LHC-ATLAS 実験
レベル 1 ミューオントリガーのための
データ収集システムのアップグレード

東京大学大学院理学系研究科　物理学専攻
素粒子物理国際研究センター　坂本研究室
浦野　祐作

2015 年 1 月 5 日

概 要

Large Hadron Collider（LHC）はスイス・ジュネーブ郊外の欧州共同原子核研究機構に建設された大型ハドロン衝突型加速器である。2012 年には重心系エネルギー 8 TeV、最大瞬間ルミノシティ0.7×10^{34} cm^{-2}s^{-1}、バンチ衝突頻度 20 MHz で稼働し、積分ルミノシティ30 fb^{-1} のデータを取得した。現在は LHC のアップグレードのために運転を停止しており、2015 年に Run-2 として実験を再開する予定である。このシャットダウン期間は目前に迫った Run-2 だけでなく、2020 年開始の Run-3 を見据えたアップグレードも行っている。

ATLAS 検出器は LHC の衝突点の 1 つに建設された大型汎用検出器で、未知の粒子の直接的な検出や素粒子標準模型を超えた物理現象の探索を目的としている。2012 年 7 月、ATLAS 実験及び CMS 実験により、先述の主目的である「未知の粒子の発見」を達成した。その後、2013 年 3 月にその粒子は Higgs 粒子であると断定され、これにより素粒子標準模型における全ての素粒子の存在が証明されたことになる。Higgs 粒子の精密な測定は未解決の物理現象を解く手がかりとなるため、LHC にはさらなる期待が寄せられている。

私が参加している ATLAS 実験の研究グループでは LHC の段階的なアップグレードに応じて、エンドキャップミューオントリガー検出器 Thin Gap Chamber（TGC）のトリガー及びデータ収集システムのアップグレードを行っている。TGC のアップグレードの目的の 1 つに、2012 年に終了した Run-1 で問題となったフェイクトリガーの削減が挙げられる。ATLAS 実験ではフェイクトリガーを削減するために、Run-3 から新しいミューオン検出器を導入する予定で、TGC グループはそれを受けて新しい読み出し系を構築する必要がある。

私はその中でも、TGC のトリガーデータが最終的に集まるモジュールである ReadOut Driver（ROD）のアップグレードに尽力しており、本論文ではプロセス間通信を用いたソフトウェアベースの ROD の検証実験について述べている。検証実験には、新しい Timing Trigger Control Readout Module 及び Sector Logic を模したエミュレータや汎用 VME モジュールを用いており、開発した ROD は要求される性能を満たしながら動作することが確かめられた。具体的には次の 2 点である。1 つは、新しい ROD には最大で入力 1856.0 Mbps、出力 1916.8 Mbps の転送能力が要求されているのに対し、エミュレータを用いた試験で処理することができた点で、もう 1 つは、スイッチングハブ及びイーサネットケーブルを用いて 2 対 1 の収集系を構築して、汎用 VME モジュールから約 380 M イベントのデータを収集できた点である。つまり、ソフトウェアベースの ROD は適切なプラットフォーム及び周辺機器を用意することで、Run-3 における読み出し系への組み込みが可能であることを示唆している。

目 次

第 1 章 LHC と ATLAS 実験 **7**

1.1 LHC 加速器 7

1.1.1 LHC 加速器の概要 7

1.1.2 LHC 加速器を用いた実験 9

1.1.3 LHC のアップグレード計画 10

1.2 ATLAS 実験 11

1.2.1 ATLAS 検出器の全体像 11

1.2.2 内部飛跡検出器 13

1.2.3 カロリメータ 14

1.2.4 ミューオンスペクトロメータ 15

1.2.5 超電導マグネットシステム 17

第 2 章 レベル 1 エンドキャップミューオントリガーシステム **19**

2.1 ATLAS におけるトリガー及びデータ収集システム 19

2.1.1 レベル 1 トリガー 19

2.1.2 レベル 2 トリガー及び EF 21

2.2 TGC の概要 22

2.2.1 TGC の構造及び動作原理 22

2.2.2 TGC の配置 24

2.3 トリガースキーム 24

2.3.1 p_T 測定の原理 24

2.3.2 コインシデンス処理 25

2.4 エレクトロニクス 26

2.4.1 システム全体のデータフロー 26

2.4.2 リードアウト系のエレクトロニクス 30

2.5 Run-3 のリードアウト系でアップグレードするエレクトロニクス 33

2.5.1 Sector Logic 33

2.5.2 ReadOut Driver 34

2.6 Run-3 に向けて 39

2.6.1 ミューオントリガーシステムの問題点 39

2.6.2 ミューオン検出器のアップグレード 40

第 3 章 プロセス間通信を用いた ReadOut Driver の開発 **42**

3.1 Run-3 におけるリードアウトスキーム 42

3.1.1 SiTCP 43

3.1.2 ROD に対する入力フォーマット 43

3.2 Software ROD 開発の目的 45

3.3 Software ROD に要求される性能 45
3.4 プロセス間通信 47
3.4.1 ネットワークを経由したプロセス間通信 47
3.4.2 同一マシン内のプロセス間通信 48
3.4.3 ロック機構 49
3.5 システムの構成 50
3.5.1 RingBuffer 54
3.5.2 TTC/SLCollector プロセス 55
3.5.3 EventBuilder プロセス 57
3.5.4 ReadoutMaster プロセス 59

第 4 章 ReadOut Driver の性能評価 61
4.1 エミュレータを用いた性能評価 61
4.1.1 エミュレータ 62
4.1.2 Software ROD の評価に際して 62
4.1.3 通信単位の性能評価 63
4.1.4 動作試験 68
4.2 汎用 VME モジュールを用いた接続試験 73
4.2.1 汎用 VME モジュール PT7 73
4.2.2 テストベンチ及び収集系で用いる機器 73
4.2.3 動作試験 74
4.3 Distributor を用いた性能評価 75
4.3.1 Distributor 75
4.3.2 動作試験 77
4.4 結論 78

第 5 章 まとめ 79

図 目 次

1.1 LHC の外観 7
1.2 LHC の前段加速器 8
1.3 検出器の外観 10
1.4 検出器の外観 10
1.5 ATLAS の座標系 11
1.6 ATLAS 検出器の全体像 12
1.7 素粒子に対する検出概念図 12
1.8 内部飛跡検出器の構成 13
1.9 カロリメータの全体像 14
1.10 LAr 電磁カロリメータの構造 14
1.11 ミューオンスペクトロメータの全体像 15
1.12 ミューオンスペクトロメータの配置 16
1.13 MDT の構造 16
1.14 CSC の構造 16
1.15 RPC の構造 17
1.16 超電導マグネットシステム 17

2.1 トリガー及び TDAQ スキーム 20
2.2 レベル 1 トリガーシステム 20
2.3 TTC システム 21
2.4 セクター 22
2.5 トリガーセクター及びサブセクター 22
2.6 TGC の構造 23
2.7 TGC のパラメータ 23
2.8 TGC の Triplet と Doublet の構造 23
2.9 TGC のレイアウト 24
2.10 p_T 算出の概念図 25
2.11 Coincidence Window の例 26
2.12 レベル 1 トリガーメニューと p_T の対応関係 26
2.13 TGC エレクトロニクスの全体像 27
2.14 TGC のトリガー系 28
2.15 TGC のリードアウト系 29
2.16 TGC のコントロール系 29
2.17 ASD Board の写真 30
2.18 PS Board のブロック図 31
2.19 PP ASIC のブロック図 31
2.20 TGC のリードアウトライン 32

2.21 ゼロサプレスの概念 32
2.22 SSW ボードの写真 33
2.23 SL の写真 33
2.24 ROD の写真 34
2.25 ROD のブロック図 35
2.26 ROB 入力フォーマット 36
2.27 フォーマッティングの過程 37
2.28 TGC における ROB 入力フォーマット 38
2.29 MDT 及び CSC のカウントレート 39
2.30 フェイクトリガーを起こす荷電粒子の飛跡の例 40
2.31 $d\theta$、dL カットの概念 41

3.1 Run-3 における New-SL のリードアウトスキーム 42
3.2 New-TTC/SL 及び Software ROD によるリードアウトスキーム 43
3.3 Run-3 における TTC フォーマット 44
3.4 Run-3 における SL フォーマット 44
3.5 1/12 セクターから ROD までのスキーム 45
3.6 ソケット通信の手順 47
3.7 共有メモリの概念 48
3.8 システム全体の機能のユースケース図 50
3.9 「イベントデータフラグメントを収集する」のユースケース図 51
3.10 「イベントデータを送信する」のユースケース図 52
3.11 Software ROD の模式図 53
3.12 RingBuffer のコントロール部及びデータ部 55
3.13 TTC/SLCollector プロセスのクラス構成 55
3.14 ソケットバッファからの読み出しで切れたイベントデータフラグメントの対処 . . 56
3.15 EventBuilder のクラス構成 57
3.16 EventBuilder の動作 58
3.17 ReadoutMaster のクラス構成 59
3.18 メインフローのシーケンス図 60

4.1 エミュレータを用いた収集系の構成 61
4.2 TCP ソケット通信の構成 63
4.3 SLEmulator1 台のときの TCP 通信の結果 64
4.4 SLEmulator を 6〜18 台用いた TCP 通信の結果 65
4.5 SLEmulator を 36 台用いた TCP 通信の結果 66
4.6 RingBuffer 単体テストの概略図 66
4.7 RingBuffer を 1 個用いた通信 67
4.8 RingBuffer を 37 個用いた通信 67
4.9 エミュレータの台数を変化させたときの ROD の出力転送速度 69
4.10 台数に応じた転送速度の理論値と測定値 69
4.11 ダンプしたイベントデータの一部 70
4.12 エミュレータを用いたグラフ 70
4.13 エミュレータを用いたグラフ 71

4.14 ステータスブロックの 1 要素目 72
4.15 PT7 の全体像 73
4.16 テストベンチマシン elec22 74
4.17 接続試験の様子 75
4.18 2 対 1 の接続試験の結果 75
4.19 Distributor の概念図 76
4.20 共有メモリの構造 76
4.21 Distributor を用いた接続試験 77
4.22 分岐数と転送速度の関係 78
4.23 Distributor を用いた接続試験の結果 78

表 目 次

1.1 LHC のパラメータ及び設計値 8
1.2 Long Shutdown におけるパラメータの変遷 11

3.1 TTC 及び SL の出力転送速度 45
3.2 DAQ PC の台数と必要な転送能力の関係 46

4.1 login.icepp のパラメータ 63
4.2 Emulator の配置 68
4.3 DAQ PC の台数とカバーできる最大データサイズ 71
4.4 login.icepp のパラメータ 74
4.5 Distributor の動作確認の結果 77

第1章　LHCとATLAS実験

私が参加しているATLAS実験では、世界最高エネルギーを誇るLHCとその衝突点の1つに建設された大型汎用検出器のATLAS検出器を用いて、新しい物理現象を探索している。その中でも、私は2020年開始のRun-3に向けた研究開発に従事している。本章では、前提知識となるLHC及びATLAS実験について解説する。

1.1　LHC加速器

はじめに、ATLAS実験で使用されているLHC加速器について述べる。

1.1.1　LHC加速器の概要

LHCは、スイス･ジュネーブ郊外の欧州共同原子核研究機構（CERN: the European Organization for Nuclear Reseach）の地下に建設された世界最大の陽子 – 陽子衝突型加速器である。現在はTeV領域の物理研究が可能な唯一のエネルギーフロンティア加速器として稼働し、世界中の研究者が参画している。LHCの全体像を図1.1に示す。

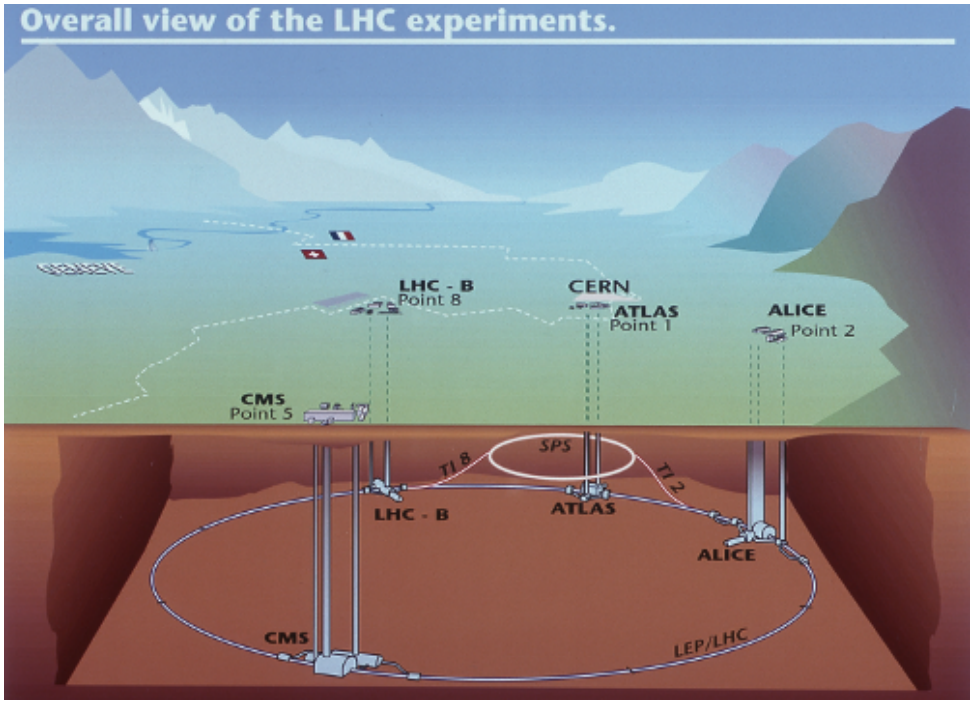

図 1.1: LHCの外観 [6]。CERNの地下約100mにあるトンネル内に設置されている。

LHC は、かつて Large Electron Positron collider（LEP）が稼働していたトンネルに設置されており、トンネル内に超電導二重極マグネットを多数並べることで、史上最高エネルギーのハドロン衝突を可能にしている。シンクロトロン放射 [1] によるエネルギー損失は質量の逆 4 乗に比例するため、電子を加速させる LEP よりも、電子の 1840 倍の質量をもつ陽子を加速させる LHC の方がエネルギー損失が極めて少ない。そのため、LEP よりもはるかに高いエネルギー領域で実験を行うことが可能になっている。

表 1.1: LHC のパラメータ及び設計値 [5]。現段階ではビームエネルギー及び瞬間ルミノシティは設計値に達していない。

パラメータ	設計値
リング周長	26.7 km
ビームエネルギー	7.0 TeV
瞬間ルミノシティ	10^{34} cm^{-2}s^{-1}
バンチ間隔	24.95 ns
バンチ ID 総数	3564
バンチ数	2808
偏向磁場	8.33 T
衝突角度	200 μrad

LHC の主要なパラメータ及び設計値を表 1.1 に載せる。現段階では、ビームエネルギーと瞬間ルミノシティが設計値に達しておらず、今後アップグレードされる予定である。Run-1 でのバンチの衝突頻度は 20 MHz であったが、Run-2 以降では 40 MHz にアップグレードされるため、背景事象の増加が予想される。そのため、興味のある物理現象をいかに効率よく、かつ正確にトリガー・データ収集できるかが鍵となる。

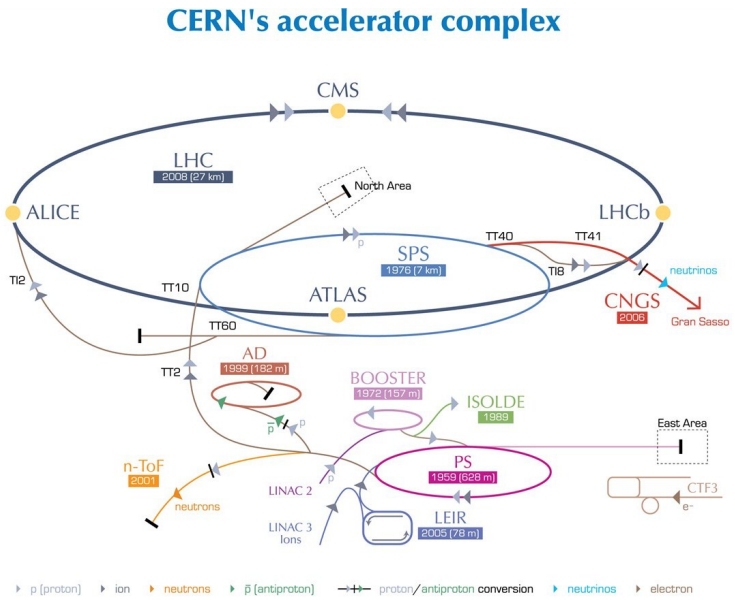

図 1.2: LHC の前段加速器 [6]。段階的に陽子を加速し、LHC へ入射する。

[1] 荷電粒子が磁場中を通過したときに電磁波を放射する現象。

LHC 加速器では、複数の加速器を接続して段階的にエネルギーを上げることで、ビームエネルギー 7 TeV という高エネルギーの加速を実現する。図 1.2 に前段加速器の全体像を示す。ここで、メインリングである LHC に入射されるまでの流れを述べる。まずはじめに、陽子は LINAC2 で 50 MeV まで加速する。その後、PS BOOSTER で 1.4 GeV、Proton Syncrotron（PS）で 25 GeV と上昇していく。そして、Super Proton Syncrotron（SPS）でバンチと呼ばれる陽子の群れを蓄積し、450 GeV まで加速する。最終的に LHC へ流入し、7 TeV まで加速する。

1.1.2 LHC 加速器を用いた実験

LHC リングには 4 つのバンチ衝突点が存在し、それぞれの衝突点に実験施設が設置されている。これらの実験施設において、次の 6 つの物理実験が展開されている。それぞれの実験を ATLAS 実験、CMS 実験、ALICE 実験、LHCb 実験、TOTEM 実験（TOTal Elastic and diffractive cross section Measurement）、LHCf 実験（Large Hadron Collider forward）と呼ぶ。TOTEM 実験は CMS 実験と連携し、光学定理を基盤として陽子衝突全断面積の測定を行っている [13]。また、LHCf 実験は LHC の陽子衝突点から超前方へ飛来するガンマ線、中性子を測定している [14]。以下に、残りの 4 つの実験で用いられる検出器の外観を図 1.3、図 1.4 に載せ、各実験について簡潔に説明する。

ATLAS 実験

ATLAS 実験（A Toroidal LHC ApparatuS）では、新粒子の発見や新しい物理の探索を目的とした衝突実験が行われている。2012 年に設計値の約半分である重心系エネルギー 8 TeV の運用を行い、同年 7 月、新粒子の存在を 5σ の確度で示唆した [1]。現在は、LHC のシャットダウン期間に伴い、ATLAS 検出器もアップグレードしている段階である。ATLAS 実験及び検出器の詳細は次節以降で述べる。

CMS 実験

CMS 実験（Compact Muon Solenoid）は ATLAS 実験と同様に、大型汎用検出器を用いて新粒子の発見や新しい物理の探索を目的とした衝突実験である。この実験で用いられる CMS 検出器は高さ 15 m、長さ 21 m で、名前の通り ATLAS 検出器より一回り小さい。磁場 4 T のソレノイドマグネット、$PbWO_4$ シンチレータを用いた電磁カロリメータが組み込まれているなど、ATLAS 検出器とは異なった思想で設計されている。CMS 実験グループは、ATLAS 実験グループと同時期に Higgs 粒子とみられる粒子を発見した [2]。

ALICE 実験

ALICE 実験（A Large Ion Collider Experiment）では、重イオンを用いた衝突実験を行っている。現在の LHC の長期シャットダウン期間を経て、重心系エネルギー 5.1 TeV、瞬間ルミノシティ 10^{27} $\mathrm{cm^{-2}s^{-1}}$ の鉛原子核同士の衝突実験が開始される予定である。鉛の原子核は陽子より重く、高エネルギー領域での衝突によって生じるクォーク・グルーオン・プラズマ（QGP）の性質を解明することを目的としている。

LHCb 実験

LHCb 実験（Large Hadron Collider Beauty）では、b クォーク物理に特化した実験を行っている。陽子陽子衝突により前方に　b クォーク及び c クォークを含む粒子が大量に生成され、その粒子の崩壊事象を観測・検証することで CP 対称性の破れやそれに付随する新しい物理を探索している。

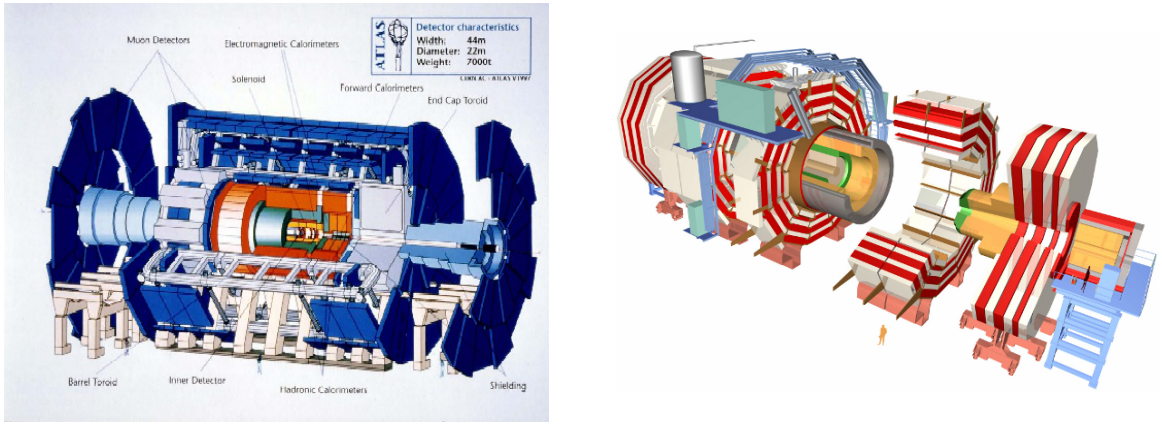

図 1.3: 検出器の外観 [6]。左：ATLAS 検出器、右：CMS 検出器

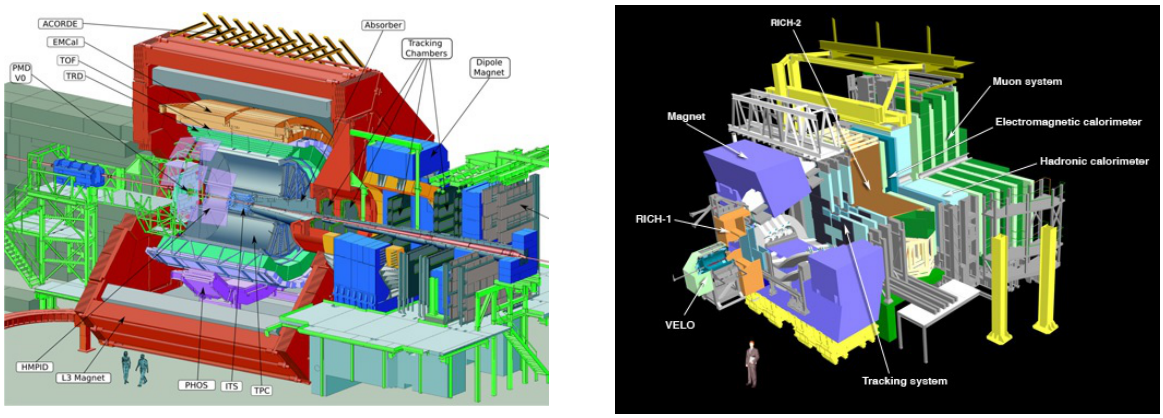

図 1.4: 検出器の外観 [6]。左：ALICE 検出器、右：LHCb 検出器

1.1.3 LHC のアップグレード計画

LHC は High Luminosity LHC（HL-LHC）に向けて計画的にシャットダウン期間（Long Shutdown）を設け、その期間に重心系エネルギーや瞬間ルミノシティなどのアップグレードを行う。HL-LHC は 2030 年までに積分ルミノシティ3000 fb^{-1} のデータを取得することを目標とし、Higgs 粒子の性質の精密測定や超対称性粒子の探索などを行う。各 Long Shutdown における重心系エネルギーと瞬間ルミノシティの変遷を表 1.2 にまとめる。以下のようにパラメータを段階的にアップグレードするため、各 Long Shutdown では装置の補修、取替などを行う。

- Long Shutdown 1
  加速管同士の接合の強化、弱くなった磁石の交換といった補修が主な作業となる。
- Long Shutdown 2
  LINAC2 から LINAC4 への移行、PS BOOSTER のエネルギー増強といった前段加速器のアップグレードを中心にメンテナンスが行われる。
- Long Shutdown 3
  ルミノシティを 5×10^{34} $\mathrm{cm}^{-2}\mathrm{s}^{-1}$ へアップグレードするために、大口径磁石や超電導クラブ空洞といった新技術を導入する。

表 1.2: Long Shutdown におけるパラメータの変遷 [10]。3 回のシャットダウン期間を設けて段階的にパラメータを上げていく。

期間	Run	重心系エネルギー [TeV]	瞬間ルミノシティ [$\mathrm{cm^{-2}s^{-1}}$]
2011-2012	Run-1	8	0.6×10^{34}
2013-2014		[Long Shutdown 1]	
2015-2017	Run-2	13〜14	1×10^{34}
2018		[Long Shutdown 2]	
2020-2022	Run-3	14	2×10^{34}
2023		[Long Shutdown 3]	
2024-		14	5×10^{34}

1.2 ATLAS 実験

本節では大型汎用検出器を用いた ATLAS 実験について述べるが、その前に ATLAS の座標系について説明する。ATLAS ではいくつかの座標系が使われており、いずれも座標の原点は陽子と陽子の衝突点（IP: Interaction Point）と定義されている。我々に馴染み深い xyz 直交座標系は、z 軸をビーム方向、x 軸を LHC リングの中心方向、y 軸をそれらと垂直な方向とした右手系である。このとき $z > 0$ の領域を A-side、$z < 0$ の領域を C-side と呼ぶ。また、原点からの天頂角を θ と定義したとき、擬ラピディティ $\eta = -\ln(\tan(\theta/2))$ という変数もよく使われる。

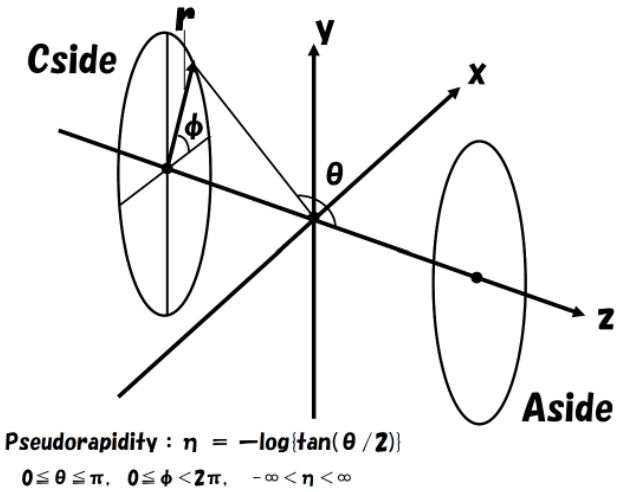

図 1.5: ATLAS の座標系 [30]。検出器のカバー領域の区別に擬ラピディティ η をよく用いる。

1.2.1 ATLAS 検出器の全体像

ATLAS 検出器（図 1.6）は直径 25 m、長さ 44 m、重量 7,000 t の円筒形をした巨大な複合検出器である。一般的な衝突型加速器と同様に、内側から内部飛跡検出器、電磁カロリメータ、ハドロンカロリメータ、ミューオンスペクトロメータという構成をとる。また、ATLAS 検出器にはソレノイドマグネット及びトロイドマグネットの 2 種類の超電導マグネットが組み込まれており、荷電粒子の運動量測定に使用されている。

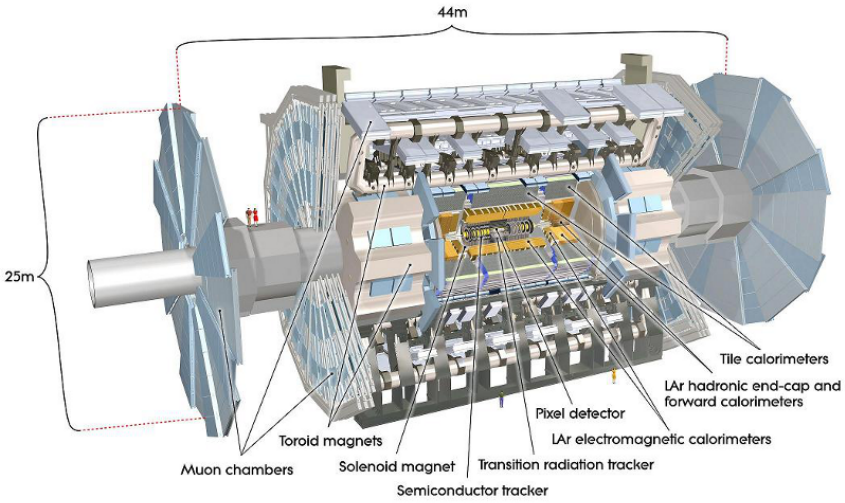

図 1.6: ATLAS 検出器の全体像 [7]。複数の検出器で構成された巨大な検出器である。

ATLAS 検出器は、素粒子に対する各検出器の検出概念図 1.7 に従って構成されている。飛跡検出器はソレノイドマグネットの磁場によって曲げられた粒子の飛跡を再構成し、運動量の精密な測定を行う。2 種類のカロリメータにおいては、電磁カロリメータは電磁シャワーを用いて電子と光子のエネルギー及び放出角度の精密な測定を行い、ハドロンカロリメータはハドロンシャワーを用いてジェットの精密な測定を行う。ミューオンスペクトロメータは最外層に位置しており、カロリメータを透過するミューオンの運動量及び位置を測定する。次項以降では、各検出器の特徴を述べる。

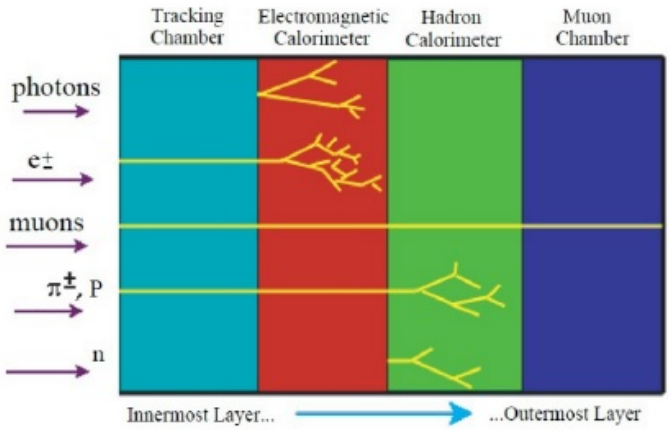

図 1.7: 素粒子に対する検出概念図 [34]。ミューオンは透過力が強いため、それを捕捉するミューオンスペクトロメータは最外層に設置されている。

1.2.2 内部飛跡検出器

内部飛跡検出器（ID： Inner Detector）は ATLAS 検出器の最内部に設置されており、そのカバー領域は $|\eta| < 2.5$ と非常に広範囲である。主な役割としては、素粒子の飛跡再構成及び運動量測定を担っている。配置は図 1.8 のようになっており、内層のシリコン検出器と外層の遷移放射検出器によって構成されている。シリコン検出器は非常に細分化されたチャンネルによって優れた位置分解能を有しており、Pixel 検出器と SCT の 2 種類の検出器で成り立っている。

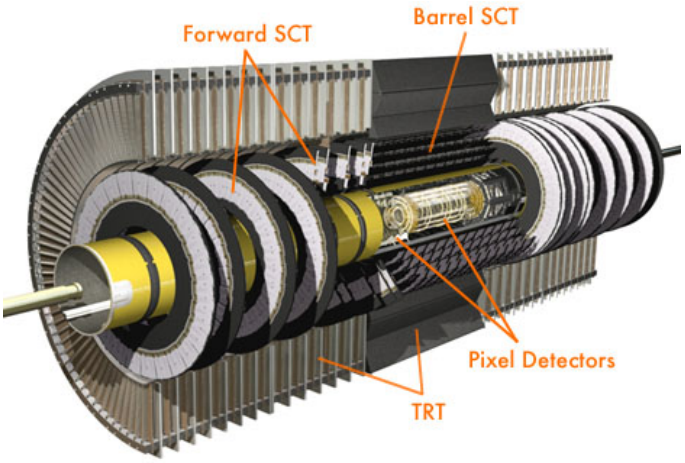

図 1.8: 内部飛跡検出器の構成 [12]。広い範囲をカバーし、粒子の飛跡再構成及び運動量測定を行う。

Pixel 検出器

Pixel 検出器はシリコンを用いた半導体検出器であり、衝突点から最も近い距離に設置されている。”ピクセル”とはセンサーを $50 \times 400 \mu m^2$ に分割したときの 1 ユニットを指し、1 つ 1 つのピクセルが 2 次元読み出しが可能な検出器として働く。この特徴により、優れた位置分解能を実現している。Pixel 検出器はバレル領域（$|\eta| < 1.05$ の領域）、エンドキャップ領域（$1.05 < |\eta| < 1.9$ の領域）ともに 3 層構造で、衝突点から近い順に B-Layer、Layer-1、Layer-2 と呼ばれる。

SCT

SemiConductor Tracker（SCT）は読み出しシリコンストリップを持った検出器で、Pixel 検出器の外側に位置している。 1 つのモジュールは 2 枚のシリコンストリップセンサーで構成され、各々のセンサーを 40 mrad だけ傾けて設置することで 2 次元の位置情報を提供する。

TRT

Transition Radiation Tracker（TRT）は遷移輻射 [2] を利用した検出器である。検出器内部には半径 4 mm のストローチューブがあり、各チューブの中には混合ガス（Xe 70 %、CO_2 27 %、O_2 3 %）が充填されている。各ストローチューブのドリフト時間を測定することで 130 μm の位置分解能が得られる。バレル領域が 4 層、エンドキャップ領域が 9 層という構造になっている。

[2]誘電率の異なる 2 つの物質の境界面において、荷電粒子が電磁放射をする現象。

1.2.3　カロリメータ

一般的なカロリメータは電子や光子、ジェットなどのエネルギー及び放出角度の測定をする役割を担う。ATLAS 検出器においても同様に、電磁カロリメータが電子と光子の情報を、ハドロンカロリメータがジェットの情報を提供する。ATLAS 検出器のカロリメータは LAr（液体アルゴン）電磁カロリメータ（EM）、バレル部タイルハドロンカロリメータ（Tile）、エンドキャップ部 LAr ハドロンカロリメータ（HEC）、フォワード部 LAr カロリメータの 4 種類で成り立っている。図 1.9 にカロリメータの全体像を示す。

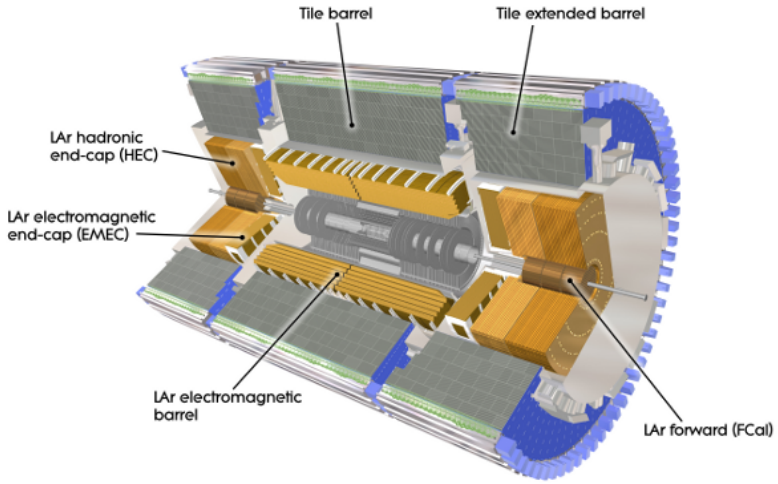

図 1.9: カロリメータの全体像 [7]。電子や光子、ハドロンなどのエネルギー測定を行う。

電磁カロリメータ

ATLAS における電磁カロリメータは LAr と鉛の吸収体を組み合わせたサンプリングカロリメータである。アコーディオン構造が特徴で、ϕ 方向に対するモジュール間の隙間をなくし不感領域を排除している。バレル領域の電磁カロリメータは 3 層構造で、それぞれ役割が異なる。第 1 層では η の精密測定、第 2 層では電磁シャワーの測定、第 3 層では電磁シャワーとハドロンシャワーの区別を行う。また、エンドキャップ領域の電磁カロリメータは各サイドに 1 枚ずつホイールとして設置されている。

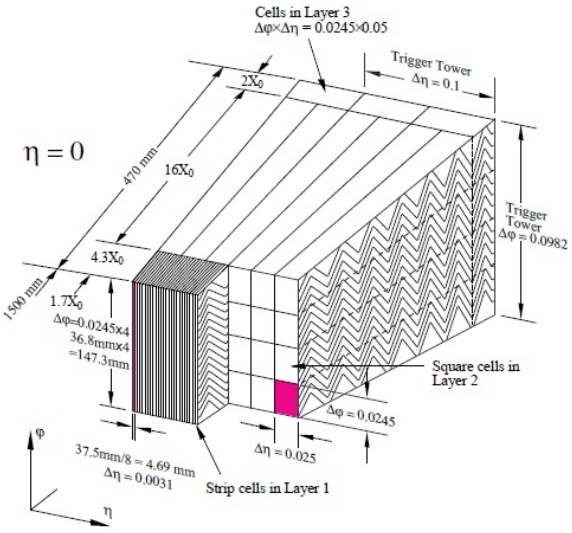

図 1.10: LAr 電磁カロリメータの構造 [7]。アコーディオン構造により不感領域を排除している。

ハドロンカロリメータ

ハドロンカロリメータは電磁カロリメータで通過してしまうハドロンを捕捉する役割を担う。バレル領域とエンドキャップ領域で構造が異なり、バレル領域をタイルカロリメータ、エンドキャップ領域を Hadronic Endcap Calorimeter（HEC）と Forward Calorimeter（FCal）が担当する。タイルカロリメータはタイル状のプラスチックシンチレータと鉄の吸収体の組み合わせで成り立っている。シンチレータの両端に波長変換ファイバーが取り付けられ、光電子増倍管によって信号を読み出す。HEC は電磁カロリメータと同様の LAr を用いたサンプリングカロリメータであるが、吸収体として銅を採用している点で異なり、32 個のくさび形モジュールを組み合わせて円環状を成している。エンドキャップ領域の中でも、超前方の領域をフォワード領域（$1.9 < |\eta| < 2.4$）と呼ぶが、そこでは銅とタングステンの吸収体と LAr からなる FCal が設置されている。

1.2.4　ミューオンスペクトロメータ

ATLAS が目指す物理にはミューオンが密接に関わっており、ATLAS 実験においてミューオンスペクトロメータは重要な役割を果たしている。ミューオンスペクトロメータは軌跡精密測定用の Monitored Drift Tube（MDT）と Cathode Strip Chamber（CSC）、トリガー用の Resistive Plate Chamber（RPC）と Thin Gap Chamber（TGC）の 4 種類の検出器と超電導マグネットシステムで構成される。ミューオンスペクトロメータの全体像を図 1.11 として以下に示す。

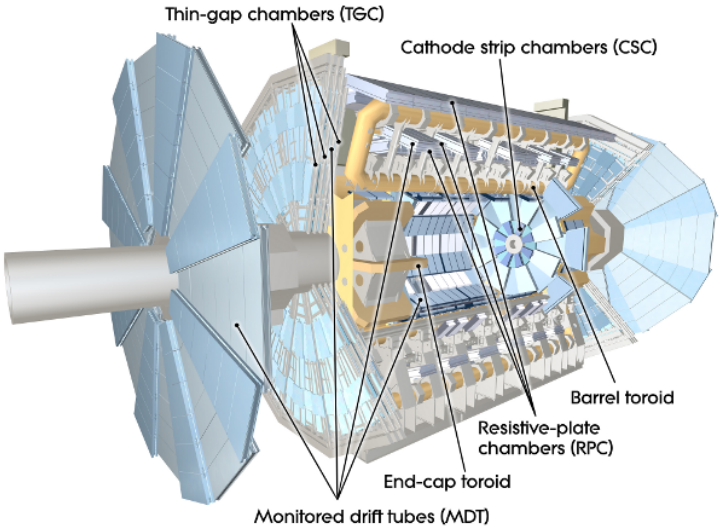

図 1.11: ミューオンスペクトロメータの全体像 [7]。$z \sim 7$ に Small Wheel、$z \sim 13$ に Big Wheel がある。

ミューオンスペクトロメータは ATLAS 検出器の最外部に配置され、これらに内包されるように超電導トロイダルマグネットが設置されている（配置は図 1.12 に示す。）。超電導トロイダルマグネットは ϕ 方向の磁場を作っており、理想的にはミューオンは $r-z$ 平面内で曲げられるが、実際は磁場の不均一性から ϕ 方向にも運動する。TGC の説明は次章に譲り、ここでは MDT、CSC、RPC の 3 つの検出器について述べる。

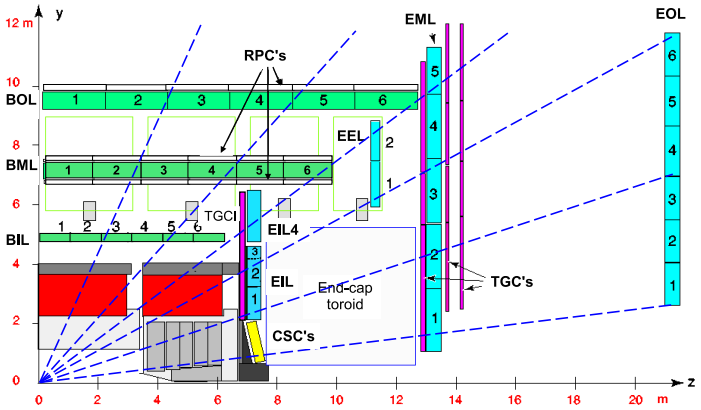

図 1.12: ミューオンスペクトロメータの配置 [7]。

MDT

MDT は混合ガス（Ar 93 %、CO_2 7 %）が封入された直径 30 mm のカソードチューブを作動させる飛跡検出器である。カソードチューブの中には半径 50 μm のアノードワイヤが張られており、チューブを 3 本または 4 本に積層したものが 1 つのチェンバーとなる。この検出器はバレル領域、エンドキャップ領域の広範囲をカバーし、$r-z$ 方向成分を精密に測定する。位置分解能は 60 μm で、総チャンネル数は 30 万チャンネルに上る。

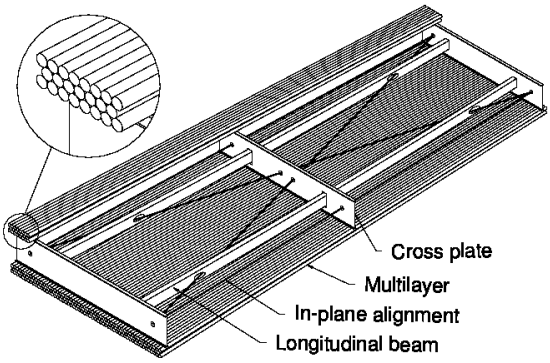

図 1.13: MDT の構造 [9]。バレル、エンドキャップ領域の広範囲をカバーする。

CSC

CSC はトラッキング用のカソードストリップ読み出し MWPC で、片サイドに大小 8 枚ずつのチェンバーが交互に並んだ構造をしている。放射線強度の高いフォワード領域に設置され、MDT より高いカウントレートで読み出す。ワイヤ間隔が 2.54 mm、ストリップ間隔が 5.08 mm という構造をしており、ドリフト時間が 30 nsec 以下、位置分解能 60 μm 及び時間分解能 7 ns という性能を持つ。

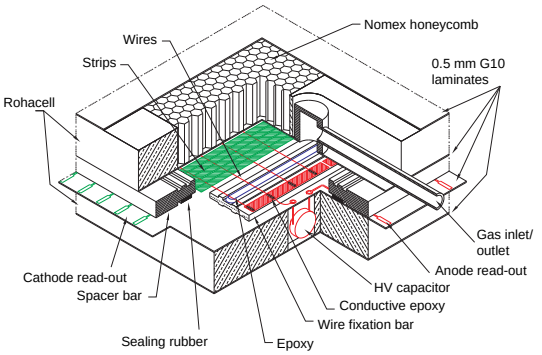

図 1.14: CSC の構造 [9]。放射線強度の高いフォワード領域に設置されている。

RPC

RPC はバレル領域に設置されている検出器で、運動量測定とトリガー判定を行う。resistive plate には混合ガス（$C_2H_2F_4$ 94.7 %、iso-C_4H_{10} 5 %、SF_6 0.3 %）によるガスギャップが存在し、2 次元（$\eta \times \phi$）のストリップ読み出しをする。

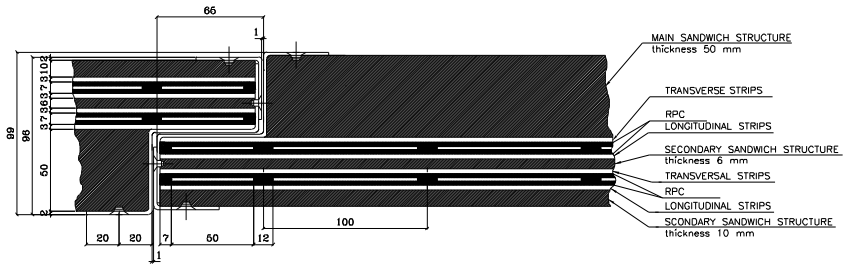

図 1.15: RPC の構造 [9]、バレル領域に設置されたトリガー用検出器である。

1.2.5　超電導マグネットシステム

ATLAS では 3 つの超電導マグネットにより特徴的な磁場を発生させ、その磁場によって曲げられた荷電粒子の運動量を測定する。超電導マグネットシステムの構成を図 1.16 に示す。中央にソレノイドマグネットが設置され、側面を囲むようにオクタント対称のバレルトロイドマグネットが、底面を覆うようにオクタント対称のエンドキャップトロイドマグネットが配置されている。

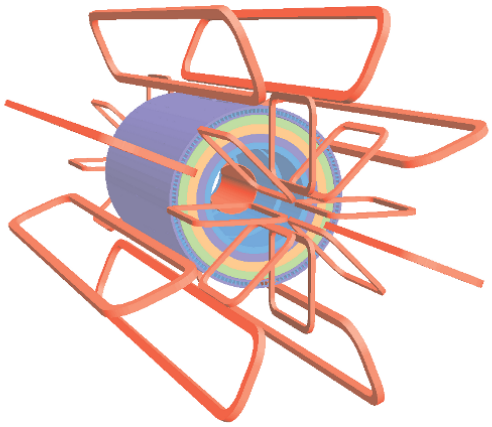

図 1.16: 超電導マグネットシステム [7]。トロイドマグネットはビーム軸方向から見てオクタント対称に配置されている。

ソレノイドマグネット

超電導ソレノイドマグネットは内部飛跡検出器とバレル領域の電磁カロリメータの間に置かれ、ビームパイプに沿った方向に 2 T のソレノイド磁場を作る。直径 2.4 m、長さ 5.3 m の円筒形マグネットで、内側の飛跡検出器に用いられる。

トロイドマグネット

トロイドマグネットはバレル領域とエンドキャップ領域にあり、ビーム軸方向から見て互いに 22.5° の間隔で配置されている。バレルトロイドマグネットは 8 つの超電導コイルにより ϕ 方向に約 0.5 T の磁場を発生させ、荷電粒子を η 方向に曲げる。バレルトロイドマグネットによって曲げられた荷電粒子は先述の RPC と MDT により捕捉される。また、エンドキャップトロイドマグネットもバレルと同様に ϕ 方向に約 1 T の磁場を発生させることで荷電粒子を曲げ、TGC、MDT、CSC で荷電粒子の情報を得る。

第2章　レベル1エンドキャップミューオントリガーシステム

レベル1ミューオントリガーシステムはバレル領域をカバーするRPCとエンドキャップ領域をカバーするTGCで構成され、ウィークボゾンの崩壊などによるhigh-p_Tミューオンを捕捉するために非常に重要である。TGCではトリガー判定の他に、ミューオンのr、ϕ方向の測定も行っている。本章では、はじめにATLASにおけるトリガーシステムの概要を述べ、次にTGCの概要、トリガースキーム、TGCで使用されるエレクトロニクスの順に説明していく。

2.1　ATLASにおけるトリガー及びデータ収集システム

LHCでは40.08 MHzの衝突頻度でビームを衝突させており、1度のバンチ衝突で数10個の陽子 – 陽子衝突が起こる。その陽子同士の衝突によるイベントレートは約1 GHzに上る。しかし、1イベントあたりの記録に要するデータ量が$\sim$ 1.5 MByteという見積もりに対して、記憶装置及び計算機資源の制約から300 MByte/s程度しか記録できない。1 GHzで起こるイベントは膨大なバックグラウンドを含んでいるため、いかに効率よく、かつ正確に興味のある物理現象を選出できるかが鍵となる。ATLAS実験ではこれを実現するため、トリガーを3段階に分けてイベントレートを絞っている。トリガーを順にレベル1トリガー、レベル2トリガー、イベントフィルター（EF）と呼び、最終的にイベントレートを700 Hz程度にまで落とす。レベル1トリガーは2.5 μsという厳しいレイテンシの条件をクリアするために高速なハードウェア処理が行われ、レベル2トリガー及びEFはより高精度なトリガーを発行するためにソフトウェアによる処理が行われる。本節では、各トリガー及びデータ収集システム（TDAQ：Trigger and Data Acquisition）について述べる。

2.1.1　レベル1トリガー

レベル1トリガーシステムはTGC、RPC、カロリメータの各検出器とCentral Trigger Processor（CTP）、Timing Trigger and Control distribution system（TTC）から構成される。TGC、RPCからはp_T情報が提供され、電磁及びハドロンカロリメータからは精度を落としたエネルギー情報が提供される。CTPにはMU（muon）、EM（electromagnetic）、J（jet）等のメニューが用意されており、そのメニューを参照してトリガーを発行するか否かを決定する。発行されたトリガーをLevel-1 Accept（L1A）信号と呼び、パイプラインメモリ（Level-1 Buffer）へ送られる。L1A信号は衝突が発生してから2.5 μs以内に発行されることになっており、レベル1では少なくとも100段構造のパイプラインメモリを用意し、25 ns $\times$ 100 = 2.5 μsの間だけ読み出しデータを保持できるようになっている。各検出器はL1A信号に対応したヒット情報をパイプラインメモリから読み出し、その後デランダマイザというバッファに格納する。デランダマイザはL1A信号の発行のばらつきを吸収して、後段のReadOut Driver（ROD）へEvent IDを付加したデータを送信する。

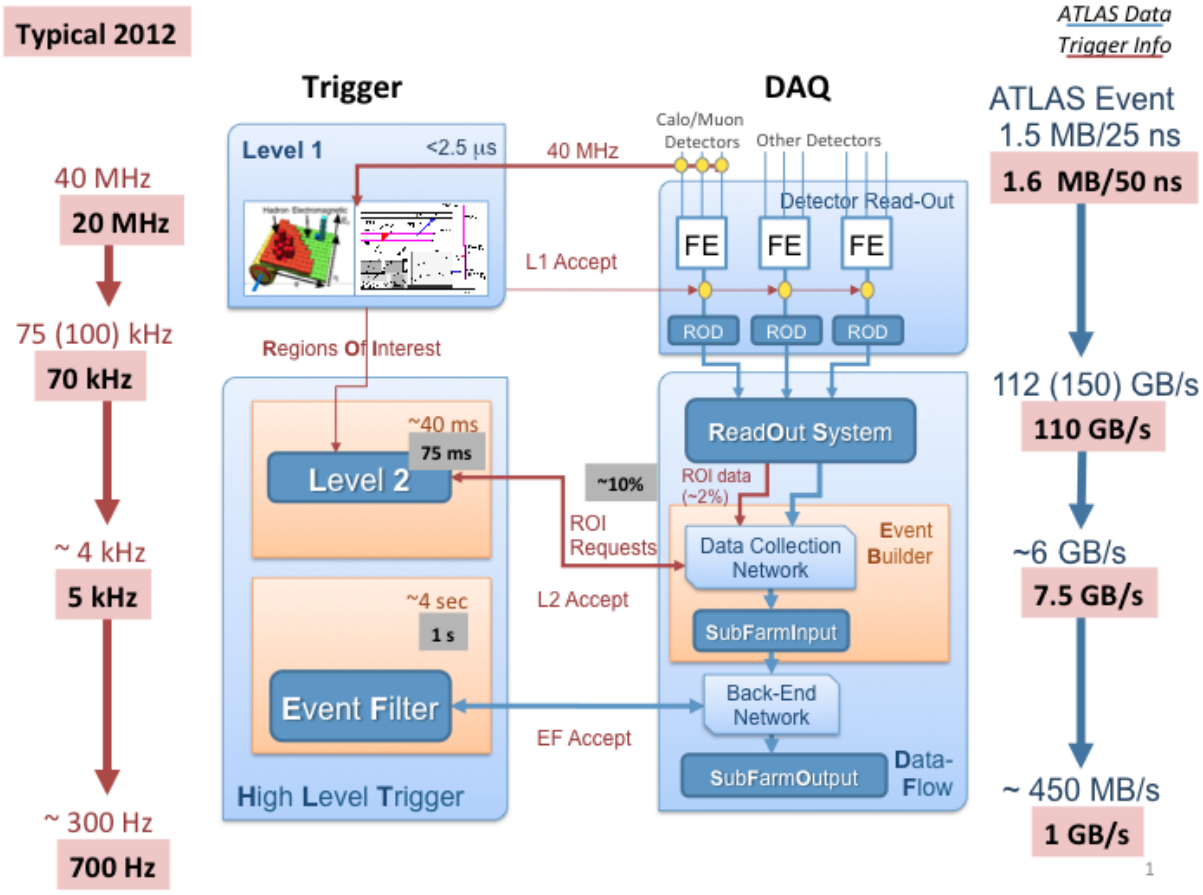

図 2.1: 2012 年におけるトリガー及び DAQ スキーム [7]。イベントレートは 20 MHz から最終的に 700 Hz まで絞られる。

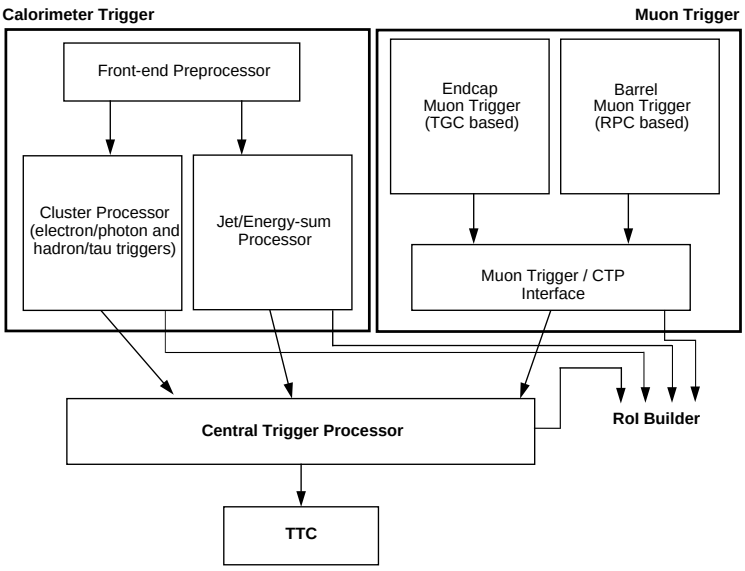

図 2.2: レベル 1 トリガーシステム [7]。TGC 及び RPC から発行されるミューオントリガーは MUCTPI（Muon Trigger / CTP Interface）を介して CTP に送られる。

TTC

TTC は各検出器のフロントエンドエレクトロニクスと同期をとり、トリガー信号を分配するために使用される。同期には Bunch Crossing（BC）クロック、トリガー信号には L1A、Event Counter Reset（ECR）、Bunch Counter Reset（BCR）等がある。L1A 信号が発行されたヒット情報には BCID[1]、L1ID[2] を付加して後段のシステムに送信する。

図 2.3 のように TTC は Local Trigger Processor（LTP）、TTCvi、TTCvx、ROD busy で構成される。LTP は各検出器ごとに配置され、CTP とタイミング・トリガー情報及び busy 信号を送受信する。TTCvi はクロック以外の情報を LTP から受け取り、トリガーを A-channel、その他の信号を B-channel に分離して TTCvx に送信する。TTCvx はこの 2 種類の信号を多重化して光信号により ROD 上の TTCrx 及び各検出器のフロントエンドエレクトロニクスに分配する。ROD busy は ROD のデータ処理が追いつかない場合に発行される信号で、LTP を介して CTP に送信される。CTP は ROD busy 信号を受け取り次第、L1A 信号の発行を停止する。

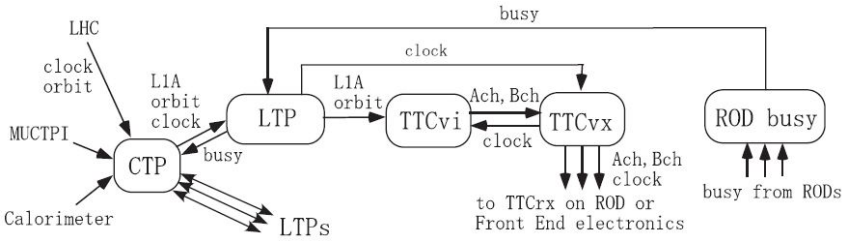

図 2.3: TTC システム [31]。ROD は TTCvx から TTC 情報を受け取り、処理が追い付かないときは LTP へ busy 信号を出力する。

2.1.2 レベル 2 トリガー及び EF

レベル 2 トリガー及び EF はソフトウェア用いて、レベル 1 トリガーよりも高精度なトリガー処理を行う。

レベル 2 トリガーは MDT、CSC、内部飛跡検出器による精密なトラック情報とカロリメータの高精度な情報を用いてトリガー判定を行う。トリガー判定に許されるレイテンシは 40 ms と短いため、粒子の検出された領域とその周辺だけの情報を使う。この領域を Region of Interest（RoI）と呼び、RoI 情報はレベル 1 トリガー情報とともに Level-2 SuperVisor（L2SV）を介して Level-2 Processor（L2P）に割り振られる。L2P は割り振られた情報と ReadOut System（ROS）からのイベント情報を統合して Level-2 判定を行う。判定により発行されるトリガーを Level-2 Accept（L2A）信号と呼び、L2SV を経由して Data Flow Manager（DFM）に送り、さらに DFM は Sub Farm Input buffer（SFI）へデータを渡す。SFI では全検出器のデータを統合する。

EF では最終的なトリガー判定を行い、イベントレートを ∼ 700 Hz まで絞り込む。レベル 2 では RoI に限定してトリガー判定を下していたが、EF では全ての領域における情報を用いてトリガー判定を行っており、約 1600 台の PC を用いた大規模システムの並列処理により膨大なデータを捌く。EF Accept 信号が発行されたヒットデータは Sub Farm Output（SFO）バッファに渡され、最終的にディスクへの書き込みが行われる。

[1]バンチに関する ID
[2]L1A に関する ID

2.2 TGC の概要

本節では TGC の概要について説明するが、その前に TGC のセクターについて簡単に述べる。TGC はエンドキャップ領域に配置されたディスク状の検出器で、ϕ 方向に 12 分割したユニットをセクターと呼び、セクター単位でデータの処理を行う。図 2.5 のそれぞれの台形をトリガーセクター、それをさらに細かく分割したものをサブセクターと呼ぶ。図 2.4 に照らし合わせると、1 つのセクターにはフォワード領域のトリガーセクターが 2 つ（ϕ_0、ϕ_2）、エンドキャップ領域のトリガーセクターが 4 つ（$\phi_0 \sim \phi_3$）の計 6 つのトリガーセクターで構成されていることがわかる。また、サブセクターはトリガーの最小単位であり、ワイヤ・ストリップともに 8 チャンネルで成り立っている。エンドキャップ領域では 37（R 方向）×4（ϕ 方向）= 148 サブセクター、フォワード領域では 16（R 方向）×4（ϕ 方向）= 64 サブセクター となっている。

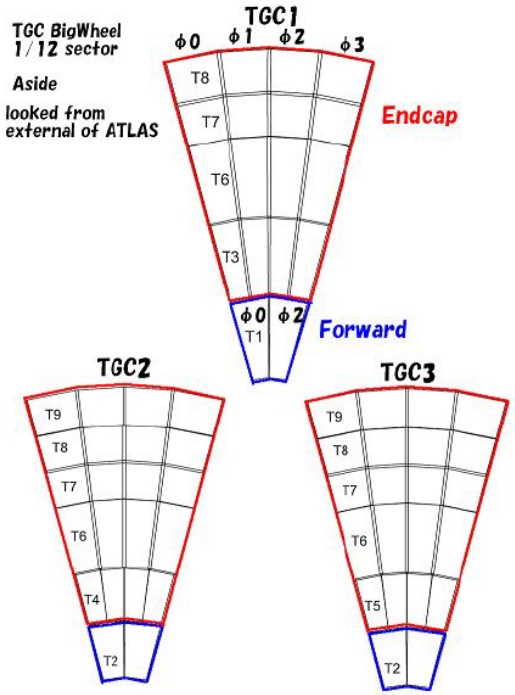

図 2.4: セクター [31]。1 枚のチェンバーは T1~T9 という区分で大きさが異なる。

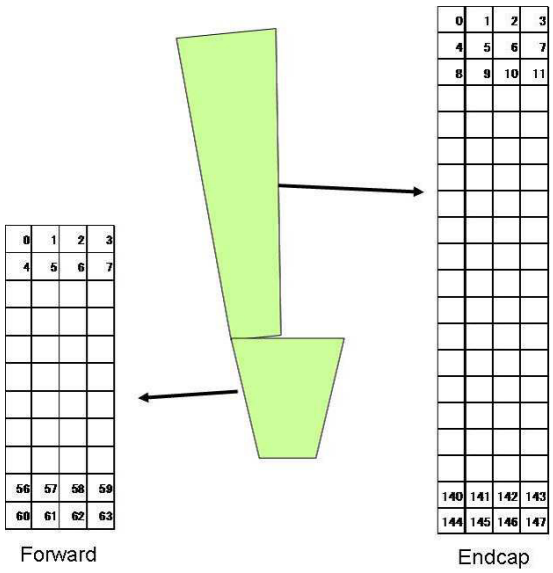

図 2.5: トリガーセクター及びサブセクター [31]。サブセクターに対応した領域を RoI（Region of Interest）と呼び、η 方向に 2 つ、ϕ 方向に 4 つの計 8 つの RoI を SSC（Sub Sector Cluster）と呼ぶ。

2.2.1 TGC の構造及び動作原理

TGC はアノードワイヤとカソードストリップの 2 次元読み出しをする MWPC である。図 2.6 に TGC の構造を、表 2.7 に TGC のパラメータを載せる。アノードワイヤ間隔 1.8 mm よりもアノードワイヤ – カソードカーボン面間隔 1.4 mm の方が短くなっている。ワイヤの間隔を狭くすることにより電子のドリフト時間を短くし、バンチの衝突間隔 25 ns 以内を実現している。チェンバー内には CO_2 を電離ガス、n-pentane をクエンチャーとした混合ガスが充填されている。ワイヤ 6~20 本を 1 つのチャンネルとして読み出し、1 つのストリップを 1 つのチャンネルとして読み出す。

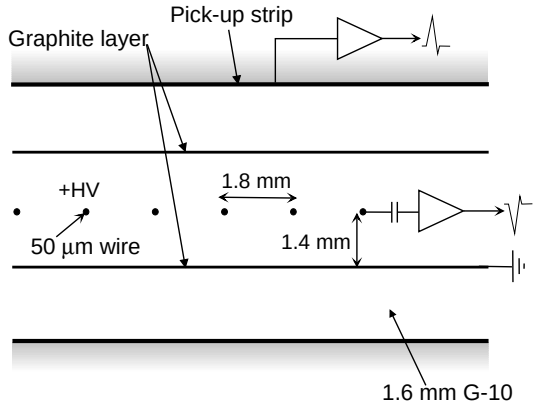

図 2.6: TGC の構造 [7]。ワイヤ・ストリップ間は 1.4 mm と短いため、25 ns 以内に応答する。

図 2.7: TGC のパラメータ [7]

パラメータ	設計値
ガスギャップ	2.8mm
アノードワイヤ間隔	1.8mm
アノードワイヤ半径	50μm
印加電圧	3100V
ワイヤ-ストリップ間隔	1.4mm
ガス成分	CO_2/n-pentane (55 : 45)

TGC には図 2.8 のようなガスギャップを 2 層持つ Doublet と 3 層持つ Triplet がある。Doublet は 2 層のワイヤ面と 2 層のストリップ面で読み出し行う。Triplet には 2 層目のストリップ面が存在せず、3 層のワイヤ面と 2 層のストリップ面で読み出しを行う。このような多層チェンバーを用いて各層のコインシデンスをとることによりバックグラウンドからの寄与を少なくしている。

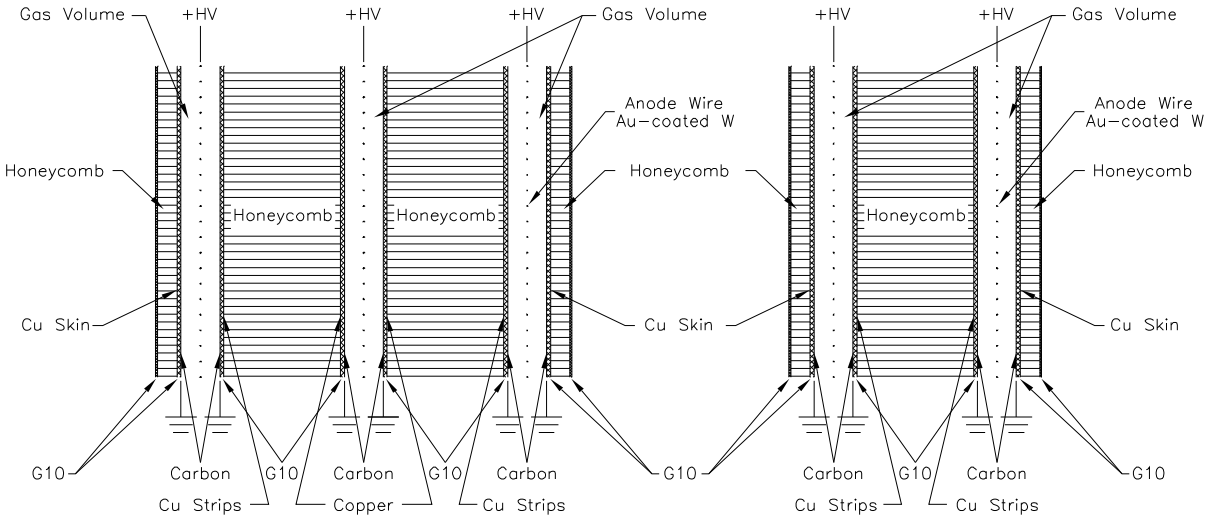

図 2.8: TGC の Triplet と Doublet の構造 [7]。各 TGC チェンバーのガスギャップ間をハニカムで支えている。

荷電粒子がチェンバーを通過すると、チェンバー内に封入された混合ガスが電離・イオン化されて、粒子の飛跡に沿った一次電子が生成される。一次電子はアノード – カソード間の印加電圧によってドリフトしながら加速され、アノードワイヤ近傍になると急激に加速される。これにより一次電子はガス分子の電離エネルギーを超えて二次電子を生成する。この繰り返しをタウンゼント電子なだれと呼び、TGC ではこれによる信号をアノードワイヤから読み出し、それと同時にストリップでは誘起した電荷を読み出す。また、荷電粒子はガス分子を電離させるだけでなく励起させることもあり、その際に紫外線を放射する。紫外線はガス分子から二次電子を生じさせ、そ

こからタウンゼント電子なだれが起きる。それを抑制するため、クエンチ効果[3]のある n-pentane が封入されている。

2.2.2 TGC の配置

TGC システムは M1（triplet）、M2（middle）、M3（pivot）と EI（Endcap Inner）、FI（Forward Inner）の 5 種類のディスク状チェンバーで構成される。M1 は Triplet チェンバー、M2、M3、EI、FI は Doublet チェンバーで、トリガー判定には M1、M2、M3 の合計 3 ステーションのコインシデンスを用いる。また、エンドキャップトロイドマグネットの内側に位置する EI、FI を Inner Wheel、エンドキャップ領域の最外層に位置する M1、M2、M3 と MDT を合わせて Big Wheel（BW）と呼ぶ。

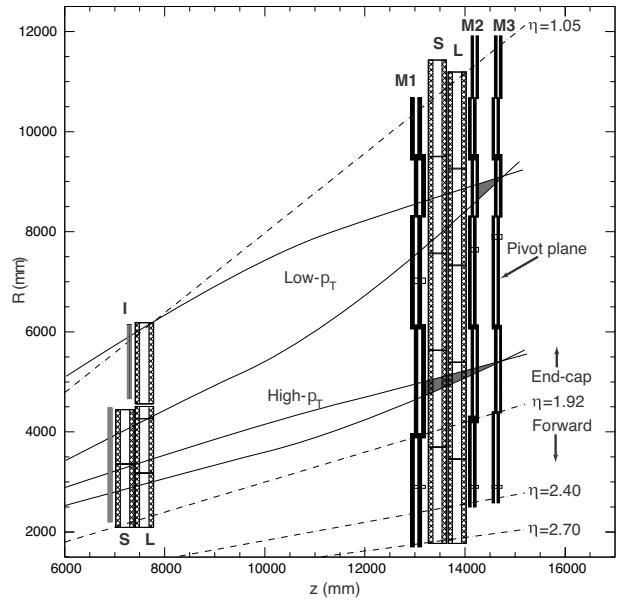

図 2.9: TGC のレイアウト [7]。IP から見て $z \sim 13$ m に M1、$z \sim 14$ m に M2、$z \sim 14.5$ m に M3 が配置されており、M1 と M2 の間には MDT が差し込まれている。

2.3 トリガースキーム

TGC ではトロイドマグネットによる磁場で曲げられたミューオンの飛跡のずれを測定して、トリガー判定を行う。本節ではトリガー判定に至るまでの手順を説明する。

2.3.1 p_T 測定の原理

p_T 測定の概念図 2.10 を以下に示す。TGC ではトロイドマグネットの磁場で曲げられたミューオンの飛跡を測定することで、ミューオンの p_T を算出してトリガー判定を行う。p_T 算出の流れは以下のようになっている。

[3]紫外線を吸収する効果。

1. 陽子同士の衝突により生成されたミューオンはエンドキャップトロイドマグネットの磁場で曲げられる。

2. 磁場を通過した後、ミューオンは M1、M2、M3 の順にヒットする。

3. M3 のヒット位置と衝突点を結んだ直線はミューオンが無限運動量を持った場合の飛跡と考えられる。この直線と実際の飛跡のずれ $(\Delta R, \Delta\phi)$ から p_T を算出する。

p_T が高いほど磁場で曲げられにくく、飛跡のずれ ΔR、$\Delta\phi$ は小さくなる。1.2.5 項でも触れたが、理想的には荷電粒子は ϕ 方向に曲がらないため $\Delta\phi = 0$ であるが、実際は $\Delta\phi$ も少なからず存在する。

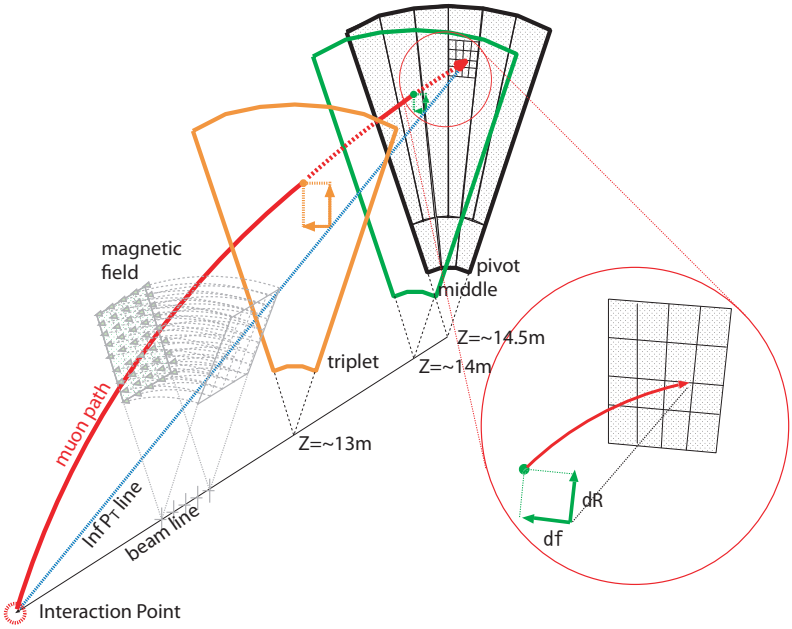

図 2.10: p_T 算出の概念図 [31]。IP と M3 の衝突点を結んだ直線と実際の飛跡のずれから p_T を算出する。

2.3.2 コインシデンス処理

TGC で発行されるトリガーは SLave Board（SLB） ASIC、High-p_T（HpT）、Sector Logic（SL）を経て段階的にコインシデンス処理が行われる。ここでは、各エレクトロニクスで行われるコインシデンス処理について説明するが、エレクトロニクス自体の説明は次節以降に譲る。

1. SLB ASIC
   はじめのコインシデンス処理は SLB ASIC で行われる。Doublet 及び Triplet 内でワイヤ・ストリップ（WD/SD、WT/ST）が独立に行われるため、4 種類の SLB ASIC が用意されている。WD/SD では 3/4 コインシデンス、WT では 2/3 コインシデンス、ST では 1/2 コインシデンスが取られ、さらに WD と SD ではコインシデンスマトリックスによって low-p_T 情報が生成される。

2. HpT
   次のコインシデンス処理では、Doublet と Triplet 間でワイヤ及びストリップについて HpT で処理される。コインシデンスマトリックスにより High-p_T 情報を生成し、p_T の高いものを選択して SL に送る。

3. SL
   最後のコインシデンス処理では、SL でワイヤ・ストリップ間の情報を統合する。ここでは、Coincidence Window（CW）と呼ばれる特別なコインシデンスマトリックスを用いて、最終的な p_T 情報を生成する。

レベル 1 ミューオントリガーではミューオンの p_T が 6 段階に分けられたトリガーメニューが用意されている（表 2.12）。CW は、SL に届いた飛跡のずれ $(\Delta R, \Delta\phi)$ をレベル 1 トリガーメニューに照らし合わせて図 2.11 のようにマップさせたもので、SL に Look Up Table（LUT）として実装されている。

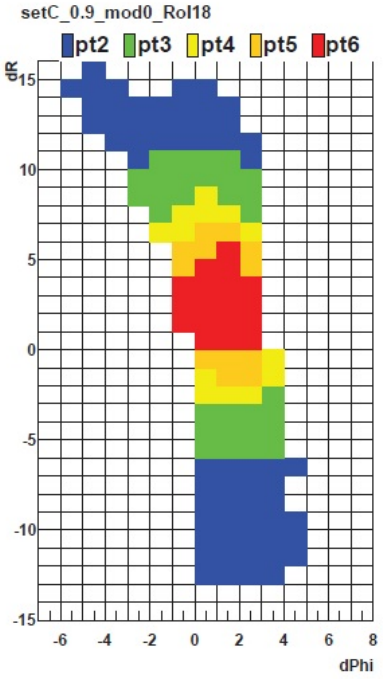

図 2.11: Coincidence Window の例 [30]

L1 menu	pt	閾値
L1_MU0	pt1	p_T 6 GeV未満
L1_MU6	pt2	p_T 6 GeV以上
L1_MU10	pt3	未使用
L1_MU11	pt4	p_T 11 GeV以上
L1_MU15	pt5	p_T 15 GeV以上
L1_MU20	pt6	p_T 20 GeV以上

図 2.12: レベル 1 トリガーメニューと p_T の対応関係 [34]

2.4 エレクトロニクス

本節では TGC で用いられるエレクトロニクスの概要について述べる。

2.4.1 システム全体のデータフロー

TGC エレクトロニクスのシステム全体像を図 2.13 に示す。TGC エレクトロニクスシステムは 3 種類のフローに分類され、それぞれトリガー系、リードアウト系、コントロール系と呼ぶ。ここ

では、各フローで用いられるエレクトロニクスの詳細な説明はせず、おおまかな流れを説明するだけに留める。

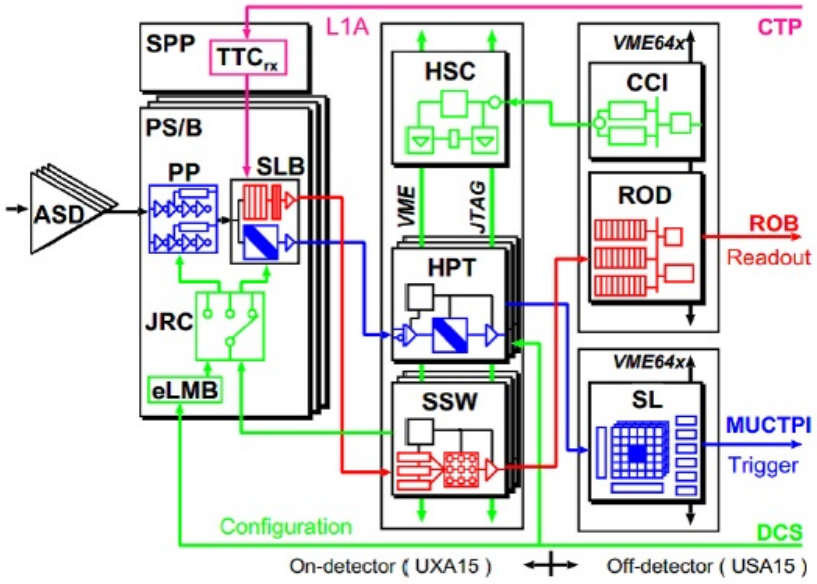

図 2.13: TGC エレクトロニクスの全体像 [34]。青線がトリガー系、赤線がリードアウト系、緑線がコントロール系のフローを表す。

トリガー系

トリガー系（図 2.14）はレベル 1 ミューオントリガーの判定を行うフローで、ここで生成されたトリガーは最終的に MUCTPI に送信される。

- Amplifier Shaper Descriminator（ASD）
  ASD は TGC で観測したアナログ信号を増幅、整形、デジタイズする。このデジタル信号がある基準の電圧を超えた場合、Low Voltage Defferential Signal（LVDS）によって後段の Patch panel and Slave board ASIC Board（PS Board）に送られる。PS Board は 1/2 セクターごとに設置されており、17 個の Doublet 用 PS Board、10 個の Triplet 用の PS Board が繋がれている。
- Patch Panel（PP） ASIC
  PS Board に組み込まれた ASIC で、TOF[4] やケーブル長の違いによるシグナルの遅延を吸収し、TTC から提供される BC クロックと同期をとり、バンチ識別を行う。
- SLB ASIC：トリガー
  SLB ASIC も PP ASIC と同様に PS Board 上に組み込まれている。M1 の 3 層によるコインシデンス処理、M2 と M3 の 4 層によるコインシデンス処理を独立に行い、各判定結果を TGC の外縁クレートに収納されている HpT へ LVDS で送信する。
- HpT
  HpT では各 SLB ASIC から送られてくるコインシデンスの結果を統合して、それを光ケーブル経由で USA15 というコントロールルームに送信する。

[4]Time Of Flight の略、粒子の飛来する時間。

- SL
  USA15 には SL が設置されており、ワイヤ – ストリップのコインシデンス処理、飛跡のずれ ΔR、$\Delta\phi$ からの運動量判定を行い、RoI 情報と運動量情報を含んだトリガーを MUCTPI へ知らせる。

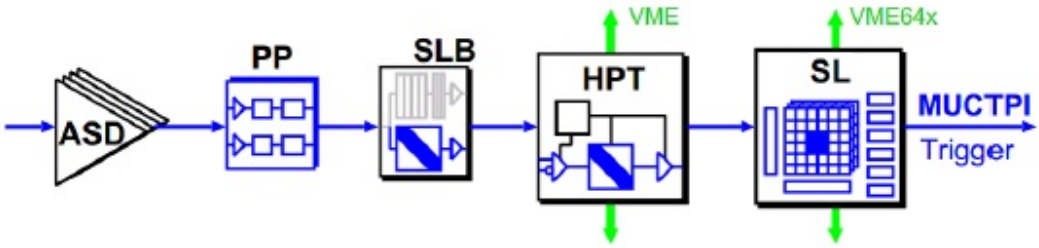

図 2.14: TGC のトリガー系 [34]。ASD 及び PP で生成されたデータは SLB、HpT、SL を経て段階的にコインシデンス処理が行われる。

リードアウト系

リードアウト系（図 2.15）で読み出されるデータは ROD へ集められる。ASD、PP の動作はトリガー系と同様なので、ここでは説明を省略する。

- SLB ASIC：リードアウト
  PP ASIC で調整されたデータは SLB ASIC 内の Level-1 Buffer に格納され、トリガーが発行されるまで保持される。CTP から出力された L1A 信号は Service Patch Panel（SPP）上の TTCrx で受信し、SLB ASIC にトリガーが発行されたことを伝える。SLB ASIC は L1A 信号に対応するヒットデータとその前後を合わせた 3 バンチ分のデータをデランダマイザへ蓄える。デランダマイザではトリガーの発行頻度のばらつきを吸収して、後段の Star SWitch（SSW）へ送信する。
- SSW
  SSW は複数の SLB ASIC からデータを収集し、データを圧縮した後に光ケーブルを通して後段の ROD へ送信する。
- ROD
  ROD は複数の SSW からのデータを収集し、各データに付属の ID と TTC から送られてくる情報との整合性をチェックして、ROS が束ねる ReadOut Buffer（ROB）にデータを送信する。ROD はサブシステムのリードアウトラインの最終ポイントに位置し、それ以降に位置する ROS はレベル 2 の橋渡しをする。
- ROS 及び ROB
  レベル 1 ではないが、ROD の後段モジュールということで合わせて説明する。ROS は特定のサブシステム（例えば、TGC）の ROD からのデータを受け取り、レベル 2 トリガーからデータを要求されるまで保存する。ROS 自体は複数のギガビットイーサネット NIC（Network Interface Card）がインストールされたマルチポートな PC で、ROD からのデータを受け取るためのバッファである ROB が 1 台の ROS につき 4 つ組み込まれている。ROD と ROB の間は CERN が開発した S-Link[26] という規格でデータ通信を行っており、ROB は S-Link 入力を 3 つまで受け付ける。

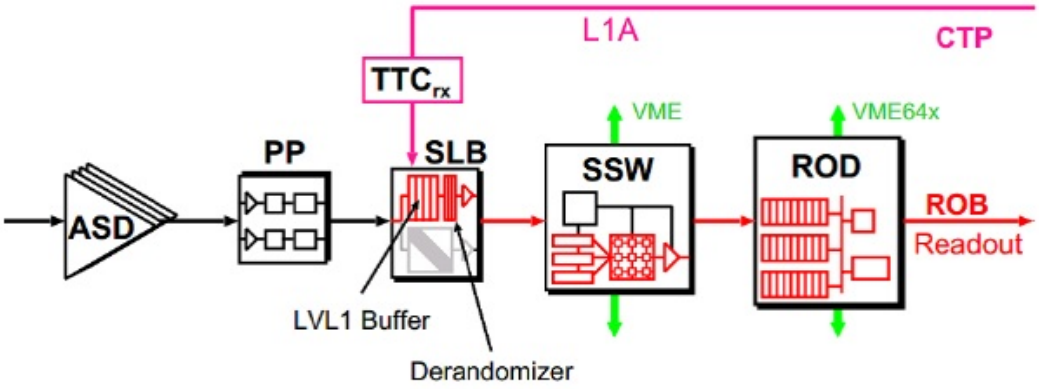

図 2.15: TGC のリードアウト系 [34]。L1A 信号が発行されたデータは SSW、ROD の順に収集される。

コントロール系

コントロール系を図 2.16 に示す。USA15 に設置されている VME モジュールと TGC の外縁クレートに置かれている VME モジュールは制御の方法が異なる。USA15 の VME モジュールは PC で直接制御し、TGC の外縁クレートの VME モジュールは High-p_T SSW Controller（HSC）経由で Crate Control Interface（CCI）の遠隔操作によって制御する。HSC も TGC の外縁クレートに配置されており、クレート上の VME マスターとして働いている。また、CCI は USA15 から光ケーブルを使って HSC を操作する。TGC に直接設置されている PS Board は 2 種類の制御経路があり、1 つは JTAG プロトコルを使って SSW 経由で制御する経路、もう 1 つは Detector Control System（DCS）から制御する経路である。DCS は PS Board の他に TGC の外縁クレート上のモジュールについても制御する役割を担う。

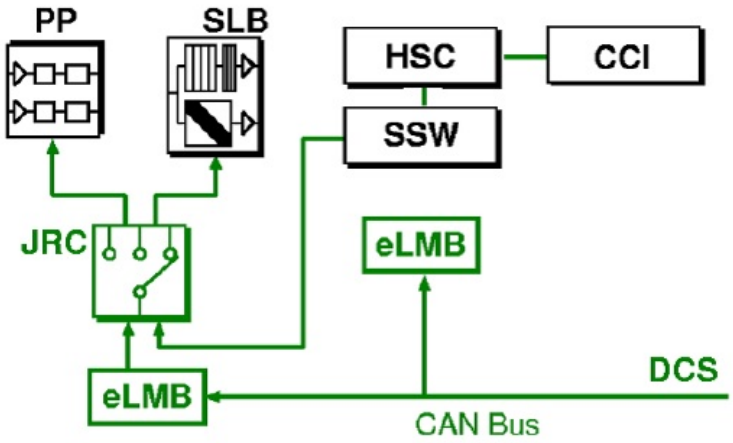

図 2.16: TGC のコントロール系 [34]。クレートの場所によって制御方法が異なる。

2.4.2 リードアウト系のエレクトロニクス

本節ではリードアウトラインで使用されるエレクトロニクスに関して説明する。ただし、本研究の対象である ROD については次節以降に説明する。

ASD Board

ASD Board は TGC の側面に設置され、4 チャンネル分の信号を処理する ASD ASIC を 4 つ搭載している。つまり、1 つのボードあたりで 16 チャンネル分の信号を処理する。ASD の主な役割は 2 つあり、1 つは TGC からのアナログ信号を増幅、整形、デジタイズして閾値電圧を超えた信号を出力することで、もう 1 つはトリガー信号を受けてテストパルスを出力することである。ASD Board は PS Board とツイストケーブルで接続されており、電源、閾値電圧、テストパルス用のトリガーは PS Board から供給される。

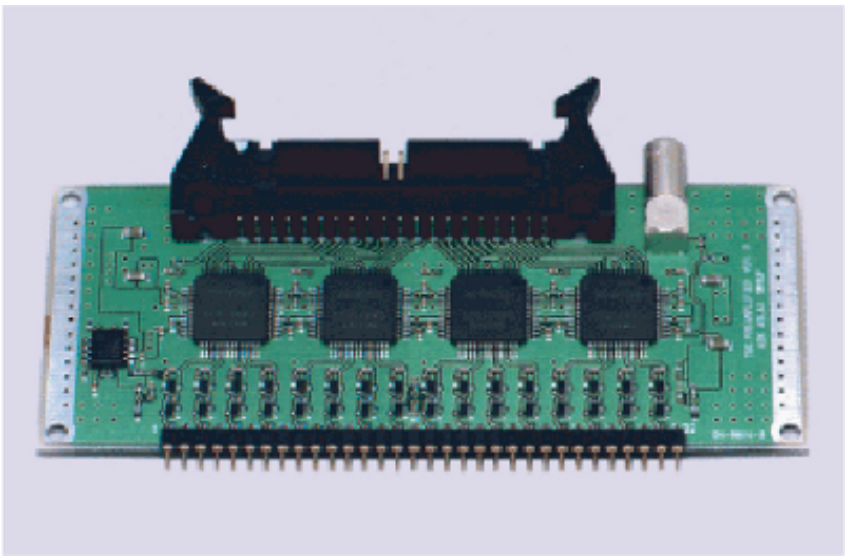

図 2.17: ASD Board の写真 [20]

PP

PP は PS Board に組み込まれた ASIC で、ASD Board から送られる信号のタイミング調整及びバンチ識別を行っている。PP に届く信号は TOF やケーブル長の差により到達時間がばらつくが、各信号にディレイをかけることでタイミング調整を行っている。このディレイは図 2.19 の variable delay 回路によって実現され、$25/32 = 0.78$ ns 単位として 0 ～ 24.2 ns の範囲で調整することができる。ディレイによって調整された信号は、バンチ識別が行われる。具体的には TTC から提供される LHC クロックと同期をとり、各バンチクロッシングで生じるデータを同じクロックにそろえる。また、PP にはテストパルス用のトリガーを発行するテストパルス回路が搭載されている。1 つの PP につき 32 チャンネルを処理することができ、2 つの ASIC Board を処理することに相当する。

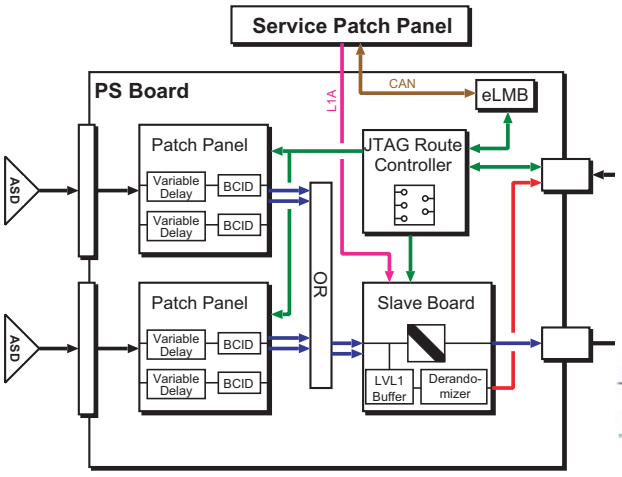

図 2.18: PS Board のブロック図 [34]

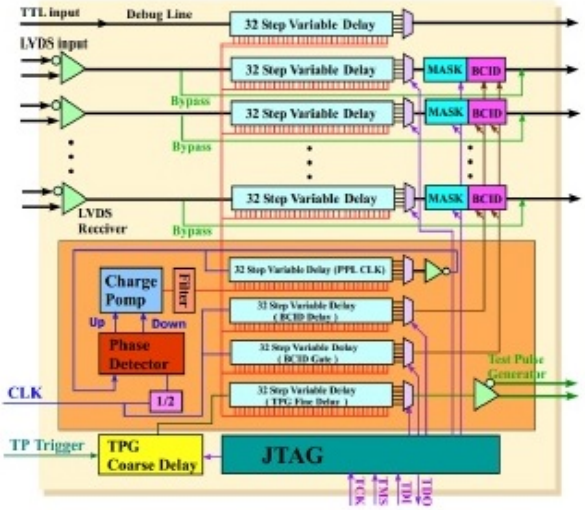

図 2.19: PP ASIC のブロック図 [21]

SLB ASIC

SLB ASIC はトリガー部及びリードアウト部で構成され、トリガー部で処理された情報は HpT へ、リードアウト部で処理されたデータは SSW へ送られる。

- トリガー部は 5 種類のコインシデンスマトリックス（ワイヤ-Doublet、ワイヤ-Triplet、ストリップ-Doublet、ストリップ-Triplet、EI/FI）を用いてデータの処理を行う。M1 には 2/3（ストリップは 1/2）、M2 と M3 には 3/4 のコインシデンス条件が課される。また、その他に PP ASIC からの信号に対して 1/2 クロック単位でディレイをかける機能、チャンネルをマスクする機能、連続したチャンネルからの入力信号を 1 つの出力信号として絞り込むデクラスタリング機能、SLB ASIC 以降のエレクトロニクスの動作確認やタイミング調整用のテストパルスを出力する機能といった、多彩な機能を持ち合わせている。
- リードアウト部は主に Level-1 Buffer とデランダイマイザで構成され、L1A 信号に対応したヒットデータの読み出しを行う。Level-1 Buffer は幅 212 bit、深さ 128 段のシフトレジスタで、25 ns × 128 段 = 3.2 μs だけデータを保持できる。Level-1 Buffer に格納されるデータは 160 bit の入力データ、40 bit のトリガーデータ、12 bit のバンチカウンタという内訳となっている。CTP により L1A 信号が発行された場合、L1A 信号に対応するデータとその前後合わせた 3 バンチ分のデータに、4 bit のイベントカウンタが付加されてデランダマイザに移される。その後、3 バンチ分のデータはシリアル変換され、4 bit 幅の LVDS で SSW に送信される。

SSW

SSW は複数の SLB ASIC のリードアウト部からデータを収集し、データを圧縮して ROD への負担を減らす。リードアウト用の SLB ASIC は PS Board 及び SL 上に設置されており、それに応じて SSW も図 2.20 のように 2ヶ所のクレートに収められている。PS Board からの読み出しは TGC の外縁クレートに収められている SSW で行い、SL からの読み出しは USA15 にあるクレートに収められている SSW で行う。

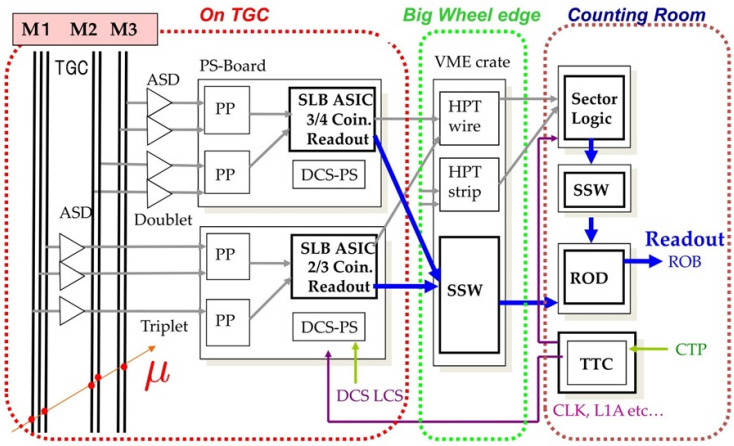

図 2.20: TGC のリードアウトライン [30]。SSW は TGC の外縁クレート及び USA15 にある専用クレートで SLB ASIC から読み出しを行う。

SSW は ROD へデータを送信する前に、ゼロサプレスという圧縮方法でデータ量の削減を図る。ゼロサプレスの概念を図 2.21 に載せる。ゼロサプレスは、データを 8 bit 単位のセルと呼ばれる塊に分割して、各セルに順次アドレスを付加し、ゼロのセルを消去する圧縮方法である。もちろん、この圧縮方法はデータにゼロが多く含まれている程効果的であり、効率よくデータを減らして ROD への負担を軽減している。

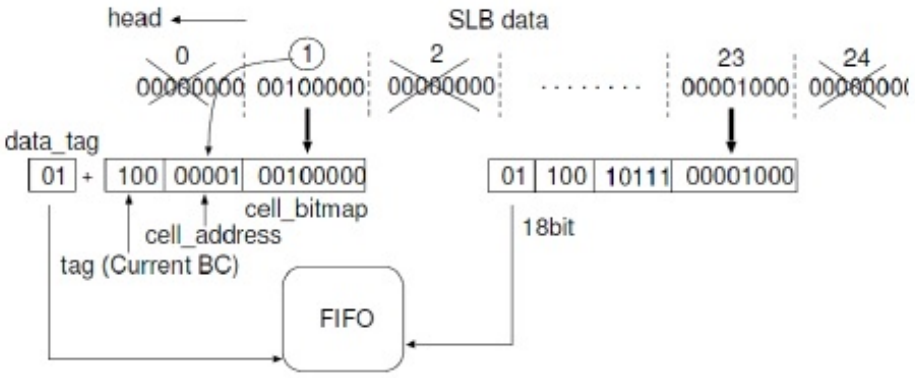

図 2.21: ゼロサプレスの概念 [24]。8 bit 単位のセルに分けて、0 のセルは消去していく。

SSW ボードを図 2.22 に載せる。SSW 上には 3 種類の FPGA が載せられており、それぞれ受信用（RX FPGA）が 6 つ、送信用（TX FPGA）1 つ、制御用（Control FPGA）1 つといった内訳になっている。RX FPGA は SLB ASIC から送られるシリアルデータを受信し、そのデータをパラレルへ変換する。その後、パラレルデータは RX FPGA 内部でゼロサプレスが行われ、圧縮後に TX FPGA へ渡す。TX FPGA では圧縮データを SSW 出力フォーマットに変換し、最終的に G-Link[19] 経由で ROD にデータを送信する。

∠
∠

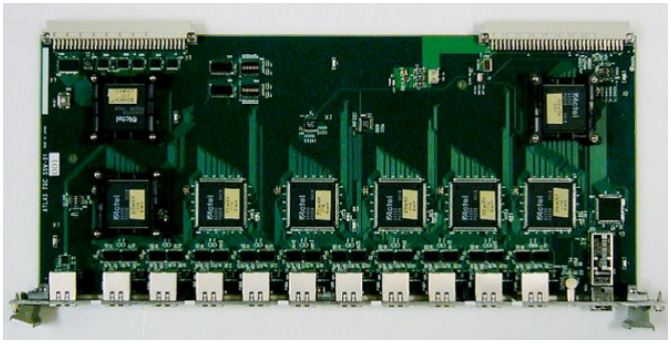

図 2.22: SSW ボードの写真 [24]。送受信用、制御用の FPGA が計 8 つ搭載されている。

2.5 Run-3 のリードアウト系でアップグレードするエレクトロニクス

Run-3 におけるリードアウト系では SL と ROD をアップグレードする予定である。ここでは、現行の SL 及び ROD について説明し、アップグレードした新しい SL 及び ROD については次章の冒頭で述べる。

2.5.1 Sector Logic

SL は TGC トリガーシステムのトリガー情報が最終的に集まるモジュールで、2 トリガーセクターのデータを処理する。SL にはエンドキャップ（EC）及びフォワード（FW）の 2 種類が存在し、2 枚の EC-SL と 1 枚の FW-SL が 1 つのセクターのデータ処理を行う。HpT から届けられる r 方向（ワイヤ）と ϕ 方向（ストリップ）の情報を統合した $r-\phi$ コインシデンス情報を生成し、ミューオンのトラック情報を再構成する。再構成されたトラックは 6 段階の p_T 閾値で分類され、p_T が高い 2 つを選別する。最終的に、2 つのミューオンの p_T 情報と RoI 情報を MUCTPI へ渡す。

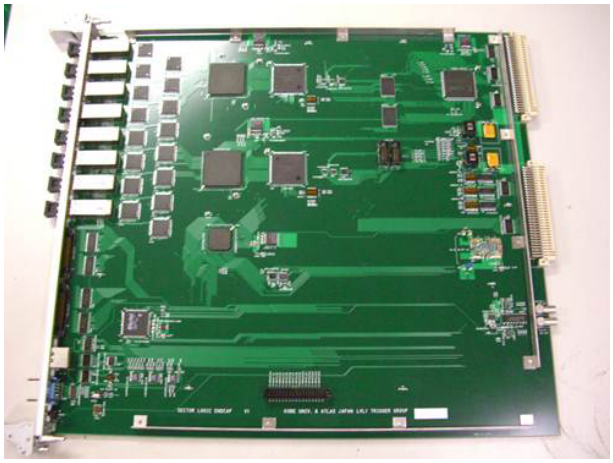

図 2.23: SL の写真 [38]

2.5.2 ReadOut Driver

本節では、本研究の対象である ROD について説明する。ROD は各サブシステムの最終的なデータ集積ポイントであり、TGC では SSW からデータを収集する（図 2.20）。ROD は各サイドに 12 台ずつ設置され、1 台の ROD で 1/12 セクターの読み出しを行っている。ROD の主な役割は以下の項目になっている。

- 複数の SSW からのデータ収集を行う。
- 受信したデータに格納されている ID のチェックする。
- 収集したデータフラグメントに対して ROB 入力フォーマット（図 2.26）へ変換する。
- 規定のフォーマットに変換した後に、S-Link という規格に則って ROS へ送信する。
- ROD 内のデータ処理が追いつかなくなった場合に busy 信号を発行する。

最後の項目の busy 信号は 2.1.1 項で述べたように LTP を介して CTP へ通知され、CTP はその通知を受けて L1A 信号の発行を中断する。L1A 信号が止められている間、ROD はフルになったバッファ内のデータを全て処理して空にすることで、データの取りこぼしを防いでいる。

エレクトロニクス

ROD は Xilinx 社製の Virtex II FPGA を搭載した 6U サイズの VME モジュールで、USA15 の VME クレートに設置されている。モジュールの側面には 4 枚のメザニンカードを挿すことができ、このうち 2 つは G-Link コネクタを載せたメザニンカードで SSW からの読み出しを行っている。1 枚のメザニンカードには G-Link コネクタが 6 個接続されており、1 つの ROD で最大 12 台の SSW から収集でき、稼働中は最大 10 個のコネクタが使用されていた。また、残りの 2 つのメザニンカードうち 1 つは S-Link コネクタを載せたデータの送信、もう 1 つは TTC シグナルの受信に使用されている。

図 2.24: TGC における ROD の写真 [25]。イスラエルの研究グループが制作したモジュールで、Run-2 以降も使用される予定である。

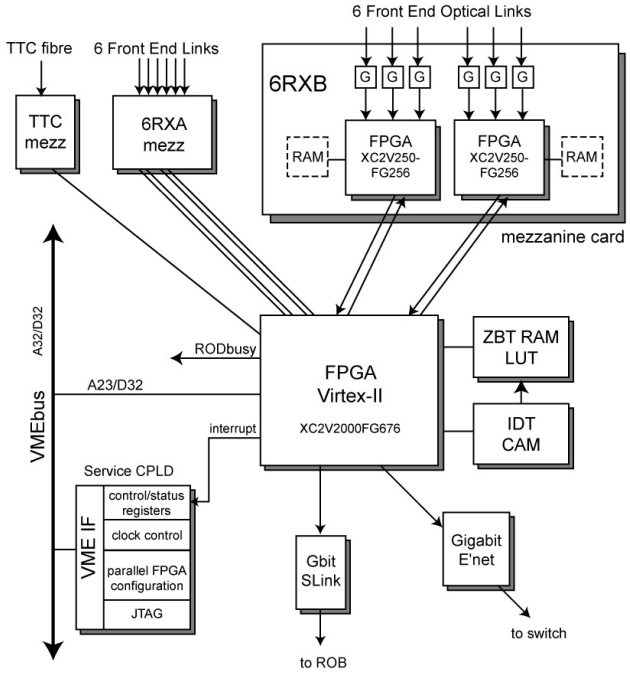

図 2.25: ROD のブロック図 [25]。受信用メザニンカード 2 枚で最大 12 台の SSW からデータを収集することができる。

出力データフォーマット

ATLAS 検出器の各サブシステムで取得したイベントデータフラグメントは、最終的に 1 つのイベントデータへまとめられる。図 2.27 のように、イベントデータフラグメントは各収集ポイントで段階的にヘッダ及びトレイラが付加される。ROD、ROB、ROS、Sub-detector、Event の順でデータはフォーマッティングされ、ROD から ROB へのフォーマットは図 2.26 のようになっている。ROB 入力フォーマットはヘッダ、ステータス、データ、トレイラの 4 ブロックで構成され、ステータスブロック及びデータブロックは各サブシステムで異なる。ROB 入力フォーマットは 4 Byte = 1 word を基本単位として成り立っており、フォーマット全体で 19 word = 76 Byte となる。

ここで TGC における ROB 入力フォーマット（図 2.28）と照らし合わせながら、サブシステムに共通の ROB 入力フォーマットのヘッダブロック及びトレイラブロックの各要素について説明する。

- ヘッダブロック
  - **Start of header marker**
    データの先頭を示す word で、ROD の場合は 0xee1234ee で表される。
  - **Header size**
    ヘッダサイズは Byte 数ではなく word 数が格納され、TGC ROD の場合はヘッダサイズが 9 word となる。
  - **Format version number**
    上位 16 bit に ATLAS バージョン、下位 16 bit に TGC バージョンという構成をとり、各領域には 0x0300 が格納される。

- **Source ID**
  sub-detector ID と Module ID の各 8 bit で構成され、残り 16 bit は reserve 領域で 0x0 が格納されている。TGC の場合、sub-detector ID には A-side を示す 0x67 及び C-side を示す 0x68 の 2 通りが用意され、さらに Module ID はデータがどのセクター由来かを示す。
- **Run number**
  Run number は実験回数に依存する数値である。この領域には上位 8 bit に Run type、下位 24 bit に Run number が格納される。
- **その他の要素**
  Extended Level-1 ID と Bunch crossing ID は TTC から供給される ID を格納する。Extended Level-1 ID は 24 bit の L1ID と 8 bit の ECR ID で構成され、Bunch crossing ID には 12 bit の BCID がそのまま入れられる。また、Level-1 Trigger type 及び Detector event type にはそれぞれその名のタイプが格納されるが、Detector event type は現在使用されていない。

• トレイラブロック

- **Number of status element**
  ステータスブロックの構成要素の数を格納する。TGC の場合は、ステータスブロックに 5 word 分の要素があるため、この領域には ”5 ” が入れられる。
- **Number of data element**
  データの word 数を格納する領域である。
- **Status block position**
  各サブシステムによってステータスブロック及びデータブロックの位置関係が異なる。TGC の場合はステータスブロックがデータブロックより先に配置されるため、ブロックポジションは ”0 ” となる。

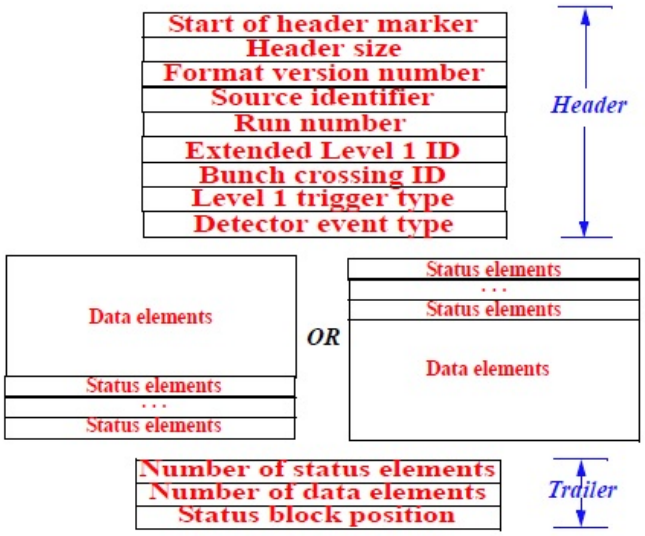

図 2.26: ROB 入力フォーマット [25]。ROD で収集されたイベントデータフラグメントはこのフォーマットに変換された後に ROB へ送信される。

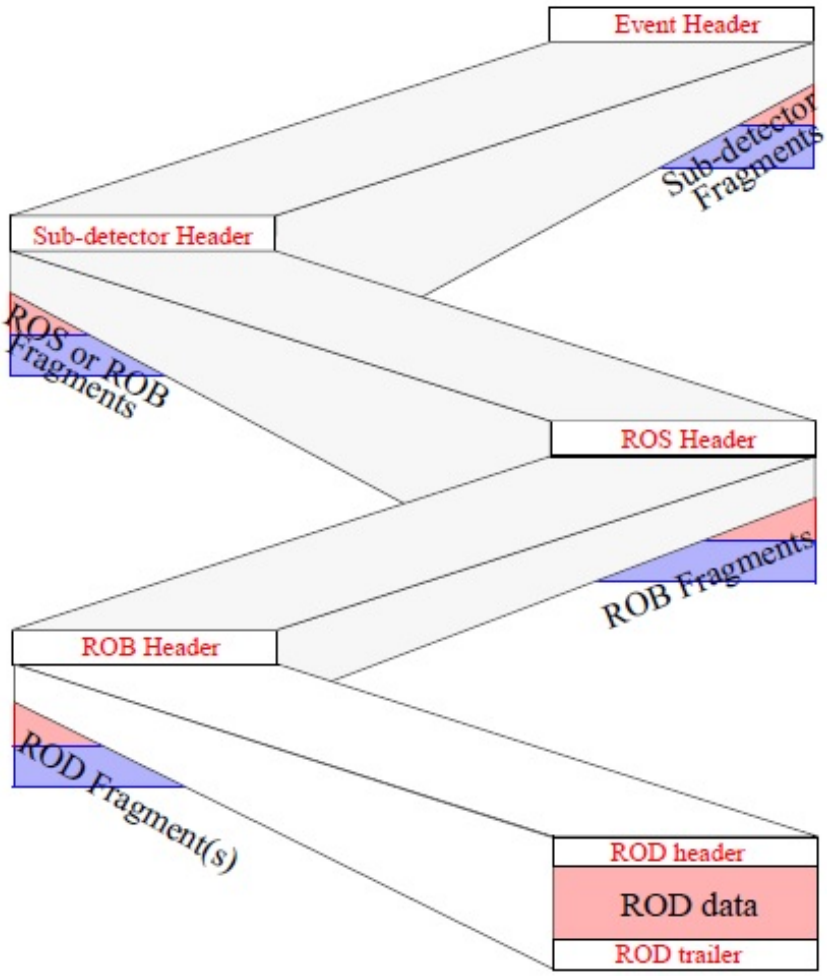

図 2.27: フォーマッティングの過程 [25]。ROD、ROB、ROS、Sub-detector、Event の順でデータフラグメントが統合されていく。

	Data word				Comments
	31..24	23..16	15..8	7..0	
Frame	x'B0F0xxxx'				event frame word (control mode word)
Hdr0	x'EE1234EE'				start of header marker for ROD data
Hdr1	*reserved*	*reserved*	header size = 9		words (excluding the x'B0F0xxxx' word)
Hdr2	ATLAS format version=3.0		TGC format version=3.0		i.e.: ATLAS=0x03'00, TGC=0x03'00
Hdr3	0	x'67' or x'68'	0	sector[12..1]	source id: x'67' / x'68' = A / C endcap;
Hdr4	Run type	Run number			
Hdr5	Level-1 ID				High byte is Extended Level-1 ID
Hdr6	*reserved*	*reserved*	Bunch crossing ID[11..0]		
Hdr7	*reserved*	*reserved*	*reserved*	Trigger type	
Hdr8	Detector event type				not used yet
Status	First status word: specific | generic				≠0: event is **not** OK. See Table 2, & [ref. 1]
Status	TGC ROD event status				See Table 3.
Status	ROD VME filter bits		Star Switch timeout or dropped status		one bit per SSW; Filter:1 = accepted. SSW: 1 = dropped or timed-out (see Table 4)
Status	Local status word		presence		Presence indicates which of the following fragments are present[a]. See Tables 5 & 6.
Status	orbit count				orbit count; zero for first L1AID.[b]
Data	Fragment ID	"raw" data word count[c]			fragment ID =1, length in words
Data	Fragment ID	"readout format" hit data word count			fragment ID =2, length in words[d]
Data	Fragment ID	"readout format" tracklet data word count ("tracklet"= 3/4 or 2/3 coincidence)			fragment ID =3, length in words
Data	Fragment ID	"chamber format" hit data word count			fragment ID =4, length in words
Data	Fragment ID	"chamber format" tracklet data word count			fragment ID =5, length in words
Data	Fragment ID	HipT output word count			fragment ID =8, length in words
Data	Fragment ID	Sector Logic word count			fragment ID =9, length in words
Data	raw data, hit, tracklet, sector logic, etc. fragments, in the order of the word counts.				See [ref. 6] and[ref. 8](raw) and Tables 7 to 10.
Data	...				
Data	last raw data, hit or tracklet word				
Trail0	number of status elements = 5				
Trail1	number of data elements				
Trail2	Status block position = 0, i.e. data follows status				
Frame	x'E0F0xxxx'				event frame word (control mode word)

図 2.28: TGC における ROB 入力フォーマット [25]。ステータスブロックには収集されたデータフラグメントの状態が記述され、エラーが生じた場合はエラー内容に応じた値を格納する。

2.6 Run-3 に向けて

LHC のアップグレード計画で述べたように、LHC は HL-LHC に向けて段階的な高輝度化が進むため、25 ns ごとに生じるバンチ衝突あたりのイベントレートが上昇する。LHC のアップグレードに対して、現行の検出器及びトリガー、データ収集システムでは負担が大きく、パイルアップの増加による測定精度の悪化や、検出器の放射線損傷の進行が早まる問題などがある。この現状を受け、ATLAS 検出器も適宜アップグレードする必要があり、私が所属しているミューオングループもトリガーシステムのアップグレードを行っている。LHC の Long Shutdown に応じて、ATLAS のアップグレード期間を phase-0、phase-1、phase-2 と呼び、本項では phase-1 について説明する。

2.6.1 ミューオントリガーシステムの問題点

現段階のミューオントリガーシステムの主な問題点として、以下の 2 点が挙げられる。

- **Small Wheel のカウントレートの限界**
  図 1.12 の $z \sim 7$ m にあるエンドキャップ領域の円盤状ステーションを Small Wheel（SW）と呼ぶ。イベントレートの上昇によるバックグラウンドを削減するために、SW を用いたトリガー判定の導入が考えられているが、ルミノシティの増加によって現行機（MDT+CSC）の検出効率が限界を迎えると予測されている（図 2.29）。現行機ではルミノシティ $\sim 1 \times 10^{34}$ cm^{-2}s^{-1} 程度で検出効率の限界を迎え、ルミノシティが $\sim 2 \times 10^{34}$ cm^{-2}s^{-1} へアップグレードされる Run-3 を視野に入れると、もう 1 桁上のオーダーまで耐えられる検出器が必要である。

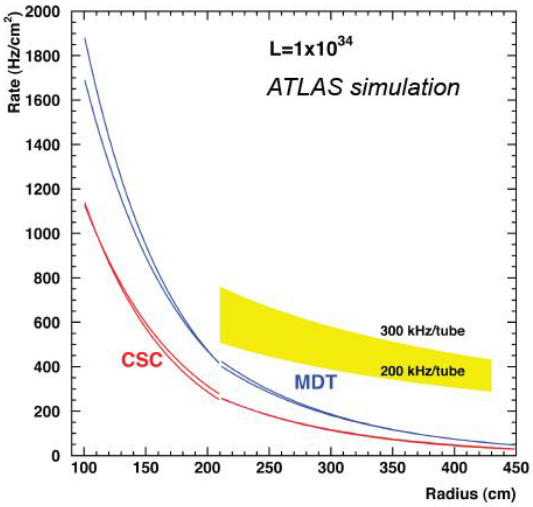

図 2.29: MDT 及び CSC のカウントレート。現行機のデザインでは $\sim 1 \times 10^{34}$cm^{-2}s^{-1} が限界で、オーダーとしてはもう 1 桁必要である [47]。

- **フェイクトリガー**
  フェイクトリガーは衝突点由来でない荷電粒子に対して発行されるトリガーで、2 つの原因が考えられる。1 つは 2 次粒子によるトリガーで、陽子や中性子によって生じる。これらの粒子は磁場領域付近のビームパイプや磁石などに衝突することで荷電粒子を生成し、その荷電粒子はエンドキャップトロイド磁場によって曲げられてあたかも high-p_T ミューオンのよう

な飛跡で BW にヒットする。もう 1 つは、ビームパイプから飛来する粒子によるトリガーである。現在のビームパイプにはシールドが施されていない箇所が存在し、そこに粒子がヒットすることで high-p_T に似た飛跡の粒子が BW にヒットすることがある。

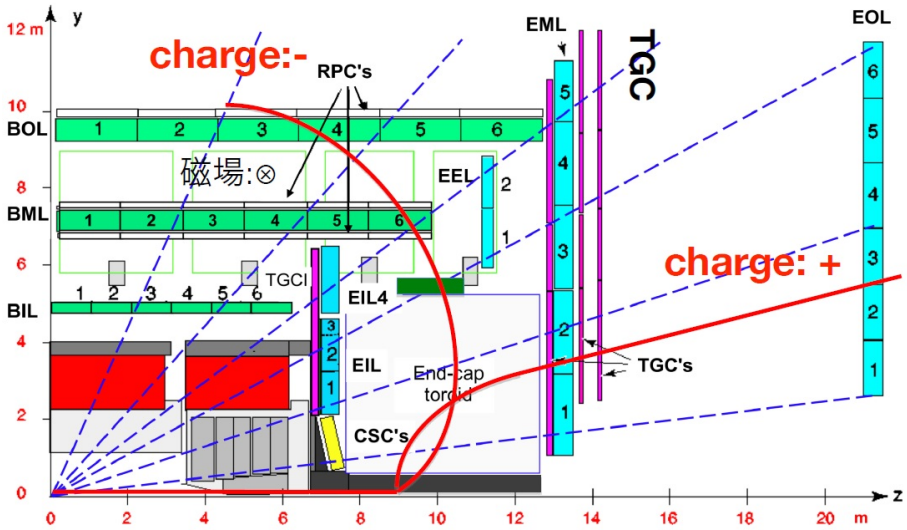

図 2.30: フェイクトリガーを起こす荷電粒子の飛跡 [48]。ビームパイプから飛来する粒子の飛跡の例。

2.6.2 ミューオン検出器のアップグレード

phase-1 では SW を New Small Wheel（NSW）に取り替える計画がある。NSW には Micro Megas（MM）と small TGC（sTGC）で組まれることが決まっており、性能としてはこれまでの MDT + CSC 系では耐えられないルミノシティにも耐え、位置及び時間分解能を維持することが要求される。

NSW では粒子のヒット情報に加えて、角度情報 $d\theta$ を使用することでフェイクトリガーの大幅な削減が見込まれている。まず、フェイクトリガーの原因の 1 つであるビームパイプ由来の荷電粒子は、IP から見て NSW の外側から飛来するので、NSW にヒットがない時点でフェイクであると断定できる。また、2 次粒子は角度情報 $d\theta$ によって、以下の手順で削減することができる。

図 2.31 の変数を用いて dL は以下のように定義される。$d\theta$ は IP とセグメントを結ぶ直線に対する飛跡の角度である。本来の high-p_T ミューオンの場合は、$d\theta \simeq 0$ なのでカットをかけることができ、フェイクトリガーの削減を図ることができる。また、dL は SW 及び BW における (η, ϕ) の差で、これも本来の high-p_T ミューオンであれば $dL \simeq 0$ となってカットをかけることができる。

$$dL = \sqrt{(\eta_{RoI} - \eta_{Seg})^2 + (\phi_{RoI} - \phi_{Seg})^2} \tag{2.1}$$

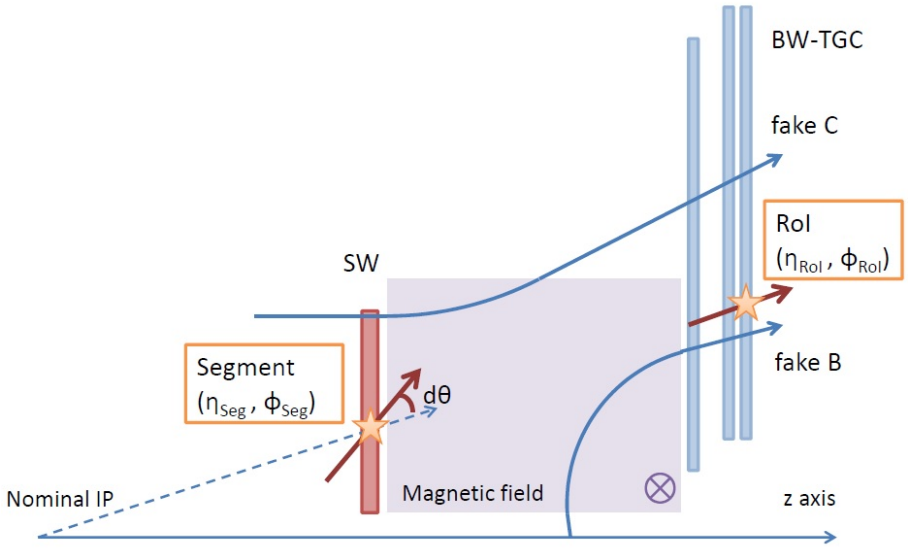

図 2.31: $d\theta$、dL カットの概念

以上のように、NSW の導入によってフェイクトリガーを削減できると期待されている。フェイクの削減には精度の高いトリガーが必要であり、より大きなデータを処理しなければならない。しかし、現在の読み出し系で使われているモジュールのままでは NSW を加味したデータを処理する能力はなく、それに応じて読み出し系もアップグレードが必要となる。

第3章　プロセス間通信を用いたReadOut Driverの開発

本研究では現行のハードウェアベースのRODに代わる、プロセス間通信を用いたソフトウェアベースのRODの開発について詳しく述べる。本章以降では便宜的に、現行のRODをHardware ROD、プロセス間通信によるRODをSoftware RODと呼ぶことにする。

3.1　Run-3におけるリードアウトスキーム

Run-3からはNSWの導入により、NSW情報とBW情報のコインシデンス処理をするブロックが必要になる。そこで、TGCグループでは現行のSLに代わるNew-SLの開発を行っている[28][30][31]。Run-3におけるNew-SLのリードアウトスキームを図3.2に示す。New-SLにはSLB ASICのリードアウト部の機能及びSSWの機能を移し、1つのFPGA上で実現する予定である。出力はSiTCP（Silicon TCP）技術によるTCP/IP通信で行われ、現行のSSWに代わり市販のスイッチングハブで収集系を構築する。また、ROD上に設置されているTTCrxはNew-SLと同様に、SiTCPを用いてRODにTTC情報を送信する（便宜的にNew-TTCと呼ぶ）。スイッチングハブで収集されたデータはイーサネットケーブルを介してDAQ PCへ送信される。

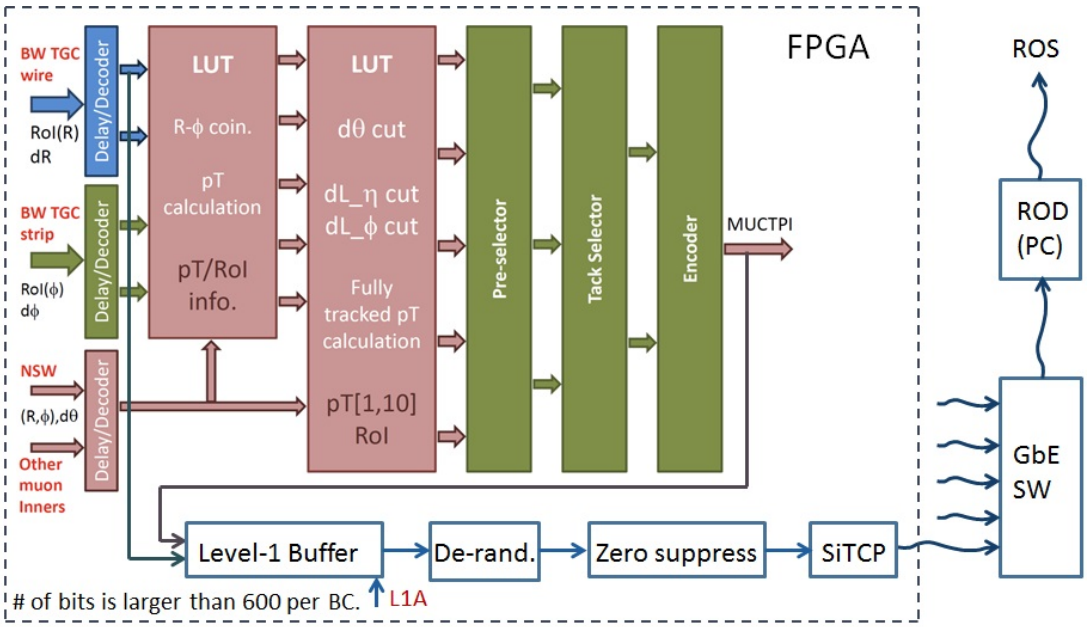

図 3.1: Run-3における New-SLのリードアウトスキーム [44]。New-TTC/SLのデータはイーサネットケーブル及びスイッチングハブを通じてDAQ PCで収集される。

3.1.1 SiTCP

Silicon TCP（SiTCP）は FPGA 上に構成されるネットワークプロセッサで、FPGA とイーサネットを接続する技術である。SiTCP は見る立場によって振る舞いが異なり、FPGA 上のユーザ回路からは FIFO のように見え、PC からはイーサネットを介して TCP サーバのように見える。そのため、一般的に普及したイーサネット及びスイッチングハブを用いてデータ収集系を構築することができる。SiTCP はハードウェアベースの処理であるため、イーサネットの転送上限値で通信することが可能となっている。

3.1.2 ROD に対する入力フォーマット

新しいリードアウトスキームでは New-SL に NSW 情報が入力として加わるため、それに応じて ROD への入力フォーマットを変更する必要がある。DAQ PC は New-TTC と New-SL からスイッチングハブを介してデータを受け取るため、TTC フォーマット及び SL フォーマットの 2 種類を定義する必要がある（図 3.1）。

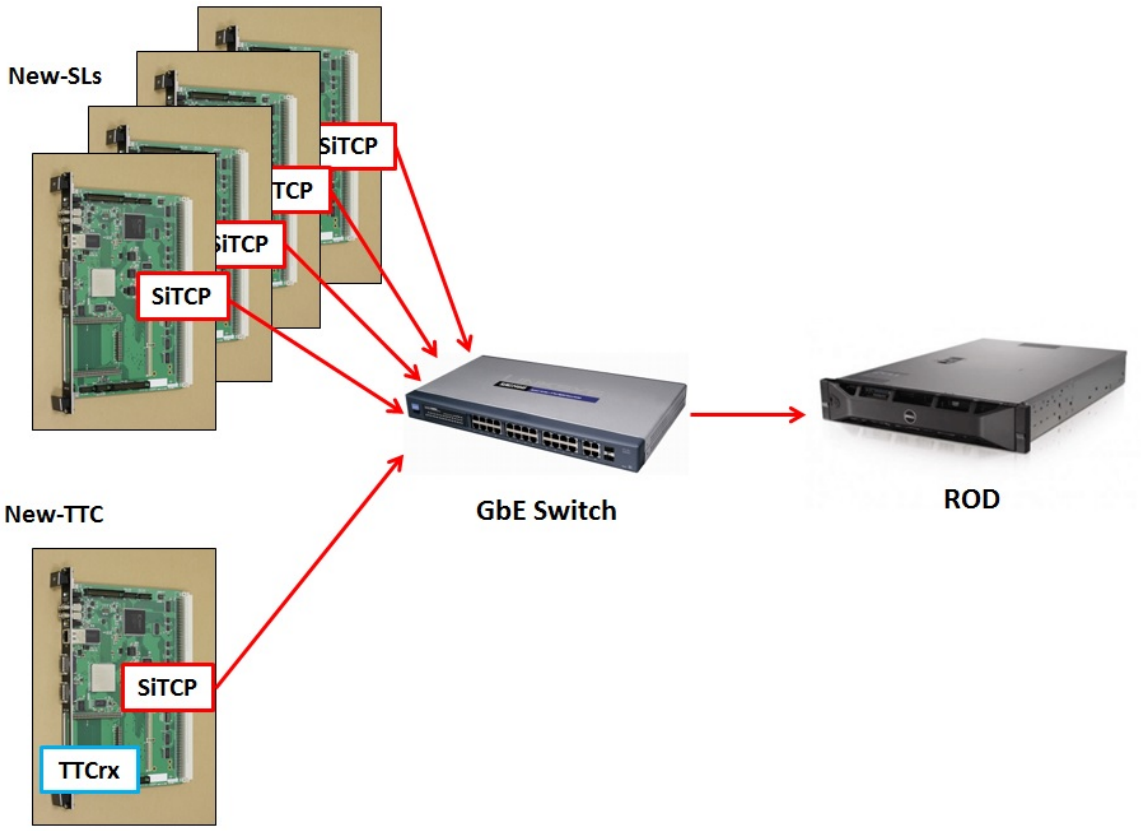

図 3.2: New-TTC/SL 及び Software ROD によるリードアウトスキーム。各コンポーネントはイーサネットケーブル及びスイッチングハブで接続されている。

以下に TTC/SL フォーマットを図 3.3、図 3.4 に示すが、要求によっては変更される可能性があることに留意されたい。

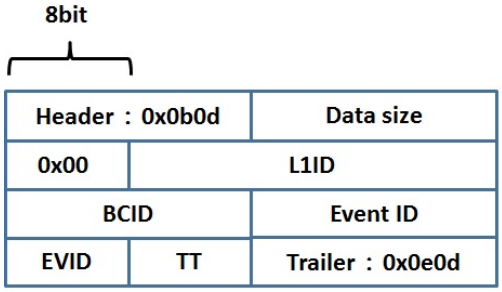

図 3.3: Run-3 における TTC フォーマット

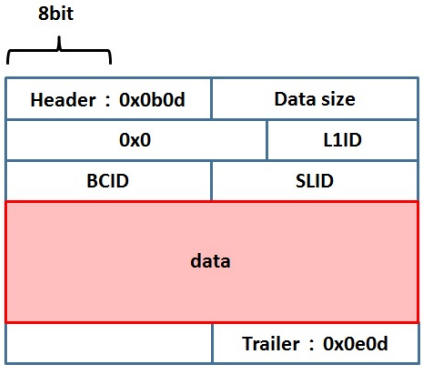

図 3.4: Run-3 における SL フォーマット

2 種類のフォーマットの構成は基本的に同じで、ヘッダにいくつかの ID を格納する。以下に ID 系の説明を簡単にまとめる。

- **L1ID**
  L1ID は TTC 上でカウントされた L1A 信号の受信回数を表す。TTC フォーマットでは 24 bit 分の領域が用意されているが、実際はリセット信号（ECR）によって 12 bit しか使用されない。SL フォーマットについても同様で、12 bit の領域のみ使用されて word の残りは 0 になる。

- **BCID**
  BCID はバンチ衝突に同期したカウント信号で、12 bit の領域に格納される。L1ID と同様にリセット信号（BCR）をかけられてループする。

- **Orbit ID**
  Orbit は LHC1 周分のバンチ衝突が終わったときに出力される信号で、Orbit ID は Orbit が出力された回数をカウントしたものである。L1ID と同じリセット信号がかけられる。

- **SLID**
  SLID はどの SL 由来のデータフラグメントかを判別するための ID である。

- **Trigger Type（TT）**
  Trigger Type は L1A 信号の種類を表す。

ID 系以外の要素としては 16 bit の領域をもつデータサイズがあり、ここには word 数を格納する。例えば TTC フォーマットの場合、1 word = 4 Byte と定義しているので、データサイズは 4 word となる。また、SL フォーマットのデータブロックはゼロサプレスにより圧縮されるので、データサイズは可変長となる。

Run-3 における L1A 信号の発行頻度は 100 kHz であるため、New-TTC 及び New-SL からは表 3.1 の転送速度でデータが送信される。New-SL の出力転送速度は、2012 年 11 月 29 日から 2013 年 2 月 14 日の Run で取得した SSW のデータを元に算出した値である [28]。本研究では、New-SL の最大出力転送速度 51.2 Mbps（1 イベントあたり最大で 16 word = 64 Byte）を処理できるような ROD を目標とする。

表 3.1: TTC 及び SL の出力転送速度 [28]。NSW 情報は現段階では未確定であるため、SL からの最大転送速度は Run1 の Raw data を元に算出している。

	転送速度 [Mbps]		
	最小値	平均	最大値
TTC	12.8 (fixed)		
SL	12.8	14.4	51.2

3.2 Software ROD 開発の目的

Hardware ROD は ATLAS 実験専用に作られたハードウェアモジュールで、複数の SSW からデータを収集する。しかし、Hardware ROD は実験専用であるため、将来的な技術革新を取り入れるのが難しい。また、モジュール自体に問題が見つかった場合は、修繕までに相当な時間を要するため非効率である。

Run-3 におけるリードアウトスキームでは New-TTC/SL の出力に SiTCP の組み込みを予定しており、それによって後段の ROD はイーサネット及びスイッチングハブを用いたソフトウェアベースの収集系として構築することが可能となった。ROD をソフトウェアベースにするメリットとしては、プラットフォームを選ばないので収集系に汎用的な機器を取り入れることができ、将来的な技術革新に柔軟に対応できる点が挙げられる。本研究では、汎用的な機器を用いた収集系を構築するため、ROD をプロセス間通信を用いたデータ収集ソフトウェアとして開発する。

3.3 Software ROD に要求される性能

Hardware ROD は片サイドあたりに 12 台設置されており、TTCrx モジュールから TTC 情報を受信して、1/12 セクター分のトリガー情報を処理する。これを Run-3 のリードアウトスキームに当てはめたものが図 3.5 である。

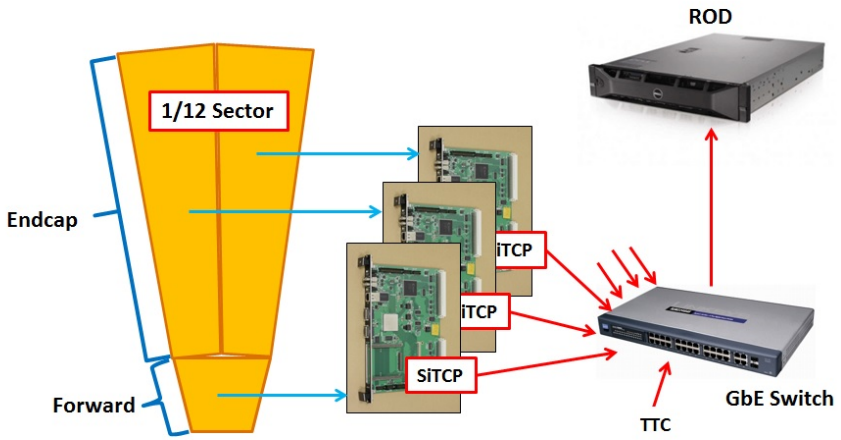

図 3.5: 1/12 セクターから ROD までのスキーム。1/12 セクターあたり 3 枚の SL ボードが担当して、ROD へデータを転送する。

New-SL は片サイドあたり 36 台（全体で 72 台）設置される予定で、3 台で 1/12 セクターのトリガー情報を処理することを考慮して、Software ROD を載せた DAQ PC は 1 台で少なくとも 1/12 セクター × 2 分のトリガー情報を処理する能力を持ち合わせているとよいと考えられる。つまり、New-TTC の役割を果たす汎用 VME モジュール 1 台と New-SL の役割を果たす汎用 VME モジュール 6 台以上に対するスイッチングハブと DAQ PC1 台の収集系となり、DAQ PC は少なくとも 7 台からデータを収集する必要がある。

片サイドに対する Software ROD を載せる DAQ PC の配置としては、以下の表 3.2 のように考えられる。DAQ PC は少なくとも 1/12 セクター × 2 分のトリガーデータを処理し、最大で片サイド分のトリガーデータを処理するので、DAQ PC の配置としては 12 の約数分になる。

転送速度

3.1.2 項で述べたように ROD に対して、New-TTC からは 12.8 Mbps、New-SL からは最大で 51.2 Mbps の転送速度でトリガーデータが送られてくる。また、ROD で収集されたイベントデータフラグメントは ROB 入力フォーマットに変換されるが、フォーマット自体のサイズは 19 word = 76 Byte であるので、L1A の発行頻度 100 kHz を考慮してフォーマット自体に必要な転送能力 T は 76 Byte × 8 × 100 kHz = 60.8 Mbps となる。つまり、Software ROD が持つべき転送能力は以下のようになる。ただし、n は New-SL の台数を表している。

$$T = 12.8 + 51.2 \times n + 60.8 \text{ [Mbps]} \tag{3.1}$$

この式を用いて、片サイドに配置する DAQ PC に対応した 1 台あたりの転送能力をまとめたものが以下の表 3.2 である。

DAQ PC の台数	New-SL の台数	転送速度 [Mbps]
1	36	1916.8
2	18	995.2
3	12	688.0
4	9	534.4
6	6	380.8

表 3.2: 片サイドあたりの DAQ PC の台数と必要な転送能力の関係。1 台の DAQ PC には 1/12 セクター単位で処理をさせたいので、配置は 12 の約数分になる。

よって、Software ROD は 380.8 Mbps 以上の転送能力をもつことを最低条件として満たし、エンドキャップ領域の片サイドを 1 台の DAQ PC でカバーできるような転送能力 1916.8 Mbps を目標として単機能性に優れた設計及び開発を行う必要がある。

3.4　プロセス間通信

高速なデータ処理を実現するため、DAQ PC 内では一連のデータ処理を分散化する必要がある。データ処理の分散化にはマルチプロセス及びマルチスレッド機能の使用が考えられ、その場合にはプロセス間及びスレッド間でデータパッシング、同期を行わければならない。このデータパッシング、同期方法をプロセス間通信（Inter Process Communication：IPC）機構と呼ぶ。本節ではネットワークを介したプロセス間通信、同一マシン内のプロセス間通信を説明し、通信をコントロールするロック機構について述べる。

3.4.1　ネットワークを経由したプロセス間通信

異なるマシンのプロセス同士の通信にはネットワークを経由する必要がある。ネットワークを介した通信にはソケットや RPC が挙げられるが、これらはネットワークに限らずローカルホスト上でも通信が可能である。

ソケット

ソケットは一般的な OS で提供される標準的な機能で、現在普及しているインターネット上で通信をするために欠かせない存在となっている。ソケットの形態にはいくつか種類があり、ドメイン[1]、ソケットタイプなどによって決まる。ドメインにはローカル通信、IPv4 といった種類があり、ソケットタイプにはストリーム、データグラムなどがある。Software ROD は SiTCP サーバと TCP/IP 通信を行うため、IPv4 のストリーム型ソケットを使用することとなる。ソケット通信はクライアントサーバモデルに基づいており、図 3.6 のように明示的に手順を呼ばなければならない。DAQ PC を図 3.6 に当てはめるとクライアント側になり、SiTCP サーバに接続を要求してサブシステムのデータを読み込むといった流れになる。

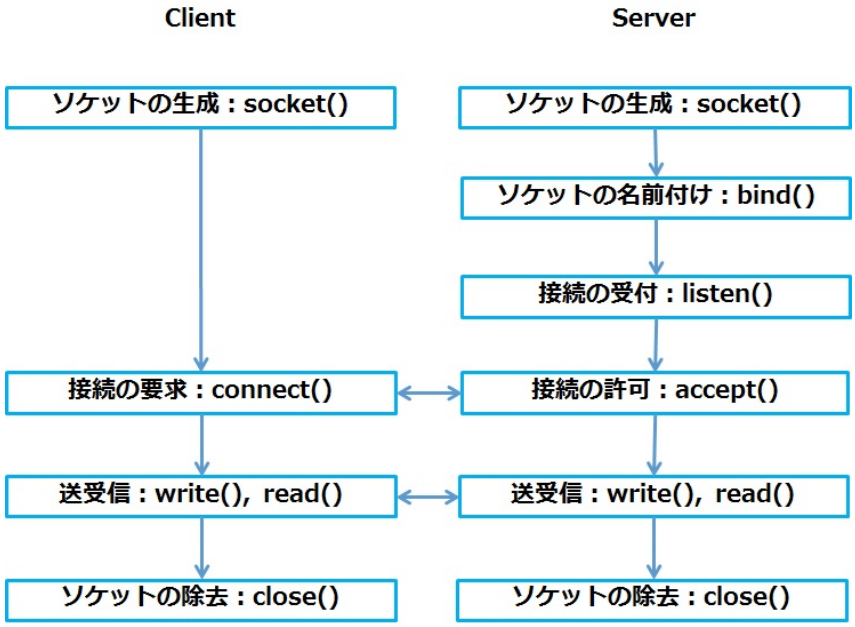

図 3.6: ソケット通信の手順

[1] ソケットの通用する範囲、プロトコルファミリとも呼ぶ。

RPC

RPC（Remote Procedure Call）は異なるホスト上の手続きを呼び出すことができ、ソケットのように明示的な通信手順を意識せずに済む。基本的な動作としては、クライアント側は手続きをリモートサーバに送信し、サーバはその要求を処理してクライアントに戻す。主に、分散システムを構築する際に使用される。

3.4.2 同一マシン内のプロセス間通信

ローカルホストでのみ使用可能なプロセス間通信について説明する。今から紹介するパイプや共有メモリはシンプルな形態であるがゆえに、ソケットや RPC のような複雑な手順を踏まなくてもよいというメリットがある。

パイプ

パイプは親子関係[2]のあるプロセス間でのみデータパッシングを行うことができる。パイプを生成するとメモリに専用のバッファが確保され、そのバッファに対する読み出し口及び書き込み口がディスクリプタ[3]として用意される。そのため、親子関係のあるプロセス間では子プロセスにディスクリプタがコピーされ、それによって情報の共有ができるようになっている。

シグナル

シグナルはある事象が発生したことをプロセスに知らせる働きを持つ。基本的にシグナルは非同期に発生するため、ソフトウェア割り込みの一種に分類される。シグナルの送受信も広い意味でプロセス間通信と呼ぶことができる。

共有メモリ

プロセスのアドレス空間はプロセスごとに異なるが、共有メモリは複数のプロセスに共通のメモリ領域を各プロセスのアドレス空間にマップすることで実現される。共有メモリはアドレス空間の共有により、カーネルを経由することなくデータパッシングを行うことができるため、プロセス間通信の中では最速である。しかし、プロセス間あるいはスレッド間で共有されたメモリは同時にデータを変更することは危険で、次項のロック機構を用いて排他制御を図る必要がある。

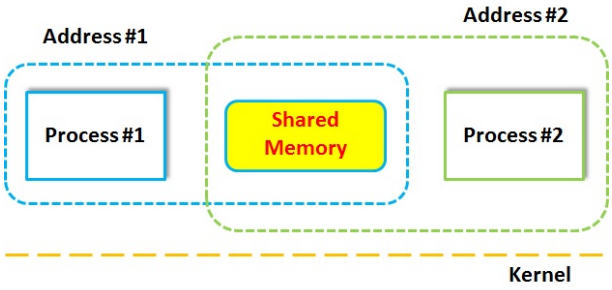

図 3.7: 共有メモリの概念 [49]。プロセスに固有なアドレス空間を共有することでプロセス間の通信を実現している。

[2] fork() 関数によって生成された子プロセスと親プロセスの関係。
[3] 記述子。識別子とともに様々な情報が含まれている。

3.4.3 ロック機構

ロック機構には様々な種類のロックが存在し、それぞれ異なった同期を提供する。Software RODではプロセス間通信の中で最速の共有メモリを用いてデータパッシングを行うが、共有メモリを用いた処理は大部分がクリティカルセクション[4]であるため、同期を行う必要がある。つまり、共有メモリを用いたデータパッシングの転送能力は種々のロックの能力に依存すると言い換えることができる。また、ロックを使用して複数の処理をコントロールする場合には、デッドロックと呼ばれるプロセスあるいはスレッドの膠着状態に注意しなければならない。ここでは、各ロックについて説明する。

相互排除変数

相互排除変数（Mutex）は最も基本的な同期形態で、複数のプロセスで共有されているリソースに対して同時に処理されないように保証する。プロセス間で同期を図る場合、相互排除変数は共有メモリ上に配置されなければならない。

条件変数

条件変数（Condition Variable）は待ちを目的とした同期機構で、ある条件が真になるまで待機することができる。基本的な動作は条件変数に対する wait 操作（待機）、signal 操作（通知）である。条件変数には常に相互排除変数が付随し、相互排除変数と同様に共有メモリ上に配置することによってプロセス間での使用を可能にしている。

リードライトロック

リードライトロック（Read-Write Lock）は相互排除変数と同様にクリティカルセクションの保護を行うが、リソースへのアクセスという観点で特徴が異なる。相互排除変数の場合、リソースへのアクセス権は 1 つのプロセスあるいはスレッドに限定されていた。それに対してリードライトロックの場合は、リソースに対する書き込みと読み出しが区別できるとき、つまりリソースを変更しないアクセスのときは複数のプロセスあるいはスレッドがリソースにアクセスできる。これも相互排除変数と同様に共有メモリ上に配置されることでプロセス間のロックを行うことができる。

レコードロック

レコードロックはファイルに対して働くロックである。ロックされるファイルはプロセスからディスクリプタによって参照され、ロックはカーネルによって管理される。ファイルはプロセスに共有のリソースであるため、相互排除変数や条件変数などとは異なり、レコードロックを共有メモリ上に配置しなくても使用できる。

セマフォ

セマフォは自身が保有する値によって複数のプロセスあるいはスレッドを同期する。セマフォの基本的な動作は、セマフォ値の加算（V 操作）及び減算（P 操作）で、プロセスがクリティカルセクションに進入するときに使用される。具体的には、P 操作後のセマフォ値を参照して、セマフォ値が 0 以上であればそのままクリティカルセクションへ入ることができ、セマフォ値が 0 未満であれば 0 以上になるまで待機する、といった動作になる。セマフォは相互排除変数と条件変数の拡張版と捉えることができる。

[4] 複数のプロセス及びスレッドからの同時アクセスによって、メモリ破壊などの危険が予測される領域。

ユーザが設計する同期は多種多様で、それに応じてロック機構も様々な種類のロックが発展してきた。ここで、各ロックに要する処理時間について述べる。相互排除変数、リードライトロックはシンプルな同期形態であるがゆえに、同期自体にかかる時間は比較的短い。それに対して、レコードロックとセマフォは高度な操作をできるが、その分同期処理に時間がかかってしまう。共有リソースに対するプロセス数あるいはスレッド数が増加するほど、その処理時間の差は顕著である [49]。Software ROD では高速なデータ処理が求められているので、相互排除変数、条件変数、リードライトロックを中心としたマルチプロセスの構成を練っていく。

3.5 システムの構成

システム全体の機能のユースケースを図 3.8 に載せる。ユースケース図は対象のシステムと外部との境界や関係を明確にした図で、アクターとユースケースによって構成される。アクターはシステムの外部に位置するエンドユーザや外部システムをモデル化したもので、ユースケースはアクターから要求された機能を表したものである。

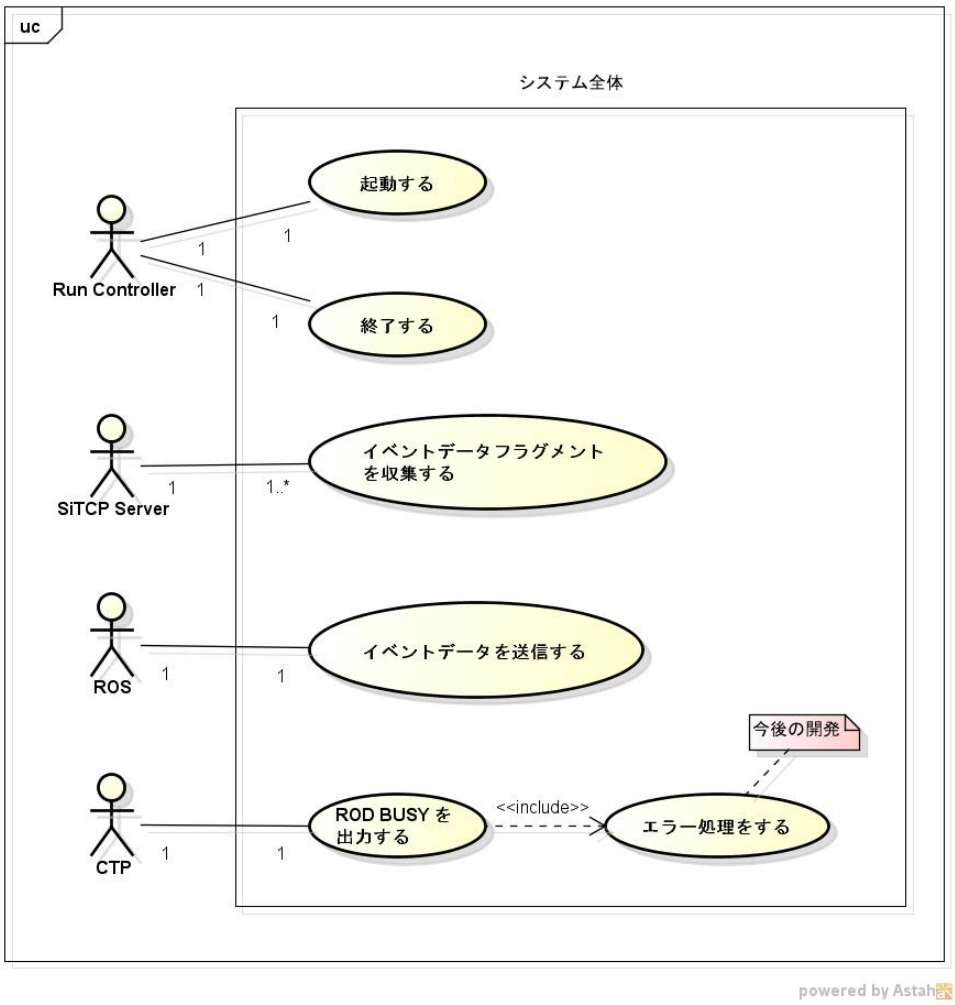

図 3.8: システム全体の機能のユースケース図。システムの主要な機能をピックアップした簡潔な図になっている。

ここで示したユースケース図はシステム全体の主要な機能をピックアップしたもので、図自体は簡潔なものとなっている。ユースケース図 3.8 におけるメインフローは SiTCP サーバと関連した ”イベントデータフラグメントを収集する ”、ROS と関連した ”イベントデータを送信する ” の 2 つのユースケースである。2 つのユースケースについて、もう少し詳しく見ていく。

ユースケース :「イベントデータフラグメントを収集する」

このユースケース図 3.9 は、New-TTC/SL モジュールに構築された SiTCP サーバからイベントデータフラグメントを収集するというユースケースをシナリオに沿って具体化したものである。Software ROD のデータ収集プロセスは、まずはじめに SiTCP サーバと接続を確立した後にデータを受信し、次に受信した各イベントデータフラグメントのヘッダ及びトレイラをチェックして、最終的にデータをバッファに保存する、という手順を踏む。

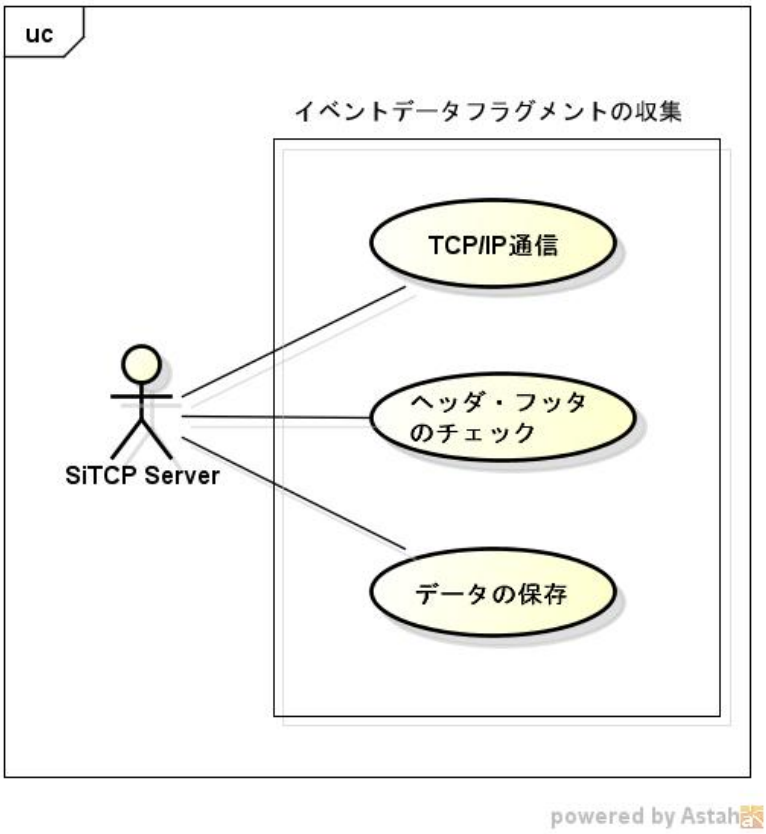

図 3.9: 「イベントデータフラグメントを収集する」のユースケース図。アクターとして SiTCP サーバとなっており、New-TTC/SL を意味している。

ユースケース :「イベントデータを送信する」

このユースケース図 3.10 ではイベントビルディングを行う。イベントビルディングは、New-TTC から受信した TTC 情報を参照して New-SL から受信したデータの整合性をチェックすること、ROB 入力フォーマットへ変換することをユースケースとして包含している。

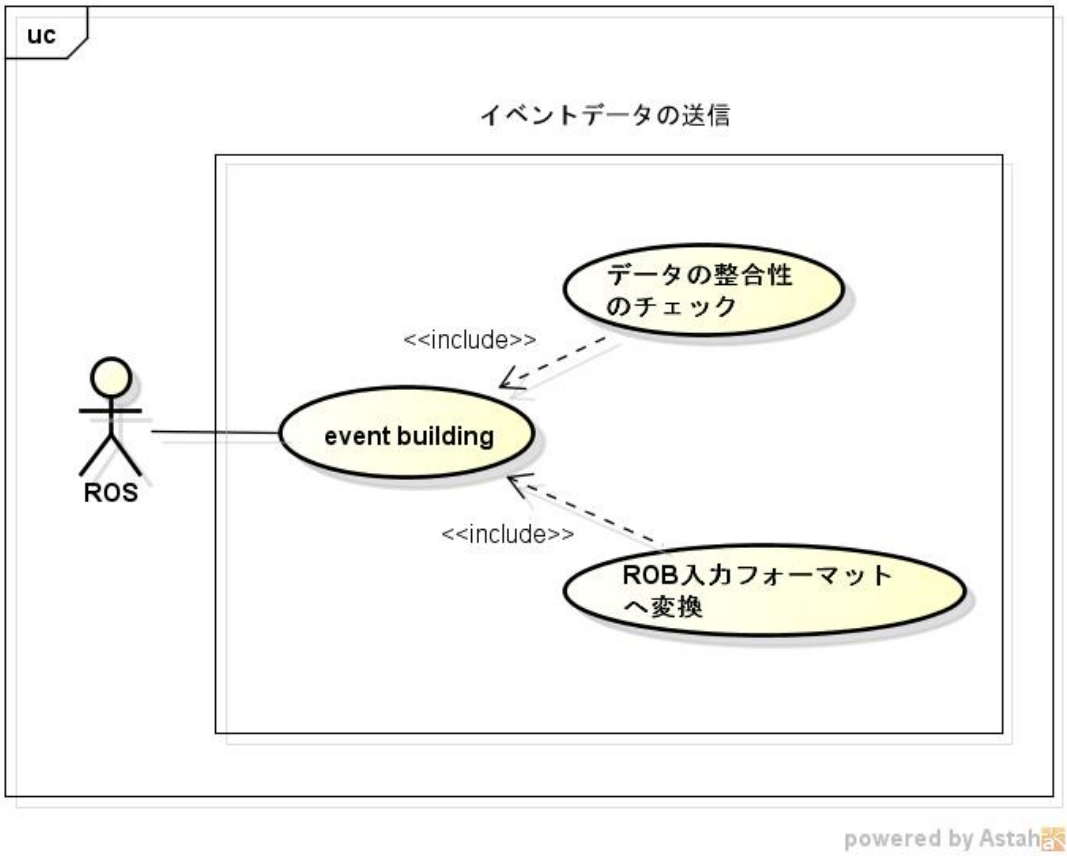

図 3.10: 「イベントデータを送信する」のユースケース図

これらのユースケース図をもとに描いた Software ROD の模式図が図 3.11 である。システムはシステム全体を管理・制御する ReadoutMaster、TCP/IP 通信によってスイッチングハブからデータを収集する Collector、複数のデータフラグメントに対してイベントビルディングを行う EventBuilder によって構成される。

- ReadoutMaster
  ReadoutMaster は SiTCP サーバの数に応じて Collector 及び EventBuilder を生成し、自身はマスターとなって Collector 及び EventBuilder を管理・制御する。具体的には、マスターは主にシステム全体に影響するエラーの対処を行い、エラーの内容によっては外部システムに通知する。

- Collector
  Collector は 1 つの TTCCollector と複数の SLCollector に分類され、各 Collector は SiTCP サーバと 1 対 1 のソケット通信を行う。

- EventBuilder
  EventBuilder は各 Collector からイベントデータフラグメントを収集して、L1ID のそろったイベントデータフラグメントを 1 つのイベントデータとして ROB 入力フォーマットへ加工する。

- RingBuffer
  Collector 及び EventBuilder が独立に動作をするためにはその間に何らかのバッファが必要で、Software ROD では RingBuffer という両端が論理的に接続されたバッファを Collector と EventBuilder の間に挟む。RingBuffer は各 Collector に対して 1 つずつ生成され、EventBuilder は複数の RingBuffer からデータを順番に読み出す。RingBuffer を Collector の数だけ配置した理由としては、各 SiTCP サーバから届くイベントデータフラグメントの到着時間のばらつきを吸収するためである。

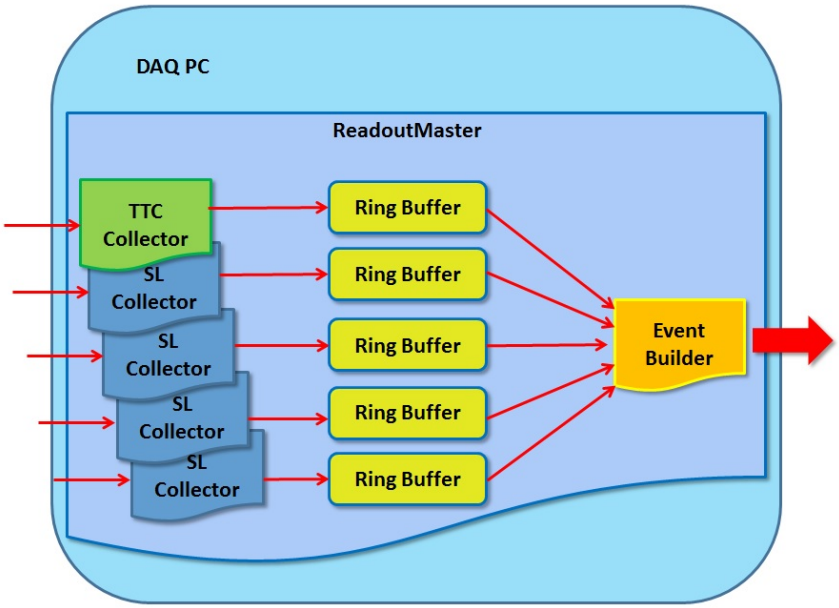

図 3.11: Software ROD の模式図。SiTCP サーバの台数に応じて Collector が展開され、EventBuilder によってデータがイベントビルディングされる。

Software ROD では高速なデータ処理を実現するため、メインフローのユースケースをマルチプロセスあるいはマルチスレッドによって、処理を並行化する必要がある。処理の並行化において、マルチプロセスを適用するか、マルチスレッドを適用するかはシステムに対する要求に依存するが、Software ROD ではマルチプロセスを採用する。マルチプロセスはマスターとなるプロセスさえ生きていれば、システム全体が落ちるという事態を防げることが多いため、マルチスレッドよりも安全性が高い。メインフローのユースケースでは対象とするシステムが SiTCP 及び ROS と別々であるため、安全性を考慮してマルチプロセスを選んだ。また、ハードウェアによるデータ処理は往々にしてエラーが生じやすく、サブシステムの最終ポイントに位置する ROD はそのエラーを受けやすいという点も挙げられる。スキーム上で発生したエラーは、DAQ PC においてはプロセスの異常終了という形で現れる危険性がある。仮にマルチスレッドで各スレッドが SiTCP サーバと TCP/IP 通信を行っていたとすると、プロセスの異常終了によりスレッドは全て落ちてしまい、全てのコネクションが切れてデータをロスする結果となる。

また、マスタープロセスである ReadoutMaster はシステム全体の管理及び制御を行うが、管理項目の中でも CTP への ROD busy の通知が非常に重要である。Software ROD における ROD busy は RingBuffer がフルの状態に生じ、Collector から発行される。そのため、マスタープロセ

ス内に ROD busy 専用のスレッドを生成して、busy 信号を待機するのがよいと考えられる。システムに全体に影響する他のエラーについても同様に、専用のスレッドで対処することでエラー処理の確実性が上がる。

3.5.1 RingBuffer

RingBuffer は共有メモリの両端を論理的に接続したバッファで、メモリへのアクセスコントロールを相互排除変数と条件変数で行っている。共有メモリは 2 層の領域に分割され、それぞれコントロール部とデータ部として使用される。コントロール部にはデータ部を制御するための情報が格納されており、データ部には Write offset と Read offset に従って収集されたデータが格納されている。Write offset と Read offset はコントロール部の情報によって決まり、Collector プロセスは Write offset を参照して書き込み、EventBuilder は Read offset を参照して読み出す。

コントロール部

コントロール部にはバッファ全体の排他制御を行う相互排除変数とプロセス間で通知を行う条件変数が埋め込まれ、その他にも書き込み及び読み出しに必要な情報の領域が用意されている。コントロール部の情報は基本的に 3 つの要素で構成されているが、今後の要求に応じて柔軟にカスタマイズすることができる。以下に各変数の説明を行う。

- Nwriter/Nreader
  データ部に対する書き込み及び読み出しの制御は相互排除変数によって行われるが、データ部にアクセスを試みているプロセスは Nwriter（書き込み待ちプロセス数）と Nreader（読み出し待ちプロセス数）で管理される。Nwriter/Nreader が 0 以下の場合はデータ部にそのままアクセスすることができ、1 以上の場合は条件変数に wait 操作がかけられ、データ部に対する書き込みあるいは読み出しが終わるまでプロセスは待機させられる。
- Nword_in/Nword_out
  Nword_in/Nword_out はデータ部に対して書き込まれた word 数及び読み出された word 数を表す。また、Read/Write offset にはこの変数が用いられ、word 単位のオフセットが格納される。
- Count_in/Count_out
  Count_in/Count_out はデータ部に対して書き込み及び読み出しが行われた回数を表す。

データ部

データ部には Collector プロセスによって収集されたデータが書き込まれ、データの先頭にはデータの word 数を格納した領域が 1 word 分だけ付加される。データ部は論理的に両端が接続されており、Write offset 及び Read offset に従ってデータの書き込みと読み出しを行う。書き込み時に Write offset が Read offset に追いついた場合、もしくは読み出し時に Read offset が Write offset に追いついた場合に条件変数に wait 操作をして他プロセスから通知されるまで待機する。Write offset が Read offset に追い付く場合はバッファがフルになっている状態で busy となるので、マスタープロセスに対して通知する仕組みとなっている。現在はデータ部のサイズを 40 MByte としており、データは最大でも 64 Byte なので、$40\ \mathrm{MByte}/(64+4)\ \mathrm{Byte} = 588,235$ event だけイベントデータフラグメントを保持できる。ただし、4 Byte はデータサイズ用に付加された word である。

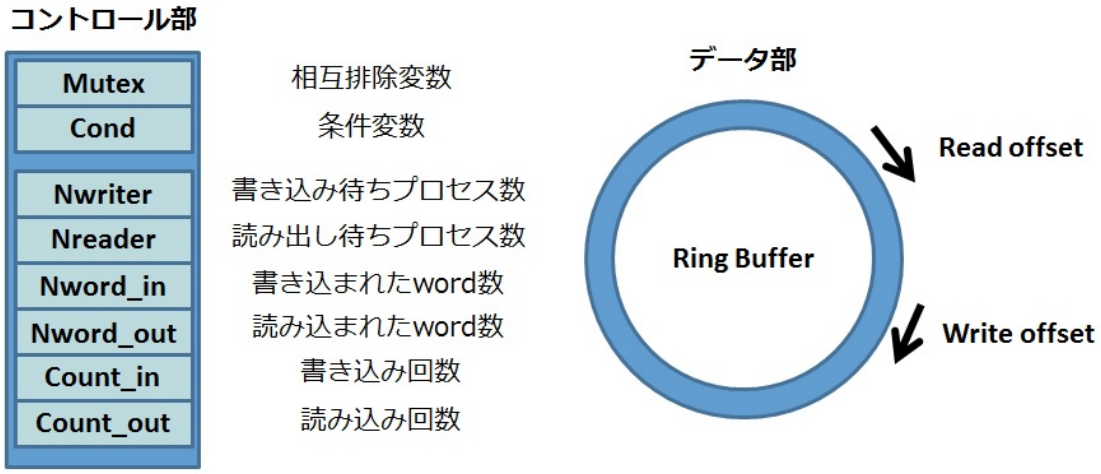

図 3.12: RingBuffer のコントロール部及びデータ部

3.5.2 TTC/SLCollector プロセス

TTC/SLCollector は TCP/IP 通信を介して ROD 入力フォーマットに従ったイベントデータフラグメントを受信し、RingBuffer へ格納するプロセスである。以下に TTC/SLCollector のクラス構成を図 3.13 に示す。

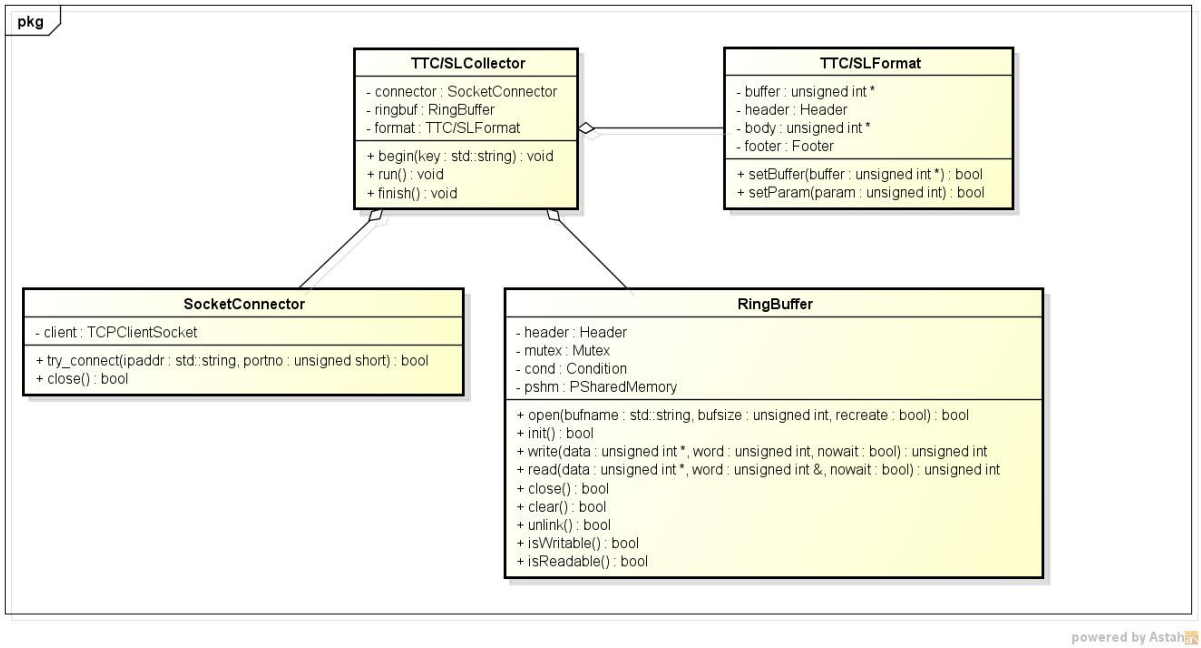

図 3.13: TTC/SLCollector プロセスのクラス構成。TTC/SLCollector クラスは、SocketConnector クラス、RingBuffer クラス、フォーマット用クラスを集約している。

Collector は ReadoutMaster によって起動され、SiTCP サーバと通信するための IP アドレス及びポート番号、RingBuffer へアクセスするためのバッファ名といったパラメータを受け取る。パラメータを受け取った後は、SiTCP サーバとの接続を確立し、RingBuffer へ書き込み可能な状態にしておく。

Collector のメインフローでは、ソケットで受信したデータを RingBuffer へ流すだけの単純な動作をする。SiTCP サーバからのデータの受信と RingBuffer への書き込みは同一スレッドで行われ、書き込みの前に各イベントデータフラグメントのヘッダ及びトレイラをチェックするが、SLCollector の動作には少し工夫が必要になる。New-SL からのイベントデータフラグメントは、ゼロサプレスにより可変長サイズで送信される。Collector はソケットバッファから一定量のサイズのデータ群を読みだすが、可変長サイズであるためデータ群の最後のイベントデータフラグメントが途中で切れてしまう場合がある。もちろん、イベントデータフラグメントが切れたままではイベントビルディングができないので、読み出しの際に調整しなければならない。Software ROD のデータ群の読み出し箇所は、Collector によるソケットバッファからの読み出し及び EventBuilder による RingBuffer からの読み出しであるが、複数の RingBuffer からの読み出しを行う EventBuilder よりも、Collector に処理を任せた方がよい。そこで、ソケットバッファからの読み出しで切れたイベントデータフラグメントの対処として、イベントデータフラグメントで切れて読み出された部分を次の読み出しに合流させる方法を採った（図 3.14）。

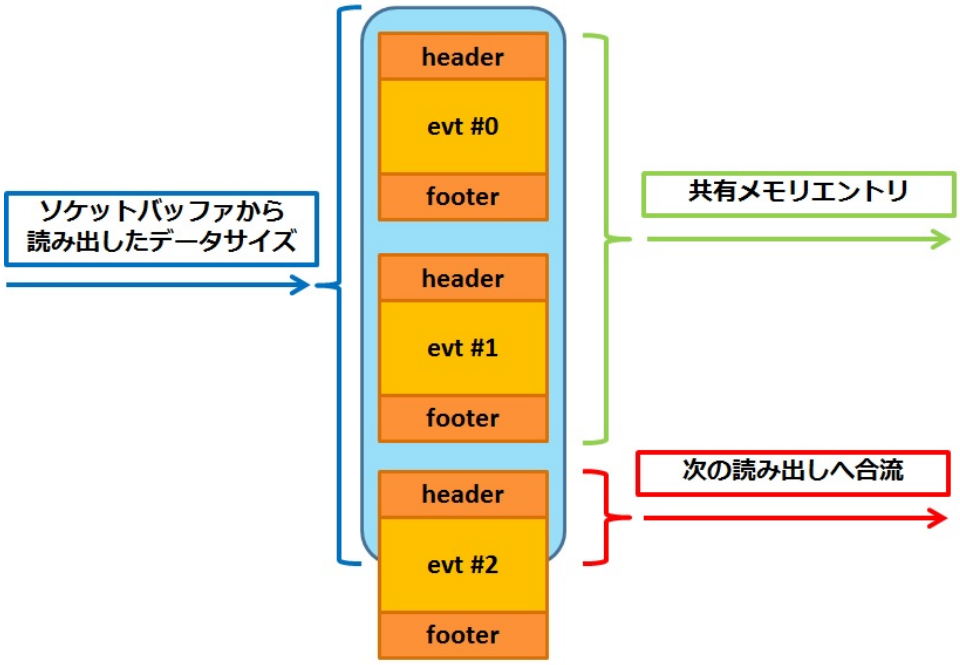

図 3.14: ソケットバッファからの読み出しで切れたイベントデータフラグメントの対処。ソケットバッファからは複数のイベントデータフラグメントが読み出される（青線）。ヘッダ及びトレイラをチェックするときにトレイラが確認されない部分を切れたイベントデータフラグメントとして判別し、その部分を次の読み出しの先頭へ合流させる（赤色）。残りの調整されたデータ群は共有メモリエントリとして RingBuffer へ書き込まれる（緑色）。

3.5.3 EventBuilder プロセス

EventBuilder は複数の RingBuffer からデータを読み出し、イベントビルディングを行うプロセスである。以下に EventBuilder プロセス内のクラス構成を図 3.15 に示す。

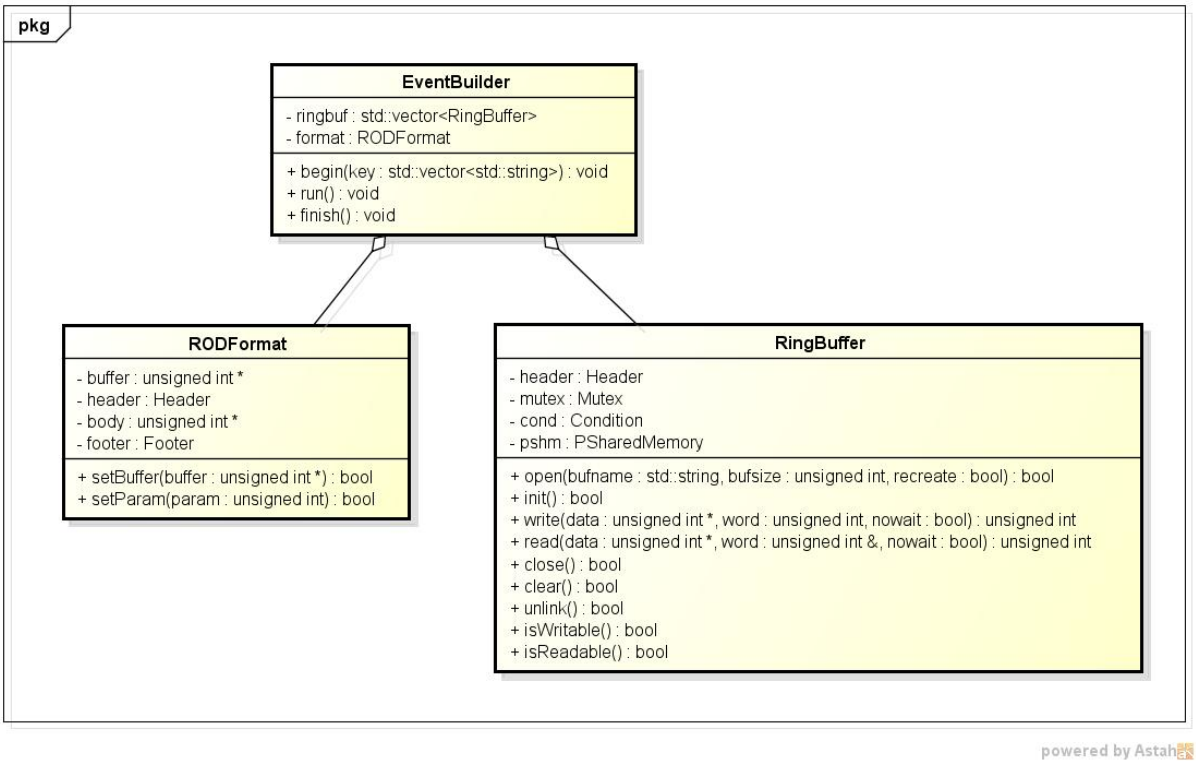

図 3.15: EventBuilder のクラス構成。EventBuilder クラスは RingBuffer クラスを集約しており、複数の RingBuffer オブジェクトを保有している。

以下に EventBuilder の動作をシーケンス図として図 3.16 に示す。EventBuilder は複数の RingBuffer から TTCCollector、SLCollector01、… 、SLCollector36 の順番でデータ群を読み出す。読み出されたデータ群はさらにイベントデータフラグメント単位に分割して、順番に 1 イベント分のデータをそろえる。各イベントデータフラグメントは TTC 情報をもとに L1ID、BCID のチェックが行われ、チェックを通過したイベントデータフラグメントはテンポラリバッファへ順にコピーされる。1 イベント分のイベントデータフラグメントの ID をチェックし終えた後、再度 TTC 情報をもとに ROB 入力フォーマットに変換する。これを RingBuffer から読み出したデータ群内のイベントデータフラグメントが全て処理されるまで繰り返し、処理し終えた後は再度 RingBuffer から読み出してデータ群を補充する。EventBuilder プロセス内では RingBuffer からの読み出しがボトルネックとなるので、RingBuffer からはなるべく大きいサイズで読み出して共有メモリへのアクセス回数を減らす必要がある。最終的に、変換後のデータは ROS へ送信されるが、これらの一連の動作は同一スレッドで行われる。

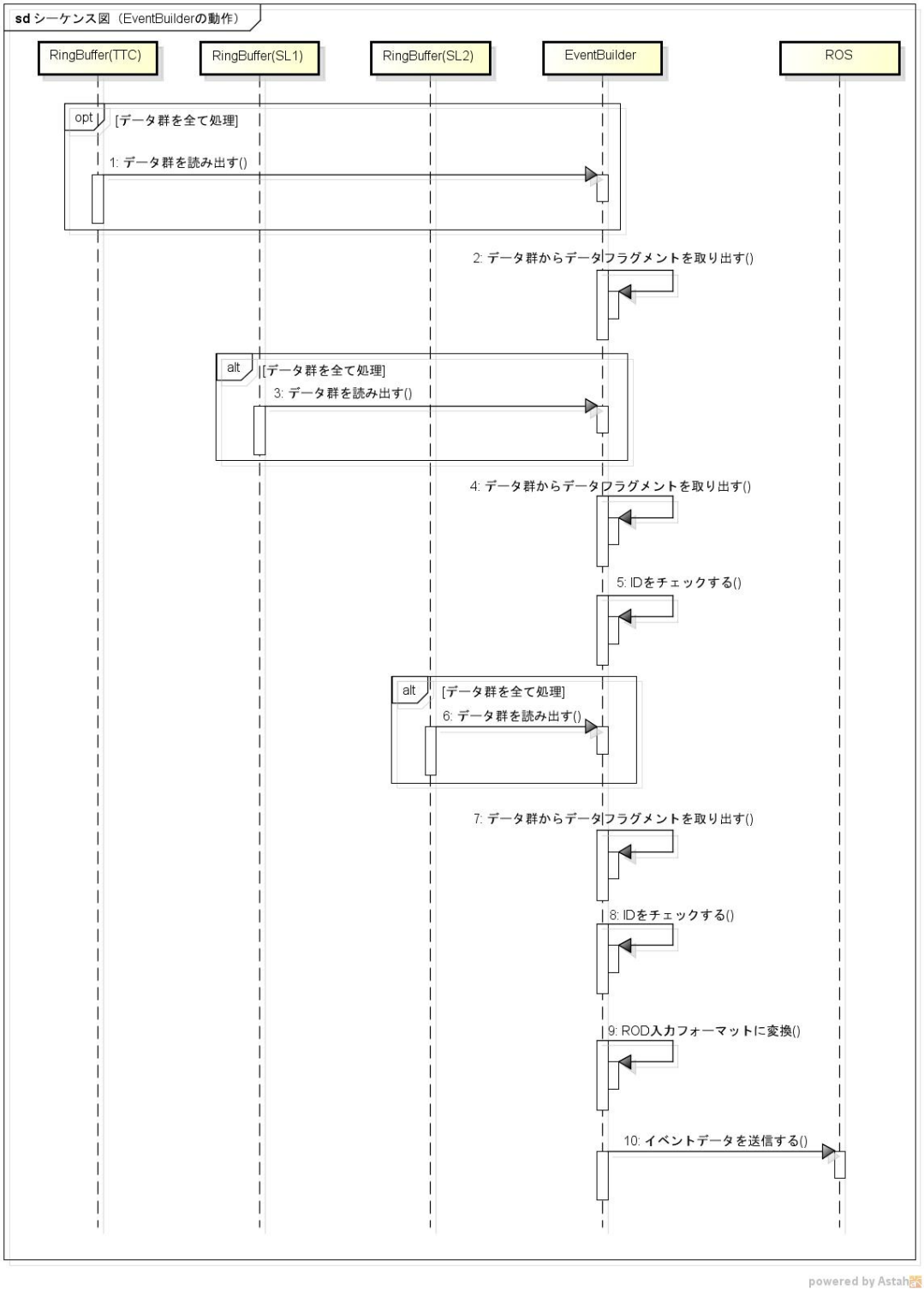

図 3.16: EventBuilder の動作。はじめに TTCCollector 用の RingBuffer から TTC 情報を取得して、順次 SLCollector のデータフラグメントを読み出す。その後に ID をチェックして、ROB 入力フォーマットに変換する。

3.5.4 ReadoutMaster プロセス

ReadoutMaster は TTC/SLCollector 及び EventBuilder を生成し、各プロセスを管理・制御するプロセスである。以下にクラス構成を図 3.17 として示す。

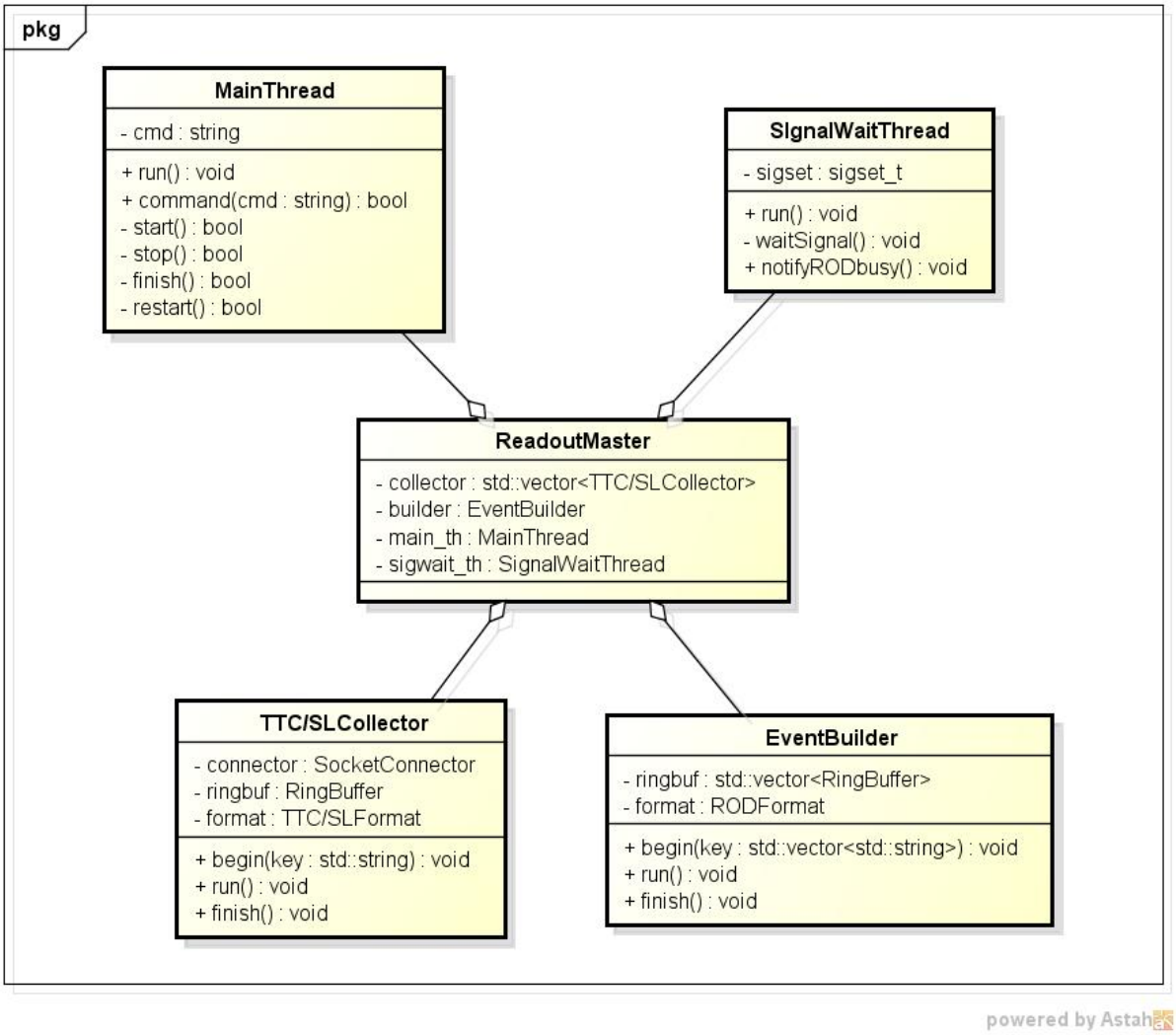

図 3.17: ReadoutMaster のクラス構成

ReadoutMaster はデータベースとなるクラスから IP アドレス、ポート番号、バッファ名といったパラメータを取得して複数の RingBuffer を初期化した後に、各プロセスにパラメータを与えて独立に動作させる。ReadoutMaster は Collector 及び EventBuilder から見て監視するプロセスという立場になり、さらに内部にいくつかの単純な動作をするスレッドが設置されている。現段階では MainThread、SignalWaitThread の 2 つであるが、システム全体を監視するスレッドを柔軟に追加することができる。MainThread は他のスレッドを監視する役目を担い、さらに将来的には Software ROD に対する外部システムとの窓口にする予定である。SignalWaitThread は子プロセスからのシグナルを捕獲するスレッドで、主に RingBuffer がフルの状態に発行されるシグナル（プロセス間通信で解説したシグナル）、つまり ROD busy を受け取る役割を果たす。データ収集時のシーケンス図 3.18 を載せる。

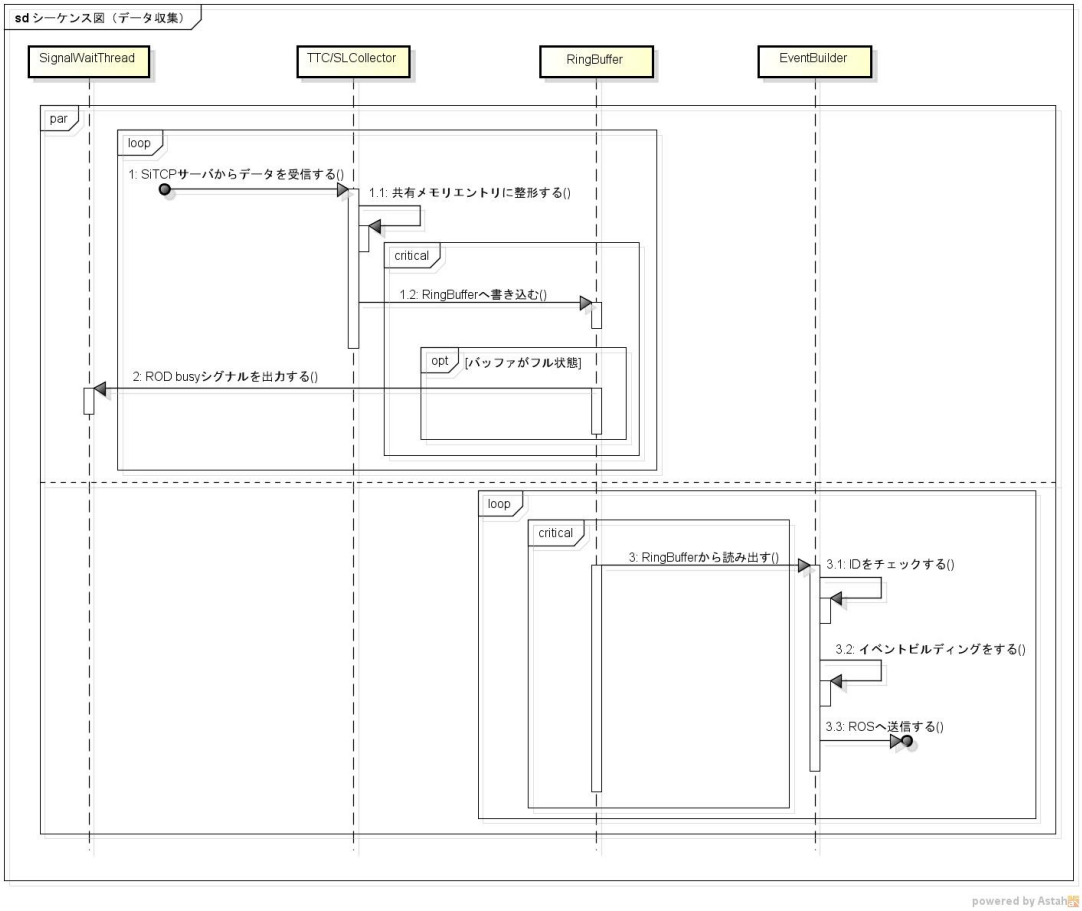

図 3.18: メインフローのシーケンス図

第4章　ReadOut Driverの性能評価

New-SL は現行のスキームで使用されているエレクトロニクスを使わず、現在は汎用 VME モジュールを用いてロジックの設計・動作試験を行っている [28][30][31]。現段階では汎用 VME モジュールは片サイド分の台数もないため、Software ROD の性能評価には工夫が必要である。本研究ではエミュレータや Distributor などを用いて Software ROD の性能評価を行った。

4.1　エミュレータを用いた性能評価

本節ではハードウェアと接続する収集系の試験に先立ち、New-SL 及び New-TTC の役割を果たすエミュレータを作成して擬似的に収集系を構築した。性能評価にエミュレータを用いる目的としては、スイッチングハブのボトルネックといったハードウェアによる不確定要素を除いて Software ROD が正しく要求された性能で動作するか検証するためである。収集系の構成を図 4.1 に示す。テストベンチについては後述するが、収集系には複数のテストベンチマシンに login.icepp を使用しており、便宜的に各 login.icepp に対して login.icepp01 というように番号付けをする。エミュレータは TTCEmulator、SLEmulator、ROSEmulator の 3 種類で、それぞれ TCP サーバとして Software ROD からのコネクション要求を待つ。エミュレータの数は最大で 38 台（TTC：1 台、SL：36 台、ROS：1 台）となるため、動作試験の際にはエミュレータを複数の login.icepp に分散させる。

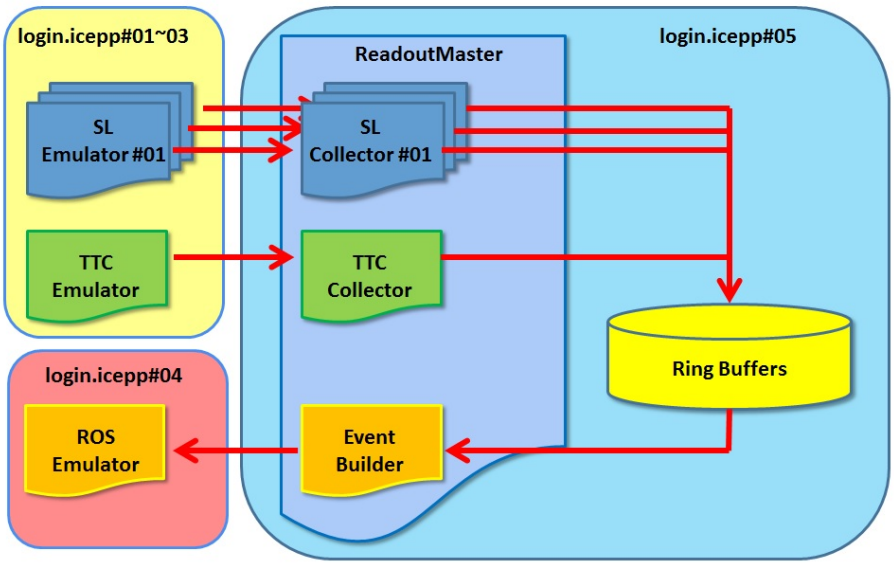

図 4.1: エミュレータを用いた収集系の構成。複数のテストベンチマシンを使用して、37 対 1（TTC: 1 台、SL：36 台）の収集系を構築する。

4.1.1 エミュレータ

エミュレータはソケットプログラミングを用いた擬似的なサーバである。TTC/SLEmulator は ROD 入力フォーマットに従ったデータを出力し、ROSEmulator は ROD によって変換されたデータを受信する。

TTC/SLEmulator

TTC/SLEmulator はデータソースとなるエミュレータで、適当なデータにヘッダ及びトレイラ情報を付加して 1 イベントデータフラグメントごとに TCP ソケットで送信する。TTC/SLEmulator では L1A の発行頻度を再現するために、内部でタイマー機能を用いて一定間隔でデータを発送するようにコードが組まれている。具体的には、タイマーは 100kHz = 10μs 間隔で自プロセスにシグナルを発行し、イベント生成・送信のループではタイマー由来のシグナルをキャッチするまで待機する。シグナルをキャッチした後は、イベントデータフラグメントを生成して TCP ソケットで送信しており、ここまでが 1 つのループとなる。また、New-SL のデータがゼロサプレスによって可変長サイズになることを加味して、SLEmulator のデータサイズはランダムに設定できるようになっている。

ROSEmulator

ROSEmulator は”ROS”という名前がついているが、ROS の機能を実装したエミュレータではなく ROD から出力されるデータのダンプや転送速度の算出を行うためのエミュレータとなっている。リードアウトスキームの構成に倣って、本試験では便宜的に ROSEmulator と名づけている。Software ROD が正しく動作しているか検証するためには Software ROD に対する入出力データが重要であるため、Collector プロセスとこの ROSEmulator はデバッグモードとしてデータのダンプと転送速度の算出を行うコードが組み込まれている。

4.1.2 Software ROD の評価に際して

Software ROD は第 3 章で述べた要求を満たして動作しなければならない。その中でも高速な動作によってデータロスを起こすことなくデータ収集することが最重要目標で、動作試験の最初のステップとしてシステム内のボトルネックを考察することが肝要である。Software ROD 内のデータフローでボトルネックになると考えられるのは、プロセス間通信によるデータパッシングと EventBuilder によるイベントビルディングである。イベントビルディングはシステムの主軸なので、システム全体の試験で性能評価することとして、まずはじめに通信単位の性能評価を行いシステム全体の試験に支障がないことを確かめる。

性能評価の指標

Software ROD はマルチプロセスで動作し、プロセス間は TCP ソケット及び RingBuffer を用いたデータパッシングを行うため、転送速度の測定が重要となる。データの転送速度を以下のように定義する。

$$\text{転送速度 [Mbps]} = \frac{\text{読み出されたデータサイズ [Mbit]}}{\text{読み出しに要する時間 [s]}} \tag{4.1}$$

Software ROD では基本単位として word を用いているが、転送速度を算出する際は 1word = 32bit の換算式を使用する。

表 4.1: login.icepp のパラメータ。10Gbps 通信が可能なサーバ。

パラメータ	性能
CPU	Intel Xeon E5-2680
コア数	8
動作周波数	2.7GHz
メモリ	16GB / node
NIC	10Gbps / node
Disk	600GB SAS × 2
OS	Scientific Linux CERN 6.4

テストベンチ

エミュレータを用いた動作試験では東京大学で使用されているサーバの login.icepp をテストベンチマシンとして用いる。テストベンチマシンの性能を表 4.1 に載せる。エミュレータを用いた試験では複数台の login.icepp 上で擬似的に収集系を構築する。各テストベンチマシンは 10 Gbps 通信が可能であるため、38 対 1 通信を実現することができる。

4.1.3 通信単位の性能評価

Software ROD 全体の動作試験に先立ち、プロセス間通信の性能評価とエミュレータの TTC / SLEmulator の動作確認を行う。ボトルネックになる箇所としてプロセス間通信を挙げたが、その中でも TCP ソケットはネットワークを介して異なるマシンのプロセスと通信するため、最も考慮すべき箇所と考えられる。逆に、RingBuffer は相互排除変数と共有メモリを用いた実装となっているため、システム内では高速に動作すると考えられるが、ここでは RingBuffer の動作確認を兼ねてデッドロック等のバグがないか検証する。

TCP ソケット

はじめに、TCP ソケット通信の性能評価と SLEmulator の動作確認を行った。この試験では SLEmulator の台数及びデータサイズを変化させて測定を行う。試験の構図 4.2 を以下に示す。図中の tcpcheck はデータのダンプ及び転送速度の算出を行うクライアントプロセスである。SLEmuator の台数は 1 台、6 台、9 台、12 台、18 台、36 台と変化させ、各台数でデータサイズを増やして転送速度の推移を見る。

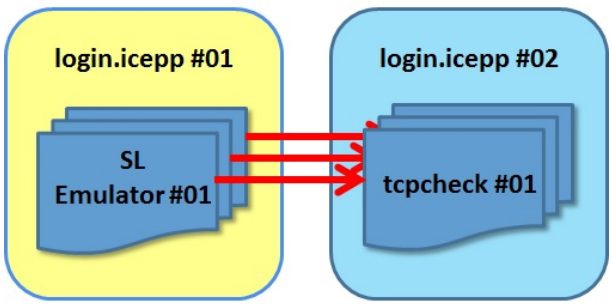

図 4.2: TCP ソケット通信試験の構成。tcpcheck プロセスで転送速度を算出する。

確認すべきポイントは以下の項目である。この試験ではテストベンチマシンの評価も兼ねており、SLEmulator を増やした場合に各 SLEmulator はどの程度の帯域を使うことができるかを測定し、エミュレータを用いたシステム全体の動作試験に支障がないことを確認する。

- SLEmulator は指定されたデータサイズに応じた転送速度で出力できるか
- テストベンチマシン間で使用できる帯域はどの程度か
- TCP ソケット通信で十分な転送速度を達成できるか

まず、SLEmulator が 1 台の場合の結果を図 4.3 を載せる。グラフは横軸に SLEmulator で指定した 1 イベントデータフラグメントのデータサイズ、縦軸に tcpcheck で算出した転送速度をとった。各測定点は 20 回測定して、その平均値を算出したものである。login.icepp 間は 10Gbps の通信が可能であるが、実際は TCP/IP やイーサネットのオーバーヘッダを考慮した実効転送速度 9.5 Gbps が使用できる範囲となる。図 4.3 を見ると、測定値はデータサイズが 1300 word 付近で直線から外れ、転送速度は 5.1 Gbps 程度で上限となっていることがわかる。転送速度の上限値を迎えたときに top コマンドで SLEmulator 及び tcpcheck の CPU 使用率を調べたところ、それぞれ約 37 %、約 91 %の使用率となっていた。つまり、tcpcheck での処理が頭打ちになっているということである。しかし、SL1 台の最大出力転送速度 51.2 Mbps に対して TCP ソケットの性能を確認するには十分である。これにより、TCP ソケットはシステム全体の動作試験に支障をきたさず、さらにエミュレータ自体も十分な性能を持っていることが確かめられた。

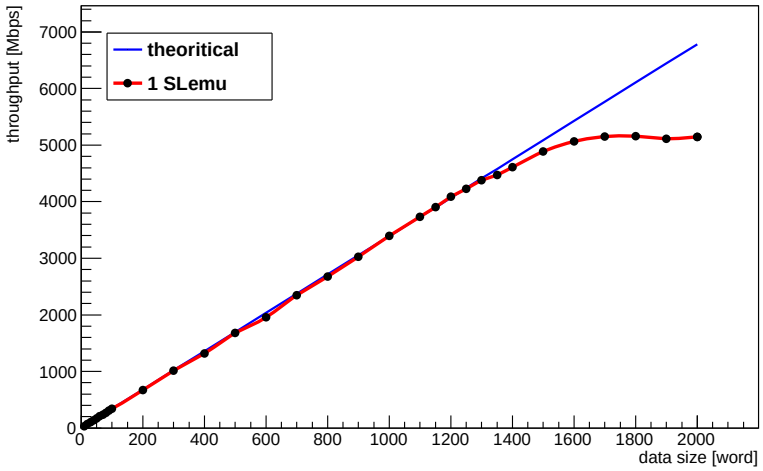

図 4.3: SLEmulator1 台のときの TCP 通信の結果。青色の直線はイベントデータフラグメントのサイズに応じた SLEmulator の出力する転送速度で、赤色の折れ線は測定値を表している。

次に、SLEmulator を複数台用いて並列に 1 対 1 通信を行った場合の結果を図 4.4、図 4.5 に載せる。グラフは SLEmulator01 に着目したグラフであるが、他の SLEmulator に関しても同様の結果である。各 SLEmulator は対応する tcpcheck と独立に TCP ソケット通信を行っているが、SLEmulator を 1 台だけ用いた通信と比べて早々に転送速度の上限を迎える結果となった。これはネットワークの帯域を各 SLEmulator が等分して使用しているためである。SLEmulator を 18 台用いた通信において、SLEmulator1 台あたりの転送速度の上限は $\sim$ 530 Mbps となっており、$530 \times 18 = 9.54$ Gbps で帯域が等分配されていることが確認できる。その他も同様で、6 台の場合は $1610 \times 6 = 9.66$ Gbps、9 台の場合は $1050 \times 12 = 9.45$ Gbps、12 台の場合は $785 \times 12 = 9.42$ Gbps となって帯域を等分配している。$\sim$ 530Mbps という上限値は ROD への入力要求値 51.2Mbps を優に超えており、18 台までの SLEmulator を 1 つの login.icepp 上に配置しても試験が可能であることを確認できた。

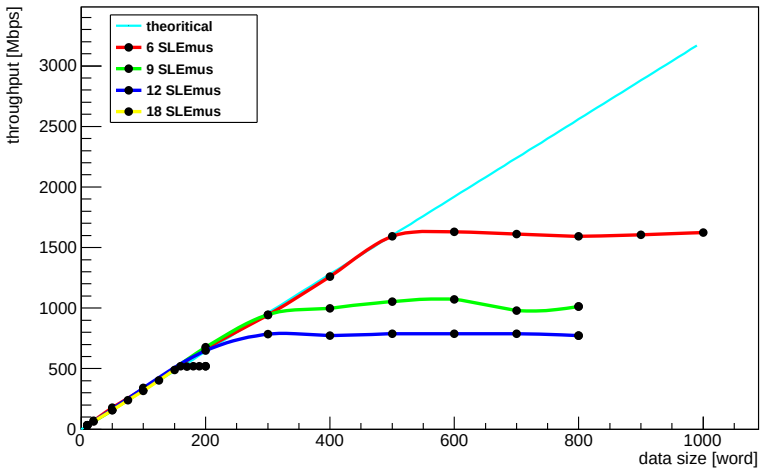

図 4.4: SLEmulator を 6〜18 台用いた TCP 通信の結果。水色の直線は図 4.3 と同様にイベントデータフラグメントのサイズに応じた出力の転送速度で、赤色の折れ線が 6 台、緑色の折れ線が 9 台、青色の折れ線が 12 台、黄色の折れ線が 18 台の場合の測定値を示している。各測定点は 20 回測定した平均値を表す。

最後に SLEmulator を 36 台用いた通信について説明する。どちらの場合も転送速度の上限値は $\sim$ 260 Mbps となっている点では同じで、やはり帯域を等分配していることがわかる。しかし、直線に従っているかどうかという点で違いがある。1 つの login.icepp 上に全 SLEmulator を配置した場合は転送速度が直線よりも下回った結果となっており、各 SLEmulator は十分な動作をしていないことがわかる。理由としては、login.icepp 上のリソースの競合が発生してプロセスの処理に待ちが生じてしまったためと考えられる。3 つの login.icepp に分割した場合は帯域の上限値に達するまで直線に従っており、これは十分な性能を発揮できているといえる。

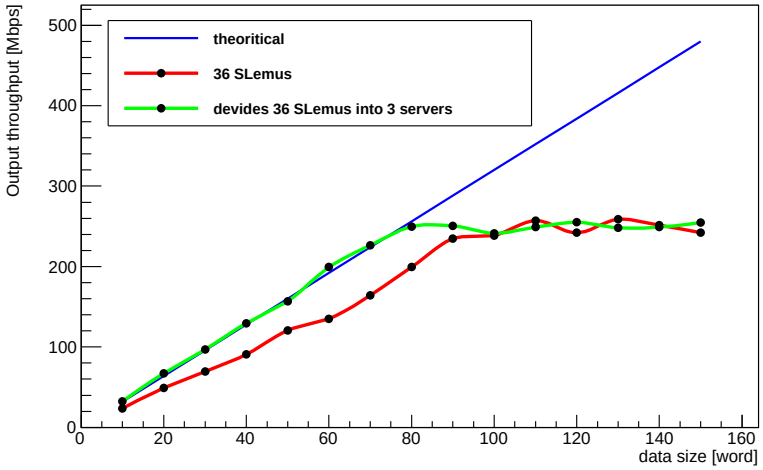

図 4.5: SLEmulator を 36 台用いた TCP 通信の結果。青色の直線については図 4.3、図 4.4 と同様である。また、赤色の折れ線は 36 台の SLEmulator を 1 つの login.icepp 上に配置したもので、緑色の折れ線は 3 つの login.icepp に 12 台ずつ配置したものである。各測定点は 20 回測定した平均値を表す。

RingBuffer

先にも述べたように RingBuffer はプロセス間通信で最速の共有メモリとロックの中では軽量な動作をする相互排除変数、条件変数で成り立っているため、システムのボトルネックになるとは考えにくいが、本節では RingBuffer を 1 個、37 個使用した場合の通信について考察して、システムのボトルネックにはなりえないことを確認する。通信の概略図 4.6 を以下に載せる。

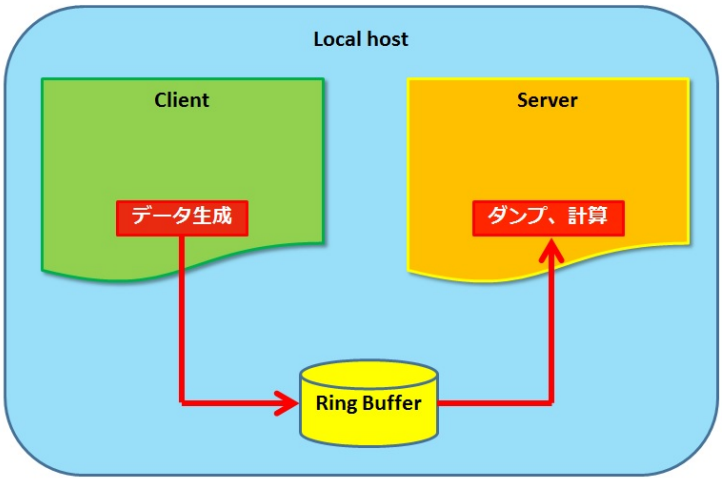

図 4.6: RingBuffer 単体テストの概略図。クライアントプロセスでデータを生成し、サーバプロセスで転送速度を算出する。

通信テストは TCP ソケットの場合と同様にクライアント・サーバの 1 対 1 通信を行う。クライアントプロセスは SLEmulator と同等の機能を有しており、L1A の発行頻度に合わせてイベントデータフラグメントを RingBuffer へ書き込む。サーバプロセスはクライアントプロセスによって書き込まれたデータ群を読み出し、データのダンプ及び転送速度の算出を行う。測定結果を図 4.7、図 4.8 を載せる。

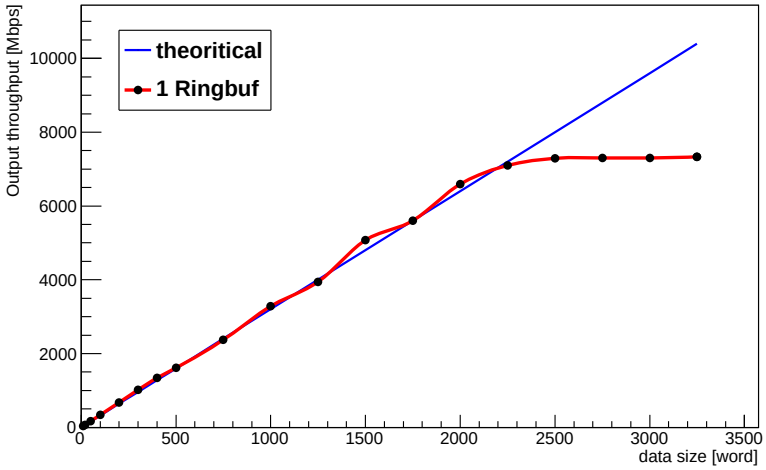

図 4.7: RingBuffer を 1 個用いた通信。青色の直線はデータサイズに応じた転送速度で、赤色の折れ線は測定値を表している。各測定点は 20 回測定した平均値を表す。

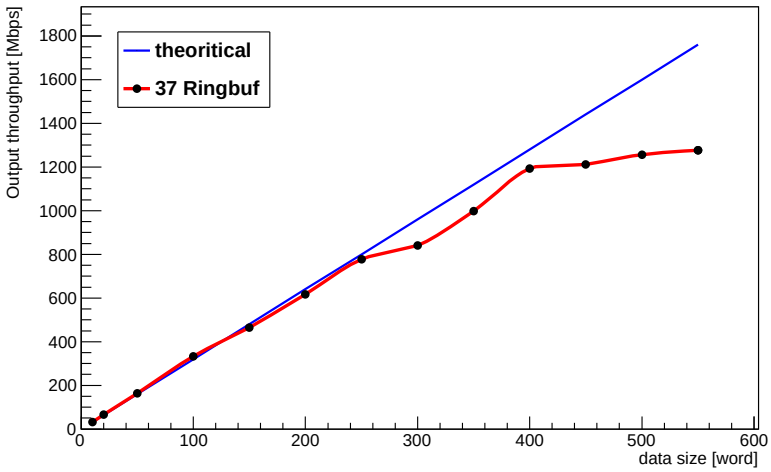

図 4.8: RingBuffer を 37 個用いた通信。青色の直線はデータサイズに応じた転送速度で、赤色の折れ線は測定値を表している。各測定点は 20 回測定した平均値を表す。

RingBuffer を 1 個だけ用いた通信ではデータサイズが 2250 word 付近で転送速度の上限を迎える。転送速度の上限値は ∼ 7.1 Gbps で、RingBuffer は非常に高速な通信ができることがわかる。さらに、37 個の RingBuffer を用いて並列に動作させたところ、データサイズが 250word 付近になるところまでは直線に従って転送速度も増加しているが、それ以降は増加はするものの予想より下回る転送速度となった。TCP ソケットの場合と同様、37 個の並列動作によって CPU などのリソースの競合が原因と考えられる。New-SL の転送速度が高々51.2 Mbps であるのに比べて、RingBuffer は並列動作の環境下でも最大 ∼ 800 Mbps までカバーできるため、TCP ソケットともども十分な性能を有しているといえる。

以上の通信単位の性能評価を踏まえて、エミュレータを用いた収集系には 5 台の login.icepp を用いることとする。TCP ソケットの性能評価から 1 台の login.icepp あたり 18 台の SLEmulator を問題なく動作させることができると述べたが、余裕を持たせるために 3 台の login.icepp01-03 に TTC/SLEmulator を分配する。例えば、Emulator を 19 台用いる場合、login.icepp01 に TTCEmu+SLEmu の計 13 台を配置し、残りの 6 台の SLEmu を login.icepp02 に配置する。図 4.4 から 12 台の場合は 1 台の SLEmulator あたり 800 Mbps 程度まで出力することができるため、試験としては十分である。また、ROSEmulator、Software ROD はそれぞれ login.icepp04、login.icepp05 に配置し、計 5 台のテストベンチマシンで構成する（表 4.2）。

表 4.2: プロセスの配置。例えば、Emulator を 19 台用いる場合、13 台の TTC+SLEmu を login.icepp01 に、残りの 6 台の SLEmu を login.icepp02 に配置する。

テストベンチマシン	配置
login.icepp01	TTCEmu, SLEmu01 ∼ 12
login.icepp02	SLEmu13 ∼ 24
login.icepp03	SLEmu25 ∼ 36
login.icepp04	ROSEmu
login.icepp05	ROD

4.1.4 動作試験

5 台の login.icepp 上で構築した収集系で Software ROD の動作試験を行った。動作試験の項目は以下の 4 つである。

1. 入力要求値に対して TTC + SL を何台までカバーできるか
2. 入力データをどの程度まで処理できるか
3. ランダムなデータサイズで正規の動作を行うことができるか
4. 正しくエラー処理が行われているか

試験 1：Software ROD が収集できる TTC/SL の台数

試験 1 では ROD に対する入力要求値に則って TTC/SLEmulator を動作させ、その状況で Software ROD が何台の TTC+SL をカバーできるか検証する。動作試験の結果を図 4.9 に載せる。図 4.9 は横軸に TTC/SLEmulator の台数、縦軸に ROSEmulator が算出した転

送速度をとったグラフで、TTC/SLEmulator の台数を 3 台ずつ増やしたときの転送速度をプロットしている。また、青色の直線は以下の式に従っている。

$$T = 54.5 \times n + 13.6 + 60.8 \text{ [Mbps]} \tag{4.2}$$

T は転送速度、n は SLEmulator の台数を表し、TTC/SLEmulator の転送速度は補正した値を用いている。TTC/SLEmulator は内部でタイマー機能を用いて擬似的に L1A を発行しているが、その精度はマイクロ秒単位であるため、タイマー機能自身が動作する時間やシグナルを捕獲る時間を加味すると正確には 10 μs にはなっていない。現在はタイマーの間隔を 9 μs に設定しており、残りの 1 μs はタイマー由来のシグナルの発行及び捕獲に割り当てている。シグナルの発行及び捕獲自体はさほど時間がかからず、L1A の発行頻度は 9 μs 台となっているため、第 3 章で述べた転送速度よりも幾分か大きくなっている。TTC/SLEmulator の転送速度を tcpcheck で求めると、それぞれ 13.6 Mbps、54.5 Mbps となっている。つまり、タイマー機能を用いた SLEmulator の出力転送速度は、データサイズに応じた転送速度の 1.06 倍で出力されることがわかる。本項の試験では、データサイズに応じて出力される転送速度に対して補正係数 1.06 を用いることとする。この補正された転送速度を用いて直線を描き、さらに測定値をプロットしたところ、図 4.9 のようにきれいに重なる結果となった。また、図 4.11 はダンプしたデータの一部を表しており、赤色の領域が ROB 入力フォーマット、黄色の領域がデータブロックとなっている。この図から見て取れるように、全てのイベントデータフラグメントはイベントビルディングされており、Software ROD は規定の動作をしていることがわかる。よって、Software ROD はレベル 1 リードアウトスキームで要求されている性能を持ち合わせていることが確かめられた。

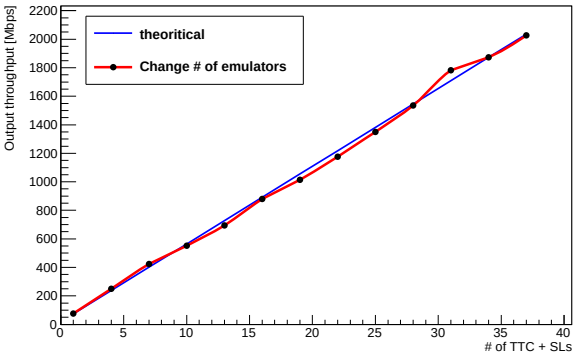

図 4.9: エミュレータの台数を変化させたときの ROD の出力転送速度。入力要求値（TTC : 12.8 Mbps、SL : 51.2 Mbps）に応じてきれいに比例している。

台数	理論値 [Mbps]	測定値 [Mbps]
1	74.4	77.4
4	237.8	250.5
7	401.3	423.1
10	564.7	552.9
13	728.1	694.6
16	891.6	879.8
19	1055.0	1013.9
22	1218.4	1176.3
25	1381.9	1350.5
28	1545.3	1535.3
31	1708.7	1782.3
34	1872.2	1872.6
37	2035.6	2026.6

図 4.10: 台数に応じた転送速度の理論値と測定値。各測定点は 20 回測定した平均値を表す。

図 4.11: ダンプしたイベントデータの一部。ROB 入力フォーマットの間にカウントアップしたデータが格納されているデータブロックが挟まれている。

試験 2：Software ROD の最大処理能力の測定

試験 1 の結果から、Software ROD は 37 台の TTC/SL から入力要求値を満たしながらデータを処理することが可能であるとわかった。ここでは TTC/SLEmulator の台数を固定し、SLEmulator のイベントデータフラグメントのデータサイズを変化させて、Software ROD の転送能力の上限値を測定した。エミュレータの台数を 10 台、13 台、19 台、37 台に固定した場合の試験結果をそれぞれ図 4.12、図 4.13 に載せる。各測定点は 20 回測定した平均値を表す。この 4 パターンは、ATLAS 検出器の片サイドに配置される 36 台の New-SL に対して考えられうる DAQ PC の台数で、DAQ PC が 4 台、3 台、2 台、1 台に対応している。各図の直線は式 4.2 に従って描かれており、各台数において上限に達するまでは、測定値はこの直線に従っていることが見て取れる。

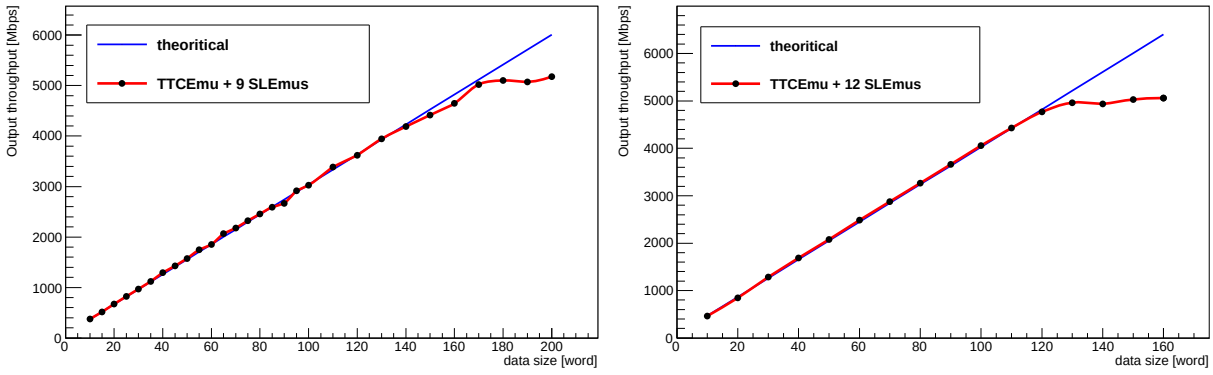

図 4.12: エミュレータを用いたグラフ。左：10 台のエミュレータ、右：13 台のエミュレータ

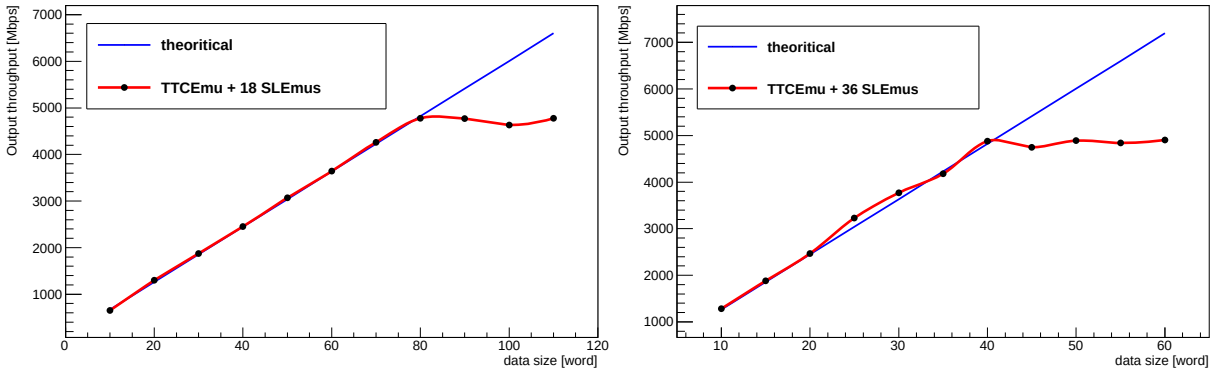

図 4.13: エミュレータを用いたグラフ。左：19 台のエミュレータ、右：37 台のエミュレータ

New-SL の最大データサイズが 16 word = 64 Byte であるのに対して、片サイドを 1 台の DAQ PC でデータ収集してもデータサイズを 40 word = 160 Byte 程度までカバーできることがわかった。

表 4.3: DAQ PC の台数とカバーできる最大データサイズ。少なくとも 160 Byte 程度のデータを処理できる。

DAQ PC の台数 [/side]	データサイズ [Byte]
4	∼ 680
3	∼ 480
2	∼ 320
1	∼ 160

試験 3：ランダムなデータサイズによる動作確認

ここまでは SLEmulator のイベントデータフラグメントのサイズを固定して試験してきた。つまり、1 つのイベントデータに含まれる全てのデータフラグメントがヒットデータであるという、本番の実験とはかけ離れた設定で試験していた。要求された性能以上に Software ROD を酷使しても耐えられる性能があることを試験 1、2 で確認したが、ここでは本番の実験に倣ってデータサイズをランダムにしても正しく動作するか確認した。ヒットデータの出現頻度を 100 回に 1 回程度にし、上限を 100 word としたランダムサイズなヒットデータを送信するように設定したが、データサイズが可変になっても Software ROD 内の収集アルゴリズムにはなんら影響はないので、ここでは簡単に試験結果について述べる。

1/100 程度の頻度でヒットデータが送信されるとすると、平均転送速度は 14.3 Mbps となる。結論としては Software ROD は正しく動作しており、瞬間的な大入力 100 word = 400 Byte にも対応できている。よって、試験 1∼3 の結果から、メインフローにおいて Software ROD が十分な性能を持っているということが確かめられたことになる。

試験 4：エラー処理の確認

試験 1∼3 ではメインフローについて試験してきたが、ここではエラーが正しく処理されているか確認する。主要なエラーの項目を以下に挙げる。

- ROD busy

  ROD busy は Software ROD の処理が追いつかなくなった場合に発行される信号で

ある。Software ROD 内に配置されているバッファはソケットバッファと RingBuffer であることは既に述べた通りであるが、処理のボトルネックがイベントビルディングであることを踏まえると、RingBuffer でシグナルを発生させるべきだと考えられる。RingBuffer は Write offset が Read offset に追い付きそうになったとき、条件変数に wait 操作をかけて RingBuffer に書き込めるだけの空きができるまで待つが、その直前に kill() 関数で ReadoutMaster にシグナルを送信する。ReadoutMaster はスレッドでシグナルをキャッチして所定の処理をするが、現段階ではキャッチするだけに留まっている。

ここでの動作確認は、EventBuilder を秒単位でスリープさせて処理を遅らせ、RingBuffer がフルになる状態を作り出すことで行う。スレッドでシグナルをキャッチしてファイルに書き出すようにしたところ、数 10000 イベント処理したあたりからファイルへ書き出されるようになった。よって、ROD busy は正しく発行されていることがわかった。

- ID チェック

  L1ID 及び BCID が間違っていた場合に正しく処理できているか確認する。まずはじめに BCID であるが、第 3 章で説明したフォーマットには Current BCID が格納されているため、前後 1 バンチ分の BCID のチェックを行う。3 バンチ分の BCID が一致しなかった場合、ROB 入力フォーマットのステータスブロックにエラーフラグを立てる。また、L1ID についても同様で一致しなかった場合は ROB 入力フォーマットのステータスブロックにエラービットを立てる。

  1000 イベントごとに間違った BCID 及び L1ID を混ぜ込んだところ、規定のエラー処理を行って動作を続けた。間違った L1ID、BCID なので、ステータスブロックの 1 要素目の 0 bit 目及び 1 bit 目が反応することになる。他の bit には Hardware ROD 特有のエラーも存在しているため、今後は Software ROD に合わせたエラーを対応させなければならない。

31..15
Specific

Bit	Specific error conditions
0	
1	
2	
3	
4	
7..5	
15..8	

15..0
Generic

Bit	Generic error conditions
0	incorrect BCID
1	incorrect L1AID
2	Timeout occurred in at least one of the FE links. Fragment is incomplete.
3	Data may be incorrect, see TGC ROD event status word (Table 3).
4	An overflow in one of the ROD internal buffers has occurred. The fragment is incomplete.
7..5	*reserved*
15..8	*reserved*

図 4.14: ステータスブロックの 1 要素目 [25]。0 bit 目、1 bit 目に BCID 及び L1ID に関するエラーフラグを立てる。

4.2　汎用 VME モジュールを用いた接続試験

エミュレータを用いた性能評価によって Software ROD は満たすべき条件をクリアしつつ動作をすることが確かめられた。それに続き、本項では汎用 VME モジュールとスイッチングハブを用いて、より本番に近い状況で Software ROD が正しく動作するか検証する。

4.2.1　汎用 VME モジュール PT7

汎用 VME モジュール PT7 は New-SL の試作機として開発されたモジュールで、FPGA と CPLD が 1 つずつ載せられている。FPGA は Xilinx 社製の Kintex-7 XC7K325T-2FFG900、CPLD は同じく Xilinx 社製の CoolRunner-II CPLD XC2C256-7PQ208 が採用されている。PT7 には様々な I/O が FPGA 及び CPLD に接続されており、その中には収集系に必要な Ethernet PHY も含まれている。また、PT7 の側面には TTCrq コネクタがあり、そこに TTCrq を接続することで TTC 信号を受信することが可能となっている [30][31]。

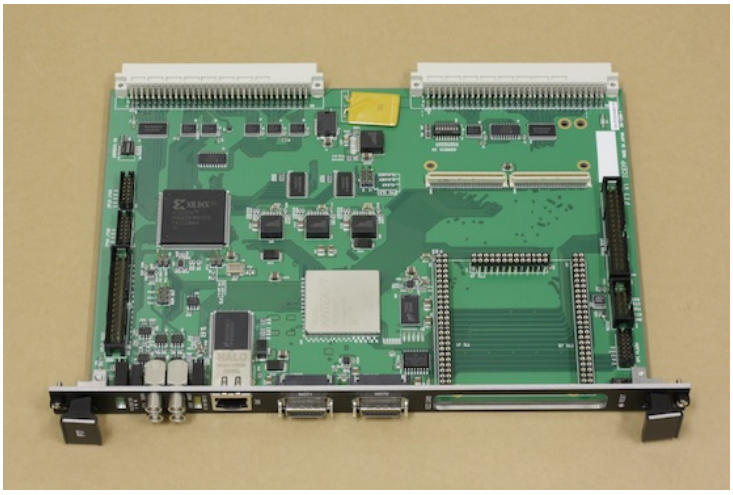

図 4.15: PT7 の全体像 [31]。Ethernet PHY や TTCrq コネクタといった I/O がついている。

4.2.2　テストベンチ及び収集系で用いる機器

汎用 VME モジュールを用いた接続試験のため、高エネルギー加速器研究機構（KEK）にてテストベンチを構成した。主なコンポーネントは、汎用 VME モジュール PT7、スイッチングハブ、イーサネットケーブル、DAQ PC である。PT7 には TTCrx が接続されており、光ケーブルによって TTCex から L1A 信号や BC などを受信するが、KEK には TTCrx 用のケーブルが 2 本しかないため、ここでは 2 対 1 の収集系を構築することとする。

テストベンチ

汎用 VME モジュールを用いた試験では KEK に設置されている elec22 という PC をテストベンチマシンに採用した。elec22 の性能を表 4.4 に示す。elec22 は後述のイーサネットケーブルとスイッチングハブで各 PT7 とつながっており、2 対 1 の収集系を構築している。また、各汎用 VME モジュールをコントロールして TTC 関連のモジュールのパラメータを変更することができる。

図 4.16: テストベンチマシン elec22。KEK で使用する PC である。

表 4.4: login.icepp のパラメータ

パラメータ	性能
CPU	Pertium Dual-Core E6300
コア数	2
動作周波数	2.8GHz
メモリ	20GByte
NIC	1Gbps
OS	Scientific Linux CERN 5.9

スイッチングハブ

BUFFALO LSW4-GT-8NS を用いる。

- ポート：10/100/1000BASE-T×8
- スイッチングファブリック：16Gbps
- データ転送速度：1000BASE-T 1,488,095pps

イーサネットケーブル

3m の Cat.6 ケーブルを用いる。

4.2.3 動作試験

本動作試験では 1 枚の PT7 は TTC 用（TTC PT7）、もう 1 枚の PT7 は New-SL 用（SL PT7）として 2 対 1 の収集系を構築する（図 4.17）。それぞれのロジック開発は本研究と並行して進められており、Software ROD と同様に単体試験を終えている。本試験では 3 つの要素を接続して収集系を構築したとき、単体試験と同様に正規の振る舞いをするか検証する。SL PT7 は実験同様にランダムなデータサイズでデータを生成・送信できることが望ましいが、FPGA 内部でランダムサイズなイベントデータフラグメントを生成するのは難しいため、ここでは SL PT7 のデータサイズを 64Byte に固定して動作させる。動作試験の結果を表 4.18 に示す。

KEK のテストベンチに設置されている TTC 関連のモジュールはソフトウェアから L1A の発行頻度をいくつか選択できるが、仕様上設定した値とは異なる発行頻度で出力される。今回は 100 kHz で出力されるように設定したところ、実際は 140 kHz で出力されていたためこれまで述べてきた転送速度よりも高い値となっている。

図 4.17: 接続試験の様子。PT7 を 2 台とスイッチングハブをイーサネットケーブルでつないで、elec22 でデータを収集する。

図 4.18: 2 対 1 の接続試験の結果。TTC モジュールの仕様により L1A 信号が 140kHz で発行されているため、転送速度が大きくなっている。

	TTC PT7	SL PT7	ROD
転送速度 [Mbps]	18.3	73.5	178.6

Software ROD はプロセス間通信を用いたパイプライン処理のため、今回のように ROD busy を出力しない（今回のテストベンチでは ROD busy シグナルの発行と同時に L1A 信号の発行を止める術がない）場合は ROD 内のバッファがフルになるとその前のモジュールにも影響を与える。具体的には ROD が busy 状態になると、SiTCP サーバが保持している FIFO からデータが溢れることとなる。FIFO からデータが溢れるとイベントデータフラグメントが中途半端な位置で切れてしまうため、Software ROD は処理をし損なう。しかし、この動作試験では最終的に 380M イベント以上のイベントデータを処理することができたため、ROD busy を起こすことなく正しく動作することが確認できたといえる。

4.3 Distributor を用いた性能評価

本章の冒頭でも述べたとおり、現段階では汎用 VME モジュールの枚数は 37 台も存在しないため、リードアウトスキームを再現することはできない。汎用 VME モジュールを使用し、かつ 37 対 1 の収集系を再現するため、ここでは Distributor というデータを分配するプロセスを作成した。

4.3.1 Distributor

Distributor の概念図 4.19 を載せる。Distributor は内部に複数のスレッドを保持し、スレッド間のデータパッシングを共有メモリを介して行っている。スレッドは 1 つの Writer Thread と複数の Reader Thread に分類され、それぞれ SiTCP サーバ及び Software ROD と 1 対 1 通信を行う。各スレッドの動作は単純で、Writer Thread は SiTCP サーバとのコネクションを確立した後にソケットでデータを受信し、共有メモリへ書き込む。Reader Thread は共有メモリに格納されたデータを読み出し、Software ROD へ TCP ソケットを介して送信する。Distributor と Software ROD

はローカルな TCP/IP 通信を行うが、これまでの試験で述べてきたように転送速度 51.2 Mbps に対してそれぞれ十分な性能を有しているので、ローカルな通信をしても問題はないと考えられる。

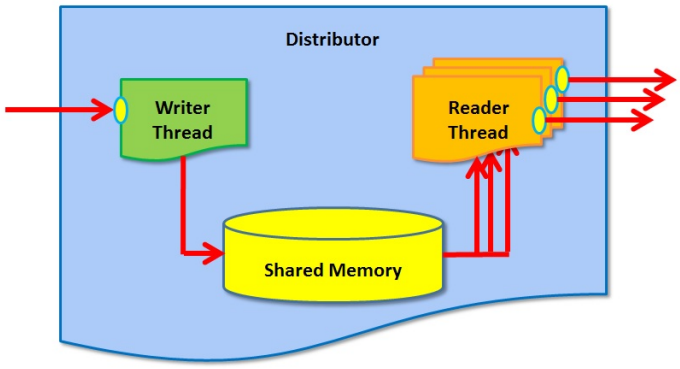

図 4.19: Distributor の概念図。単一の Writer Thread が複数の Reader Thread へデータを配る。

共有メモリの構造を図 4.20 に示す。共有メモリの構造は RingBuffer と同様にコントロール部とデータ部に分けられており、各スレッドはコントロール部の情報を参照しながら書き込み及び読み出しを行う。コントロール部には相互排除変数、条件変数、リードライトロックが埋め込まれており、相互排除変数はコントロール部を保護し、リードライトロックはデータ部を保護する。今回のように共有リソースに対して書き込み及び読み出しが区別できる場合はリードライトロックを用いることができ、リードライトロックによって Reader Thread は同時にデータ部にアクセスしてデータを読み出すことができる。そのため、Reader Thread 間でデータを送信するタイミングがほぼ同時となり、Software ROD の立場で見ると 1 つのデータソースがあたかも複数のデータソースが存在しているかのように見えるという仕組みである。

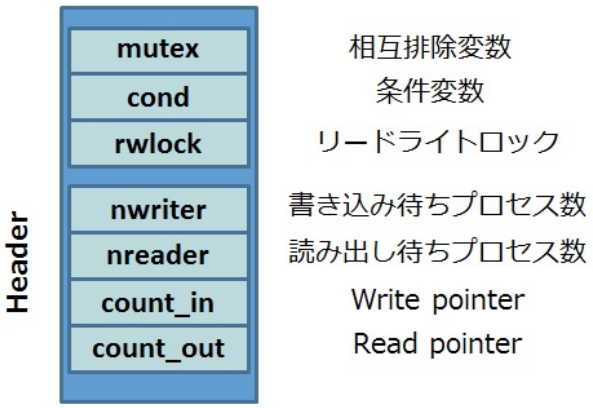

図 4.20: 共有メモリの構造。相互排除変数、条件変数、リードライトロックを用いてメモリ全体の保護を行う。

動作確認

汎用 VME モジュールを用いた動作試験に導入する前に、Distributor が I/O レートを 100 %に維持したままどの程度分配できるかを調べる必要がある。入力には本章で使用した SLEmulator を 1 つ、出力先には tcpcheck を複数配置して動作確認を行う。Reader Thread の数を変化させたときに、各 Reader Thread からの転送速度が SLEmulator の転送速度と同程度であれば正しく動作しているといえる。動作確認の結果を表 4.5 に示す。結論としては、Distributor は 36 個の分岐に対しても I/O レートを 100 %に保ったまま通信できることがわかった。つまり、2 枚の汎用 VME モジュールが（1 枚は TTC 用、もう 1 枚は SL 用）あればリードアウトスキームを擬似的に再現できる性能を有しているといえる。表は 36 台のうち 4 台をピックアップしたものだが、その他の Reader Thread も ～ 73Mbps 程度で、同様に I/O レートを 100 %に保ったままデータを分配することができている。

表 4.5: Distributor の動作確認の結果。その他の Reader Thread も同様な転送速度である。

スレッド	Writer	Reader			
スレッド番号		1	12	24	36
転送速度 [Mbps]	73.1	72.8	73.1	72.9	72.7

4.3.2 動作試験

先述の汎用 VME モジュールを用いた収集系に対して、ここでは DAQ PC 内部に Distributor を挟むことで擬似的に 3 台以上と通信をする。テストベンチマシン elec22 の共有メモリ用に割り当てられるメモリは 4 GByte で RingBuffer の作成は 24 個までとなるため、本動作試験では 10 台、13 台、19 台の転送速度を測定した。動作試験の構成を図 4.21 に示す。

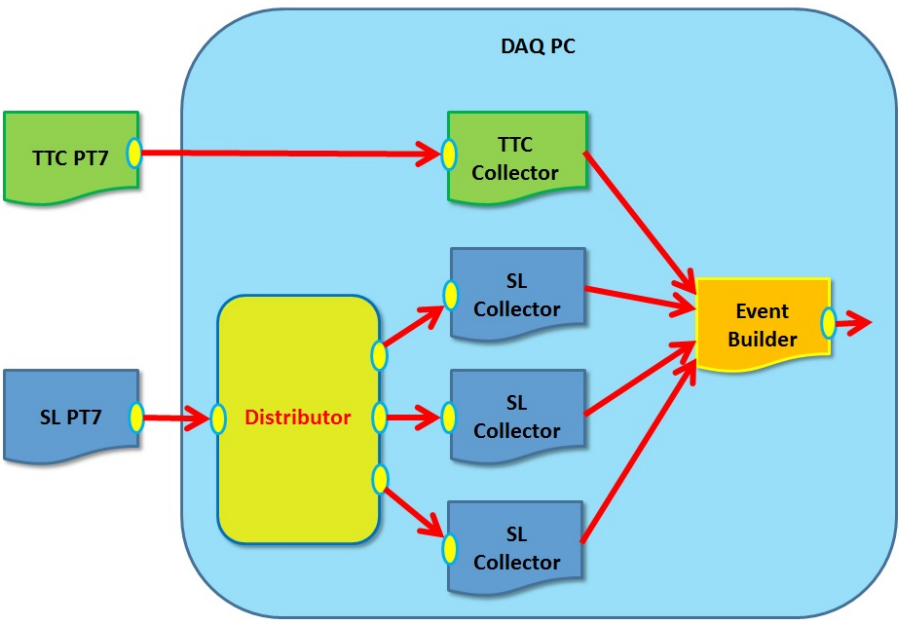

図 4.21: Distributor を用いた接続試験。Distributor で SL PT7 からのデータを分岐させている。

先ほどと同様に TTC PT7 は ∼ 18Mbps で出力し、SL PT7 は ∼ 73 Mbps で出力した場合の結果を図 4.21、表 4.23 にまとめる。転送速度は Distributor の分岐数（+TTC1 台）にきれいに比例しており、エミュレータを用いた試験と同様の結果が得られた。擬似的な収集系ではあるが、汎用 VME モジュールとの多対 1 通信でも正しく動作していることが確認できた。

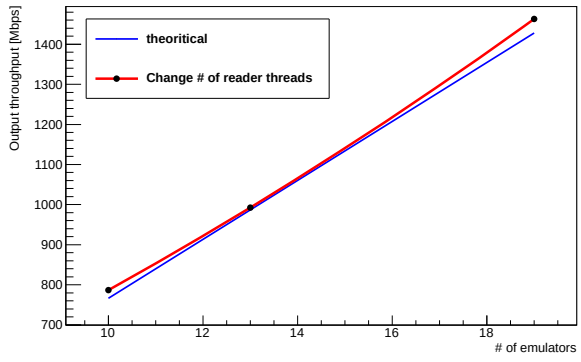

図 4.22: 分岐数と転送速度の関係。分岐数は Reader Thread の数（＋TTC）に対応している。

Distributor の分岐	転送速度 [Byte]
9(+1)	786.9
12(+1)	992.4
18(+1)	1463.0

図 4.23: Distributor を用いた接続試験の結果。表中の (+1) は TTC1 台分を表している。

4.4 結論

本章ではエミュレータ及び汎用 VME モジュール PT7 を用いて Software ROD の性能評価を行った。

エミュレータを用いた動作試験で得られた結果は第 3 章で述べた要求を十分に満たしており、Software ROD の有用性が確かめられた。具体的には、1 台の TTCEmulator 及び 6 台の SLEmulator からデータを収集して、出力転送速度 380.8 Mbps で動作することが最低限満たすべき要求であったが、動作試験の結果から 1 台の TTCEmulator と 36 台の SLEmulator からデータを収集することができ、最大で ∼ 5.0 Gbps の転送能力を有することがわかった。

また、KEK でテストベンチを構築して本番により近い状況で動作試験を行った。2 台の汎用 VME モジュールに TTC 及び New-SL のリードアウトロジックをダウンロードし、各 VME モジュールと DAQ PC をスイッチングハブ経由で接続したところ、エミュレータの試験と同様に、正しく動作することが確かめられた。これは、適切な DAQ PC やその他の周辺機器を選ぶことにより、Run-3 におけるリードアウトスキームが実現可能であることを示唆している。

今後の課題としては、Software ROD を使用するためのインフラストラクチャの整備、エラー処理、外部システムとの接続について考えていく必要がある。

第5章　まとめ

ATLAS 実験は HL-LHC に応じて段階的にアップグレードされる。TGC で起きるフェイクトリガーを削減するため、Run-3 からは ATLAS 検出器の内側に New Small Wheel が導入される予定である。New Small Wheel の導入により TGC が処理すべきデータが増加するため、それに応じてリードアウトスキームもアップグレードする必要がある。

本研究では Run-3 に向けたリードアウトスキームにおける ReadOut Driver のアップグレードを行い、現行のハードウェアベースの ReadOut Driver に代わる、ソフトウェアをベースとしたプロセス間通信による ReadOut Driver を開発した。

エミュレータを用いた試験によって次のことがわかった。ハードウェアベースの ReadOut Driver は 1 台で 1/12 セクター分のデータ処理を行うのに対して、ソフトウェアベースの ReadOut Driver は DAQ PC1 台で片サイド分のデータ量を捌くことができる。具体的な数値としては、新しい TTC モジュール 1 台、新しい SL36 台からの入力要求値 1843.2Mbps に対して、ATLAS フォーマットを考慮した転送速度 1904.0Mbps を達成しており、最大でも ∼ 5000 Mbps まで処理できる。ただし、スイッチングハブの耐久性や帯域幅といった物理的な要素は除いている。

また、汎用 VME モジュールとの接続試験を行い、実際にスイッチングハブやイーサネットケーブルを使用して収集系を構築した。接続試験では新しい TTC 及び SL のロジックを搭載した汎用 VME モジュールを 1 台ずつ使用して 2 対 1 とし、L1A の発行頻度を 140kHz としてデータを収集することができた。具体的には TTC から 18.3 Mbps、SL から 73.5 Mbps の転送速度で ReadOut Driver に入力するのに対し、ROB 入力フォーマット分 85.12 Mbps を加味して ReadOutDriver から 178.6 Mbps で出力することができた。接続試験により、スイッチングハブ及イーサネットケーブルを用いて少なくとも 2 対 1 の収集系が構築できることがわかった。

参考文献

[1] ATLAS Collaboration: Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC, 2012

[2] CMS Collaboration: Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC, 2012

[3] Latest Results from ATLAS Higgs Search
http://www.atlas.ch/news/2012/latest-results-from-higgs-search.html

[4] CERN Home Page
http://home.web.cern.ch/

[5] CERN: LHC Design Report, 2004.

[6] CERN Document Server / Photos
http://cds.cern.ch/collection/Photos

[7] ATLAS: The ATLAS Experiment at the CERN Large Hadron Collider, 2008.

[8] ATLAS: Inner Detector Technical Design Report, 1997.

[9] ATLAS: Muon Spectrometer Technical Design Report, 1999.

[10] ATLAS: New Small Wheel Technical Design Report, 2013

[11] ATLAS Photos
http://www.atlas.ch/photos/index.html

[12] ATLAS Detector Description
http://www.atlas.ch/detector.html

[13] TOTEM Home Page
http://totem.web.cern.ch/Totem/

[14] LHCf Home Page
http://hep.fi.infn.it/LHCf/

[15] TTC web page
http://ttc.web.cern.ch/ttc/

[16] James Stirling: Tevatron and LHC parton luminosity comparison plots
http://www.hep.phy.cam.ac.uk/~wjs/plots/plots.html

[17] ATLAS: Letter of Intent for the Phase-I Upgrade of the ATLAS Experiment, 2012.

[18] Revised Muon Endcap chapter of the Level-1 Trigger TDR, Rev01, 2000

[19] TGC document
https://twiki.cern.ch/twiki/bin/view/Main/TgcDocument

[20] Osamu Sasaki: Amplifer-Shaper-Discriminator ICs and ASD boards, 1999

[21] Patch-Panel ASIC, 2005

[22] R.Ichimiya, H.Kurashige, R.Kurumida: Design of the TGC Sector Logic for the ATLAS Endcap Muon Trigger System, 2013.

[23] ATLAS-Japan TGC electronics group: Slave Board ASIC Technical Document, 2005.

[24] Hiroshi Nomoto: Star Switch Spec, 2009.

[25] TGC ROD web page
https://atlas-proj-tgc.web.cern.ch/atlas-proj-tgc/ROD/TGCROD.html.

[26] CERN S-LINK Home Page
http://hsi.web.cern.ch/HSI/s-link/

[27] Yu Suzuki: Upgrade of the ATLAS Level-1 Muon Trigger for the High Luminosity LHC, 2013.

[28] 徳永孝之:LHC-ATLAS 実験 RUN3 に向けた新しいミューオントリガー装置の FPGA 読み出し開発とその性能評価, 2015

[29] 鈴木翔太: LHC-ATLAS 実験における Thin Gap Chamber での TTC System の評価と RUN3 に向けた TTC signal readout module の開発, 2015

[30] 大谷育生: ATLAS ミューオントリガーシステムのアップグレードに向けた ギガビット通信を用いたデータ収集系の検証及び読み出しプロトタイプの開発, 2013

[31] 加藤千曲: ATLAS 実験のための新しいミューオントリガー装置の研究開発, 2014

[32] 神谷隆之: ATLAS 前後方ミューオントリガーシステムのアップグレードに向けた読み出し系システムインフラストラクチャの研究開発 , 2011.

[33] 二ノ宮陽一: High-luminosity LHC へ向けた ATLAS ミューオントリガー検出器用読み出しシステムの開発 , 2011.

[34] 来見田吏恵: ATLAS 実験 RUN2 に向けたレベル 1 ミューオントリガーへの新しいアルゴリズム導入, 2014.

[35] 田代拓也: ATLAS 実験における新しいミューオントリガー回路の開発と実装, 2012.

[36] 桑原隆志: ALTAS 前後方ミューオントリガーシステムの構築, 2007

[37] 今野智之: Double Chooz 実験のためのオンラインモニターシステムの開発, 2009

[38] 門坂拓哉: ATLAS 前後方ミューオントリガーシステム Sector Logic 及びオンラインソフトウェアの開発, 2008.

[39] 平山翔: ATLAS 実験前後方レベル 1 ミューオントリガーシステムの構築と検証 , 2009.

[40] 鈴木雄太: Look Up Table 改良による ATLAS レベル 1 ミューオントリガーの性能向上の研究, 2012.

[41] 徳宿克夫: LHC/ATLAS アップグレードの展望, 2011.

[42] 中本建志: LHC アップグレード計画 (HL-LHC, LIU) と日本の貢献, 2012.

[43] 佐々木修: Muon Endcap Trigger Updates, ATLAS Upgrade Week 2012.

[44] 佐々木修: Muon Endcap Trigger in Phase I, New Sector Logic Board, 2013

[45] 内田智久: Hardware-Based TCP Processor for Gigabit Ethernet, 2008.

[46] 前田順平: Level-1 Endcap muon trigger for Run-3, 新学術領域研究会テラスケール 2014 先端加速器 LHC が切り拓くテラスケールの物理, 2014

[47] 片岡洋介: New Small Wheel Upgrade, 新学術領域研究会テラスケール 2014 先端加速器 LHC が切り拓くテラスケールの物理, 2014

[48] 隅田土詞: Level-1 Muon Trigger 全体像と現状, 新学術領域研究会テラスケール 2014 先端加速器 LHC が切り拓くテラスケールの物理, 2014

[49] W. Richard Stevens 著 篠田陽一　訳:UNIX ネットワークプログラミング 第 2 版 Vol.2 IPC: プロセス間通信, 2001.

[50] 河野清尊:C 言語による UNIX システムプログラミング入門, 2006.

[51] 三苫健太: オブジェクト指向開発トレーニングブック, 2009

謝辞

本論文を執筆するにあたり、多くの方々からご指導、ご鞭撻を賜りました。この場をお借りして感謝の意を表します。

まずはじめに、このような素晴らしい研究の機会を与えて下さった指導教官の坂本宏教授には心から感謝を申し上げます。未熟な自分を根気強く支えて下さり、ソフトウェアを中心に研究に対して丁寧なご指導を賜りました。また、研究の他にも普段の研究生活、CERN 出張、学会発表などの様々な場面で支えて頂きました。改めて、感謝申し上げます。

高エネルギー加速器研究機構の佐々木修教授には、会議で数多くのアドバイスや知識を頂きました。また、神戸大学 前田順平特命助教には本論文の執筆に至るまで、大変親身になって面倒を見て頂きました。特に、CERN 滞在時には何度も質問させて頂き、ソフトウェアに関する基礎を丁寧にご指導頂きました。深く感謝致します。さらに、首都大学東京 特別研究員の今野智之氏には、同氏の開発したソースコードから学ばせて頂きました。感謝申し上げます。

TGC グループの藏重久弥氏、内田智久氏、戸本誠氏、隅田土詞氏、石野雅也氏、青木雅人氏、堀井泰之氏を始めとした方々には研究に対するご助力を頂き、多くの刺激を得ることができました。深く感謝致します。また、田代拓也氏、山内克弥氏、樋口浩太氏、小林大氏、来見田吏恵氏を始めとした他大学の諸先輩方には CERN での生活をよりよいものにして下さいました。

研究室では結束晃平氏、二ノ宮陽一氏、大谷育夫氏、加藤千曲氏に大変お世話になりました。研究は勿論のこと、国内・CERN での研究生活や研究室での行事など、様々な場面でご指導頂きました。日頃の感謝の意を申し上げます。

同期の皆様のおかげで研究生活が大変充実したものとなりました。安達俊介氏、家城斉氏、荻野真由子氏、寺尾伸吾氏、徳永孝之氏、久島真悟氏、嶺岸優司氏、森拓晃氏、山道智博氏、山本遼氏の皆様には感謝の意を表します。また、他大学同期の鈴木翔太氏、救仁郷拓人氏、長谷川誠氏とは TGC グループとして研究に励んで参りました。感謝致します。

最後に、有意義な学生生活を送らせてくれた家族に感謝の意を述べ、謝辞とさせて頂きます。