FLIC

第4章 FTK 受信モジュールの開発構築およびテスト

この章では、早稲田大学がイタリア原子核研究所 (INFN) とともに開発を行っている FTK 受信モジュール (Input Mezzanine:IM) について詳細を記す。

4.1 Input Mezzanine 概要

IM はシリコン飛跡検出器の Pixel 検出器および SCT 検出器と、今回の Upgrade で実装予定の IBL 検出器の Read out driver(ROD) から 380 本のファイバーで Hit 情報を受信する。各ファイバーのデータレートは 2.0Gb/s のため、IM は 380 個の ROD からトータルで 760Gb/s のインプットレートでヒットを受信する事になる。

それぞれの IM には 2 つの Xilinx Spartan 6 LX150T FPGA が搭載され、それぞれの FPGA につき、2 S-Link を受信する。この 2 つの S-Link のうち 1 つが SCT ROD で、もう一方が Pixel または IBL の ROD に接続されている。ROD から送られてくるデータはイベントごとに ROD の情報を持っており、Run number と合わせてその ROD がどのシリコン飛跡検出器の各モジュールに接続しているのか判断できるように、IM には ROD number からその ROD に接続してあるモジュールが判断できる Look up table が格納されている。クラスタリングはそれぞれの内部飛跡検出器のモジュール内で行われ、32bit データとなっている。最大レートは 40MHz/event となる。

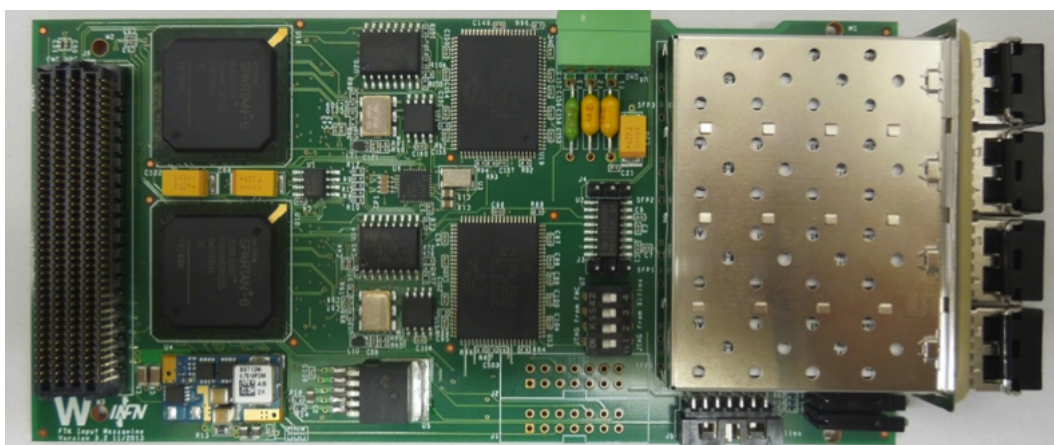


図 4.1: IM version 3.3

4.2 IM への要請

IM の Input となる ROD の Module 数とファイバーの数について Fig.4.1 に示す。IM はこれらの ROD から出力されるファイバーを全て受信する。IM は ROD から 100kHz のイベントレートでデータを受け取る。32bit データを扱い、最大 400KHz/event ほどとなる。IBL を含めたシリコン検出器の読み出しチャンネルは合計約 100M チャンネルと非常に多く、これは、ATLAS 検出器の全読み出しチャンネルの 8 割以上を占める。そのため、Table.4.1 のように ROD から出力されるファイバー数も非常に大きくなる。また、今後の増強計画に伴い高輝度環境下でのシリコン検出器の occupancy の増加を考慮し、ROD の module 数とそれに伴う S-Link ファイバーの数も増大する。IM は前述した通り、1 つの FPGA で Pixel ROD、SCT ROD をそれぞれ 1 つずつ扱う。これは、Pixel は 2 次元読み出しのため、クラスタリングの処理が非常に複雑になるため、1 つの FPGA に Pixel ROD が 2 つ入力されないようにしているためである。そのため、IM の台数は Pixel の ROD 数によって決定され、全体で 252 本の S-Link が Pixel ROD に接続されているため、IM も最低 126 台必要となる。また、ROD の出力は最大 100kHz/event であり、S-Link は 32Bit データを 40MHz で扱う。これらには、FPGA の GTP バンクを用いている。

Detector layer	ROD output (～2012)	ROD output (2015/2018)	Fibers
IBL	N/A	56 (2015)	56
Pixel B-Layer	44	44	44
Pixel Layer-1	38	76 (2018)	76
Pixel Layer-2	26	52 (2015)	52
Pixel Disk	24	24	24
Pixel Total	132	252 (～2018)	252
SCT	90	128 (～2015)	128
Total	222	380 (～2018)	380

表 4.1: シリコン検出器 ROD の Module 数とファイバーの数

4.3 IM の開発

4.3.1 ハードウェア

主にデータフローは以下の 3 つのパーツが行う。

- ・Finisar SFP transceiver (FTL8519P2BNL)[up to 2.15 Gbps]

- Xilinx Spartan 6 FPGA (XC6SLX150T)[4GTP transceiver up to 3.2Gbps]
- Samtec FMC connector (HPC 400 pins)

ROD からの S-Link input は最大 2Gb/s で受信し、FPGA にてクラスタリング及び FTK データフォーマットへの変換を行う。Spartan 6 FPGA は 150k logic cell を持ち、クラスタリングの機能を実装するのに十分な logic 数を搭載する。最終的には、200MHz DDR によってクラスタ化した Hit 情報を FMC コネクタを介してマザーボードである DF へと送信する。

IM は 4 S-Link input から 32bit 40MHz で、合計 5.1Gb/s で受信し、16 LVDS ライン、200MHz の合計 6.4Gb/s でデータを送信する。Output は Input に対して 25% の余裕を持っているため、クラスタリング処理が重いイベントが来て一次的にスタックすることはあっても、最終的にはデータレート回復できる。Fig.4.2 に IM のボードデザインを示す。

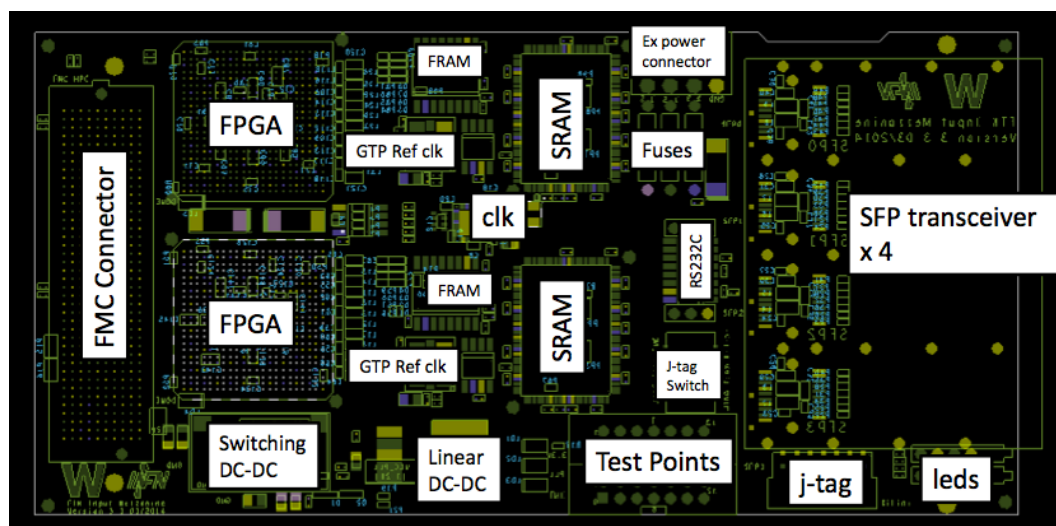


図 4.2: IM board design

IM は 380 以上のパーツが搭載されている。2つの SRAM は、テスト時に S-Link で受信しなくてもデータを扱えるよう、pseudo データの実装等も行える。また、2つの Flash-RAM は FW の実装用に、さらに、Free usage 用に 2つの Flash-RAM も搭載されている。外部から FW をダウンロードできるように J-tag コネクタも搭載されており、主にテストではこの J-tag コネクタを用いてテスト FW を実装している。また、IM スタンドアローンテスト用に External power supply も搭載されている。S-Link の隣に搭載されている 4つの LED は SFP transceiver の Link-up 確認用に、他の 8つの LED は user が自由に使用できるように、さらに 2つの青色 LED は FW が実装されているかの確認用に、DC-DC converter 横の 3つの LED は電源確認用に搭載されている。12V と 3.3V が DF より FMC コネクタを介して供給される。3.3V は直接全パーツに供給される。12V は switching DC-DC converter (12V $\rightarrow$ 1.2V) を介して 1.2V に落とされ、FPGA の core voltage へ供給される。また、3.3V から linear DC-DC converter (3.3V $\rightarrow$ 1.2V) を介して FPGA の I/O へと供給される。

Table.4.2 に IM の電源供給を示す。現在は、IM1 台につき 8W 消費する。DF からは最大

35W/IM まで割り当てられている。

	Measured power / IM (W)	typical expected source
FPGA	3.6	8 GTP transceiver (~1.0 W)
other	4.4	4 SFP transceiver (~2.5 W)
Total	8.0	

表 4.2: IM の電源供給

PCB

以下、Version4.0 について記述する。IM の FPGA で使用するメインクロックは 200MHz クロックを DF から FMC コネクタを介して受け取る。また、GTP 用に 2 つの reference クロックを搭載している。さらに、general purpose clk も搭載している。また、4 つの SFP が LVDS differential pair にて 2 つの FPGA のそれぞれ 2 つの GTP へ接続されており、合計 34 LVDS differential pair が FMC コネクタと FPGA を接続している。Table.4.3 に IM の基盤各層の概要を示す。IM は 12 層の基盤を持つが、これらの主なデータフローに使用する線は安全性の観点から、3 層目および 4 層目に配線されている。全ての signal line は GND 層に挟まれる構造になっている。

FPGA

IM には 2 つの Xilinx Spartan 6 LX150T FPGA が搭載されている。FPGA(Field Programmable Gate Array) は書き換え可能なデバイス Programmable Logic Device (PLD) の一種で、デジタル回路を集積化できる。ユーザーが回路情報を、ハードウェア記述言語 (Hardware Description Language;HDL) によって論理合成されたファームウェアをインストールすることで、ソフトウェアの様に使用することが出来る。IM にインストールするファームウェアについては後述する。FPGA に代表される集積化回路の一つに ASIC (Application Specific Integrated Circuit) があるが、ASIC は PLD とは異なり、回路を書き換えることが出来ず、特定の目的に特化した回路を提供する。ASIC はトランジスタレベルから設計する集積回路で、アナログ回路とデジタル回路の双方を実装することができる。一方で、FPGA はデジタル回路のみの実装となるが、PLD となるため、回路を作製後もプログラムを変更することができる。一般に ASIC は大量生産時に、PLD と比較して安価に入手することができる。FTK ファームウェアは、拡張性を持たせるために、その大部分が FPGA によって構成されているが、大量に実装する必要がある AMchip に関しては ASIC を採用している。Fig.4.3 に IM が使用する Spartan 6 FPGA(FGG484LX150T) のピンとその機能を、Fig.4.4 に Bank の構造を示す。

FPGA は基本的には AND 素子や OR 素子を持たず、SRAM と Dtype Flip-Flop (DFF)、スイッチ付き配線で構成される。この SRAM を用いて、4 入力の Look Up Table (LUT) として使用し、AND 素子や OR 素子を持たずとも AND 演算や OR 演算を実行することができる。これにより、素子の構造の違いによる信号の遅延がないため、基本的には遅延を気にすることな

Layers		Thickness (mm)
Layer 1(top)	Lines and pads	0.018
Pre-Preg		0.245
Layer 2	GND	0.035
Core		0.200
Layer 3	Main data flow lines	0.035
Pre-Preg		0.230
Layer 4	GND	0.035
Core		0.200
Layer 5	Main data flow lines	0.035
Pre-Preg		0.230
Layer 6	GND	0.035
Core		0.200
Layer 7	lines	0.035
Pre-Preg		0.230
Layer 8	GND	0.035
Core		0.200
Layer 9	power plane 1.2 V	0.035
Pre-Preg		0.230
Layer 10	power plane 3.3 V	0.035
Core		0.200
Layer 11	GND	0.035
Pre-Preg		0.245
Layer 12 (bottom)	lines and pads	0.018
Total		2.701

表 4.3: IM PCB layer と厚さ

く高速に動作することができる。回路をダウンロードするとは FPGA に SRAM のデータと配線スイッチを書き込むことにあたる。Xilinx 社では SRAM と DFF の組み合わせを Slice と呼んでいる。

IM の製作およびテスト

現在まで作製してきたボードを Fig.4.5, Fig.4.6, Fig.4.7, Fig.4.8 に示す。

Vesion 1.0 は 2011 年 2 月に 5 台製作された。IM の First prtotype ボードで、DF との接続ではなく Edro ボードと接続を想定して開発された。Edro ボードと接続するため、DF との接続に用いる FMC コネクタではなく QTH コネクタを搭載している。このボードを作製した当時は、DF は未だ製作されておらず、IM のマザーボードとして、既存の VME クレートと Edro ボードを用いてテストを行う為にこのようなデザインとなった。また、マイナーチェンジを行った Version 2.0 を 2012 年 4 月に 5 台製作した。これらのボードは CERN で Vertical Slice テストに用いた。Vertical Slice テストとは、実際の ATLAS シリコン検出器の ROD から実データを受信し、FTK 各プロトタイプボードを用いたテストである。この際には、全並列処理を行うのではなく、1 つの並列処理単位でのテストを行っている。

Version 3.1 は 2013 年 2 月に 5 台製作された。Version 3.1 から QTH コネクタではなく、FMC コネクタを搭載し、DF への接続が可能になった。また、FMC コネクタに変更したことにより、

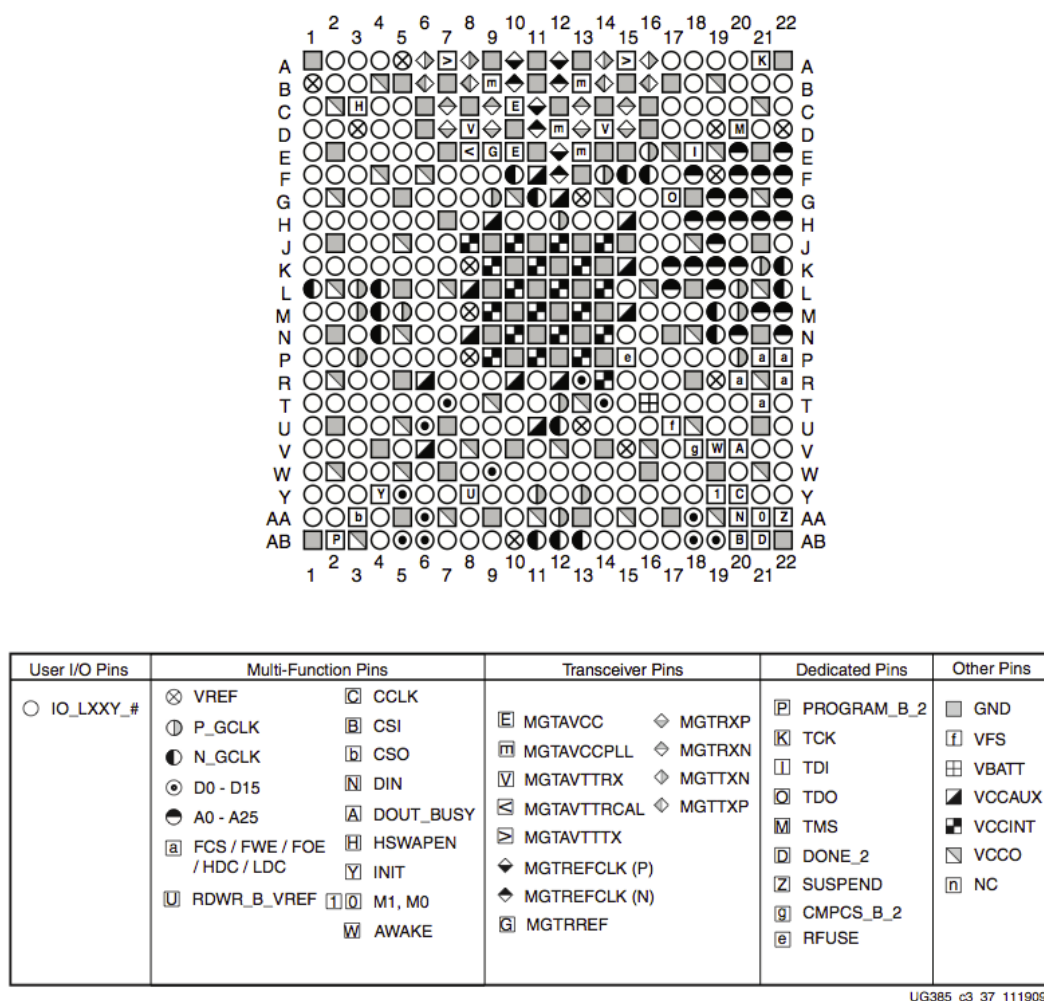


図 4.3: Spartan 6 FPGA(FGG484LX150T) のピンとその機能

電源の供給方法も変更したため、新しい DC-DC converter も搭載した。主に、大きな変更点である FMC コネクタを介した Output 部のテストを行った。また、DF とのコミュニケーションプロトコルも決定し、DF に接続してのデータフローテストを行った。KC705 には Kintex 7 FPGA (XC7K325T-2C) が搭載されている。IM Output は、200MHz の DDR 伝送を用い、32bit データを各 FPGA につき 6 LVDS differential pair を用いて送信する。6 pair のうち 4 pair がデータ用、2 pair がコントロール bit として使用される。コントロール bit には、データの始まりを表す is 1st bit、データかアイドルワードかをハンドルする Data Valid bit、データのエラー検出用の Parity bit 等を含む。4 pair での DDR 伝送となるため、200MHz の 4 cycle で 1 word 32bit データを送信する (50MHz 相当)。Input データレートが 40MHz となっているため、データが IM 内で処理がスタックしデータを損失することはない。また、データラインの他に Reset signal も FMC を介して行い、KC705 のボタンを用いて制御する。200MHz DDR 伝送では、非常に早い動作となるために、タイミング制約が重要になる。配線の長さの差異に

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22		
A		0	0	0	0	101		101		101		123		123		123	0	0	0	0			A	
B	3	0	0					101		101		101		123		123		123	0	0	0	1	1	B
C	3		0	0	0			101		101		101		123		123		0	0	0	1	1	1	C
D	3	3	0	0	0			101		101		101		123		123		0	0	0	1	1	1	D
E		3	3	3	0	0								123		0	0	0	0	0	1	1	1	E
F	3	3	3		3		0	0	0	0				123		0	0	0	0	1	1	1	1	F
G	3		3	3		3	3	0	0	0	0	0	0	0	0	0			1	1	1	1	1	G
H	3	3	3	3	3	3	3		3		0	0	0	0	0		1	1	1	1	1	1	1	H
J			3	3		3	3										1	1	1	1	1	1	1	J
K	3	3	3	3	3	3	3	3									1	1	1	1	1	1	1	K
L		3	3	3	3		3									1	1	1	1	1	1	1	1	L
M	3	3	3	3	3	3	3	3								1	1	1	1	1	1	1	1	M
N	3		3	3		3	3									1	1			1	1	1	1	N
P	3	3	3	3	3	3	3	3								1	1	1	1	1	1	1	1	P
R		3	3	3		3	2	2		2	2	2	2	2	1	1	1	1	1	1	1	1	1	R
T	3	3	3	3	3	3	2	2		2	2	2	2	2	2	1	1	1	1	1	1	1	1	T
U	3		3	3		2	2	2	2	2	2	2	2	2	2	2		1	1	1	1	1	1	U
V	3	3	3		3		2	2		2	2	2	2	2	2	2	2	1	1	1	1	1	1	V
W	3		3	3		2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	W
Y	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	Y
AA		3	3	2	2		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	AA
AB						2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	AB
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22		

図 4.4: Spartan 6 FPGA(FGG484LX150T) のバンク構造

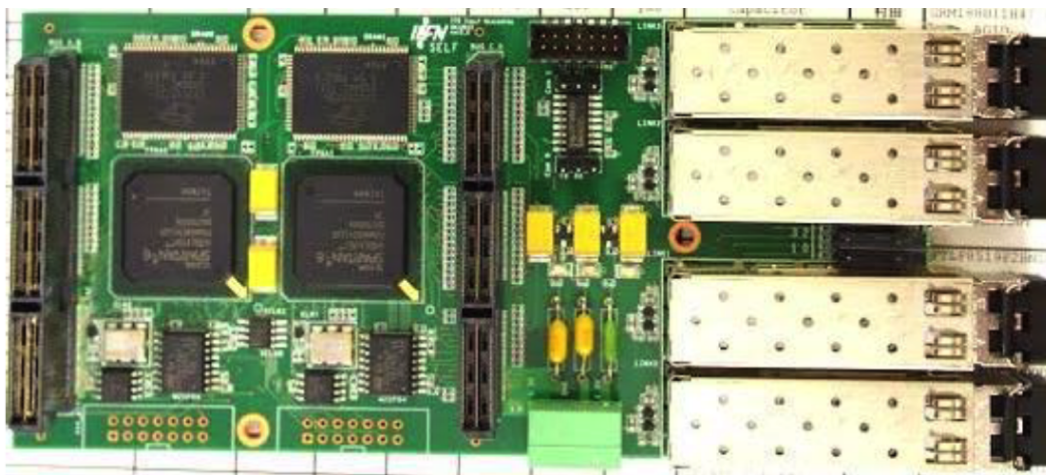


図 4.5: IM version 1.0

よるデータ同期のタイミングの差異を考慮し、正しいタイミングで正しい処理を行えるようにファームウェアの中で制約を課す必要がある。このような機能を実装した DDR 伝送を用いた Output 部のテストを行った。IM 用に、簡単なデータを生成し、FMC を通して DDR 伝送にて送信するファームウェアを作製し、IM FPGA へとインストールを行った。KC705 用には、FMC を介して DDR 伝送にてデータを受信し、FPGA 内でエラーレートを確認するファームウェアを作製し、KC705 FPGA にインストールを行いテストを行った。KC705 のファームウェア上に実装されるエラーレートは chipscope を用いて読み出しを行った。



図 4.6: IM version 3.1

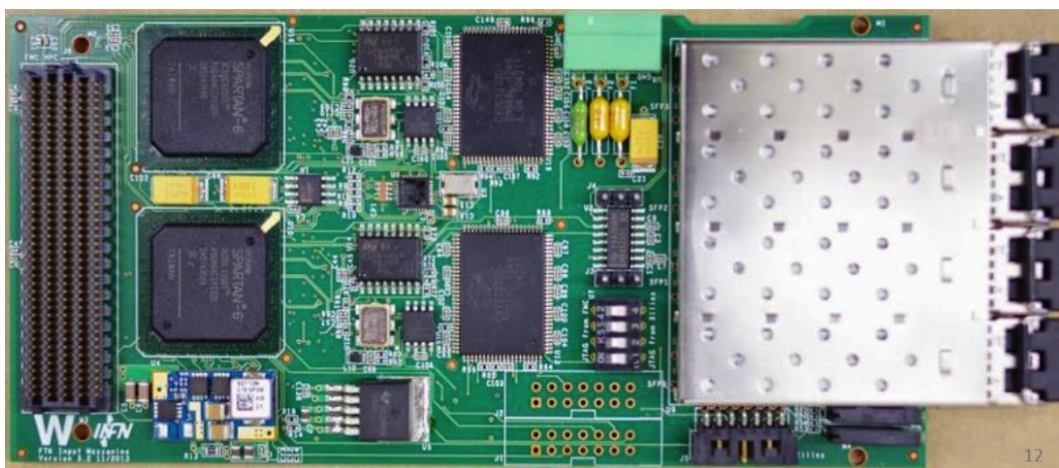


図 4.7: IM version 3.2

Fig.4.7 に Version3.2 を示す。2013 年 11 月に Version 3.2 の 10 台の作製を行った。Switching DC-DC converter に問題があったため、より電圧が安定した Switching DC-DC converter へと変更を行った。

DF は Virtex 7 FPGA を 1 つ搭載したプロダクト version (Palsur2b) ではなく、Kintex 7 FPGA を 2 つ搭載したプロトタイプ version (Palsur2a) を使用している。疑似ヒット送信モジュールから光ファイバを介して IM へ疑似ヒットを送信し、IM は S-Link を使用した GTP でデータを受信、さらに、IM 内にてクラスタリングを始めとした処理を行った後に、DF へと FMC を介して DDR 伝送を用いて送信する。IM には DF から I2C 通信を用いてアクセスできる Spy buffer を多数用意している。IM は Spy buffer へデータをコピーしている。Spy buffer にストアされた IM Input 直後のデータ、IM Output 直前のデータ、DF Input 直後のデータを

比較することにより、スムーズに debug を行うことができる。IM ファームウェアのブロック図は Fig.4.14 に示す。

また、Version 3.2 は Global Integration テストに使用した。Global Integration テストとは、FTK の各プロトタイプを CERN に持参し行うテストである。各開発機関で行っていたスタンドアローンテストでなく、隣接するボードを用いてのテストから開始し、順次拡張を行い、最終的には全ボードを統合してのテストを行う。

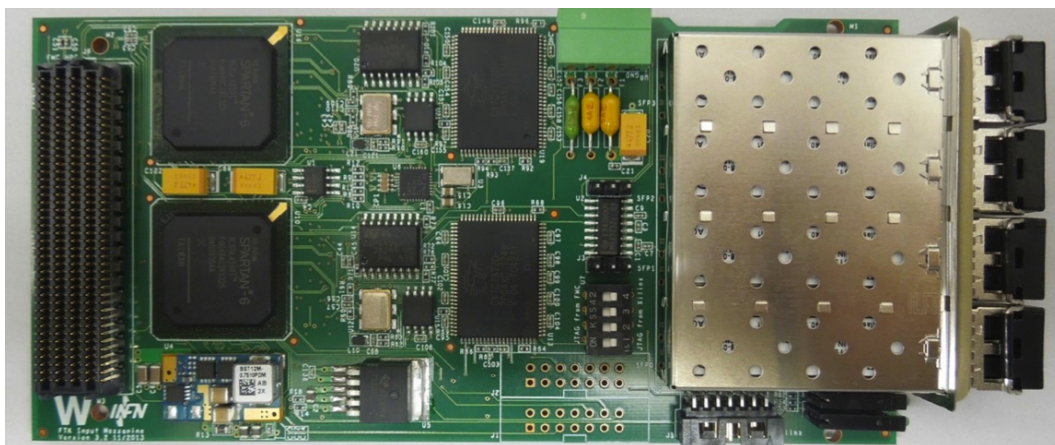


図 4.8: IM version 3.3

Fig.4.8 に Version 3.3 を示す。Version 3.3 は 2014 年 3 月に 5 台製作した。量産に向けた最終版プロトタイプで、Version 3.2 からマイナーチェンジを行った。具体的には、FMC-FPGA 間のパターンの配線を変更した。また、Version 3.3 として製作した 5 台のうち 1 台で、FMC コネクタが導通していない個体を確認したため、X 線を用いた不良解析を行った。Fig.4.9 に X 線撮影を行った FMC の写真を示す。また、導通していない FMC ピンを Fig.4.10

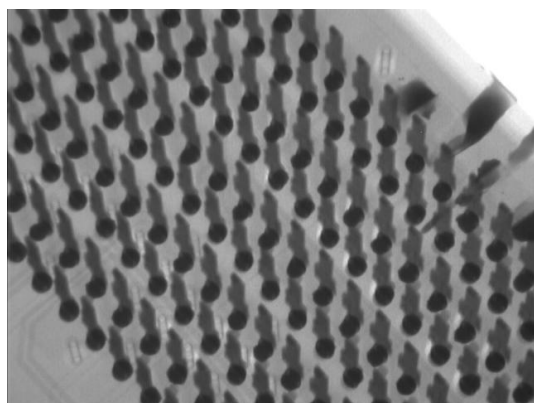


図 4.9: X 線を用いた FMC の撮影 (1)

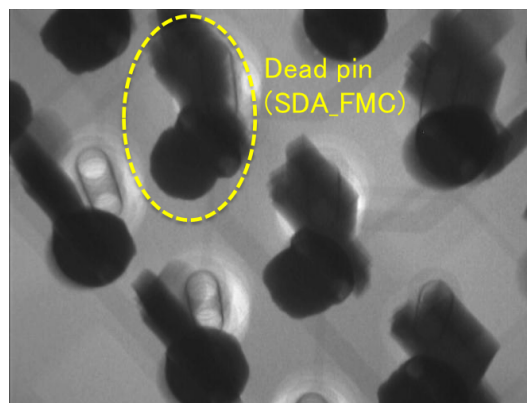


図 4.10: X 線を用いた FMC の撮影 (2)

X 線を用いた解析では導通していないピンの問題を特定することができなかったが、実体顕微鏡を用いた解析を行ったところ、問題のピン下部でクラックを確認することができた。Fig.4.11

に FMC 全体を実体顕微鏡で撮影した写真を、Fig.4.12 に問題のピンをズームして撮影した写真を示す。Fig.4.12 を見るとピン下部でクラックが起こっていることが確認できる。FMC は全 400 ピンを表面実装を行っており、全て手載せをする必要があるため実装による個体差がしやすい部品である。この結果を受けて Version 4.0 作製時の実装方法を変更した。詳細は後述する。

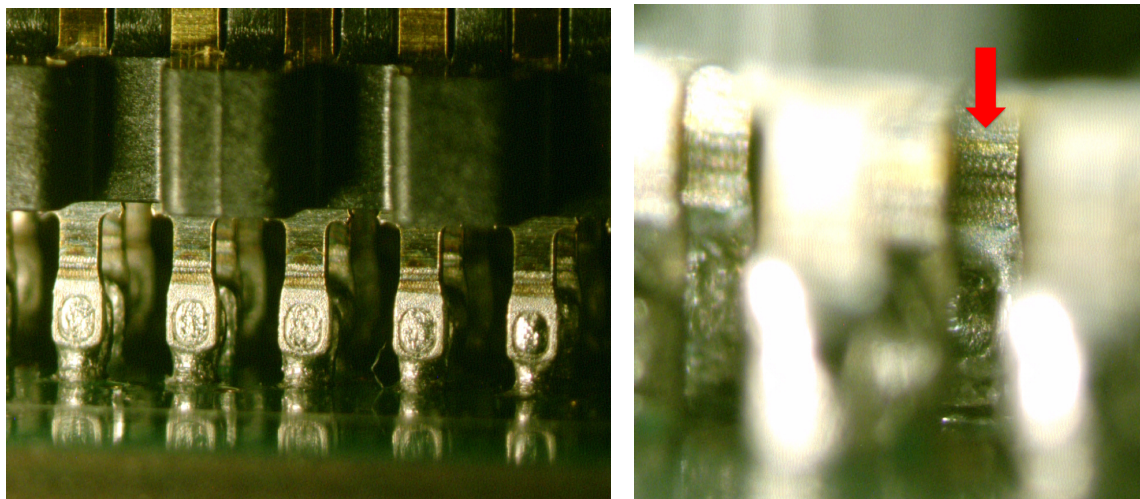


図 4.11: 実体顕微鏡を用いた FMC の撮影 (1) 図 4.12: 実体顕微鏡を用いた FMC の撮影 (2)

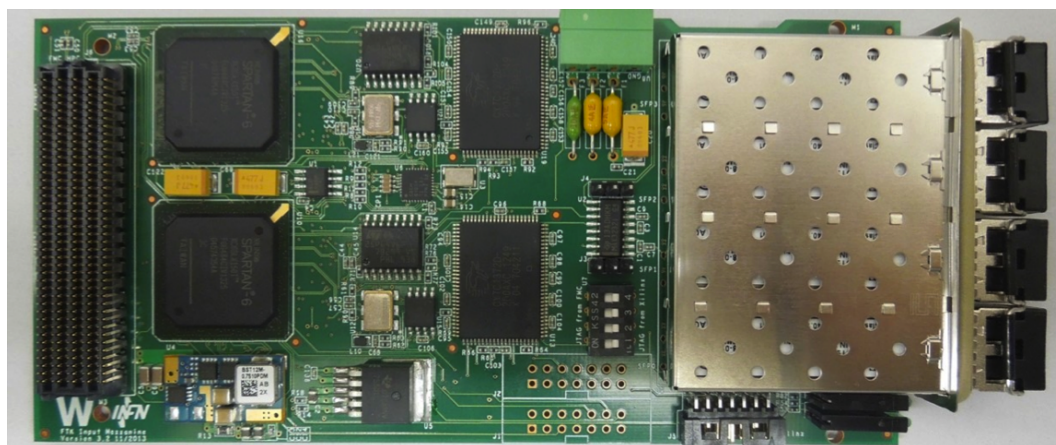


図 4.13: IM version 4.0

Fig.4.13 に Version 4.0 を示す。Version 4.0 はプロダクションバージョンであり、80 台の量産を行った。後述する Artix 7 FPGA を使用したプロダクトバージョンと 80 台ずつ量産し、合計 160 台の量産を行う。前述した通り、IM は 126 台必要なため、量産による歩留まりやスぺアを考慮し 160 台に決定した。

量産にあたって、全 380 種/IM の部品を収集した。また、Quality Control 用の試験項目を整備し、実装を行う企業で簡易的な納品前テストを行った。また、早稲田に納品後に、再度 Quality Control テストを行う予定である。主な Quality Control の試験項目を Table.4.4 に示す。

Test List	check point
基本的なテスト	
・ パーツのマウント / 方向の目視試験	-
・ 導通前抵抗チェック	電源
・ 導通チェック ・ 低負荷、高負荷での電圧値、電圧波形チェック	電源
・ FPGA, flash MemoryにF/Wを入れる	JTAG ⇄ FPGA ⇄ Flash Memory
I/Oとメモリのテスト	
・ 入力部接続試験 (S-Link, GTP I/O)	S-Link ⇄ FPGA
・ 出力部接続試験 ・ 出力部負荷試験	FMC ⇄ FPGA, CLK
・ SRAM, Flash Memory試験	SRAM, Flash Memory ⇄ FPGA ⇄ RS232
その他のテスト	
・ その他のGTP I/O接続試験	FPGA ⇄ FMC, FPGA
・ 内部クロック試験	G/P CLK
・ 4ピン電源試験	4ピン電源

表 4.4: Quality Control 試験の主な項目

前述した FMC の問題を防止するため、Version 4.0 から FMC 部分の半田のペースト方法の変更を行った。半田を塗る際には、PCB にメタルマスクを被せ、メタルマスクの上から半田を塗布する。このメタルマスクには、半田塗布部にのみ穴が開けられているため、意図した箇所にもみ半田を塗布することができる。Version 4.0 の製作時には、FMC 部分のみ半田が圧塗りになるように、メタルマスクの FMC の穴部にのみ裏からセロハンを貼った。セロハンを張ることでメタルマスクと PCB の間に隙間ができ、その分半田が圧塗りになるよう意図した。

Artix 7 を使用した IM

IBL 検出器の高い occupancy を考慮して、Artix 7 FPGA (XC7A200T) を使用した IM を INFN が製作している。主な仕様は Spartan 6 を使用した IM と同じで、Artix 7 への変更に伴い DC-DC converter も変更される。Table.4.5 に Spartan 6 FPGA (XC6SLX150T) と Artix 7 FPGA (XC7A200T) のリソース数についてまとめる。Spartan 6 に比べて Artix 7 は Slice が 46% 増加、BRAM は 172% 増加する。また、GTP の最大駆動周波数は、Spartan 6 が 3.2Gb/s であるのに対して、Artix 7 は 6.6Gb/s まで対応している。

4.3.2 ファームウェア

FPGA にインストールされるファームウェアは HDL で記述され、論理合成が行われた後、使用する FPGA に合わせた配置・配線が行われて生成される。IM にインストールするファームウェアは開発が現在進んでいる。開発中のファームウェアのブロック図を Fig.4.14 に示す。

まず、シリコン検出器からの Input を SerDes で受信する。Input データは S-Link を介して送られ、データは Serialize されているため、SerDes にて Deserialize を行う。Deserialize したデータは 32bit/word として扱われる。その後、Pixel データ、SCT データはそれぞれは

		XC6SLX150T	XC7A200T
Logic resources	Slices	23038	33650
	CLB FFs	184304	269200
Memory resources	Max RAM (Kb)	1355	2888
clock resources	Clock Management tiles	6	10
I/O resources	Max Single-End pins	540	500
	Max Differential pins	270	240

表 4.5: Spartan6(XC6SLX150T) と Artix7(XC7A200T) のリソース数

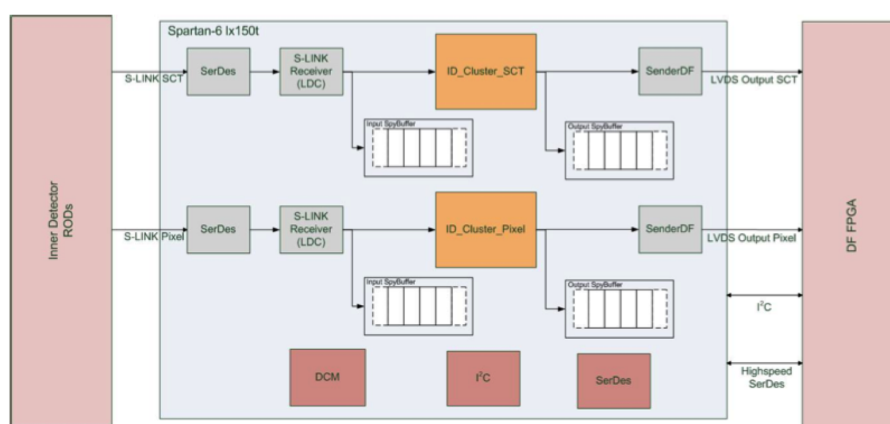


図 4.14: IM FW のブロック図

ID Cluster と呼ばれる Module でクラスタリングが行われる。その後、SenderDF と呼ばれる Module にて Output の準備を行う。SenderDF では DDR 伝送に向けて、データを分割し、コントロールワードが追加された後、出力される。ID Cluster の前後には、Spy Buffer を設けており、常にデータは Spy Buffer にコピーされ保存される。DF から FMC を通して I^2C 通信を用いてこの Spy Buffer にアクセスすることができ、クラスタリング前後のデータをモニターすることができる。

また、IM は各所に FIFO(First In First Out) を設けている。後段ボードでの処理がスタックし、データを失わないように HOLD Signal が送られてくる。HOLD signal が続く限り、IM ではデータの Output を停止し、その分 FIFO にデータを一時的に貯める。HOLD signal が落ちるとデータの Output を再開する。各所に配置された FIFO は後段から順を追って出力を停止する。この際に、FIFO が完全に一杯になる手前で 1 つ上段の FIFO の出力を停止するため、データを損失はない。FIFO を大きくするとデータをより多く貯めることができるが、その分 FPGA の LUT を占有するため、今後最適化していく必要がある。

現在開発を進めているファームウェアでの FPGA リソースの占有率を Table.4.6 に示す。

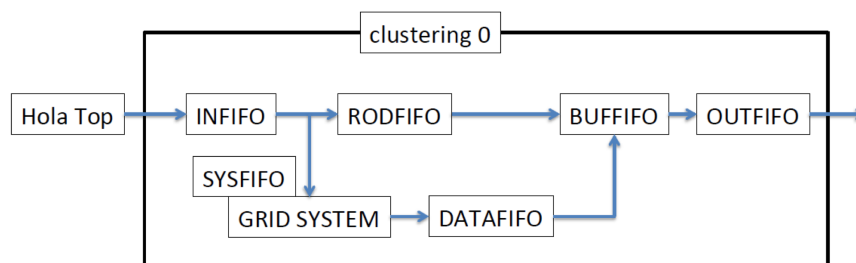


図 4.15: IM FW の FIFO の概念図

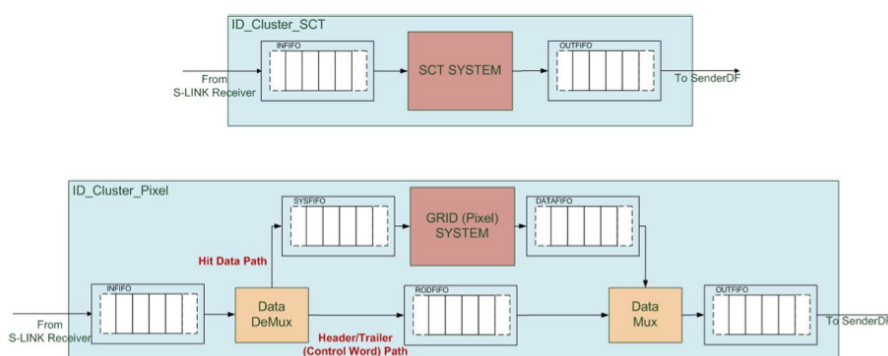


図 4.16: クラスタリングファームウェアのブロックダイアグラム

16 Engines – Spaartan6 150T – 80MHz clock			
Slice type	Used Slices	Total Slices	Used Slices(%)
Flip Flops	40611	184304	22
LUTs	50585	92152	54.9
BRAMs (18kbit)	176	268	65.7

表 4.6: 開発中の Pixel Clustering の各リソースの占有率

第5章 ハードウェアエミュレーションによる IMの性能評価

5.1 ハードウェアエミュレーションの構築

IMの開発を進める一方で、IM Outputの質の詳細な検証を行っていく必要がある。現在にいたるまでの開発では、電源部やI/Oのテストをはじめ、データフローテストや近接するボード間でのコミュニケーションテスト等、IMをハードウェアの観点でテストを行ってきた。とくに、データフローテストではL1IDが止まることなく下流まで流れて行くのか、下流よりHOLD signalを受信した際に、データを損失することなく処理できるかといった性格のテストを行ってきた。今後は、本年夏からのインストールを見据え、ハードウェアと同じ処理をソフトウェアに実装し、エミュレーションを通してハードウェアから出力されるデータの評価・保証を行う必要がある。以下にハードウェアエミュレーションの構築にあたって実装した機能を述べる。

Fig.5.1にハードウェアエミュレーションの概念図を示す。まず、Wrapperファイルと呼ばれる、Hitの位置情報が記されているtxtファイルをハードウェアエミュレーションのInputとして用いる。その後、エミュレーション側では、C++やPythonで記述したIM処理(クラスタリング処理及び、Module情報の変換)を行う。この際に、IM処理の前後でハードウェア言語への変換を行う。IM処理前をIMハードウェアのInputとして使用し、IM処理後のエミュレーションOutputは、referenceとして用いる。一方ハードウェア側では、エミュレーションから作製したIMハードウェアInputを疑似ヒット送信モジュール(QUEST)から出力し、DFに接続したIMが受信する。IMはクラスタリングやModule情報の変換等のハードウェア処理を行った後、DFへ出力する。DFは処理を行わずIMからの入力をSpy bufferへとコピーする。ROS PCからDFのSpy Bufferへアクセスし、IMハードウェアOutputを取得し、エミュレーションOutputと比較を行う。

5.1.1 データ形式の変換

以下で、Input部・Output部に分けてIMが扱うデータ形式の詳細を記す。ハードウェアエミュレーションで扱うデータはtxtデータやrootファイルであり、処理はC++やPythonを用いて行う。とくに、ハードウェアエミュレーションのInputとなるSample dataはtxtファイルとなるため、これをハードウェアInput用に0・1の電圧値から構成されるデータに変換する必要がある。IMハードウェアのデータ形式はInput(ROD data format)とOutput(FTK data format)で異なる。以下にそれぞれのデータフォーマットについて詳細を述べる。

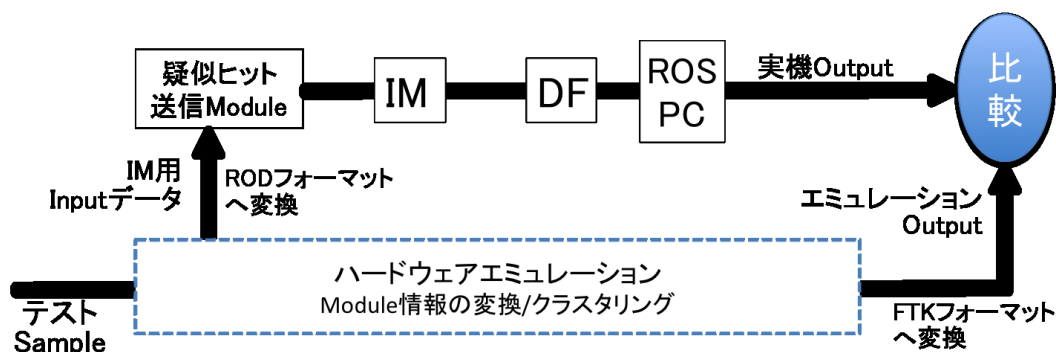


図 5.1: ハードウェアエミュレーションの概念図

Input データ形式

IM の Input は ROD が扱うデータ形式で送られてくる。1 event は、Event の始まりを表すデータや Event number、Level 1 ID (L1ID) 等のイベントに関わる情報を含んだ Header 部、Hit 情報が含まれる Data 部、イベントの終わりを表すデータやエラー情報を含んだ Trailer 部の 3 つのデータ郡で構成される。以下でそれぞれについて詳細を述べる。

- Header

Header は 10word で構成される。具体的には以下のようになる。

Word	Contents	Comment
0	0xB0FTTTTT	Beginning of fragment marker/ T = Router Trap Type Data (Removed by Router)
1	0xEE1234EE	Start of header
2	0x9	Header size
3	0x30100000	Format Version Number (Ver 3.1)
4	0x001XNNNN Pixel 0x002XNNNN SCT	Source Identifier N = Module ID, X = LS Nibble of Sub-detector ID
5	0xTTSSSSSS	Run Number: T = Run Type → 0x00 > Physics 0x01 > Calibration 0x02 > Cosmics 0x0F > Test S = Sequence within Run Type
6	0xEELLLLLL	Extended Level 1 ID: E = ECR ID, L = L1ID
7	0x00000BBB	Bunch Counter ID
8	0x000000AA	ATLAS Level 1 Trigger Type
9	0x00RR000T	Detector Event Type R = ROD or T = TIM

表 5.1: ROD Header の内訳

- Data

Data 部は SCT と Pixel により大きく異なる。SCT Data として必要な情報は、Module number、Front End chip (FE) number、Strip number の 3 つである。Data は 16 bit/word として送られる。現在は Expanded モードと呼ばれるデータ形式で送信される。今後は Condensed モードや Super Condensed モードへと変わっていく可能性がある。Expanded モードでは、Data word はさらに、Data の始まりを表す Data Header、Data の区切りとして使用される Data Trailer、

Hit の情報を含む hit cluster に分かれる。Data Header は Link number の情報を含み、ROD ID と含めて Module number を特定することができる。hit cluster は、FE number と Strip number の情報を含む。Table.5.2 に詳細を示す。16 bit/word につき、基本的には 1hit、Hit が隣り合う Strip で連続した場合のみ最大 2hit を含むことが出来る。例えば Hit のあった Strip が 3 つ連続した場合、1st hit cluster、2hit cluster、1 hit cluster という順番で送られてくる。

Name	Bits [15:0] or [31:16]
Header	001pt1bKdMMMMMM
Trailer	010zhvxxxxxxxxxxx
1 hit condensed	1FFFFCCCCCxfx0
2 hits condensed	1FFFFCCCCCsfx1
1st hit cluster expanded	1FFFFCCCCC0DDD
1 hit cluster expanded	1xxxxxxx0xxx1DDD
2 hit cluster expanded	1xxxxxxx1DDD1DDD
Hit cluster super condensed	1FFFFCCCCCNNNN
Flagged error	000xxxxxxxxFFFFEE
Raw data	011nnnxxWWWWWWW

Key:	M = link number	K = SCT condensed mode
	t = time out error	N = cluster size -1. i.e. 0000 = single cluster to 1111 = 16 hits cluster
	d = link masked by DSP	
B = BCID		n = count of raw data bits + 1
C = cluster base address		p = preamble error
D = 3 bit hit data		s = error in condensed mode data, 2nd hit
E = ABC error code		v = data overflow error
F = FE number		W = raw data
f = error in condensed mode data, 1st hit		x = Don't care (ROD fills these with 0's)
h = header trailer limit error		z = trailer bit error
L = L1ID		
SCT Super condensed mode is bit 21 of rod detector event type in ROD header (last word on 9 ROD header)		

表 5.2: ROD SCT Data Format

Pixel Data は 4 つに分かれる。Link number 等の Module 情報を含む Module Header、Hit 情報を含む Module hit、データの区切りを表す Module Trailer、エラー情報を持つ Module errors である。Table.5.3 に詳細を示す。

・ Trailer

ROD Trailer は 6 word で構成される。具体的には以下のようなになる。

Name	Bits [31:0]
Header	001PxxxxxxxxAAAAMMMLLLBBBBBBB
Trailer	010ZHVxxxxxxxxxxxxxxxxxxxxxxxx
Hit	100xFFFFTTTTTTTxxxCCCCRRRRRRR
FE Flag Error (Old)	0000FFFFxxxxxxxxxx11110FFFFFFEE
FE Flag Error (New)	0001FFFFxxx11111eeeeeeeeEEEEEEE
Raw Data	011DDDDDDDDDDDDDDDDDDDDDDDDDD
Time Out Data	001000000000000000000000000000

Key:

A = BCID Offset used in EFB error check	M = Number of MCC Skipped Events
B = BCID	N = Count of raw data bits + 1
C = Pixel Column	P = Preamble Error
D = Raw Data	R = Pixel Row
E = FE Error Code	T = Time over Threshold value
e = MCC Error Code	V = Data Overflow Error
F = FE Number	x = don't care (ROD fills these with 0's)
H = Header Trailer Limit Error	Z = trailer bit error
L = L1ID	

表 5.3: ROD Pixel Data Format

Word	Contents	Comment
0	Event Fragment Error Flags	Status 1: Bit error see EFB errors [31:0] (Table 31)
1	Error Count and Static Error Flags	Status 2: Count of words with error [15:0] TIM OK, BOC OK and ROL Status
2	0x2	Number of status words
3	Ndata	Count of data words
4	0x1	Status block position: 0 = before, 1 = after data
5	0xE0F00000	End of fragment marker

表 5.4: ROD Trailer の内訳

Output データ形式

IM の Output は FTK で扱うデータ形式で送信する。ROD data format で受信したデータを、クラスタリングを行った後、FTK data format へと変換する。FTK data format は ROD data format と同様に 1 event 中に Header、Data、Trailer の 3 つのデータ群によって構成される。以下にそれぞれについての詳細を述べる。

- Header

ROD data format の Header が 10word で構成されるのに対して、6word で構成される。ROD data format のうち FTK で使用しない情報は削除される。具体的な bit の詳細は Fig.5.2 に示す。

- Data

32 bit/ word として扱う。ROD data format と比較してクラスター化処理により圧縮される。ROD data format では、ROD ID + Link number によってシリコン Module を ID していたが、FTK data format では、シリコン Module に通し番号 (Hash ID) を割り振り、この Hash ID によってシリコン Module を扱う。また同様に、ROD data format では、FE number + Strip number で Module 内の Hit の位置を特定していたが、FTK では Strip に通し番号を割り振ること、Strip number のみで Module 内の Hit の位置を扱う。また、クラスタリング処理すること

32	31..28	27..24	23..20	19..16	15..12	11..8	7..4	3..0	comment
1	B	0	F	0	x	x	x	x	begin of fragment
0	C	A	F	E	reserved (15..0)				header word 1
0	LVL1ID (31..0)								header word 2
0	0 + 31-bit run number (30..0)								header word 3
0	0	0	L1TT (23..16)	0	TTCrcBCID (11..0)				header word 4
0	reserved (31..0)								header word 5
0	DATA (31..0)								data words
1	E	0	D	A	x	x	x	x	end of data
0	LVL1ID (31..0)								trailer word 1
0	error flags (31..0)								trailer word 2
0	reserved (31..0)								trailer word 3
0	CRC (31..0)								checksum
1	E	0	F	0	x	x	x	x	end of fragment

図 5.2: Clustering IM Output

とにより、クラスターの中央値となる Strip number と、クラスターサイズによって、Hit を表す。これにより、IM が Output する FTK data format の Data として必要な情報は、Hash ID とクラスター中央値およびクラスターサイズの 3 つとなる。ROD data format で 16 bit/word につき最大 2Hit の情報を持つが、FTK data format では、32 bit/word 中、連続する Hit はいくらでも情報を持つことができるため、最終的には情報量は圧縮される。Table.5.5 に詳細な bit の内訳を示す。

31	30..28	27	26..16	15	14	13	12	11	10..0	comment
1	rrr	reserved			0	rrr			r	module num.
0	c.width	column coord			r	r	r	row coord		Pixel cluster1
1	rrr	reserved			1	r	r	module num.		SCT module
0	H2 width	H2 empty	Hit2 coord	r	H1 width			r	Hit1 coord	SCT cluster1

表 5.5: Clustering 後の data format

・Trailer

ROD data format と同様 6word で構成されるがその内訳は異なる。具体的な bit の詳細は Table.5.2 に示す。

5.1.2 クラスタリング

ハードウェアエミュレーションにクラスタリングの一部を実装した。

SCT クラスタリング

SCTのクラスタリングはストリップと垂直な向きで1次元で行われる。SCTは表裏にセンサーを搭載した4088 Modulesで構成され、Stripのピッチは $80\text{ }\mu\text{m}$ となっており、1つのセンサーに768本のStripが存在する。HitはS-Linkにより32bit word 40MHzのレートで送られてくる。SCTのデータフォーマットは前述した通り、ROD data formatに則り、Expanded、Condensed、Super-Condensedの3つのモードが存在する。この際には、Hitの情報は16 bit wordで送られてくる。クラスタリング処理では、連続するHitをクラスターとして扱い、これらを32bit dataへと変換する。Fig.5.3にクラスタリングの例を示す。Strip-0およびStrip-1が鳴ったとすると、IMのInputとなるROD data formatでは、Strip-0、Strip-1と別々にHit情報を持つが、IMではこれをクラスターとして扱い、その中央値1.5(実際にOutputされるのは2倍の3)と、クラスターサイズ2を情報としてOutputする。この際に、クラスター中央値はStrip単位ではなく、Stripの分解能を考慮する。

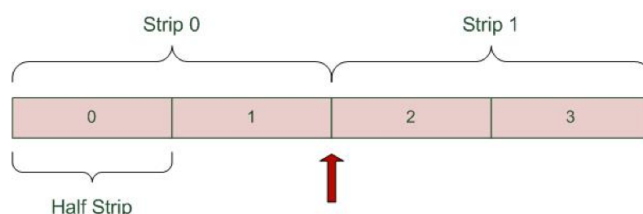


図 5.3: SCT Clustering の例

細かいデータフローについて詳細を記す。これらの処理は全てSCT ID clusterシステムで行われる。まず、SCT ID clusterシステムは32 bit データを受信する。その後、これらがどのdecoding mode(Expanded, Condensed, Super-Condensed モード)なのか判断する。その後、データがData Header なら、これらをHash IDへと変換する。このHash IDに関する処理は後述する。データがhit clusterの場合、それらのHitがSingle hitなのかクラスターを構成しているのか判断する。最初のhit clusterを受信した後、次のデータがhit clusterの場合、クラスターを構成していると判断し、さらに次のデータを待ち、hit clusterが途切れData HeaderまたはData Trailerが来たらクラスターの終わりと判断して、そのクラスターの中央値とクラスターサイズを出力する。一方、次のデータがData Headerの場合、Single hitと判断して、Strip numberの2倍を出力する。

Pixel クラスタリング

Pixel クラスタリングは2次元方向で行われる。ATLAS pixel moduleは 144×328 pixelsを持ち、これらは16のFEで読み出される。FEは8個ずつ2列に並んで搭載されている。これらのFEの読み出しはハードウェアの制約からコの字型に出力されるため、ソートを行う必要があり、データを一時的に貯める必要があり処理を複雑にしている。hitは32 bit word、40MHzでRODから出力される。Fig.5.4にPixel ModuleのFEの読み出し順番を示す。



図 5.4: Pixel module の FE の読み出し順序

IM の Pixel クラスタリングは Pixel ID cluster システムで行われる。16 の Grid システムと呼ばれる並列システムに分かれており、1 つの Grid システムが 1 つの Pixel Module を扱われ、それぞれの Grid システムは独立してクラスタリングを行う。Pixel ID cluster システムはまず、Hit Dcoder へと入る。Hit decoder はコの字型に読み出された Hit を、クラスタリングできるようにソートを行う。この際には LIFO (Last In First Out) を用いて行われる。LIFO のサイズは 512 word 分確保されている。これは Pileup 80 における Pixel module の occupancy を基に計算されている。また、LIFO サイズを超える場合はクラスターが分割され出力される。LIFO サイズを超えた場合、この情報は error flag として後段に知らされる。Hit decoder によってソートされた Hit は Grid システムへと入るが、Pixel のクラスタリングは時間がかかるため、16 個の Grid システムによって並列処理を行っている。Grid システムでは、Moving Window アルゴリズムを用いてクラスタリングを行う。Fig.5.5 に例を示す。この Window サイズを超えるクラスターは分割されてしまうため、Window サイズは予め計算されている。最初の Window の位置は最初に受信した Hit の位置に依存する。シードとなった Hit は Window 内で隣接する Hit がなくなるまで読み出される。Window を超えた場合 Window を超えたという情報が付加される。Fig.5.5 における a の図における斜線がシードの候補で、この例の場合、1 番の pixel がシードとなる。b,c の図において、隣接する Hit が読み出され、d の図において Window の端まで確認したのでこの Window での読み出しを終了する。その後、シードを変更して同じことを繰り返す。

Hit が Grid システムによってクラスターとしてまとめられた後、中心値の値を計算する。これによりデータを圧縮することが出来る。具体的には、クラスターを構成する hit のうち、hit position の行と列の最大値・最小値から、クラスターが収まる長方形を定義し、その中心値をクラスター中心値として扱い、中心値の position を行と列それぞれ出力する。Fig.5.6 に中心値の計算の例を示す。

Grid システムにてクラスター化処理を行った後、Data Merger によって、並列処理によって算出されたそれぞれのクラスターが統合される。この際には、LVL1ID を基にマージされ、全てのクラスターが揃った段階で Pixel ID Cluster から出力される。

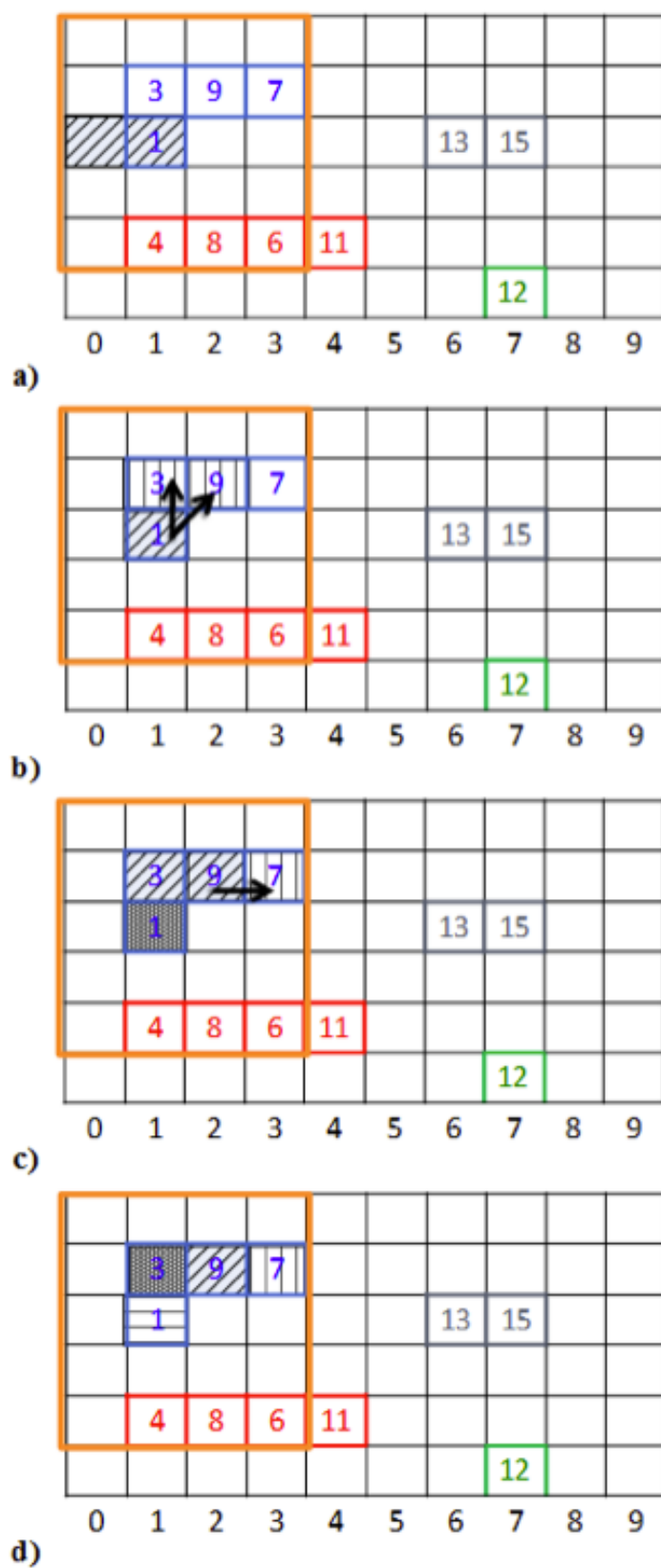


図 5.5: Moving Window を用いた Pixel クラスタリングの例

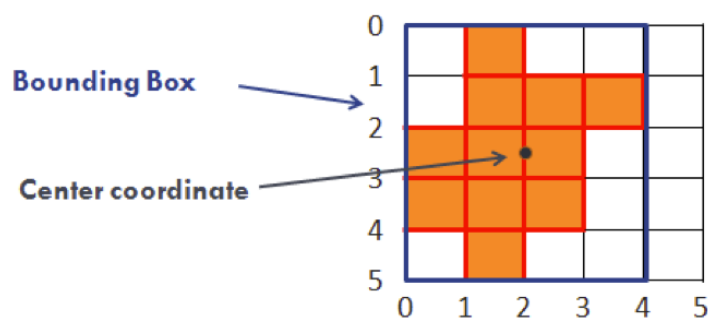


図 5.6: Pixel クラスティングにおける Centroid Calculation の例

5.1.3 モジュール情報の変換

ROD に接続されるシリコン検出器 Module は Run ごとに変更される。これは、Hit の occupancy が上がることや、Dead Module 等によるものである。シリコン検出器 Module を特定するために、ROD data format では Link number と ROD ID を用いて行っているが、FTK では、FTK 共通の通し番号 (Hash ID) を定義しており、この ROD ID + Link number と Hash ID を対応させた LUT を持っている。この変換は IM にて行われる。前述した ROD に接続されるシリコン検出器 Module の変更時には、LUT を新しいものに変更するのみで対応できる。この LUT は ATLAS Run-Control を経由して IM にダウンロードすることができる。このモジュール情報の変換の概念を Fig.5.7 に示す。

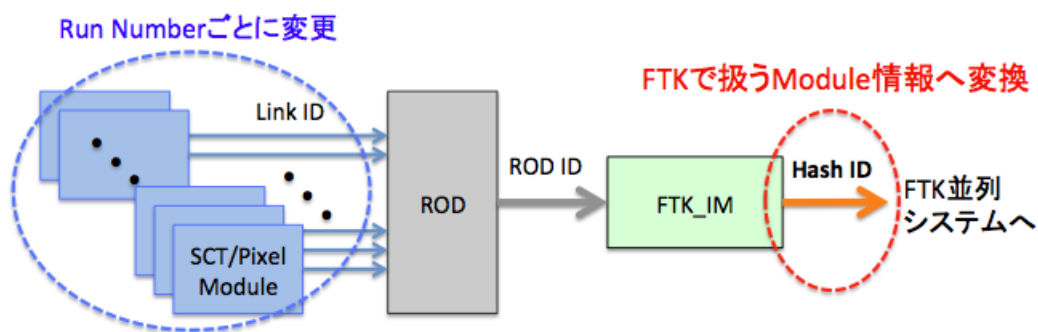


図 5.7: FTK データ形式への変換の概念図

5.2 テストスタンド

これらの処理をエミュレーションに実装し、SCT データを用いてテストを行った。ttbar は非常に density の高いため、高負荷の IM 処理を行うことが出来る。Fig.5.8 にハードウェアのテストスタンドを示す。左のハードウェアが QUEST で、エミュレーションから作製した Input を IM へと出力する。QUEST から 4 チャンネル分を S-Link によって出力することができ、各チャンネルは 1Gb/s で出力する。IM はこれらのデータを S-Link 経由で受信し、FPGA にてハードウェア処理を行い、DF へ FMC コネクタを経由して出力する。DF は Spy buffer に受信したデータのコピーを行う。その後、ROS PC から、DF に搭載されたマイクロコントローラ経由で Spy buffer を読み出す。

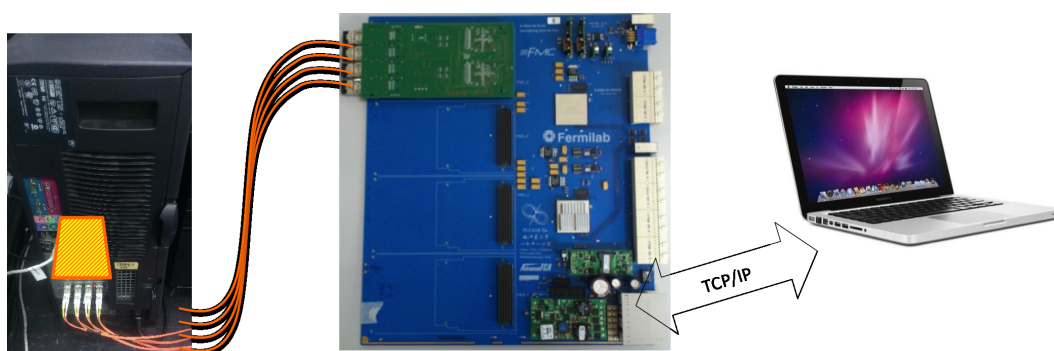


図 5.8: ハードウェアエミュレーションのハードウェアテストスタンド

5.3 テスト結果

	Hit情報の送り方	平均Hit Word数 /ev.
Raw	1 hit / word	1211
ROD	1 or 2 hit / word	1010
FTK	クラスター情報 / word	580

表 5.6: 600event における 1event 中の平均 Hit word 数

IM ハードウェア Output をエミュレーション Output と比較し、Single μ サンプル 5000 event 全てでデータが一致することを確認した。ハードウェアに実装したクラスタリングアルゴリズムや Module 情報の変換をはじめとしたハードウェア処理が予期した通り動作していることがわかった。また、Pile up 60 における ttbar サンプル 600 event を用いて同様にテストを行い完全に一致することを確認した。

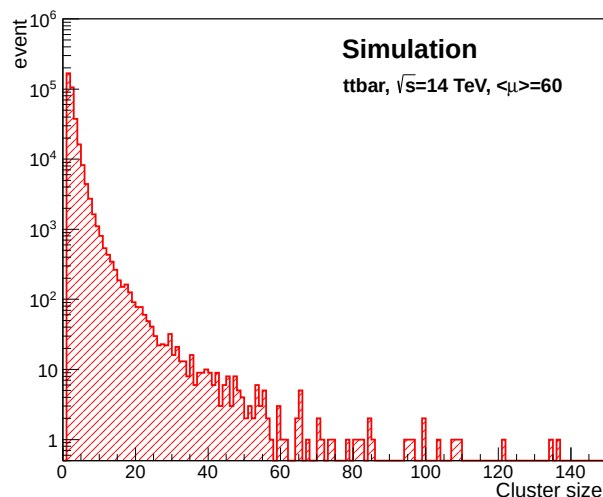
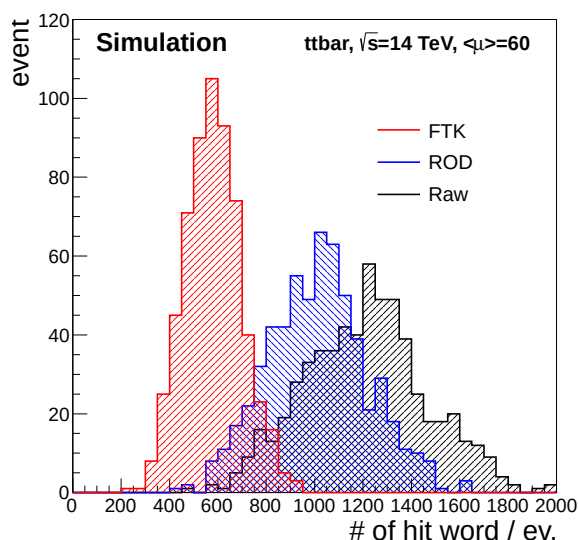


図 5.9: 各データフォーマットにおける 1event 中の Hit word 数のヒストグラム

Fig.5.9 に、ROD data format/FTK data format/Raw data format(1word につき 1Hit を送信する場合)の 1event 中の Hit word 数のヒストグラムを示す。また、Fig.5.10 にクラスターサイズのヒストグラムを、Table.5.6 に各データフォーマットにおける 1event 中の平均 Hit word 数を示す。Raw data format とは、単純に 16bit word につき 1 Hit を送信する format である。FTK data format は ROD data format/Raw data format に比較して情報が圧縮されていることがわかる。これは、ROD data format は、連続した Strip の場合 16 bit word に最大 2hit を含むため、Raw に比べて情報は圧縮され、また、FTK data format は 16 bit word にクラスター情報を含むため、ROD と比べてもさらに情報は圧縮されるためである。さらに、平均のクラスターサイズは 2.1 なため、FTK data format の 1 event 中の Hit word 数と比較して、Raw data format は約 2.1 倍の Hit word 数を持つことが予想でき、Fig.5.9、Table.5.6 が妥当なことが確認できる。Fig.5.10 を見ると、クラスターサイズが 7 以上のクラスターが 1%程度存在する。FTK data format ではクラスターサイズを表す bit 幅が、現在 3bit しか用意していないため、8 以上は全て overflow flag が立つのみで、クラスターサイズ 8 もクラスターサイズ 140 も後段では同じに見えてしまう。さらに、現在の SS サイズは数十を想定しているため、100 以上だと SS を超えてしまい、本来なら SS2 つにまたがる Hit が SS1 つとなってしまうため、パターン認識できずに Inefficiency になる可能性が考えられる。この Inefficiency を救うためには、FTK data format におけるクラスターサイズを表す bit 幅をより多く用意するか、SS サイズを大きくする必要がある。SS サイズを大きくすると AM に格納する Road の数が小さくなる一方で、Fit の回数が乗数で増加するため、Fit に時間がかかってしまう。一方、FTK data format の Hit word はまだまだ余っている bit が多く存在する。これらはまだ何に使用するのは未定だが、クラスターサイズを表す bit 幅を多くすることは十分に可能である。PU60 では 1%程度しか存在しない overflow だが、PU が今後どんどん増加していくに従ってシリコン検出器の

occupancy が増加し、この overflow が増加していくと考えられる。これらの影響については、今後、このハードウェアエミュレーションを使用して様々な Sample を Input として調べていく必要がある。

第6章 まとめと展望

本研究では ATLAS の増強計画の 1 つとして 2015 年より挿入される IM の開発とハードウェアエミュレーションを用いた IM の性能評価を行った。IM の開発では、Version 3.1 の開発から携わり、プロダクトバージョンである Version 4.0 の製作を行った。とくに、X 線・実体顕微鏡を用いた不良解析や各種テストを始め Quality Control 試験の整備など様々な開発を行った。今後は、早稲田大学にて Quality Control 試験を行った後、CERN へと郵送してインストールを行って行く必要がある。また、ハードウェアエミュレーションでは、エミュレーションの構築を行い、IM の Output の詳細検証を行った。エミュレーションには C++ や Python を用いて、クラスタリングや Module 情報の変換を実装し、これらをハードウェアで扱えるデータ形式に変換した。具体的には、1/0 の電圧値によるデータで、IM の Input では ROD data format に、Output では FTK data format に変換した。Single μ サンプル 5000 event をエミュレーションの Input として IM ハードウェアの Input データを作製し、これを Input とした IM ハードウェアの Output と、エミュレーションから作製したエミュレーション Output を比較し、5000event の全ての bit で完全に一致することを確認した。今後は、IM 以外の FTK 各ボードのテストベクターを作製し、各ボードの Output の質の保証を行っていく必要がある。また、PU60 における ttbar サンプル 600 event も同様のテストを行った。Density の高い ttbar サンプルではクラスタサイズが 7 以上となるクラスタが全体の 1% 程度存在することがわかった。クラスタサイズが大きくなると後段のパターン認識に失敗し、Inefficiency を産む可能性がある。今後、様々なサンプルでハードウェアエミュレーションを行い、この影響を詳細に検証し、FTK data format におけるデータ形式の最適化等に繋げていく必要がある。

参考文献

- [1] ATLAS Collaboration: "The ATLAS Experiment at the CERN Large Hadron Collider", 2008.
- [2] ATLAS FTK Group: "FTK: a hardware track finder for the ATLAS trigger Technical Proposal", 2010.
- [3] ATLAS Photo Page
<http://www.atlas.ch/photos/index.html>
- [4] ATLAS LuminosityPublicResults Homepage
<https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResults>
- [5] ATLAS Beam Spot Web Page
<https://atlas-beamspot.cern.ch/webapp/myplots/>

第7章 謝辞

学部4年生からの3年間をはじめ、本研究を進めるにあたり、寄田浩平准教授には終始熱心に丁寧に、ときには厳しくご指導を賜りました。未熟な私に、研究の楽しさを始め、研究に取り組む姿勢や物の考え方を教えて下さり、新たな目標を見つけるきっかけを下さいました。寄田研究室に在籍した3年間で、物理学を通して、今一度自らに向き合う時間を多く作ることができ、人間として大きく成長できたと感じております。深く感謝申し上げます。

沢山のご助言を頂き無知な私に考える機会を与えて下さり、研究の楽しさを教えて下さったばかりか、いつも丁寧に質問に対応して下さいました木村直樹氏に感謝致します。研究以外の様々な話題でも気軽にご相談に乗っていただき、研究で行き詰まっていた際には、肩の力が抜け、視点を変えるきっかけとなりました。とくに、木村氏がアリストテレス大学に行かれてからの1年間は Skype を通して、早稲田大学での進捗を常にお気遣い下さり、今は何を進めるべきなのか、何をしなければいけないのか学生同士で議論するきっかけを与えて下さいました。本当に大変お世話になりました。

また、的確なご指摘をいただき深く考えるきっかけを下さった田中雅士次席研究員に深く感謝致します。FTK ミーティングに参加された際には、おそろしく的確なご助言を下さいました。毎回質問に答えられず、田中氏の質問を先読みすることが深く考えるきっかけとなりました。それでも質問には答えられず、自分の考えの浅さを痛感致しました。

研究を進めていく上で、コンピュータ環境を整えて下さった蛭名幸二氏にも深く感謝致します。研究をはじめ、とくに私生活においても、常に俯瞰的な視点でご助言下さり、視野が狭くなりがちな自分に新しい視点を与えて下さりました。常に温かい言葉をお掛け下さり前向きになることができました。

坂本敦子秘書にも大変感謝しております。研究を進めるにあたり、多くの事務手続きがありましたが、全てスムーズに進んだのは坂本氏のおかげです。毎回、私の書類には不備があり、多大なご迷惑をお掛けしましたが、毎回、正確かつ迅速にご対応下さいました。とくに、いつも明るく笑顔で接して下さい、根詰めて研究をしていた際には何度も助けられました。本当にありがとうございました。

博士の先輩方にも大変感謝致します。学部4年生の頃から、大変お世話になり、研究に取り組む姿勢を先輩方の背中を見て教わりました。とくに、ATLAS グループとして三谷氏、桜井氏、飯沢氏には、無知な質問にも大変ご丁寧に対応して下さい、自身が何を理解していないのか把握することが出来ました。先輩方と議論できることが非常に嬉しく、研究の一つの楽しみとなっておりました。

FTK 開発に共に携わった後輩の川口氏、亘氏にも大変感謝致します。数多くの議論を行い、共に問題解決に取り組んだ日々は、非常にエキサイティングで大変充実した時間でした。これから、今後の FTK の運用に向けて多に活躍してくれることを期待しております。

また、FTKの開発にあたって、Alberto Annovi氏、Akis氏、Calliope氏、Jamison氏、奥村氏にも深く感謝致します。

同期の加地俊瑛君、川村将城君、澤村治基君、仲松弥君、鷺見貴生君、先輩の昌子貴洋氏には多大なご迷惑をおかけしたにも関わらず、忍耐強く最後まで付き合ってくださいました。共に深く議論し、多くの刺激を与えて下さり、一緒に研究できたことは大変幸せでした。本当にありがとうございました。進路は様々ですが、これからも変わらずに議論し、お互いに切磋琢磨していけたら幸いです。

Input Mezzanineの開発にあたって、非常にスピード感のある最先端の現場で研究に携わることができ、本当に幸せであったと感じております。現場にいとその感覚を忘れ、なかなか研究が進まずに苦しいと感じることもありましたが、振り返りますと研究に本気で向き合った時間は大変有意義で、本当に多くのことを学ばせていただきました。私は来年度から物理学の基礎研究から離れますが、最先端の研究に携わり世界中の多くの研究者と関わり、議論できるエキサイティングな環境に身を置けたことは非常に幸せであったと感じております。

最後に、未熟な私を大学院卒業まで支えて下さった両親、兄妹に心から深く感謝致します。