

# 修士学位論文

## LHC-ATLAS実験Run-3における 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開発

2017年2月3日

専攻名	物理学専攻
学籍番号	152s116s
氏名	谷岡 凌

神戸大学大学院理学研究科博士課程前期課程



## 概要

Large Hadron Collider (LHC) は欧州原子核研究機構 (CERN) に建設された大型陽子・陽子衝突型加速器である。この LHC は 2015 年に重心系エネルギー 13 TeV で Run-2 という期間として運転が再開された。この Run-2 は 2018 年まで続き、その後 2020 年からの Run-3 に向けたルミノシティ向上のためにアップグレード期間、Phase-1 upgrade 期間に入る。

LHC に設置された大型汎用検出器の 1 つである ATLAS 検出器は陽子同士の衝突から未知の物理を探索することを目的としている。Phase-1 upgrade の期間には、この ATLAS 検出器においてもより精密な物理事象探索のためにシステムの大幅な改良を行う。

Run-3 からは新しいミューオン検出器である New Small Wheel (NSW) を導入するため、トリガー及びデータ取得のシステムを大きく変更することが決定した。本論文ではミューオントリガー判定装置である New Sector Logic (NSL) からのデータの収集を行う Readout Driver(ROD) を開発した。

新しい ROD はプロセス間通信を用いたソフトウェアベースであり、本論文ではこのソフトウェアベースの ROD (SROD) の検証実験について述べている。検証実験には、SROD の基本的な性能をエミュレータを用いた試験を行ったほか、NSL を含む Run-3 から導入される他のシステムとも接続して実際の環境でテストビームを用いた試験を行った。

これらの試験を通して、イベントデータの収集を行えることを実証した。

# 目次

第 1 章	序論	1
第 2 章	LHC と ATLAS 検出器	4
2.1	LHC	4
2.2	ATLAS 実験	5
2.2.1	ATLAS 検出器の構成	6
2.2.2	ミューオンスペクトロメータ	7
2.2.3	ヒッグス粒子の物理	9
第 3 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	12
3.1	ATLAS におけるトリガー及びデータ収集システム	12
3.2	レベル 1 エンドキャップミューオントリガー	15
3.2.1	Thin Gap Chamber	15
3.2.2	トリガー判定の流れ	17
3.2.3	エレクトロニクス	17
3.2.4	TTC	25
3.2.5	ミューオントリガーシステムの問題点	26
3.3	ミューオン検出器のアップグレード	28
3.3.1	NSW	28
3.4	エレクトロニクスのアップグレード	32
第 4 章	TDAQ system	36
4.1	TDAQ Application	36
4.2	Information Service	38
4.3	Error Reporting Service	38
4.4	Object Kernel Support	40
第 5 章	ソフトウェアベースの Readout Driver の開発	42
5.1	SROD の開発の目的	42
5.2	Run-3 におけるリードアウトスキーム	42
5.2.1	10 GbE スイッチ	44
5.2.2	PC	44
5.2.3	PCI express card	44
5.3	SROD に要求される性能	45
5.3.1	TTC, SL からの入力フォーマット	46
5.3.2	SROD で作成する出力フォーマット	47
5.4	システムの構成	50
5.4.1	TTC/SL Collector	51



5.4.2	EventBuilder	55
5.4.3	ReadoutMaster	57
5.4.4	MessageLogger	57
5.5	システムの実装	59
5.5.1	ネットワークを経由したプロセス間通信	59
5.5.2	同一マシン内のプロセス間通信	59
5.5.3	ロック機構	62
<b>第 6 章</b>	<b>ReadoutDriver の性能評価</b>	<b>64</b>
6.1	Test Beam による性能評価	64
6.1.1	Test Beam の概要、セットアップ	64
6.1.2	Test Beam の試験結果	69
6.1.3	High Rate の試験	71
6.2	処理速度の最適化	71
6.2.1	エミュレータ	71
6.2.2	バッファサイズの最適化	74
6.2.3	処理速度の測定	78
<b>第 7 章</b>	<b>まとめ</b>	<b>81</b>
	参考文献	v

# 第1章 序論

欧州原子核研究機構 (CERN) にある Large Hadron Collider (LHC) は世界最大の陽子・陽子衝突型加速器である。LHC ではヒッグス粒子の精密測定や超対称性等の標準理論を超える新物理の発見のために最大ルミノシティを徐々に上げて運転を行っている。図 1.1 に LHC の 2015 年以降の運転計画を載せる。実験中の期間を Run と呼び、アップグレードのためにビームを停止させている期間のことを Long Shutdown (LS) と呼ぶ。Run は 2011 年から始まり、順番に Run-1、Run-2、Run-3 と呼ぶ。LS は 2019 年から LS-2、2023 年から LS-3 が予定されており、それぞれ Phase-1 upgrade、Phase-2 upgrade と呼ばれる。表 1.1 に各 Run における重心系エネルギー及び最大ルミノシティを載せる。最初の運転である Run-1 は 2011 年から始まり、重心系エネルギー 8 TeV、最大ルミノシティ  $0.6 \times 10^{34} \text{cm}^{-2}\text{s}^{-1}$  で運転し、ヒッグス粒子の発見と、そのスピンや崩壊比の検証につながった。2015 年から現在運転中の Run-2 では重心系エネルギーを 13 TeV に、最大ルミノシティを  $2.0 \times 10^{34} \text{cm}^{-2}\text{s}^{-1}$  に上げ、積分ルミノシティ  $100 \text{fb}^{-1}$  を目指して衝突を行っている。この Run-2 は 2018 年まで続き、その後 2021 年からの Run-3 に向けたアップグレード期間に入る。2021 年から予定されている Run-3 では最大ルミノシティを  $3.0 \times 10^{34} \text{cm}^{-2}\text{s}^{-1}$  に上げて、2023 年までの積分ルミノシティ  $300 \text{fb}^{-1}$  を目指し運転を行う予定である。2026 年からは加速器・測定器を大幅にアップグレードさせた HL-LHC を計画し、最大ルミノシティ  $5.0 \times 10^{34} \text{cm}^{-2}\text{s}^{-1}$  で、10 年間の積分ルミノシティ  $3000 \text{fb}^{-1}$  を目指して衝突を行う予定である。

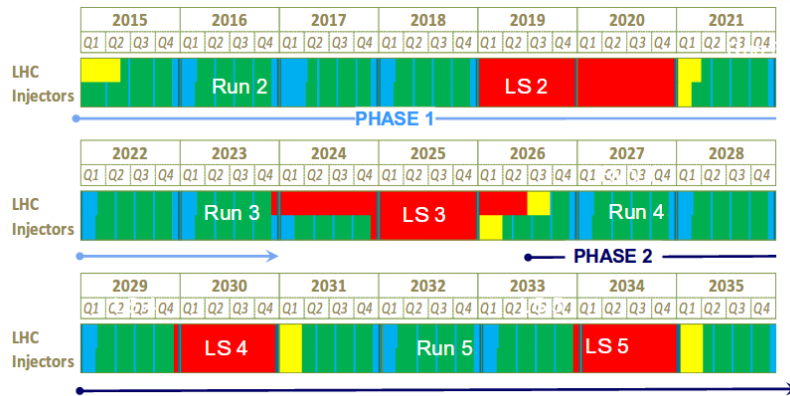


図 1.1: LHC の運転スケジュール [1]: 図中緑が運転期間を表し、赤がシャットダウン期間を表す。

A Troidal Large ApparatuS (ATLAS) 実験は LHC における主要な検出器の一つである。ATLAS 実験が目的としている物理事象は全体の事象数に比べて非常に少なく、例えばヒッグス粒子の場合、全体の物理事象のうち  $10^{-10}$  程度の割合でしか生成されない [2]。膨大な陽子・陽子衝突事象の中から稀な事象を見つけ出すために、ATLAS 実験ではトリガーと呼ばれる事象選別処理を行っている。トリガーによって目的でない物理事象を排除し、興味のある物理事象のみを選別してデータ収集を行っている。トリガーは段階的に実装されていて、レベル 1 トリガー

表 1.1: LHC の各 Run に対する重心エネルギーと最大ルミノシティ

Run	期間	重心系エネルギー [TeV]	最大ルミノシティ [ $\text{cm}^{-2}\text{s}^{-1}$ ]
Run-1	2011–2012	8	$0.6 \times 10^{34}$
Run-2	2015–2018	13–14	$2.0 \times 10^{34}$
Run-3	2021–2023	14	$3.0 \times 10^{34}$
HL-LHC	2026–	14	$5.0 \times 10^{34}$

で最初に 40.08 MHz のバンチ衝突から 100 kHz まで減らし、次のハイレベルトリガーで 1 kHz までイベントを削減する。

ミュオントリガーでは高い横方向運動量  $p_T$  を持つミュオンが生成される事象を選出する。ミュオンは透過力が高く識別が容易である。陽子・陽子衝突事象のほとんどは、QCD 反応で  $p_T$  の低いジェットが生成され、高い  $p_T$  を持つミュオンは少ない。しかし、W,Z などのウィークボソンを終状態に持つ事象は、標準模型の精密測定に重要で、高い  $p_T$  のミュオンを伴うことがある。超対称理論、余剰次元などの標準模型を超える理論においても、高い  $p_T$  のミュオンを伴うことがある。以上のことから、高い  $p_T$  のミュオンを持つ事象を選び出すためのミュオントリガーが重要である。

しかし、Run-1 においてレベル 1 エンドキャップミュオントリガーシステムでは、外側の Big Wheel (BW) だけで判断していたため、衝突由来でない荷電粒子でトリガーを発行してしまう、フェイクトリガーという問題を抱えている。フェイクトリガーを削減するために Run-2 では内部のミュオン検出器である Small Wheel (SW) やタイルカロリメータとのコインシデンス処理を追加しているが、現在のトリガーシステムのままだと Run-3 におけるイベントレートでは十分にフェイクトリガーを削減できず、レベル 1 トリガー全体の発行頻度 100 kHz のうち、ミュオントリガーの発行頻度が 28 kHz を占め、トリガーレートを圧迫してしまう。

そのため、Run-3 に向けた Phase-1 upgrade では SW を新たなミュオン検出器である New Small Wheel (NSW) という検出器に入れ替える。NSW になることで位置分解能が向上し、軌跡を測定し衝突点からのミュオンであることを確認できるため、トリガーレートを 13 kHz まで削減することが期待される。

NSW を導入するために、ミュオントリガーシステムでは Phase-1 upgrade の際にエレクトロニクスを大幅にアップグレードする。1 つはトリガー判定を行う Sector Logic を、NSW からの情報を受け取れるように New Sector Logic (NSL) として新たに開発する。そして NSL のトリガー情報を収集する ReadOut Driver (ROD) のアップグレードも行う。本論文ではこの ROD の開発について述べる。

トリガー情報の収集は、Run-2 までは Star SWitch というデータ収集のための専用ハードウェア及びそれらのデータを収集するための ROD により、ハードウェアで処理していた。しかし、これらのモジュールの開発から 10 年以上経過し、処理速度の面でも、使用する電子部品の欠如という点でも同様のものを製作できず、NSL の読み出しの再開発を行う必要があった。

そこでネットワーク機器の発展を考え、NSL からのデータ収集に市販の 10 Giga bit Ethernet (GbE) ネットワークスイッチを用いることを考えた。その場合、NSL との通信は Ethernet による TCP/IP を用いることになるが、KEK Open-It が開発した SiTCP を使用することでハードウェアから TCP/IP で通信を可能になる [3]。したがって、ROD は従来のような専用のハードウェアではなく、市販の Linux PC 上でのソフトウェア・ベースでデータ通信及びイベントビルディングといった機能の実装が行えると考えた。

ソフトウェアベースの ROD (SROD) では、レベル 1 トリガーの発行頻度である 100 kHz で

ATLAS 検出器全体の同期をとる Timing Trigger and Control system (TTC) と ATLAS の 12 台の NSL からデータを集集し、予め決めてあるフォーマットに従ってイベントビルドを行い、平均 194.0 Mbps の処理速度でイベントデータを次のシステムである ReadOut System (ROS) へ送信することが要求される。またエラーのチェックや処理が間に合わなくなってしまうときの BUSY ハンドリングが要求される。これを満たすために、近年の CPU に搭載されているマルチコア技術を活用し、ROD の処理を機能別に分割し、マルチプロセスで実装する。そのうえでプロセス間の通信として共有メモリを使用することで、高速な処理を実現する。先行研究ではこの SROD の原理的な動作は確認されているが [4]、本研究では SROD を ATLAS のフレームワークである Trigger and Data Acquisition System (TDAQ System) の中の Run Control Driver (RCD) のソフトウェアの枠組みに従って実装した。

そして 2016 年 10 月に SROD をはじめ Phase-1 のシステムを用いて CERN SPS H8 ビームラインを用いたビームテストを行った。本試験において、SROD を TDAQ System に組み入れ、NSL との初めての接続試験を行い、データを取得することに成功した。

さらにビームテストを通して SROD の中で用いられるバッファのサイズが処理速度に大きく依存することを発見し、その変化を調べることで SROD の処理速度の高速化を図り、TTC と NSL のデータを擬似的に作成して SROD に送るソフトウェアを用いて SROD の処理速度を測定した。

本論文では、2 章で LHC と ATLAS 実験について説明し、3 章で ATLAS 実験におけるミューオントリガー及びそのアップグレードについて説明し、4 章で ATLAS 実験の TDAQ システムについて説明し、5 章で Run-3 に向けた SROD について述べ、6 章で SROD の性能評価について述べる。

## 第2章 LHCとATLAS検出器

### 2.1 LHC

Large Hadron Collider (LHC) はスイス・ジュネーブ郊外の欧州原子核研究機構 (CERN) の地下 100 m に建設された世界最大の陽子・陽子衝突型加速器である。図 2.1 にその全体像を示す。LHC は、2000 年まで稼働していた Large Electron Positron collider (LEP) と呼ばれる大型電子・陽電子衝突型加速器で使用されたトンネルを再利用して建設された。陽子は電子よりも約 1800 倍の質量を持ち、加速によって生じるシンクロトロン放射は加速される粒子の質量の 4 乗に反比例するため、陽子を加速させる LHC のほうが電子を加速する LEP よりも加速時のエネルギー損失が極めて少ない。そのため、LHC では LEP よりもはるかに高いエネルギー領域での衝突実験を行うことができる。

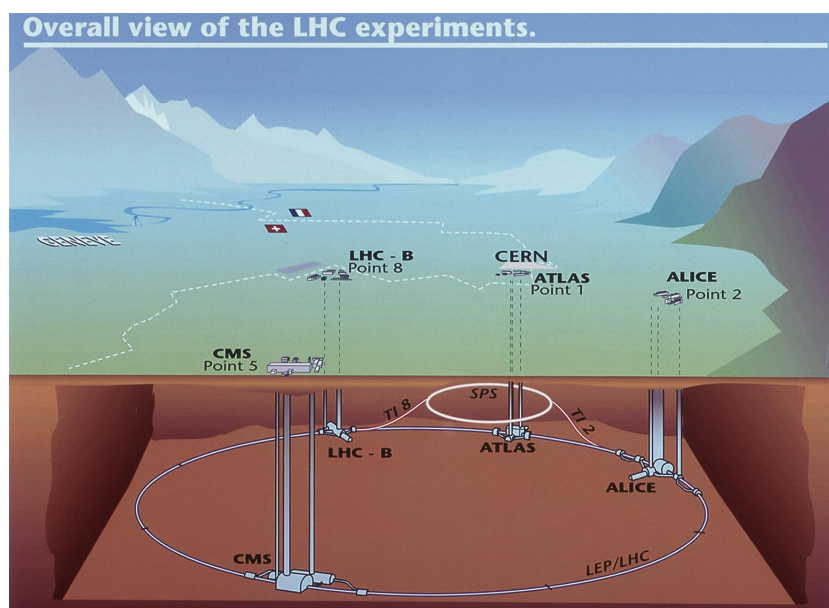


図 2.1: LHC 加速器の全体像 [5]

LHC は 2011 年に最初の運転を重心系エネルギー 7 TeV で 2 年間行い、その後しばらくの間停止していたが、2015 年から運転を再開した。LHC の現在の運転における主要なパラメータを表 2.1 に載せる。

LHC には陽子同士の衝突点が 4 箇所あり、それぞれの衝突点に検出器が設置されている。以下に代表的な実験を示す。

- A Troidal LHC ApparatuS (ATLAS)  
新粒子・新物理の探索を目的とした衝突型実験を行っている。ATLAS 実験についての詳細は後述する。
- Compact Muon Solenoid (CMS)  
ATLAS 実験と同様に新粒子・新物理の探索を目的とした衝突型実験を行っている。検出

表 2.1: 現在の LHC の主要パラメータ

パラメータ	設計値
周長	27 km
バンチ間隔	25 ns
バンチ衝突頻度	40.08 MHz
重心系エネルギー	13 TeV
最大ルミノシティ	$2.0 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$

器全体の大きさは ATLAS 検出器に比べて小さいが、より強力な磁場を持つソレノイドマグネットが用いられている。

- A Large Ion Collider Experiment (ALICE)  
重イオンを用いた衝突型実験で、鉛原子核を加速させた、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の解明を目的とした実験を行っている。
- LHC beauty (LHCb)  
陽子・陽子衝突の際に前方に多く生成される b クォーク及び c クォークの崩壊事象を観測・検証することで CP 対称性の破れ等を観測し、標準模型を超える物理の探索を目的としている。

図 2.2、図 2.3、図 2.4 にこの 4 つの実験で用いられる検出器のうち、それぞれ CMS、ALICE、LHCb の外観を載せる。

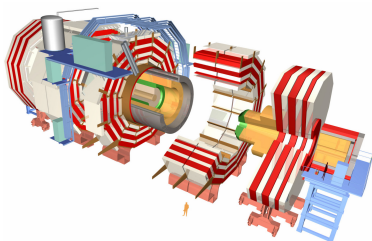


図 2.2: CMS 検出器 [6]



図 2.3: ALICE 検出器 [7]

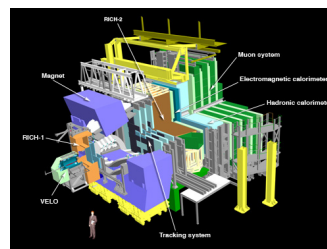


図 2.4: LHCb 検出器 [8]

## 2.2 ATLAS 実験

本節では ATLAS 実験について記述するが、その前にここで ATLAS 実験における座標系について説明しておく。ATLAS 実験では図 2.5 に示すような、衝突点を原点とした直交座標を用いており、LHC の円の中心方向を  $x$  軸、ATLAS 検出器の天頂方向を  $y$  軸にとり、そこから右手系を成す方向に  $z$  軸をとる。この  $z$  軸はビーム軸と一致する。このとき  $z > 0$  の領域を A Side、 $z < 0$  の領域を C Side と呼んでいる。また図 2.6 に示すとおり、 $xy$  平面における極座標  $r, \phi$  や、原点からの天頂角  $\theta$  から得られる擬ラピディティ  $\eta = -\ln(\tan(\theta/2))$  というパラメータも用いられる。 $|\eta| < 1.05$  の、 $z$  軸に対して平行に検出器が配置されている領域をバレル部と呼び、 $1.05 < |\eta| < 2.4$  の  $z$  軸に対して垂直に検出器が配置されている前方の領域をエンドキャップ部と呼び、 $2.4 < |\eta|$  の前方領域をフォワード部と呼ぶ。



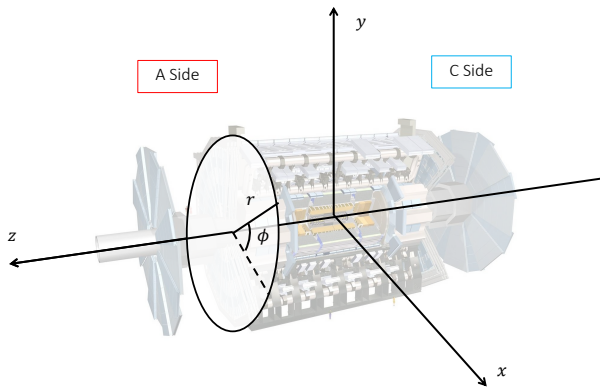


図 2.5: ATLAS 検出器の座標系

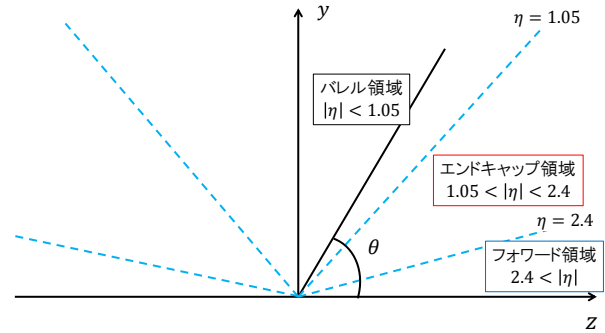


図 2.6: 擬ラピディティ

## 2.2.1 ATLAS 検出器の構成

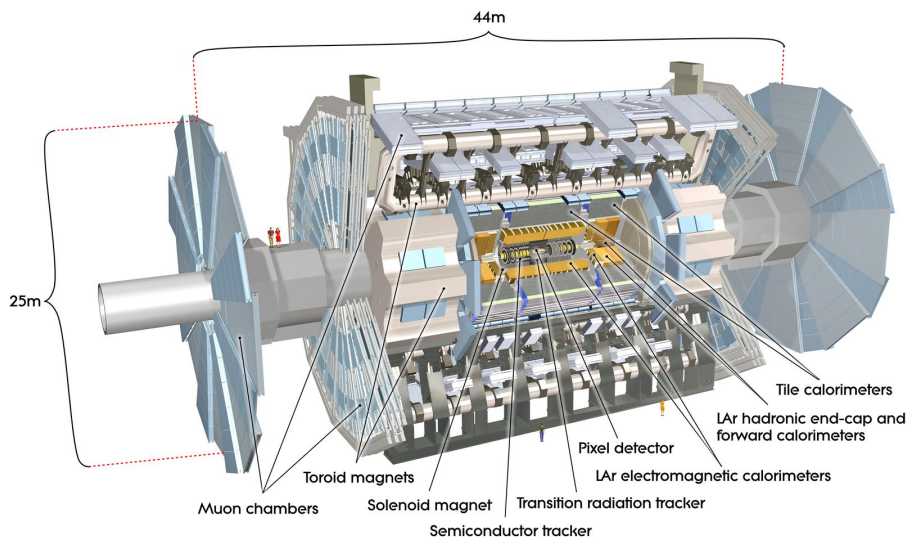


図 2.7: ATLAS 検出器の全体像 [5]

ATLAS 検出器は図 2.7 に示すような高さ 25 m、長さ 44 m の円筒形をした大型汎用検出器である。ATLAS 検出器は陽子の衝突の際に発生する様々な粒子を識別するために、複数の検出器で構成されている。図 2.8 に複数の粒子に対する検出概念図を示す。

- 内部飛跡検出器  
内部飛跡検出器はピクセル検出器、シリコン検出器、遷移輻射検出器から構成される。ソレノイドマグネットの磁場によって曲げられた荷電粒子の飛跡を再構成し、電荷と運動量を測定する。
- 電磁カロリメータ  
液体アルゴン中に鉛の板を並べたカロリメータで、液体アルゴン中での電離電子を捉えて、電磁シャワーを検出することで電子と光子のエネルギー及び飛来方向を測定する。

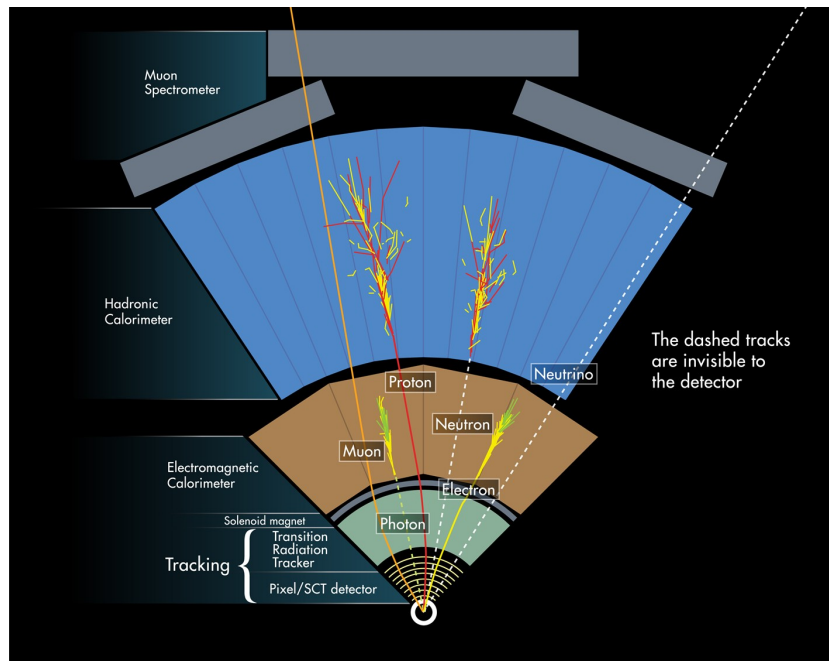


図 2.8: 素粒子に対する検出概念図 [5]

- 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を抜けたミュオン以外の粒子を検出するカロリメータで、バレル部、エンドキャップ部、フォワード部で3種類存在する。

  - バレル部
 

吸収体である鉄中にタイルシンチレータと呼ばれるタイル状のシンチレータを並べたシンチレータで、ハドロンシャワーのエネルギーを測定する。
  - エンドキャップ部
 

液体アルゴン中に銅の吸収体を並べたカロリメータで、液体アルゴン中の電離電子を検出して、ハドロンシャワーのエネルギーを測定する。
  - フォワード部
 

フォワード部のハドロンカロリメータは $z$ 軸方向に3層並べられており、内側の1層は棒状の銅の吸収体と液体アルゴンの電磁カロリメータで、残り2層は棒状のタングステン吸収体と液体アルゴンのカロリメータである。
- ミューオンスペクトロメータ
 

検出器の再外層に位置するミュオンスペクトロメータでは透過力の高いミュオンの位置と運動量の測定を行う。詳細は2.2.2で述べる。

## 2.2.2 ミューオンスペクトロメータ

本研究に関係するミュオン検出器についてより詳細に説明する。

ATLAS 実験の目指す物理事象において、ミュオンは粒子識別がしやすい。また、陽子・陽子衝突での大部分を占める QCD 反応では、 $p_T$  の低い jet が生成される。高い  $p_T$  を持つミュオンは、W, Z といったウィークボソンの崩壊と、b, c といった heavy flavor jet からのものが多い。従って、LHC の陽子・陽子衝突反応の中から、興味深い事象を捉えるためには、高い  $p_T$  を持つミュオンが有効である。



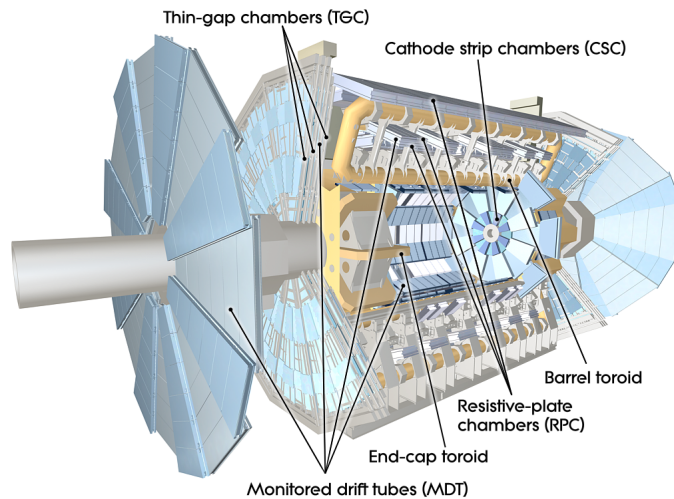


図 2.9: ミューオンスペクトロメータの全体像 [5]

図 2.9 にミューオンスペクトロメータの全体図を示す。ミューオン検出器は透過力の高いミューオンを検出するため、ATLAS 検出器の再外層に位置する。ミューオン検出器はトリガー用の Resistive Plate Chamber (RPC) と Thin Gap Chamber (TGC)、運動量測定のための Monitored Drift Tube (MDT) と Cathode Strip Chamber (CSC) から構成される。トロイド磁場を通過するミューオンの軌跡を 3 ステーションのミューオン検出器で検出器し、その運動量を測定する。その中で、RPC, TGC はバンチ識別のために時間分解能に優れていて、MDT, CSC は運動量測定用の位置の精密測定ができる。

図 2.10 にミューオンスペクトロメータの配置図を示す。

エンドキャップ部にはエンドキャップトロイドマグネットが設置されており、この磁場を通過したミューオンはローレンツ力によって曲げられるため、この曲がり具合からミューオンの運動量を測定できる。

TGC によるトリガーについては 3.2 節で説明する。

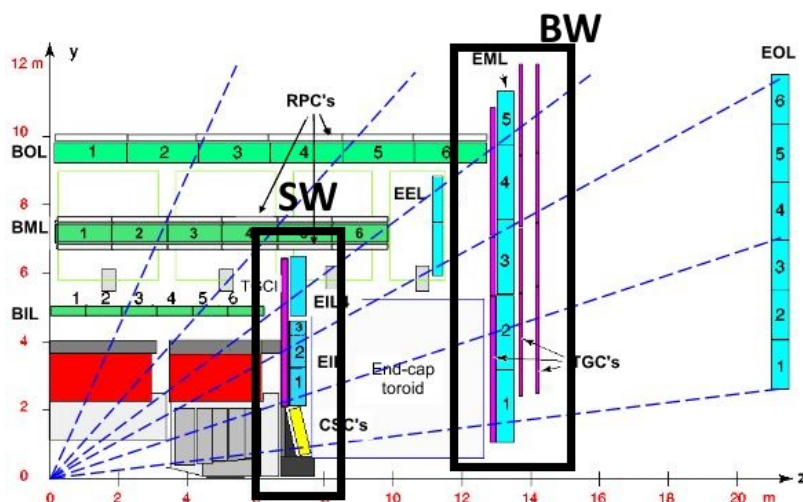


図 2.10: ミューオンスペクトロメータの配置 [9]: エンドキャップミューオンスペクトロメータのうち、 $z = 7\text{m}$  付近にある TGC、MDT、CSC の集合体を Small Wheel (SW) と呼び、 $z = 13\text{m}$  付近にある TGC、MDT の集合体を Big Wheel (BW) と呼ぶ。

### 2.2.3 ヒッグス粒子の物理

ATLAS 実験の目的は、新粒子及び新物理の探索と標準模型の精密測定である。LHC では、粒子の衝突エネルギーが 14 TeV と非常に大きいため、新粒子・新物理の探索を可能にする。

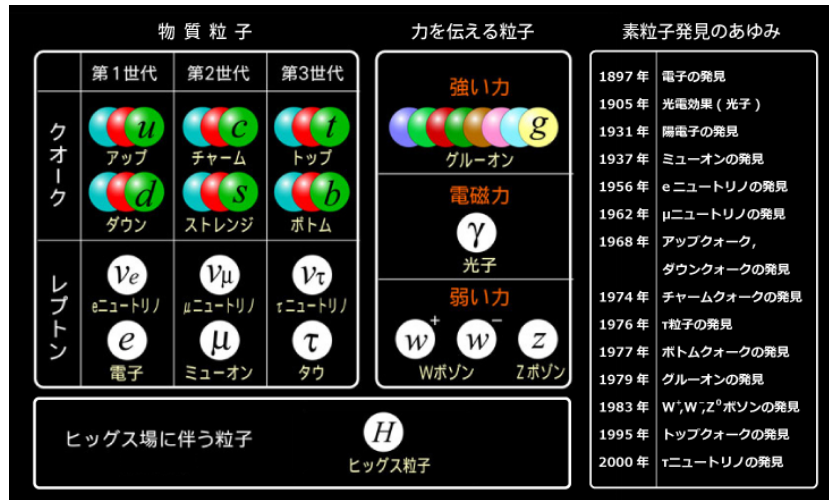


図 2.11: 標準理論の粒子 [10]

図 2.11 に、標準理論模型の粒子の一覧を示す。ATLAS 実験は CMS 実験と共に 2011 年の中で唯一未発見であったヒッグス粒子を発見し、標準理論の全ての粒子の存在の確認に大きく貢献した。

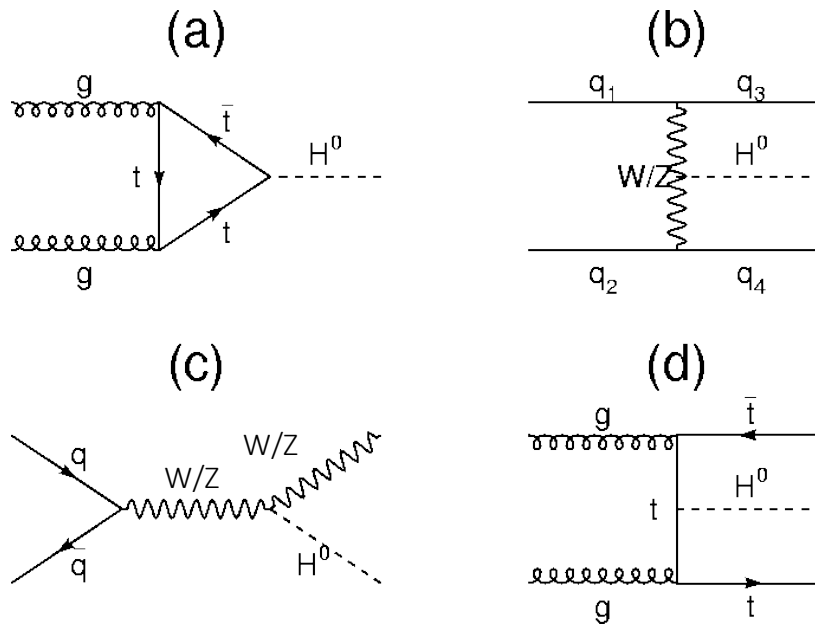


図 2.12: ヒッグス粒子の生成過程 [11]

図 2.12 にヒッグスの主な生成過程を載せ、図 2.13 にこれらの生成の断面積の重心系エネルギーによる変化を載せる。

(a) のグルーオンフュージョンはヒッグスの生成過程において最も断面積が高く、トップクォークとヒッグスの結合を持つため、結合定数の測定に用いられる。しかし、ヒッグス粒子以外の粒子が生成されないため、質量を再構成できる  $W, Z, \gamma$  以外の崩壊はバックグラウンドとの識別が難しい。

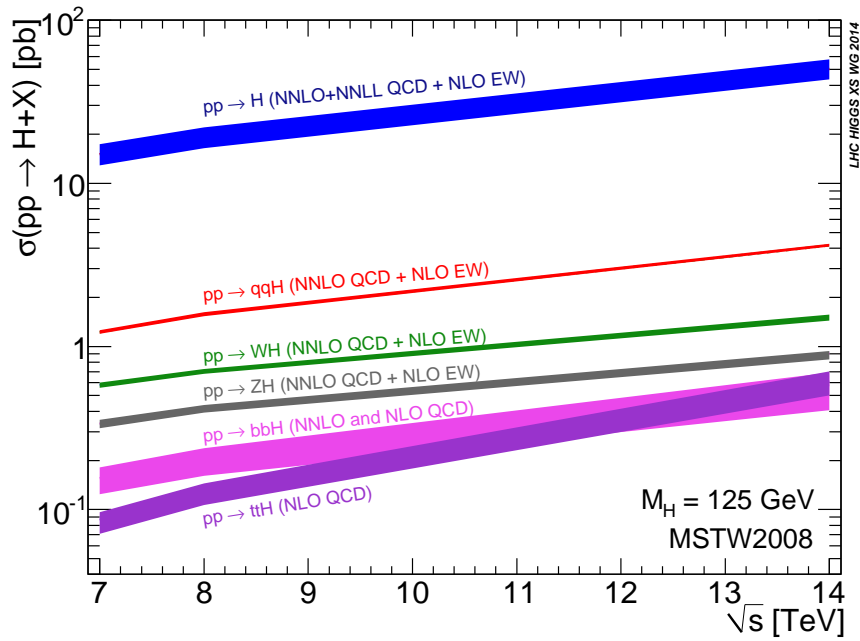


図 2.13: ヒッグス粒子の生成断面積の陽子・陽子衝突の重心系エネルギーによる変化 [12]: 横軸が重心系エネルギー  $\sqrt{s}$  を、縦軸が生成断面積をそれぞれ示す。図 2.13 の生成過程において、青が (a)、赤が (b)、明るい緑と暗い緑が (c)、紫が (d) を示す。

(b) のベクターボソンフュージョンはグルーオンフュージョンに次いで生成断面積が高く、また W と H のカップリングを持つ。(a) と異なり、終状態にクォークが生成されるためバックグラウンドとの識別がしやすいため、 $H \rightarrow \tau\tau$  の精密測定では (b) を用いる。

(c) の W,Z ボソンの制動放射によって生成される過程では (b) と同様に、終状態に W,Z ボソンが現れるため再構成がしやすいが、生成断面積が小さく、未だ観測されていない。

(d) のグルーオンから対生成されたトップクォークからヒッグスが生成される過程では、(c) よりもさらに生成断面積が小さく、観測はされていないが、この過程ではトップクォークが 2 つ生成される。トップクォークの崩壊がほとんど  $t \rightarrow Wb$  でバックグラウンドとの識別が容易であるため、トップクォークとヒッグスの結合定数は (d) を用いて解析できるようになることが理想である。

次にヒッグスの崩壊チャンネルについて、シグナルの強さ  $\mu$  の Run-1 の解析結果を図 2.14 に載せる。現在  $H \rightarrow b\bar{b}$  はまだ見つからないが、他のチャンネルは観測されている。未発見である (c) の生成過程や  $b\bar{b}$  はデータの選別に W,Z ボソンを使用する。W,Z ボソンは終状態に高い横運動量  $p_T$  を持つミュオンが現れるため、ミュオン検出器のトリガー精度を高めることが未発見の物理チャンネルの究明に大きくつながる。

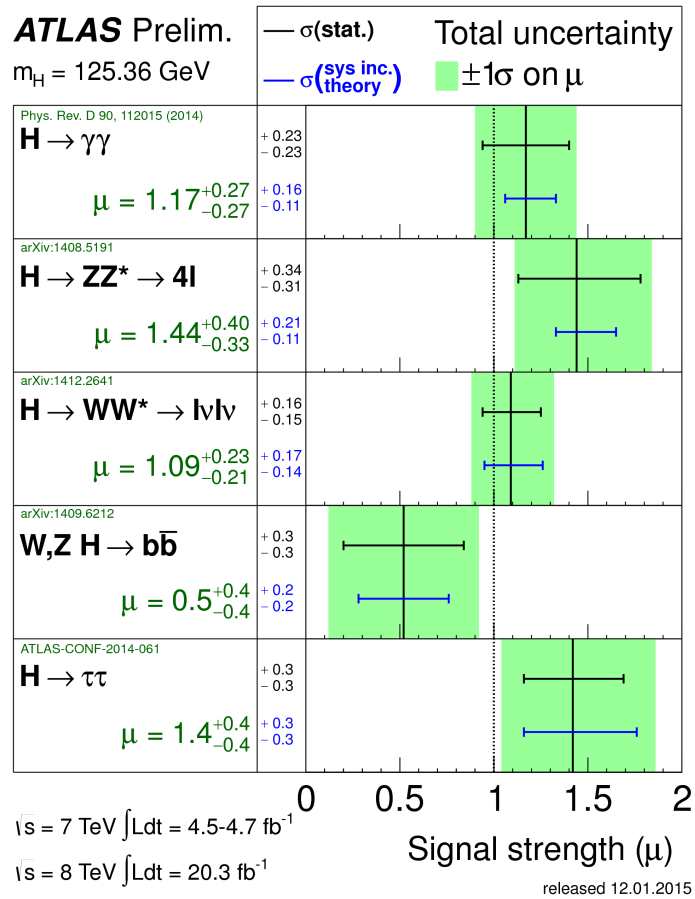


図 2.14: ヒッグス粒子の崩壊チャンネルとそのシグナル強度  $\mu$  値 [5]:  $\mu$  は観測した事象数を、理論で予想した事象数で割った値であり、 $H \rightarrow b\bar{b}$  以外のチャンネルでは 0 から  $3\sigma$  以上離れている。

# 第3章 レベル1エンドキャップミュオン トリガーシステム

## 3.1 ATLASにおけるトリガー及びデータ収集システム

ATLAS 実験では、陽子のバンチ衝突頻度が 40.08 MHz (25 nsec 間隔) と非常に多く、1つの衝突のデータ量は約 1.5 MByte と見積もられており、全てのデータを保存することは不可能である。加えて衝突事象の中には、目的となる物理事象以外の背景事象も多く含まれる。そこで ATLAS 実験では、2段階に分けてトリガーと呼ばれる事象選別処理を行って残った事象のみを保存する。図 3.1 に、現在の ATLAS 実験のトリガーシステムの概要を載せる。まず最初の段階であるレベル1トリガーでは、ハードウェアモジュールを用いた高速処理で、不要なイベントの除去を行うことで、イベントレートを 100 kHz まで落とす。そして次にハイレベルトリガー (HLT) では、ソフトウェアによる高度な解析を行うことで、興味のある物理事象のみを選別する。これらのトリガー処理により、最終的なイベントレートを 1 kHz まで減らす。

### レベル1トリガー

レベル1トリガーではミュオンの横運動量  $p_T$  や電子、光子、jet の横方向エネルギー  $E_t$  の情報を用いてイベントレートを削減する。

各検出器でトリガー判断を行った後、そのトリガー情報が Central Trigger Processor (CTP) に送られる。CTP では受け取った情報からトリガーを発行するかを決定する。

CTP から発行されたトリガーを Level-1 Accept (L1A) 信号と呼び、その信号は Timing Trigger and Control system (TTC) を通して、各検出器のフロントエンド・エレクトロニクスへ送られる。各エレクトロニクスではデータは全て Level-1 Buffer (L1 Buffer) に格納される。L1A 信号はバンチ衝突後 2.5  $\mu$ sec 以内に発行されるため、L1 Buffer は 2.5 $\mu$ sec/25 nsec = 100 回のバンチ衝突分のデータを保持するためのメモリサイズが必要である。L1 Buffer では L1A を受け取ると、それに対応したイベントのデータを後段に送り、それ以外のデータは全て棄却される。

レベル1ミュオントリガーでは、TGC と RPC からミュオンの横運動量  $p_T$  を算出し、予め設定しておいた閾値を超えるミュオン候補を Muon Trigger to CTP interface (MUCTPI) を介して CTP に送る。 $p_T$  の閾値は 6 段階設定されていて、表 3.1 にレベル1ミュオントリガーメニューを示す。

### ハイレベルトリガー (HLT)

ハイレベルトリガーでは、MDT や内部飛跡検出器からの情報を用いて、レベル1トリガーより精度の高い事象選別を行い、トリガーの判定を行う。ATLAS ではイベントの特徴に応じてトリガーメニューが用意されている。ハイレベルトリガーではトリガーメニューを参照し、レ

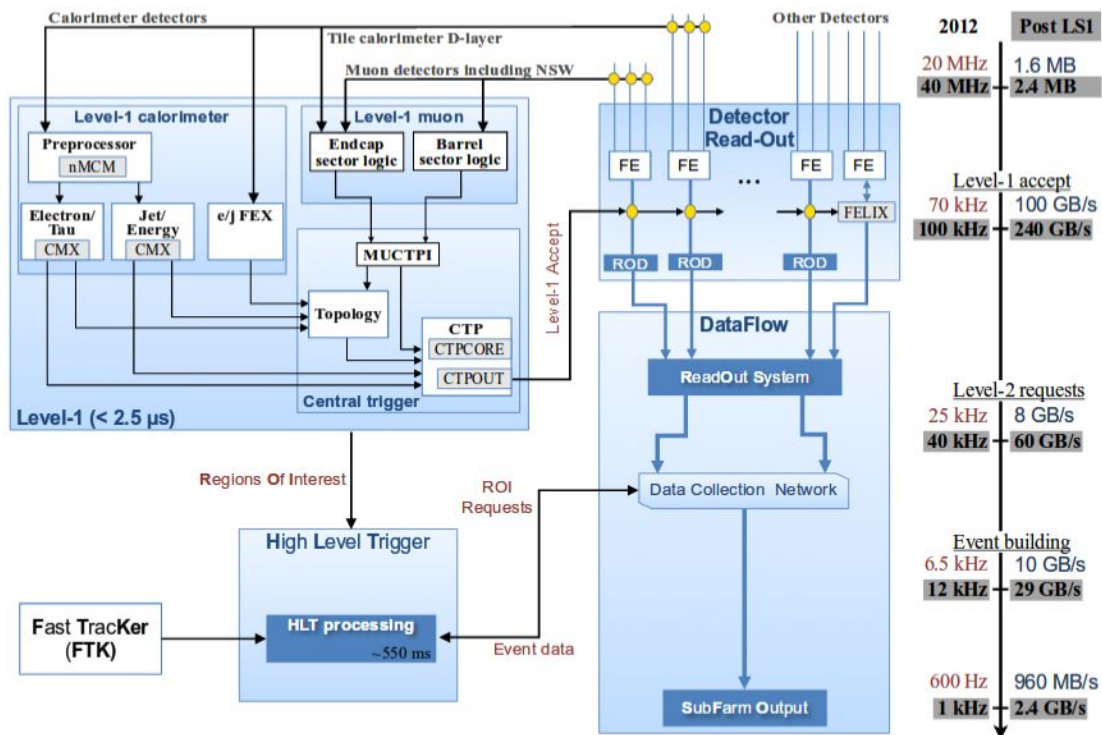


図 3.1: ATLAS 実験のトリガーシステム [13]: ミューオントリガーではバレル、エンドキャップ別々にトリガー処理を行い、トリガー情報を MUCTPI に送る。MUCTPI ではバレル部、エンドキャップ部の情報から最終的なミューオントリガー判定を行い、CTP に送信する。CTP から L1A 信号が発行されると、各フロントエンドモジュールはそのイベントに対応する情報を ROS に送信する。

表 3.1:  $p_T$  の閾値とレベル 1 ミューオントリガーメニューの対応関係

ミューオンメニュー	閾値	Rates at $1.2 \times 10^{34}$		Efficiency	
		Single $\mu$	Double $\mu$	Barrel	Endcap
L1_MU4	$p_T > 4$ GeV	1200 kHz	38 kHz	76 %	90 %
L1_MU6	$p_T > 6$ GeV	300 kHz	4.5 kHz		
L1_MU10	$p_T > 10$ GeV	135 kHz	1.5 kHz	68 %	
L1_MU11	-	49 kHz	-		
L1_MU15	$p_T > 15$ GeV	20 kHz	-		
L1_MU20	$p_T > 20$ GeV	13 kHz	-		

レベル1トリガーを通過したイベントに対してを用いてトリガー判定を下す。現在使用されているトリガーメニューのうち、ミューオンに関するものを表3.2にまとめる。

表 3.2: 現在使用されているトリガーメニューの一例: ミューオンの関わるもののみを抜粋している。

Trigger	内容	$p_T$ 下限値 (GeV)		Rate	
		Level-1	HLT	Level-1 (kHz)	HLT (Hz)
Single lepton	Single isolated $\mu$	20	26	13	133
	Single $\mu$	20	50	13	48
Two leptons	Two $\mu$ 's	$2 \times 10$	$2 \times 14$	1.5	21
		20 (one $\mu$ )	22, 8	13	30
	$e$ & $\mu$	20 ( $\mu$ )	7( $e$ ), 24( $\mu$ )	13	2
	$\tau$ & $\mu$	10(+jets), 12( $\mu$ )	25( $\tau$ ), 14 ( $\mu$ )	1.5	7
Three leptons	Three $\mu$ 's	$3 \times 6$	$3 \times 6$	15	< 0.1
		20 (one $\mu$ )	20, 2 $\times$ 4	13	4
	One $e$ + Two $\mu$ 's	$2 \times 10$ (Two $\mu$ 's)	$2 \times 10, 12$	1.5	0.2
	Two $e$ 's + One $\mu$	$2 \times 8$ ( $e$ ), 10 ( $\mu$ )	$2 \times 12, 10$	1.1	0.1

例えばシングルミューオンの場合について説明する。レベル1トリガーでは  $p_T$  が 20 GeV 以上のミューオンの候補を選出し HLT でそのミューオンの条件によってトリガーを決定する。isolated  $\mu$  は図 3.2 のような、jet から遠いミューオンを意味する。例えば  $H \rightarrow ZZ$  の場合には isolated のミューオンが 2 本検出される可能性が高い。

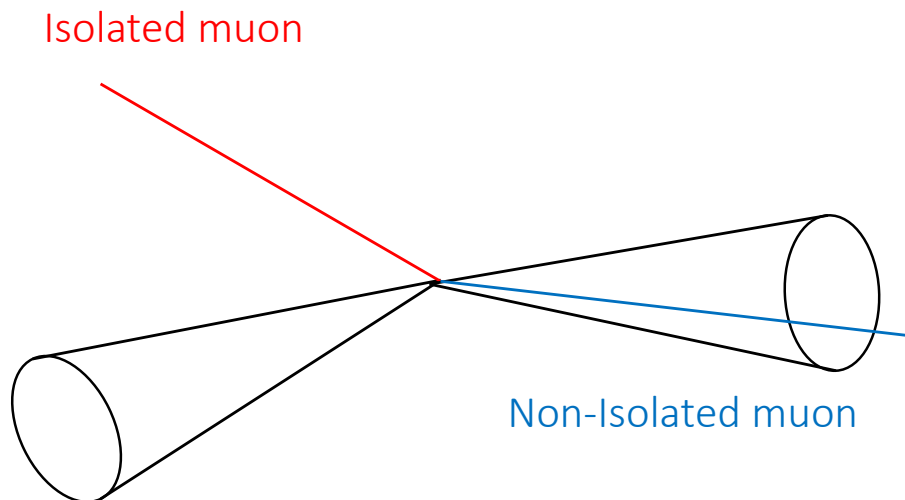


図 3.2: Isolated muon の模式図: jet と重なっていないミューオンを Isolated muon という。

しかし高いエネルギーで生成されたトップクォーク崩壊の場合には運動量の高いミューオンが生成される。図 3.3 に、 $t \rightarrow Wb$  の崩壊の模式図を示す。



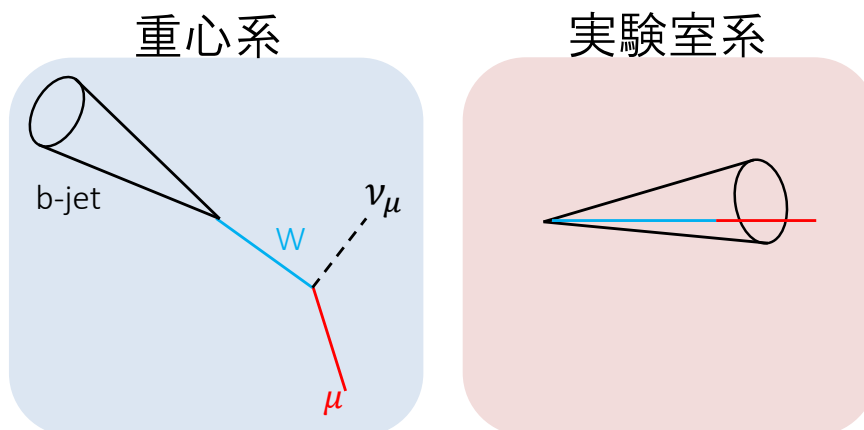


図 3.3:  $t \rightarrow Wb$  の模式図

トップクォークの崩壊では、エネルギーの高い  $W$  と  $b$  を放出し、 $W$  はミューオンとニュートリノに、 $b$  は jet となる。重心系では back-to-back に粒子が放出されるが、実験室系では boost されることにより jet とミューオンが重なる。しかし、 $W$  からのミューオンは非常に高い運動量を持つので、isolated でなくても、運動量が高いミューオンのイベントは解析に用いられる。

## 3.2 レベル1エンドキャップミューオントリガー

### 3.2.1 Thin Gap Chamber (TGC)

2.2.2 節で述べたように、レベル1エンドキャップミューオントリガーでは TGC 検出器を使用する。TGC は図 3.4 に示すような断面を持つ MWPC の一種で、ワイヤー ( $R$  方向) とストリップ ( $\phi$  方向) を用いて粒子の位置を 2 次元で読み出せる。TGC の設計パラメータを表 3.3 に載せる。TGC 検出器は、ワイヤとカソードの間隔が 1.4 mm と非常に短い。これは 25 ns の間隔で発生する衝突のデータを正しく検出するための設計である。

表 3.3: TGC の各パラメータと設計値

パラメータ	設計値
ワイヤ材質	タングステン
ワイヤ間隔	1.8 mm
ワイヤ直径	50 $\mu\text{m}$
ガスギャップ	2.8 mm
印加電圧	3.0 KV
ガス混合体	$\text{CO}_2$ / n-pentane (55 %/45 %)
ガス増幅率	$10^6$



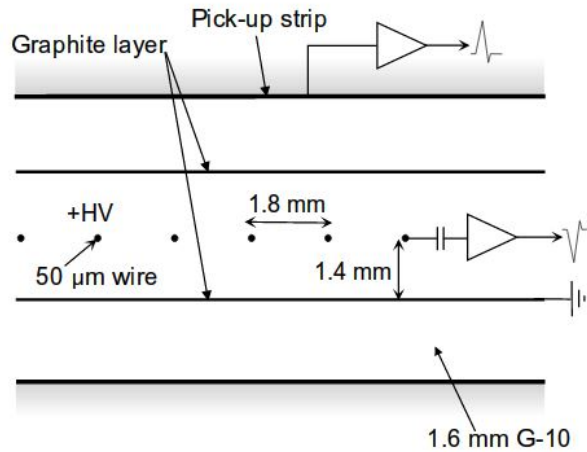


図 3.4: TGC 検出器の構造 [9]: ワイヤと垂直にストリップが配置されている。また検出器内にはガスが充填されている。

TGC における検出原理について、図 3.5 を用いて説明する。TGC 内部には  $\text{CO}_2$  と n-pentane 混合ガスが充填されており、荷電粒子がこのガス中を通過すると炭酸ガスとペンタンガスが電離され、電子とイオンが発生する。電子は正の電圧のかかっているワイヤへ、イオンはカソード面へ移動する。電子がワイヤに近づくとき電子雪崩を起こし、電子とイオンの量がさらに増幅される。これらの電子・イオン対がワイヤとカソードにそれぞれ移動することで信号が発生する。これを測定することで粒子の飛来した位置と時間を調べることができる。

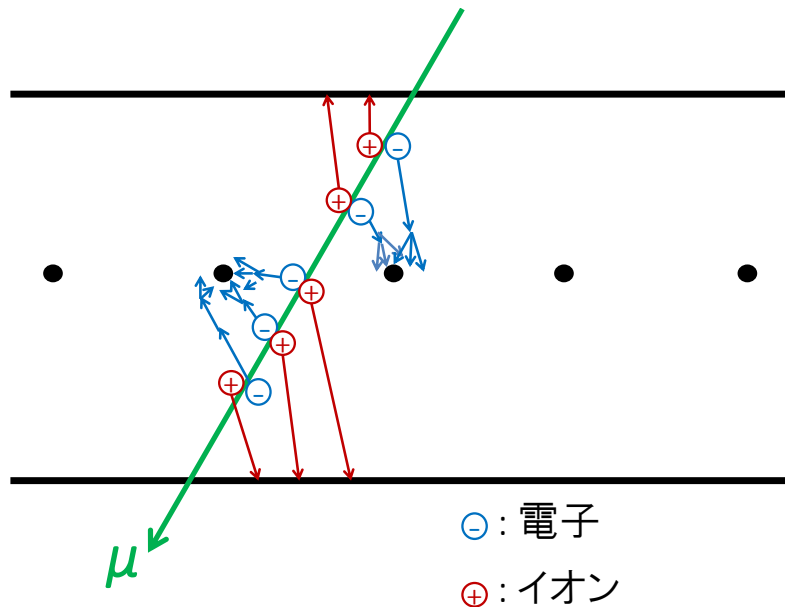


図 3.5: TGC 検出器の検出原理

また TGC にはガスギャップを 2 層持つ Doublet と、3 層持つ Triplet がある。それぞれの構造を図 3.6 に示す。Doublet は 2 層のワイヤ面と 2 層のストリップ面から読み出す。Triplet は中央の層にはストリップ面がなく、3 層のワイヤ面と 2 層のストリップ面から読み出しを行う。このように複数の層で読み出しを行うことで、ミューオンの検出効率を高く保っている。

図 3.7 に TGC の配置図を、図 3.8 に BW の写真をそれぞれ載せる。トロイド磁場より内側に位置する Inner 領域には、Endcap Inner (EI) と、Forward Inner (FI) が配置されている。EI は  $1.0 < |\eta| < 1.3$  の領域をカバーする Doublet TGC で、FI は  $1.3 < |\eta| < 2.7$  をカバーする

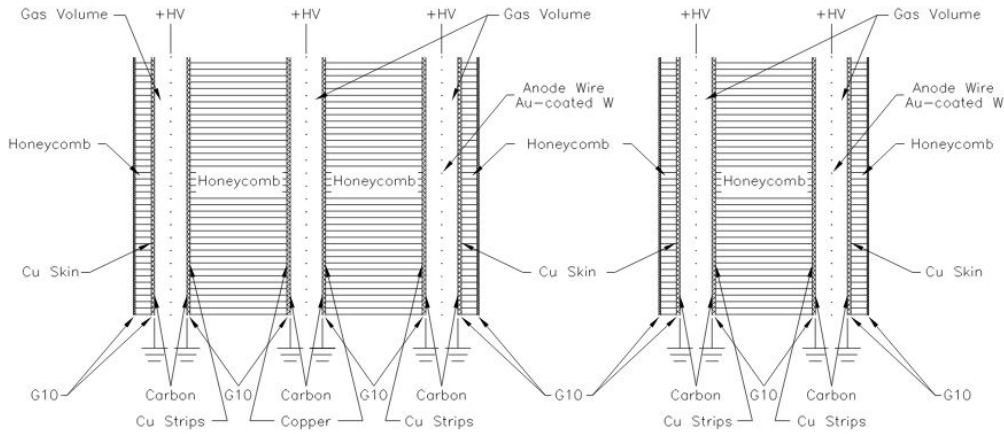


図 3.6: TGC Doublet と Triplet の構造 [9]: 左が Triplet、右が Doublet である。

Doublet TGC である。BW には  $z$  の絶対値が小さい順に M1、M2、M3 と呼ばれる TGC が配置されている。これらのうち M1 のみが Triplet で、それ以外は全て Doublet である。

また TGC では  $\phi$  方向に 12 分割した領域をセクターと呼び、TGC におけるデータ処理はこのセクター単位で行われる。図 3.9 に、セクターの図を載せる。1つのセクターはエンドキャップ部を 4つ、フォワード部を 2つの台形に分けられ、そのそれぞれをトリガーセクターと呼び、それらを図 3.10 のように  $\phi$  方向には 4 分割、 $R$  方向にはエンドキャップ領域は 37 分割、フォワード領域は 16 分割したものを Region of Interest (RoI) と呼ぶ。RoI はトリガーにおける最小単位で、1セクターにつき  $37 \times 4 \times 4 + 16 \times 4 \times 2 = 720$  個の RoI がある。

### 3.2.2 トリガー判定の流れ

TGC 検出器における  $p_T$  の算出方法について説明する。図 3.4 に、 $p_T$  の測定の概念図を載せる。衝突点から生成されたミュオンは磁場によって曲げられた後 BW の TGC を通過する。磁場によってミュオンの飛跡が曲げられる量はミュオンの運動量に依存し、高い運動量を持つミュオンほど曲がりにくい。この性質を利用して、TGC ではミュオンの運動量を算出する。まず TGC の 3 層目と衝突点を直線で結ぶ。この直線と TGC の 1 層目との交点と、実際に磁場を通過してきたミュオンが 1 層目を通過した地点との距離 ( $\Delta R, \Delta \phi$ ) から運動量を算出する。算出の際には図 3.12 に示すようなコインシデンスウィンドウと呼ばれる Look-Up Table を用いる。これは ( $\Delta R, \Delta \phi$ ) の情報からミュオンの運動量をレベル毎に分けたもので、その基準は表 3.1 に示した通りである。

### 3.2.3 エレクトロニクス

TGC で検出したデータを処理するためのエレクトロニクスは、トリガー系、リードアウト系、コントロール系の 3つのシステムに分類される。図 3.13 に、TGC 検出器におけるエレクトロニクスの概要図を示す。配置場所は TGC の側、TGC 外縁、コントロールルーム (USA15) に分類される。

以下に、それぞれのシステムの概要、およびそこで使用されているシステムについて記述する。

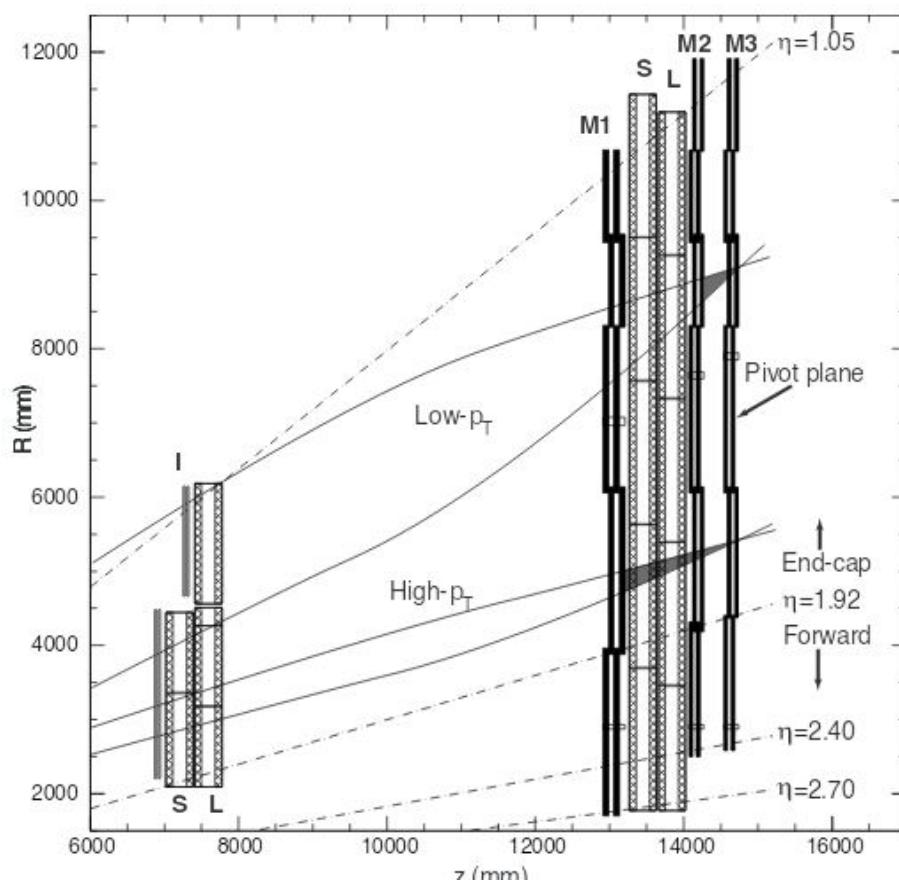


図 3.7: TGC の  $R$ - $z$  平面における配置図 [9]

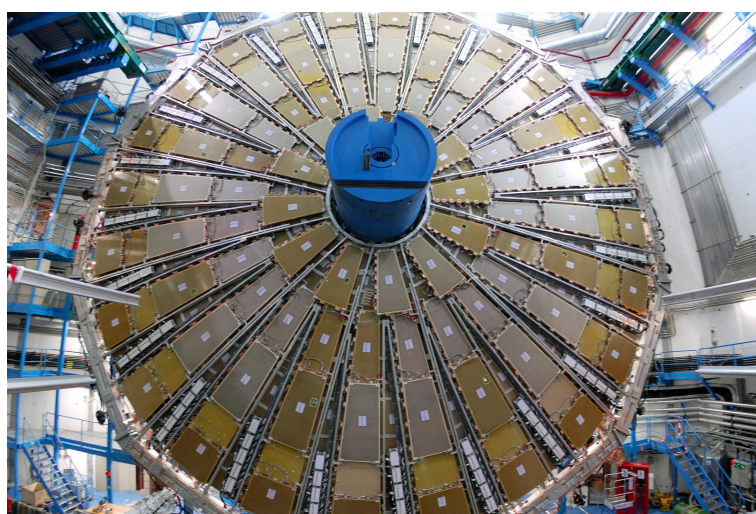


図 3.8: TGC Big Wheel の写真 [5]

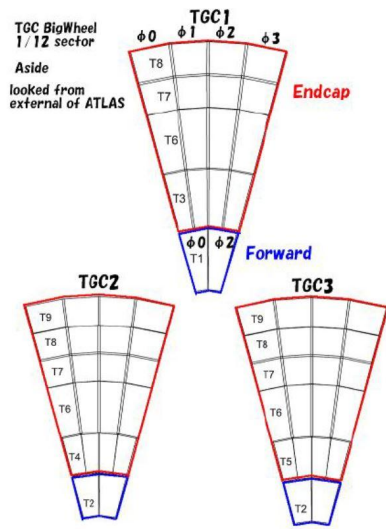


図 3.9: BW の 1/12 セクター

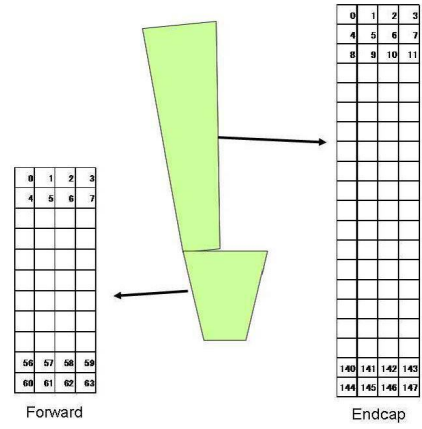


図 3.10: トリガーセクター及び RoI

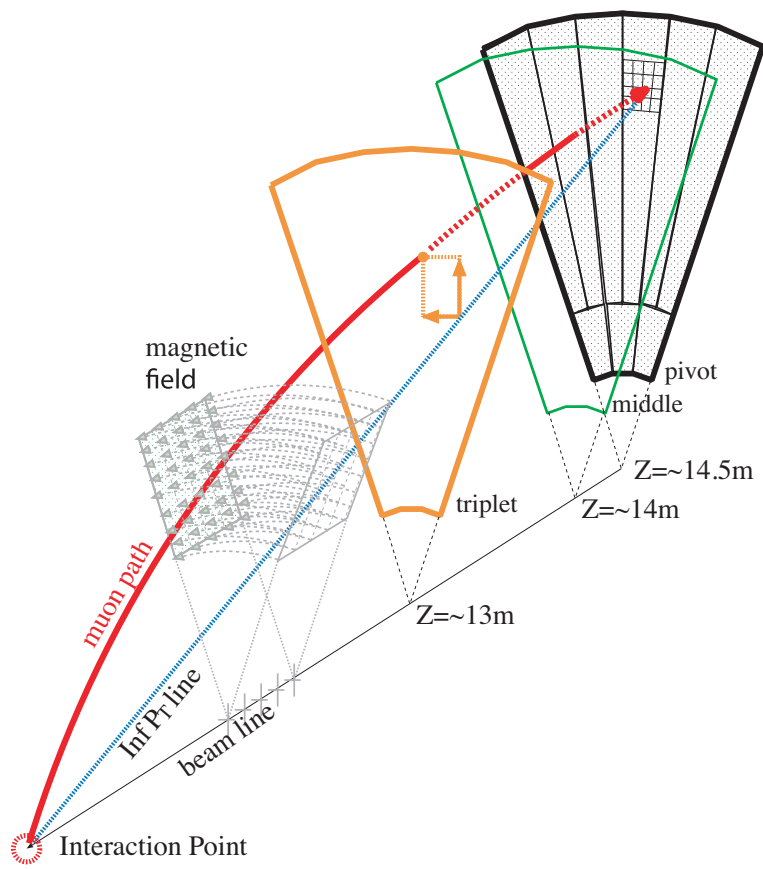


図 3.11: TGC における  $p_T$  算出の概念図

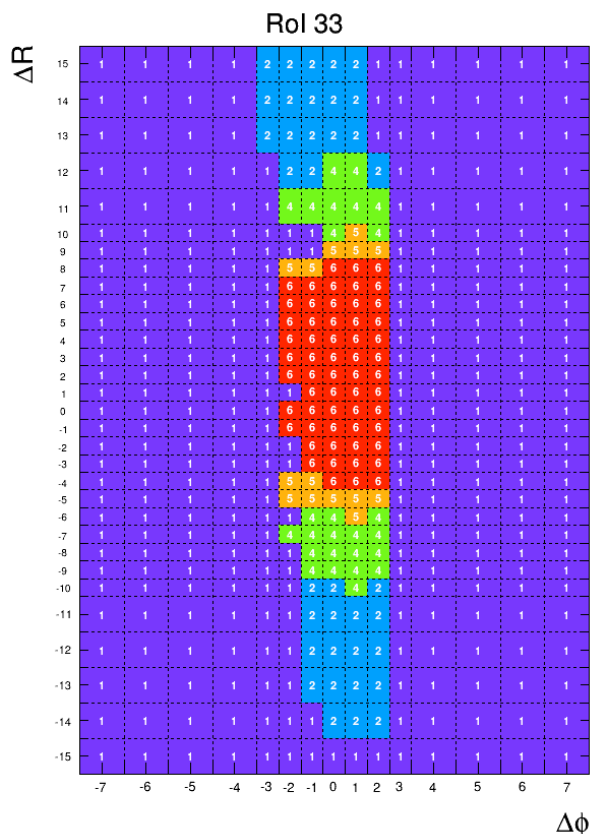


図 3.12: コインシデンスウィンドウ: $p_T$  のレベルに応じて色分けして表示されている。青が L1.MU4, 水色が L1.MU6, 緑が L1.MU11, 黄色が L1.MU15, 赤が L1.MU20 を示す。

## トリガー系

トリガー系ではレベル1 ミューオントリガーの判定が行われる。トリガー系では図 3.14 に示すように段階に分けてコインシデンス処理を行う。

- Amplifier Shaper Discriminator (ASD)

ASD Board は図 3.15 に示すような TGC 検出器の側面に取り付けられるモジュールで、TGC からのアナログ信号に対して増幅、整形したうえで、一定の閾値を超えた信号のみを PS Board に送る。また PP ASIC から送られるテストパルス信号を各チャンネルに分配する。

- PS Board

Patch Panel ASIC (PP ASIC) と SLB ASIC からなるモジュールである。PP ASIC は ASD から送られる信号を遅延させることで、TTC から送られる Bunch Crossing (BC) クロックとの同期を図る ASIC である。ASD から送られる信号は、ケーブルの長さや粒子の飛来時間といった理由から信号の到来時間に差が生じてしまうため、この PP ASIC でタイミングをそろえることは非常に重要である。また PP ASIC には TGC の信号を擬似的に生成するテストパルス源が実装されている。

また SLB ASIC はトリガー部とリードアウト部に分けられる。ここではトリガー部のみ説明し、リードアウト部は他のリードアウトエレクトロニクスと合わせて説明する。トリガー部では Doublet と Triplet それぞれにコインシデンス処理を行う。Doublet である M2、M3 ではワイヤーとストリップそれぞれに 4 層中 3 層のヒットがあるかを判定し、Triplet



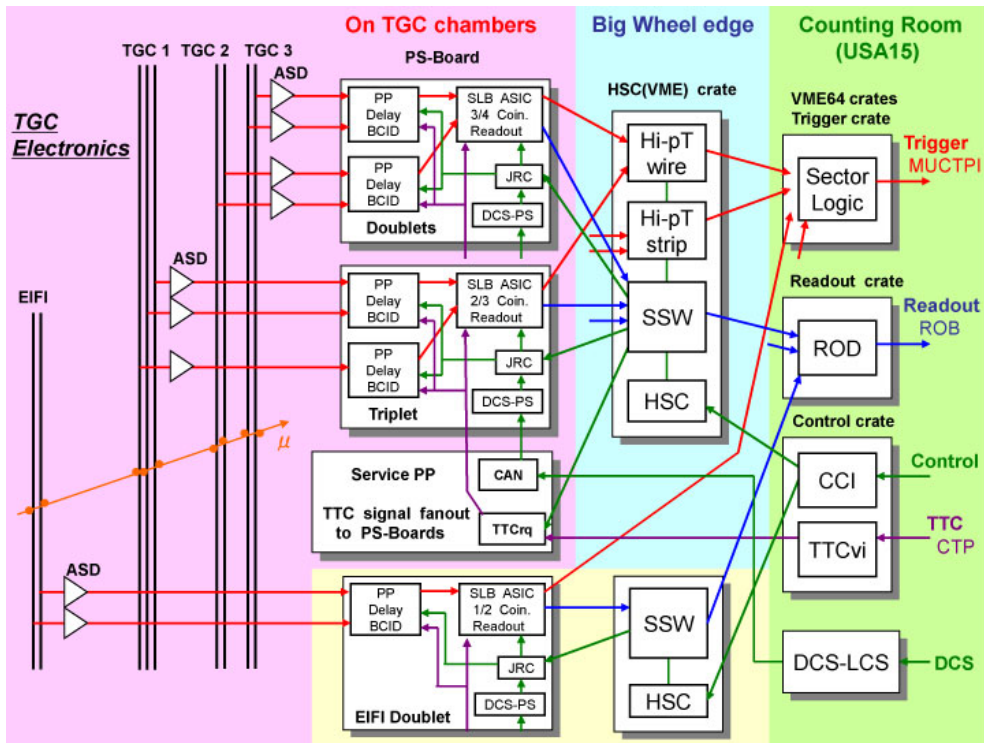


図 3.13: TGC エレクトロニクスの全体像 [14]: 図中赤で示されている部分は TGC 検出器の表面に設置されたモジュール、青で示されている部分は Big Wheel の端に設置されている VME クレータに配置されているモジュール、緑で示されている部分は ATLAS 検出器のある実験ホールから離れた USA15 というエレクトロニクスがまとめられている部屋に配置される。また矢印は信号の流れを示しており、赤がトリガー、青がリードアウト、緑が制御系の信号、紫が TTC の信号を表す。

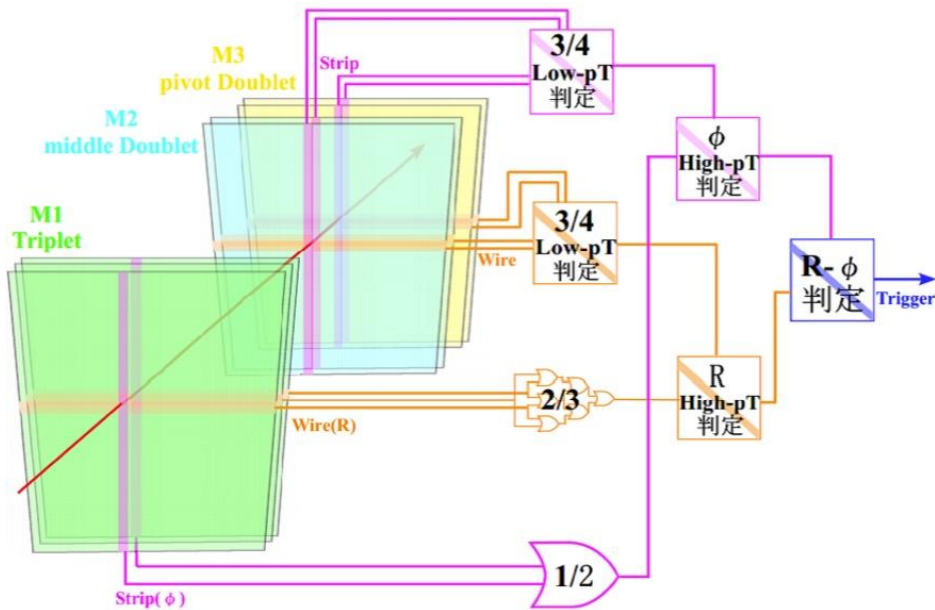


図 3.14: TGC の段階に分けられたコインシデンス処理

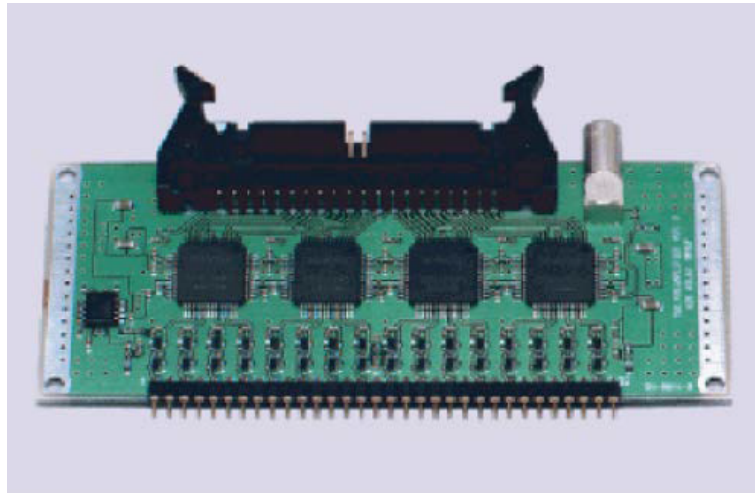


図 3.15: ASD Board [15] : ASD ボードは4つのASICを搭載している。

である M1 ではワイヤー 3 層中 2 層のヒットとストリップ 2 層のうちどちらかにヒットがあるかを判定する。コインシデンス処理を行った後、その情報は High-pT Board へ送られる。

- High-pT Board

High-pT Board では SL Board から M1 のヒット点の情報 ( $R_1, \phi_1$ ) と M3 のヒット点の情報 ( $R_3, \phi_3$ )、そして M2-M3 間の差分 ( $\Delta R_{32}, \Delta \phi_{32}$ ) を受け取り、Doublet (M2, M3) と Triplet (M1) の間のコインシデンス処理を行う。コインシデンスは、 $R$  方向と  $\phi$  方向で別々に行い、ミューオンのトラック候補のうち、 $p_T$  が最も高い 2 つを選び、M3 におけるヒット位置と M1-M3 間での差分を SL に送信する。

- Sector Logic (SL)

SL はミューオンの  $R$  判定及び  $\phi$  判定を行い、3.2.2 節で説明したような Look-Up Table を用いて  $p_T$  の最終決定を行い、MUCTPI にトリガーを発行するモジュールである。1 つのトリガーセクターにつき、最大 2 つのミューオンの  $p_T$  情報と RoI 情報を MUCTPI に送信する。

SL は図 3.16 に示すようなモジュールで、1 つの 1/12 セクターにつき、エンドキャップを 2 枚、フォワードを 1 枚、計 3 枚で処理し、片サイドで  $3 \times 12 = 36$  枚、両サイドで 72 枚用いられている。

## リードアウト系

TGC の情報を ReadOut System (ROS) に送信する。

- PS Board

PP ASIC で時間調整したデータは SLB ASIC 内の Level-1 Buffer (L1 Buffer) と呼ばれるバッファに格納される。格納されたデータは、CTP から L1A 信号を受け取ったとき、その L1A 信号に対応するデータを SSW へ送信する。

- Star SWitch (SSW)

複数の SLB ASIC からデータを受け取り、ゼロサプレスというデータ圧縮処理を行った後、G-link と呼ばれるシリアル通信規格を用いて ROD に送信する。ゼロサプレスとは全



図 3.16: Sector Logic Board

てのデータを 8 bit 毎の cell に分け、その値が 0 でない cell のみ、その cell のアドレスを追加してデータを送る処理である。図 3.17 にその概略図を載せる。0 でない cell に対して、アドレスを追加するためデータのある部分に対してはデータサイズが大きくなるが、TGC から送られてくるデータには 0 が入ることが非常に多く、このゼロサプレス処理を行うことは非常に効果的である。

- ReadOut Driver (ROD)

ROD は図 3.18 に示すようなハードウェアモジュールで、片サイドにつき 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う。ROD には SSW から受け取ったデータを集め、それらのイベント ID のチェックをした後、後段の ROS に S-link と呼ばれる通信規格を用いて送信する役割がある。ROS に送信する際には、収集したデータを決められたフォーマット (ROB フォーマットと呼ぶ) に従ってイベントデータをまとめて送信する。

## コントロール系

トリガー系、リードアウト系で使用されているモジュールの制御を行う。コントロール系はモジュールの場所によってその制御方法が異なる。

- SSW

JTAG 通信を用いて PS Board の制御を行う。また I2C 経由で TTCrq(3.2.4 節参照) の制御も行う。

- High-pT SSW Controller (HSC)

BW の端の VME クレーンを制御するモジュールである。

- Crate Control Interface (CCI)

CCI は USA15 に置かれた、HSC をコントロールするモジュールである。USA15 の CCI から HSC を経由して SSW 及び High-pT Board、そして SSW を経由して TGC 検出器近傍の各モジュールの制御を行う。



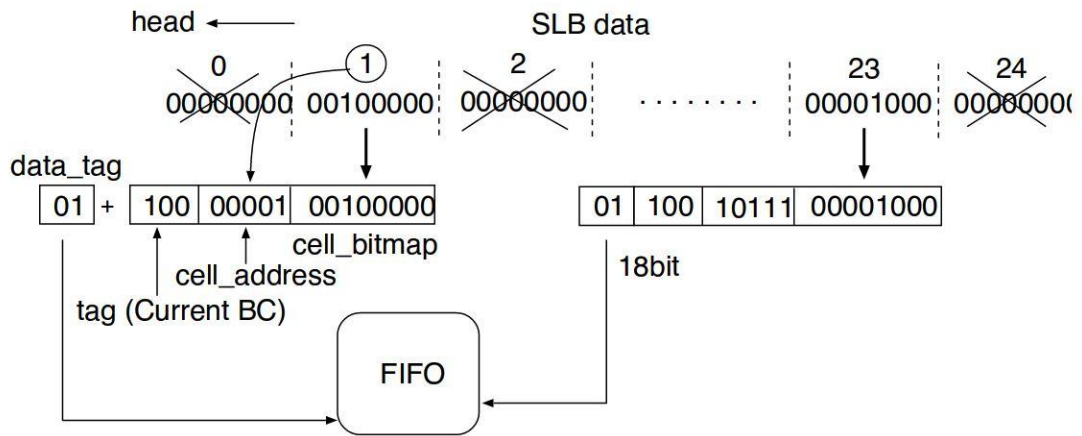


図 3.17: ゼロサプレスロジックの概略図 [16]

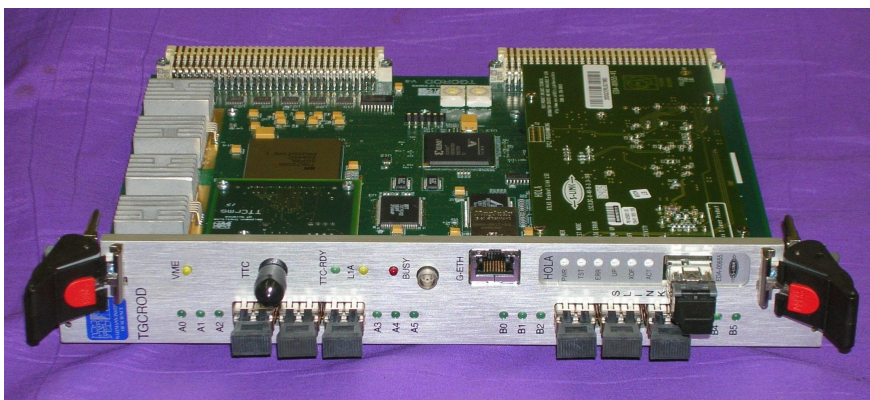


図 3.18: TGC ROD モジュール

### 3.2.4 TTC

Timing Trigger and Control (TTC) システムは、各検出器のフロントエンドエレクトロニクスの同期をとるためのシステムで、LHC クロックという同期信号や、L1A 信号をはじめとしたトリガー情報を分配する。

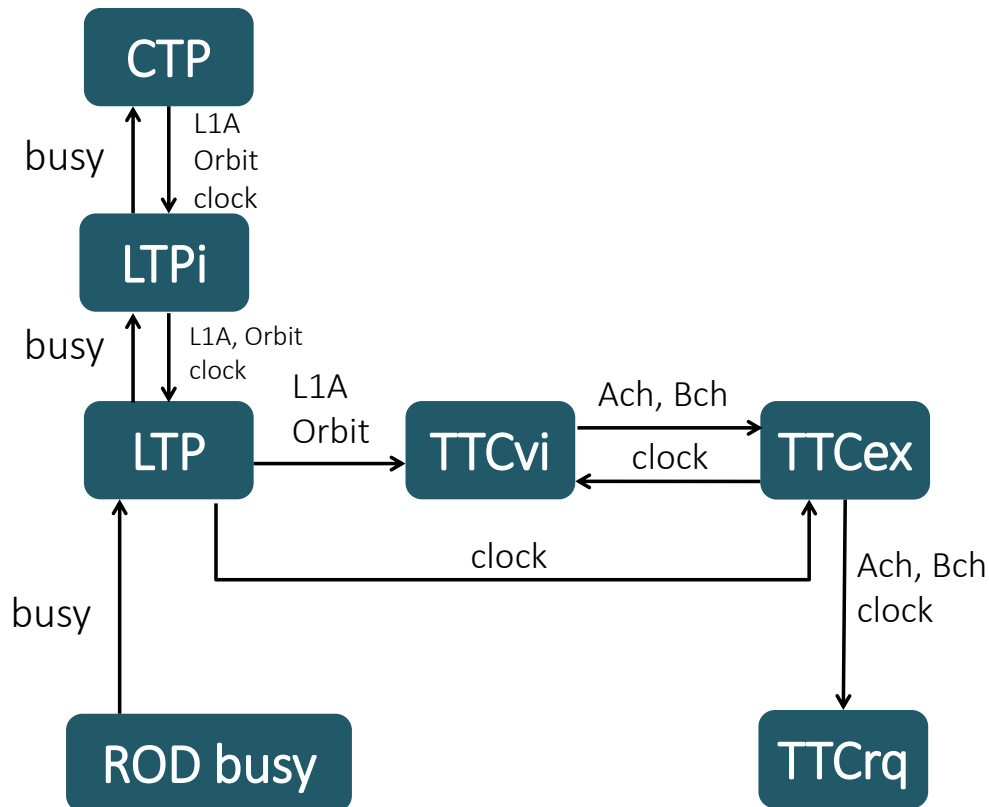


図 3.19: TTC システム: Ach には L1A と clock を、Bch にはそれ以外を送信する。

以下に TTC が各エレクトロニクスに分配する信号の種類を説明する。

- LHC clock  
LHC が配布する 40 MHz のクロックを各システムに分配する。
- L1A  
L1A は CTP から送られ、TGC の電気回路においては SLB 内の L1 Buffer からデータが読み出される。
- Orbit ID  
LHC1 周分のバンチ衝突の回数を表す ID。
- Bunch Counter Reset (BCR)  
Bunch Counter という、バンチ衝突の回数を表すパラメータのリセット信号で、Orbit ID が増加するときリセット信号が発行される。
- Event Counter Reset (ECR)  
L1ID のカウントが 0 に戻るタイミングで発行されるリセット信号。

- Trigger Type  
L1A が発行された際に、どのような条件でトリガーを発行したのかを示すパラメータ。

TTC システムの構成を図 3.19 に載せる。以下にそれぞれの説明を簡潔に説明する。

- Local Trigger Processor Interface (LTPi)  
LTPi は各検出器ごとに設置されており、CTP から L1A やクロック信号等、タイミング・トリガーに関する情報を受け取る。また ROD から BUSY 信号を受け取り、それを CTP に知らせて L1A の発行を停止させる役割も持っている。
- Local Trigger Processor (LTP)  
LTP は LTPi から信号を受け取り、TTCvi に L1A 信号と Orbit 信号を送信し、TTCex にクロック信号を送信する。また LTP はバンチカウンタのクロックや Orbit などを内部で生成する機能を持っていて、単独の試験を行う際には LTP で信号を生成して行うことができる。
- TTCvi  
TTCvi は LTP から受け取った信号をトリガーに関する情報とそれ以外でそれぞれ A-channel と B-channel に分けて TTCex に送信するモジュールである。A-channel には L1A に同期して送られる L1ID、ECR ID、BCID が、B-channel には Orbit ID や Trigger Type といった L1A の発行に関係なく送られる情報が送られる。
- TTCex  
TTCex は A-channel と B-channel に分けられた TTC 信号を各モジュールに分配する。信号を送る際には光信号に変換して送る。
- TTCrq  
TTCrq は TTCex から送られてきた光信号を電気信号に変換してモジュールに分配する装置である。

### 3.2.5 ミューオントリガーシステムの問題点

ミューオントリガーでは、図 3.20 のように、衝突点由来でない荷電粒子が TGC を通過することでトリガーが発行されてしまうという問題があり、これをフェイクトリガーと呼ぶ。フェイクトリガーが発生する主な原因は、ATLAS 検出器内に残る二次粒子がビームパイプや磁石に衝突することで生じる荷電粒子がトロイド磁場によって曲げられ、ミューオンの飛跡として検出されることが挙げられる。

このフェイクトリガーを削減するために、Run-2 では TGC 以外の、BW より内側の検出器の情報を追加して以下のコインシデンス処理を行っている。これらの処理をまとめて Inner Coincidence と呼ぶ。

- EIFI Coincidence  
Endcap Inner (EI) と Forward Inner (FI) からの、磁場で曲げられる前のミューオンの位置情報を用いてコインシデンス処理を行うことで、例えば図 3.20 のようなビームパイプから発生した荷電粒子によるトリガーを排除できる。

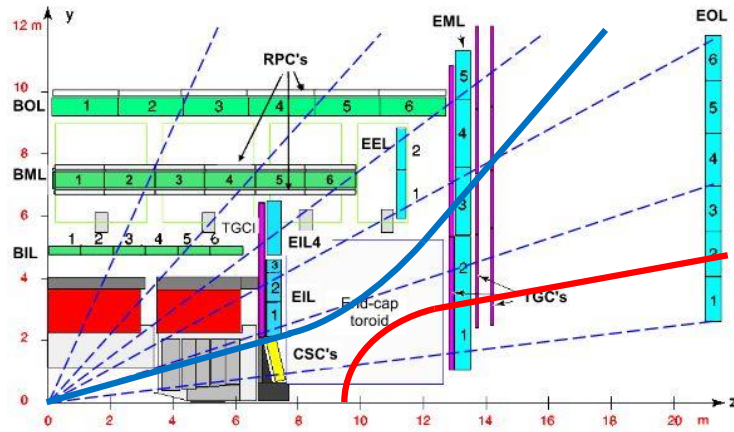


図 3.20: フェイクトリガー [17]: 衝突点由来のミュオン (図中青線) ではなく、ビームパイプ等から飛来した荷電粒子が SW-BW 間のトロイド磁場で曲がり、TGC を通過することで発行される偽りのトリガー。

- Tile Coincidence

Inner Coincidence と同様、タイルカロリメータの情報を用いてコインシデンスを行う。ミュオントリガーでは、Extended Barrel 最外層のセルの情報を用いてトリガー処理を行う。EI の部分にはバレルトロイドコイルが位置しており、図 3.21 に示すように、EIFI Coincidence を行ったときに EI 領域でヒット数が大きく低下する領域が生じてしまう [18]。そのためこの EI 領域においては、EI よりも衝突点に近いタイルカロリメータの情報を用いる。

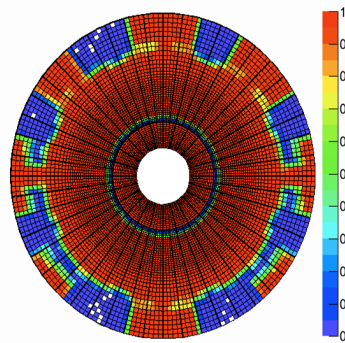


図 3.21: EI または FI でのヒットを課したときの BW のシミュレーションによるヒット分布 [18]: 外側 (EI の領域) では部分的にヒット数が低下する。

図 3.22 に、タイルカロリメータの配置図を示す。Tile Coincidence では、EI と同じラピディティ領域 ( $1.0 < |\eta| < 1.3$ ) をカバーする D5、D6 の情報を用いる。

これらのコインシデンス処理を通して、ある程度はフェイクトリガーを削減できる。しかし、現在のままでは十分に削減することができず、Run-3 のイベントレートではトリガーを圧迫してしまう。

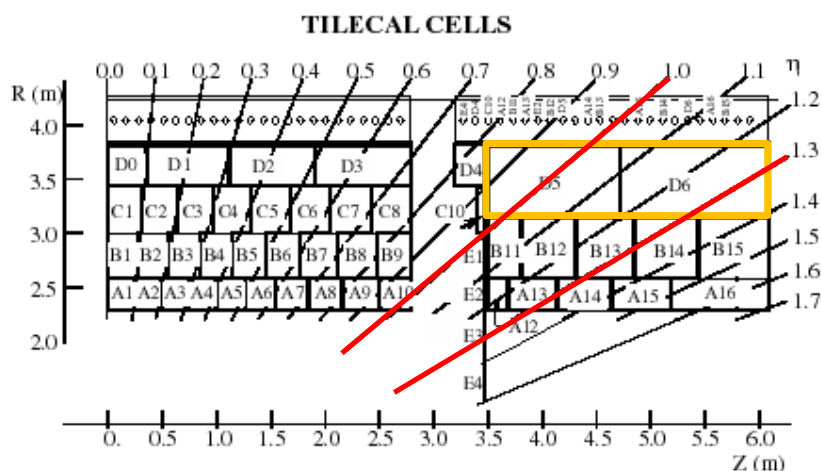


図 3.22: Tile カロリメータの配置図 [19]: Tile Coincidence では D5,D6 を使用する。

### 3.3 ミューオン検出器のアップグレード

Run-3 からはそれまで Small Wheel (SW) のあった場所に、新しく New Small Wheel (NSW) を導入する。

ミューオントリガーは NSW を用いたコインシデンス処理を追加することで、フェイクトリガーをさらに精度よく削減し、さらにこれまで TGC がカバーされていなかったフォワード領域においてもコインシデンス処理を行うことができるため、トリガーレートを抑えられることが期待される。表 3.4 にトリガー処理によるトリガーレートの変化を示す。

表 3.4: Run-3 におけるコインシデンス処理による L1\_MU20 のトリガーレート [17]: NSW によるトリガー処理で、従来のトリガーレートの 50 % 以上削減されることが期待される。

TGC-BW	TGC-BW + EIFI + Tile Coincidence	TGC-BW + NSW + Tile Coincidence
51 kHz	28 kHz	13 kHz

#### 3.3.1 NSW

NSW は図 3.23 のような形をした、Run-3 から新しく実装される検出器である。NSW は MicroMegas 検出器と small-strip TGC (sTGC) で構成され、sTGC 4 層 - MicroMegas 8 層 - sTGC 4 層という構造を持つ。NSW は

NSW で使用されている MicroMegas 検出器の断面図を図 3.24 に示す。また MicroMegas 検出器は  $\phi$  方向の測定を可能にするために、図 3.25 左に示すように水平方向に対して  $1.5^\circ$  だけ傾けたステレオストリップ層を持つ。MicroMegas8 層のうち外側 2 層が水平方向のストリップで、内側の 4 層が stereo ストリップとなる。

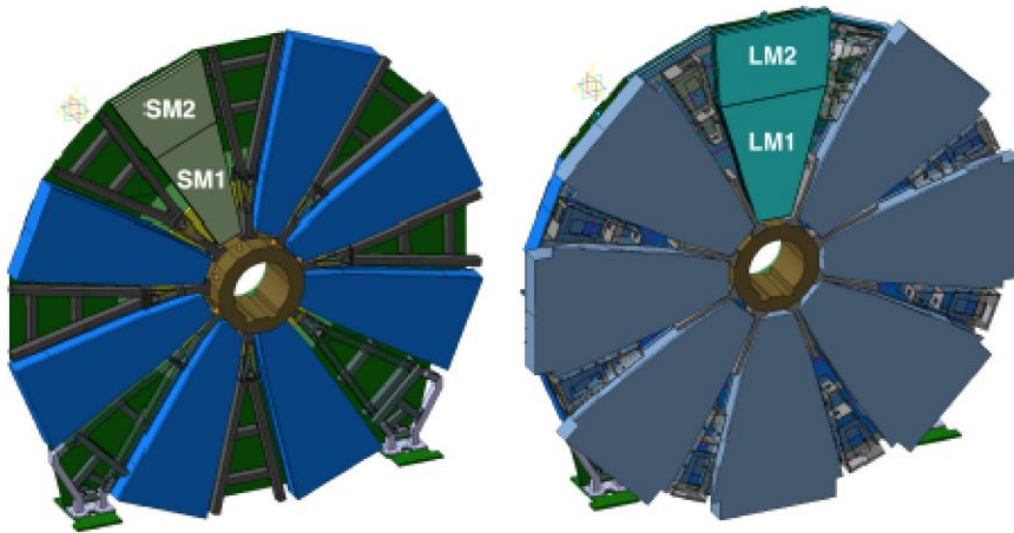


図 3.23: NSW [17] : ラージセクタとスモールセクタが各8セクタずつ、 $\phi$ 方向に $2^\circ$ 重なった状態で交互に配置されている。スモールセクタの面が衝突点に近い側になるように、現在のSWのある位置に配置される。SWは $1.3 < |\eta| < 2.0$ の領域のみカバーされ、フォワード領域はCSCがあるためカバーされていなかったが、NSWでは $1.3 < |\eta| < 2.7$ の領域をカバーする。

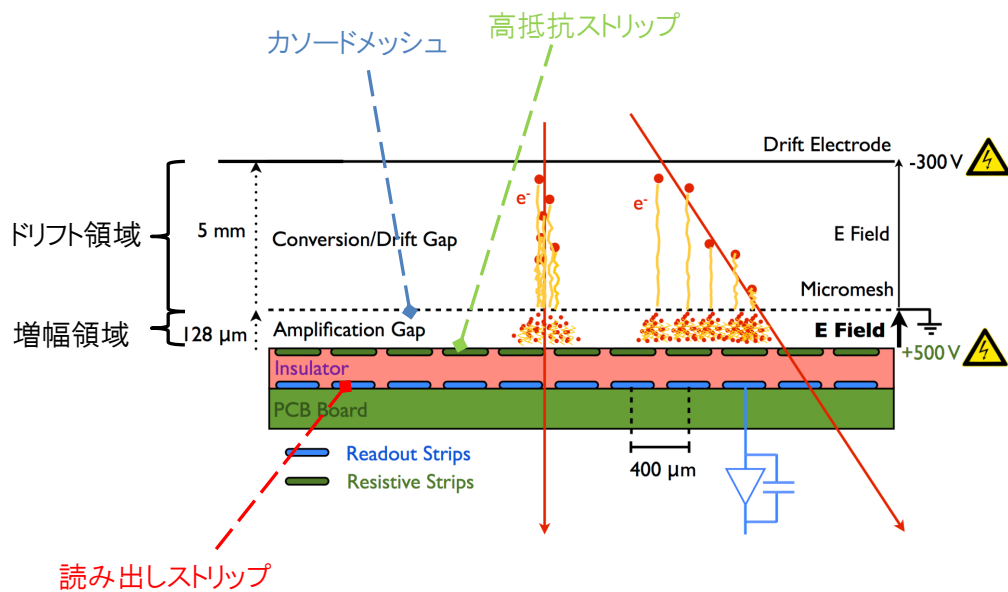


図 3.24: MicroMegas 検出器の断面図 [20] : 5 mm のドリフト領域と、128  $\mu\text{m}$  の増幅領域があり、その境界にカソードメッシュが存在する。ドリフト領域には Ar と  $\text{CO}_2$  が 93 : 7 の割合で充填されている。増幅領域の下、読み出しストリップの上面には放電を防ぐための高抵抗ストリップが並んでいる。高抵抗、読み出しストリップは同じ間隔で 400  $\mu\text{m}$  のストリップが並んでおり、ラージセクタは 0.450 mm 間隔、スモールセクタは 0.425 mm 間隔で並んでいる。

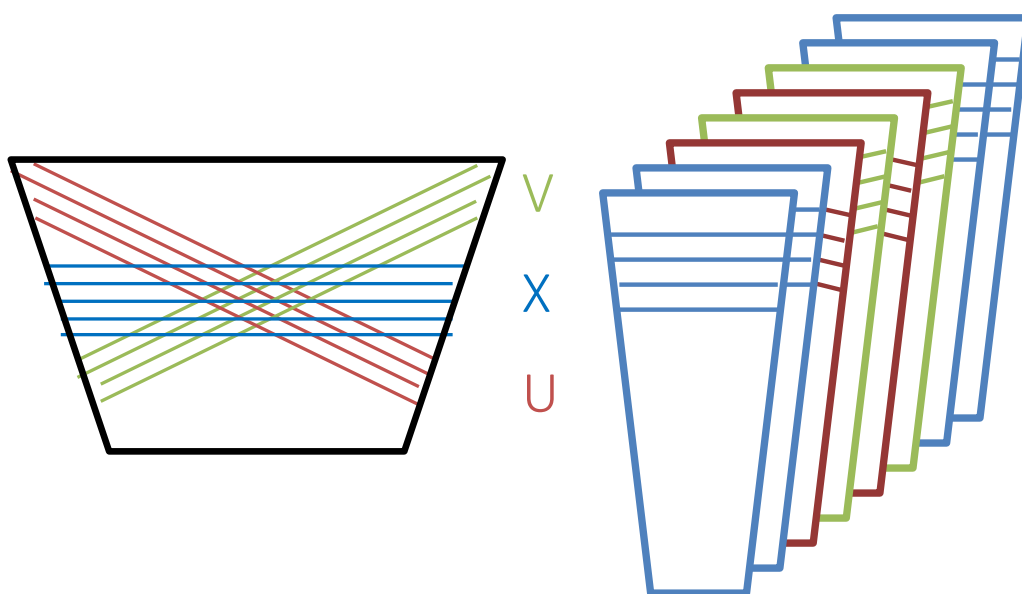


図 3.25: MicroMegas 検出器のストリップの配置 : MicroMegas 検出器は水平方向のストリップ層 X (図中青)4 枚と、 $1.5^\circ$  だけ傾けたステレオストリップ層 U(図中赤) と V(図中緑) がそれぞれ 2 枚ずつで構成される。配置は、X が外側に 2 枚ずつあり、U と V が内側に交互に並ぶ。(XXUVUVXX)



また NSW で使用される sTGC は、図 3.26 のような構成で、sTGC の各パラメータを表 3.5 に載せる。sTGC の、現行の TGC と異なる点は、sTGC はストリップピッチが 3.2 mm と小さく、TGC の早い応答だけでなく、NSW に必要な約 100  $\mu\text{m}$  の高い位置分解能が得られることである [17]。

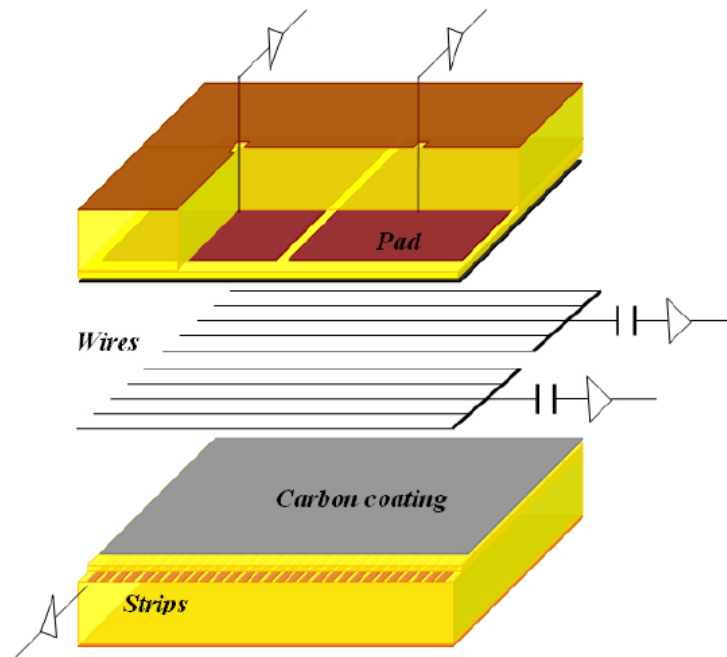


図 3.26: sTGC の断面図 [21]

表 3.5: sTGC の各パラメータと設計値 [21]

パラメータ	設計値
ワイヤ間隔	1.8 mm
ガスギャップ	2.8 mm
ストリップ-カーボンギャップ	0.1 mm
ストリップピッチ	3.2 mm
ストリップ間隔	0.5 mm
カソードプレートの表面抵抗	100 $\text{k}\Omega/\text{sq}$

NSW を用いたトリガーロジックを図 3.27 に示す。NSW では MicroMegas と sTGC でそれぞれトリガー処理を行い、独立にトラックセグメントを再構成する。その後重複するトラックセグメントを削除し、NSW における粒子のヒット情報 ( $\eta_{\text{NSW}}, \phi_{\text{NSW}}$ ) から、衝突点からの直線と NSW で検出したミューオンの飛跡の角度差  $\Delta\theta_{\text{NSW}}$  が得られる。NSL では従来のトリガーロジックに加えて、NSW から  $\eta_{\text{NSW}}, \phi_{\text{NSW}}, \Delta\theta_{\text{NSW}}$  を受け取って新しいトリガー処理を行う。この NSW からの角度情報を用いたコインシデンス処理は、衝突点由来でないフェイクトリガーを削減することが期待される。



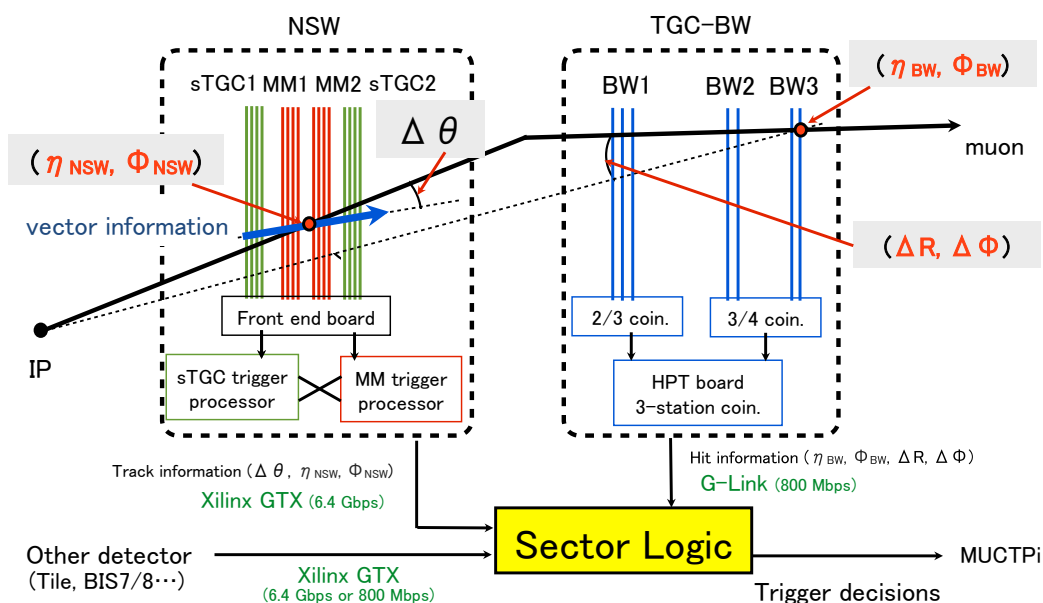


図 3.27: NSW を含めたトリガーロジック [17]:NSW は従来の SW と比べて、衝突点からの直線との角度差  $\Delta\theta$  が得られ、衝突点由来でない粒子によるトリガーを削減する。

### 3.4 エレクトロニクスアップグレード

Run-3 からは NSW が新しく導入され、それに伴ってレベル 1 ミューオントリガーのトリガーロジックが変更することになる。そのためトリガー処理を行う SL が Run-3 で新しく開発される。また SL に TTC 情報を送信する TTC Fanout Module、SL からのトリガー情報を受け取る ROD も新しく変更することになった。この中で、New SL (NSL)、TTC Fanout Module について本節で説明する。

#### NSL

Run-3 に向けて、SL をアップグレードする。NSL は図 3.28 に示すような 9U モジュールである。また SSW で行われていたゼロサプレス処理を、NSL 内で行う。SSW のデータ収集の役割は、ROD のアップグレードによって通信規格が G-link からイーサネットに変更されるため、市販のスイッチに置き換わり、SSW は Run-3 では使用されなくなる。NSL は Run-2 までの SL と同様に、1 つの 1/12 セクターを 3 枚で処理するため、片サイドにつき 36 枚用いられる。

NSL には、BW の High-pT モジュールからデータを受け取るための G-link インターフェースに加え、NSW 及びタイルカロリメータ等内部の検出器からのデータを受け取るために、高速シリアル通信プロトコルである GTX トランシーバが追加される。GTX は NSW 等のデータの受信の他に、MUCTPI にトリガー情報を送る際にも使用する。その他にも TTC 信号を受け取る 16 pin connector、試験用の LEMO IN/OUT やリードアウト用のイーサネットポートが搭載されている。

以下に、NSL に搭載されている集積回路について述べる。

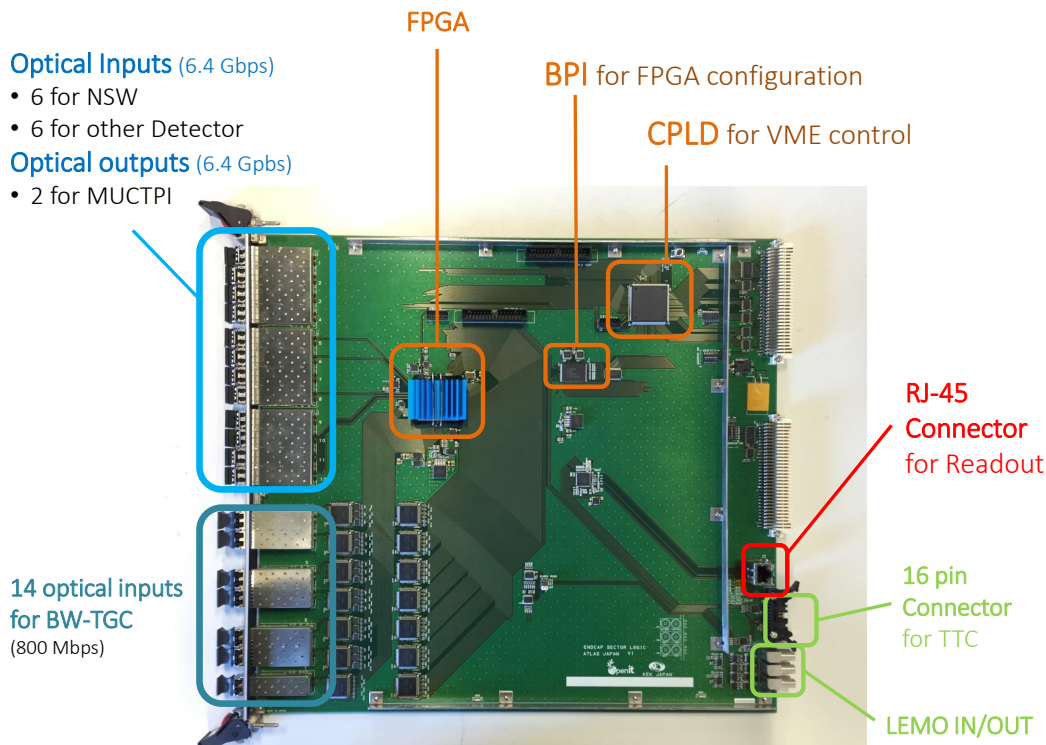


図 3.28: New Sector Logic

- Field Programmable Gate Array (FPGA)

FPGA は Xilinx 社が製造した、書き換え可能な集積回路で、NSL ではメインのトリガーロジックをこの FPGA で行う。

NSL に搭載されている FPGA(Xilinx 社製 Kintex-7 XC7K325T) には GTX トランシーバが搭載されている。

また NSL には KEK Open-It が開発した SiTCP という技術が用いられている。SiTCP は FPGA をイーサネットに接続して通信するための技術である。FPGA 側では FIFO のように扱い、PC からは TCP サーバのように扱うことができ、ハードウェア、ソフトウェア双方の開発、運用を容易にすることができる。加えて SiTCP は FPGA 内に実装し、ハードウェアで処理が行われるため、イーサネットの処理速度の上限でデータ通信を行うことが可能である。

- Byte-wide Peripheral Interface (BPI)

BPI は不揮発性のメモリで、FPGA の回路情報を記憶し、電源を投入したときに FPGA にその情報を書き込む。

- Complex Programmable Logic Device (CPLD)

CPLD は FPGA と同様の書き換え可能な集積回路である。FPGA より回路規模は小さいが、FPGA と違って電源を落としてもその回路情報が消えないという長所を持つ。

NSL では VME バスを経由して BPI、FPGA の制御を行うために用いる。

図 3.29 に、NSL のロジック図を示す。トリガーロジックでは従来のように  $R$ 、 $\phi$  コインシデンスを行って  $p_T$  を決定した後、位置情報ははじめとした NSW からの情報によるトリガー処理が追加される。トリガー判定を行った後、トリガー情報や NSW, BW からのデータを NSL 内に

ある Level-1 Buffer に格納し、L1A 信号が来た際に、それに対応するイベントデータを ROD に送る。送る際には、従来 SSW で行っていたゼロサプレス処理を NSL 内で行い、データサイズを小さくした後で SiTCP という通信プロトコルを用いて ROD に送信する。

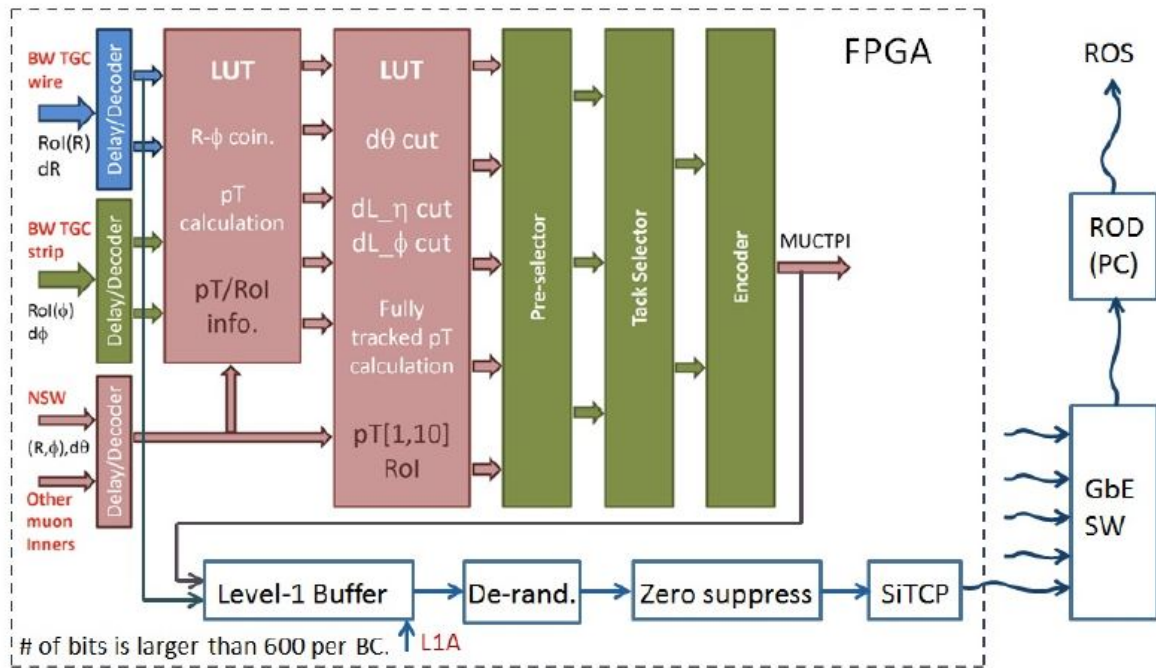


図 3.29: NSL のトリガー及びリードアウトのロジック図 [22]: NSL では従来のトリガー処理に加えて、ROD までのリードアウトの処理も行う。トリガーロジックは、従来の  $R-\phi$  コインシデンス、 $p_T$  測定に加え、NSW からの情報を使った新しいコインシデンス処理が導入される。

## TTC Fanout Module

TTC Fanout Board は図 3.30 に示すような TTC 信号を NSL に分配し、ROD に TTC 情報を送信するためのモジュールである。TTC Fanout Module から NSL へは従来通りフラットケーブルを用いて送信するが、ROD へはイーサネット通信を用いて TTC 情報を送る。

図 3.31 に TTC Fanout Board のリードアウトのブロック図を示す。TTC Fanout Board は 2 つのチャンネルに分けられた TTC 情報を受け取る。TTC Fanout Board では A-channel、B-channel から受け取った情報をそれぞれの FIFO (FIFO1, FIFO2) に格納し、1 イベント分の情報が揃ったら FIFO3 にデータを格納する。

## Readout Driver

NSL の仕様変更に伴って、読み出しの電子学も大きく変更することとなる。その中でデータの収集を行う ROD はハードウェアによる処理からソフトウェアベースに変更することになった。ソフトウェアに変更することでデータ収集ロジックの作成、改良が容易になり、PC で動作するようになるため市販のインターフェースを導入することができるようになる。

ROD については 5 章でより詳細に説明する。

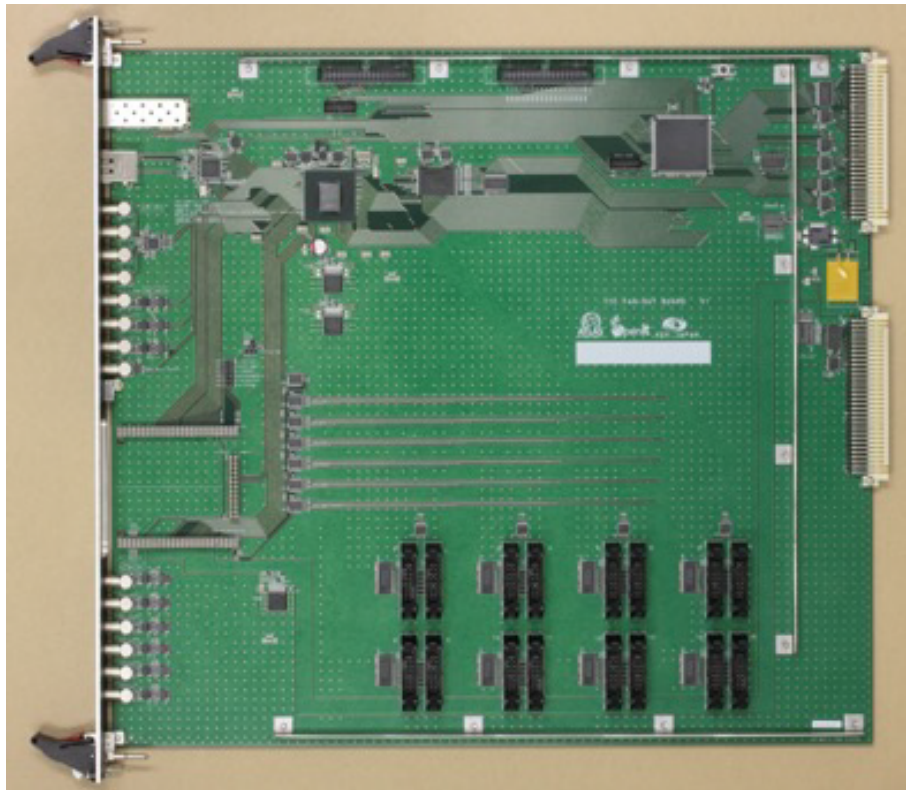


図 3.30: TTC Fanout Board の写真

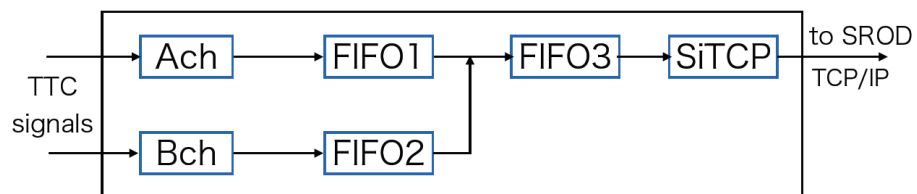


図 3.31: TTC Fanout Board のリードアウトスキーム

# 第4章 TDAQ system

本章では TDAQ のシステムについて説明する。

## 4.1 TDAQ Application

TDAQ では各モジュールを実際に動かし、制御するためのソフトウェアが用意されている。ATLAS では検出器の各モジュールの制御には、Run Control Application というものを使用している。Run Control Application は図 4.1 に示すような GUI アプリケーションで、ここから各モジュールの On/Off の設定、データ取得の開始、停止を行うことができる。

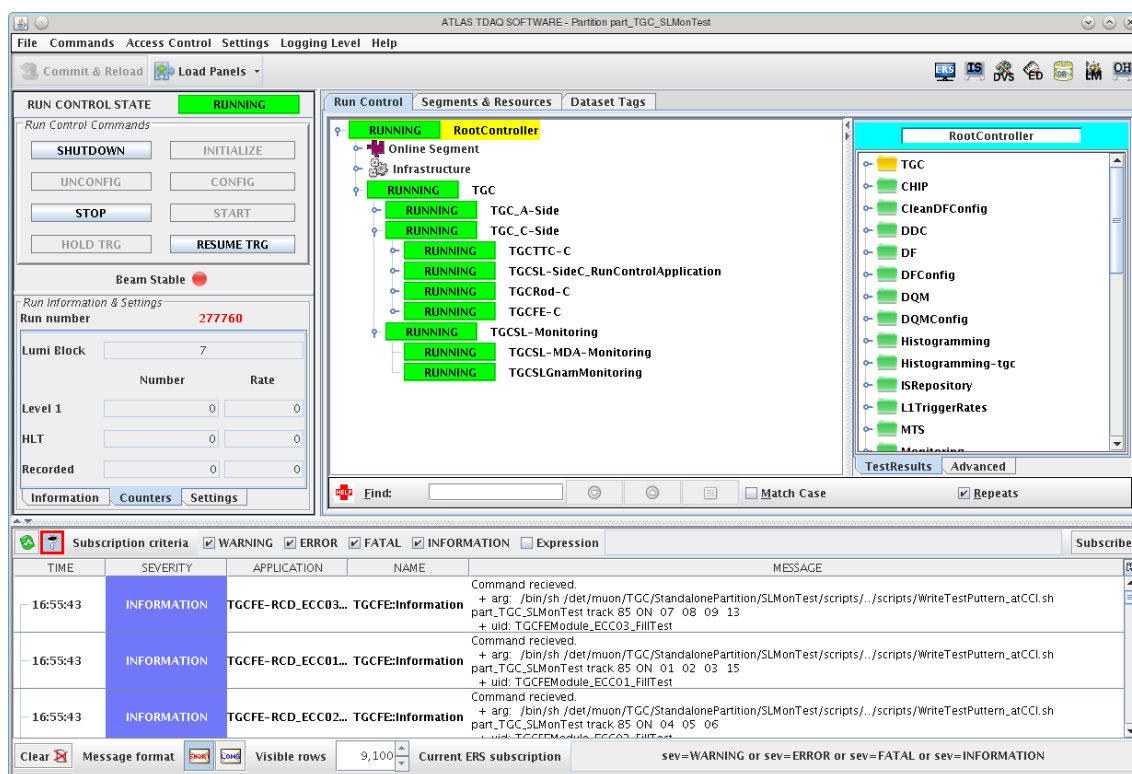


図 4.1: Run Control Application

ATLAS のシステムは partition と呼ばれる領域で分けられ、Run Controller では指定した partition 内のシステムのみ動かすことができる。partition は試験の際に他のシステムに影響しないようにするために設けられている。

partition の中ではさらに segment と呼ばれる構成要素に分けられる。この segment の中にはまた別の segment を設けたり、実際に動作する resource がまとめられている。

segment は全体としてツリー構造になっている。例として、現在の TGC のシステムの構造を図 4.2 に示す。



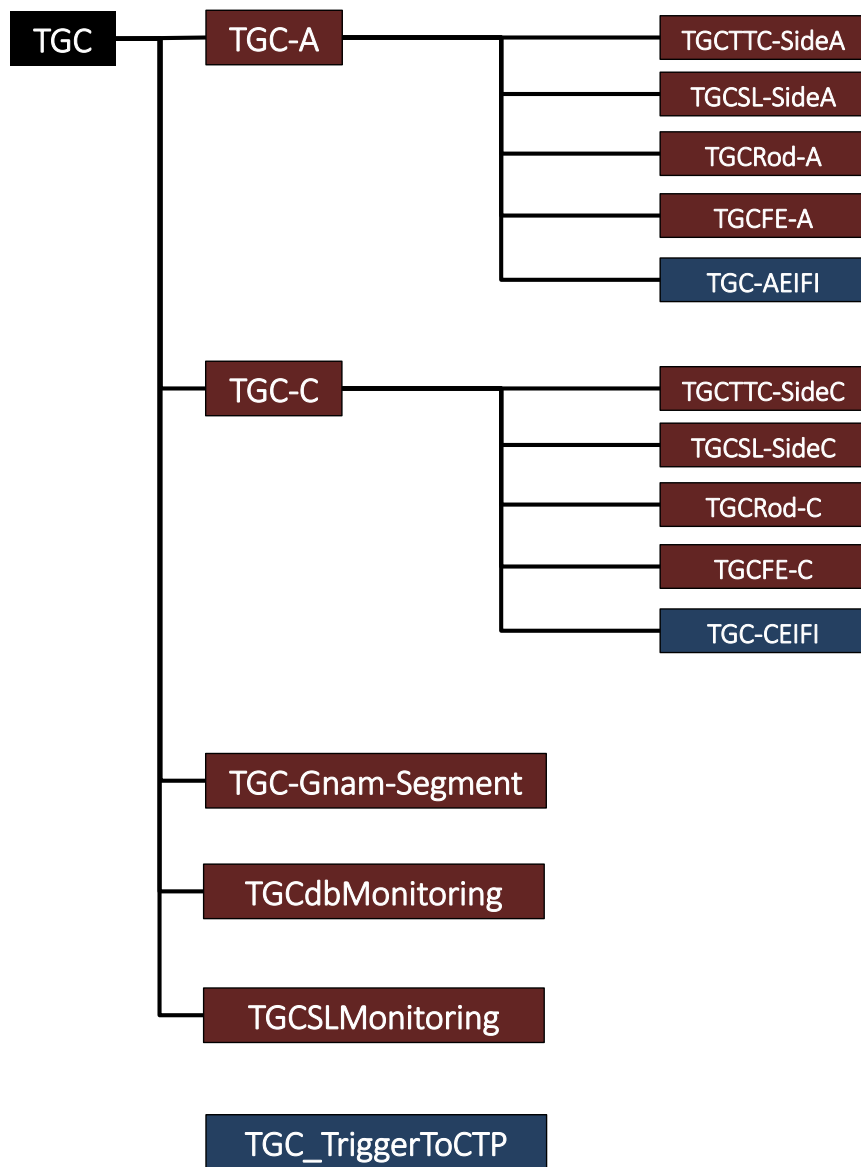


図 4.2: TGC segment: 図中赤で示されているものが segment, 青で示されているものが resource である。segment の中には resource や別の segment が入る。各 segment, resource の説明は本論文では割愛する。

また segment の中で動作する resource には様々なものがある。

- Resource Application  
基本的に TDAQ のコマンドに影響せずに動作するソフトウェア。起動、停止のタイミング (どのコマンドが入力されたときに終了するか) は設定ファイルで指定することができる。
- Run Control Driver (RCD)  
Run Controller から TDAQ コマンドを受け取り、対応した処理を行うソフトウェアである。segment のステートはこの RCD のステートを示していて、コマンドを受け取ると、各アプリケーションにコマンドを分配する。

システムを TDAQ のシステムの下で動かすためには、TDAQ で定義されたステートマシンにしなければならない。ステートマシンとは状態と状態間の遷移、及びそのときに行う動作を示すものであり、状態を表す「ステート」と、遷移を引き起こす「コマンド」からなる。図 4.3 に、TDAQ で決められているステートマシン図を載せる。はじめは NONE の状態で、boot コマンドを受け取ると起動して INITIAL の状態になる。configure コマンドを受け取るとアプリケーションは DAQ のための設定を行い、それが完了したら CONFIGURED の状態になり、そこから連続して外部のシステムとの接続を確立する connect コマンドを実行し、それが完了した CONNECTED 状態になる。

start コマンドを受け取ることで、DAQ を行う RUNNING の状態になる。また一時停止、再開のための pause、continue コマンドと、一時停止の PAUSED の状態も用意する必要がある。

さらに stop コマンドも段階分けされており、これは複数のシステムを動作させるときに、上流のシステムから順番にシステムを停止させる必要があるためである。

## 4.2 Information Service (IS)

IS は Run 中に各モジュールが取得した情報を保存するシステムである。

アプリケーションは IS にアクセスし、IS 内のパラメータを参照したり、更新することができる。

IS で保存された情報はソフトウェアで取得することができ、Online Histogramming Server (OHS) でリアルタイムにヒストグラムを作成することも可能である。

## 4.3 Error Reporting Service (ERS)

ERS は TDAQ アプリケーションのログやエラー報告等、メッセージに関する API を提供しているサービスである。

表示される情報には、アプリケーションやメッセージの内容に加え、メッセージの重要性を示す Severity がある。Severity にはレベルの低い順に DEBUG、INFO、WARNING、ERROR、FATAL があり、Severity に応じてメッセージの処理を分けることが可能である。表 4.1 に、本研究の SROD における Severity 毎の処理とその説明を載せる。stdout と stderr はログファイルへ出力することを表す。ログファイルは標準出力用 (stdout) と標準エラー出力用 (stderr) で 2 つに分けられている。message transport service (mts) は Run の最中にアプリケーションの様子を確認するためのメッセージを表示するためのシステムで、例えば図 4.1 の下部に表示される。throw は C++ の例外処理の機構に則り、FATAL のメッセージを表示した後に例外処理に以降させることができる。

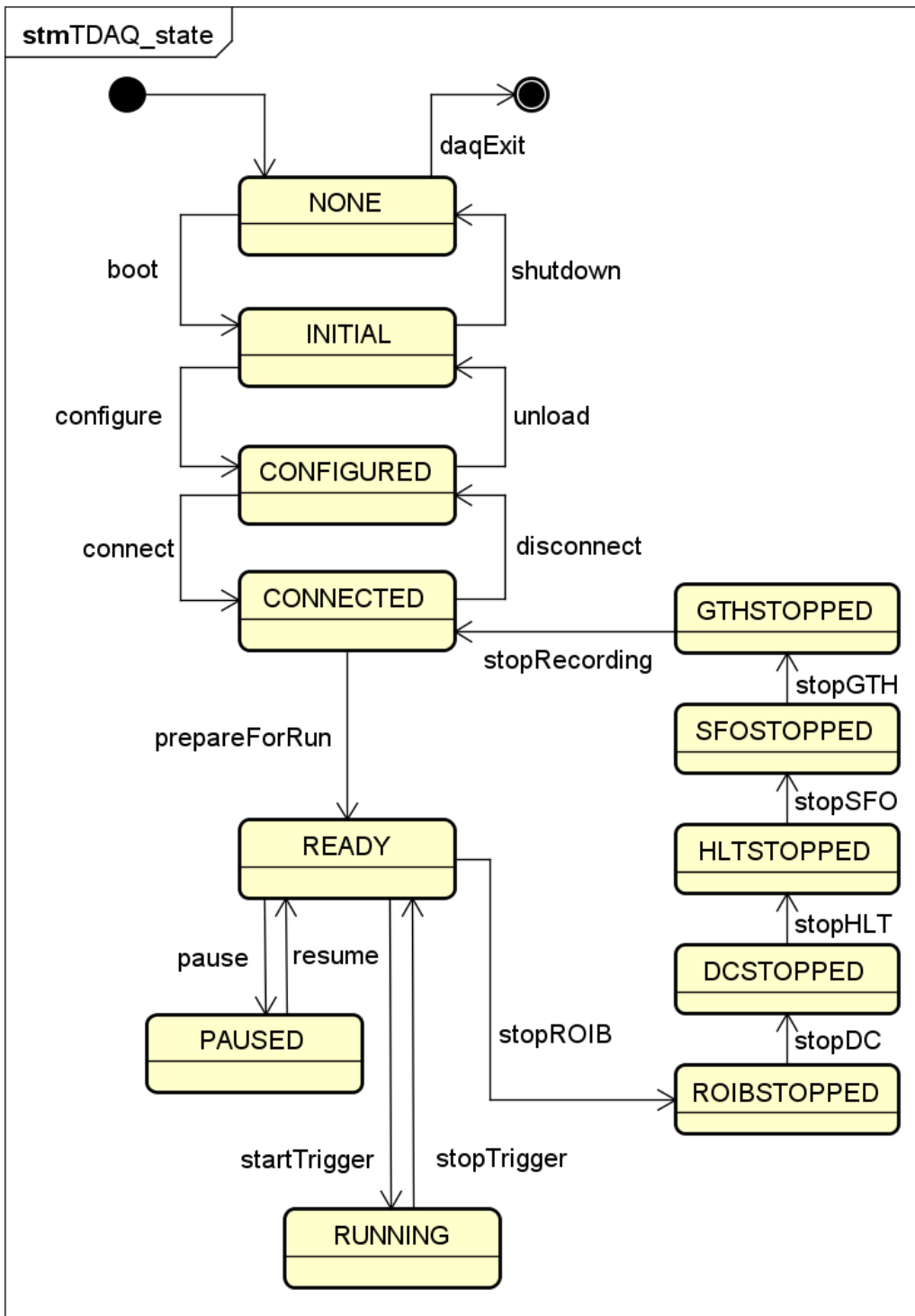


図 4.3: TDAQ のステートマシン図



表 4.1: SROD における Severity 毎の処理の一覧: 左表は SROD で設定している処理を示して  
いて、右表はその中の処理の説明を簡潔に示している。

Severity	処理	処理	説明
DEBUG	stdout	stdout	標準出力
INFO	stdout, mts	stderr	標準エラー出力
WARNING	stdout, mts	mts	メッセージを GUI に表示
ERROR	stdout, stderr, mts	throw	プログラム内で例外を投げる
FATAL	stdout, stderr, mts, throw		

## 4.4 Object Kernel Support (OKS)

OKS は各 partition や segment 等、DAQ における様々なパラメータやその関係性を示すためのライブラリである。

OKS の基本はオブジェクトと呼ばれるもので構成され、オブジェクトは Attribute と Relation で構成される。

Attribute はオブジェクトの持つ基本的なパラメータで、整数型、小数型、文字列といった基本的な型を持つパラメータがこれに該当する。

Relationship はオブジェクト間の関係を示すパラメータで、Relationship に該当するパラメータはクラスを型に持ち、パラメータには関連するオブジェクトの数と、そのオブジェクトが格納される。

OKS のデータ構造には Schema と Data が存在する。Schema には Data で使用するためのクラスの定義が記述されており、partition や segment といったクラスの構成を示している。

Data にはモジュールの機能、設定や他のクラスとの関係性が記述されていて、実際の partition や segment 等の設定を書いたファイルは Data ファイルになる。OKS のデータファイルは図 4.4 に示すような xml ファイルで記述されている。各オブジェクトはオブジェクトのクラス名と、オブジェクトの名前となる “id” を設定する必要がある。そして schema ファイルに従って Attribute と Relationship を記述する。

```

<obj class="RunControlApplication" id="MuonBT_SROD-RunControl">
  <attr name="InterfaceName" type="string">"rc/commander"</attr>
  <attr name="IPCName" type="string">"</attr>
  <attr name="ActionTimeout" type="s32">120</attr>
  <attr name="ProbeInterval" type="s32">5</attr>
  <attr name="FullStatisticsInterval" type="s32">60</attr>
  <attr name="IfError" type="enum">"Error"</attr>
  <attr name="ControlsTTCPartitions" type="bool">0</attr>
  <attr name="Parameters" type="string">"</attr>
  <attr name="RestartParameters" type="string">"</attr>
  <attr name="Logging" type="bool">1</attr>
  <attr name="InputDevice" type="string">"</attr>
  <attr name="InitTimeout" type="u32">120</attr>
  <attr name="ExitTimeout" type="u32">5</attr>
  <attr name="StartIn" type="string">"</attr>
  <attr name="RestartableDuringRun" type="bool">0</attr>
  <attr name="IfExitsUnexpectedly" type="enum">"Error"</attr>
  <attr name="IfFailsToStart" type="enum">"Error"</attr>
  <rel name="InitializationDependsFrom" num="0"></rel>
  <rel name="ShutdownDependsFrom" num="0"></rel>
  <rel name="Program">"Binary" "rc_controller"</rel>
  <rel name="ExplicitTag">" " "</rel>
  <rel name="Uses" num="0"></rel>
  <rel name="ProcessEnvironment" num="0"></rel>
  <rel name="RunsOn">"Computer" "pctgc-jdaq-01.cern.ch"</rel>
</obj>

```

図 4.4: OKS データファイルの一例

# 第5章 ソフトウェアベースの Readout Driver の開発

本章では本研究対象である、Run-3に向けたソフトウェアベースの ReadOut Driver (SROD) について説明する。

LHC Run-3では、加速器の性能を向上させ、ルミノシティを増やす予定である。ルミノシティが向上すれば、より稀な物理事象を探索することが可能になり、新物理の探索の可能性が高まる。しかし、単にルミノシティを増やしただけではバックグラウンドとなるイベントの頻度も同様に増える。そのためトリガーの性能を向上させ、背景事象の削除を行う必要がある。そのため Run-3からは新しいトリガーロジックを用いるため NSL を導入する。

ROD はイベントデータを保存するためのシステムであるが、本研究における ROD は、NSL からのトリガー情報を収集するためのものである。トリガー情報は物理解析には直接用いられるものではないが、物理データを取得するためのトリガーの性能を評価するために記録する必要がある。記録したトリガー情報からシステムが正常に働いているかを判断し、実装されたトリガーロジックを評価する。さらに、トリガー条件を改良してトリガー精度を向上させるためにもトリガー情報の収集は極めて重要である。

## 5.1 SROD の開発の目的

Run-3では、ハードウェアが変更されるため、リードアウト系もそれに伴って変更する必要がある。しかし、現在のハードウェアモジュールである ROD を利用することは、処理速度の面でも、適応する電子部品の欠如という面でも不可能である。また、ハードウェアの開発には製造・評価のターンラウンド期間が必要で、2020年に間に合わせる事が難しい。そのため Run-3 における ROD は、PC を用いたソフトウェアベースで開発することとした。ソフトウェアベースにする利点は、汎用的な機器を取り入れて開発することができること、今後の改良、修正が容易になり、将来の技術の進展に柔軟に対処できることが挙げられる。本研究では、TCP/IP や共有メモリといったプロセス間通信を用いた SROD の開発を行う。

## 5.2 Run-3 におけるリードアウトスキーム

図 5.1 に、Run-3 におけるリードアウトシステムの構成を載せる。

TTC Fanout Board は TTC から信号を受け取ると、それを各 NSL に分配する。各 NSL は、TTC から信号を受け取ると、その L1A 信号に対応したイベント情報を PC に送信する。それと同時に TTC Fanout Board は、SROD に TTC 情報を送る。NSL、TTC Fanout Board と SROD 間のデータ転送は Giga bit Ethernet を介した TCP/IP 通信を用いる。TTC Fanout Board と NSW からのデータ・コンセントレータには市販の 10 GbE スイッチを使用し、イベントビルディングは PC で行う。PC で作成したイベントデータは S-link という特別なインターフェースを介して後段の ROS に送信する。

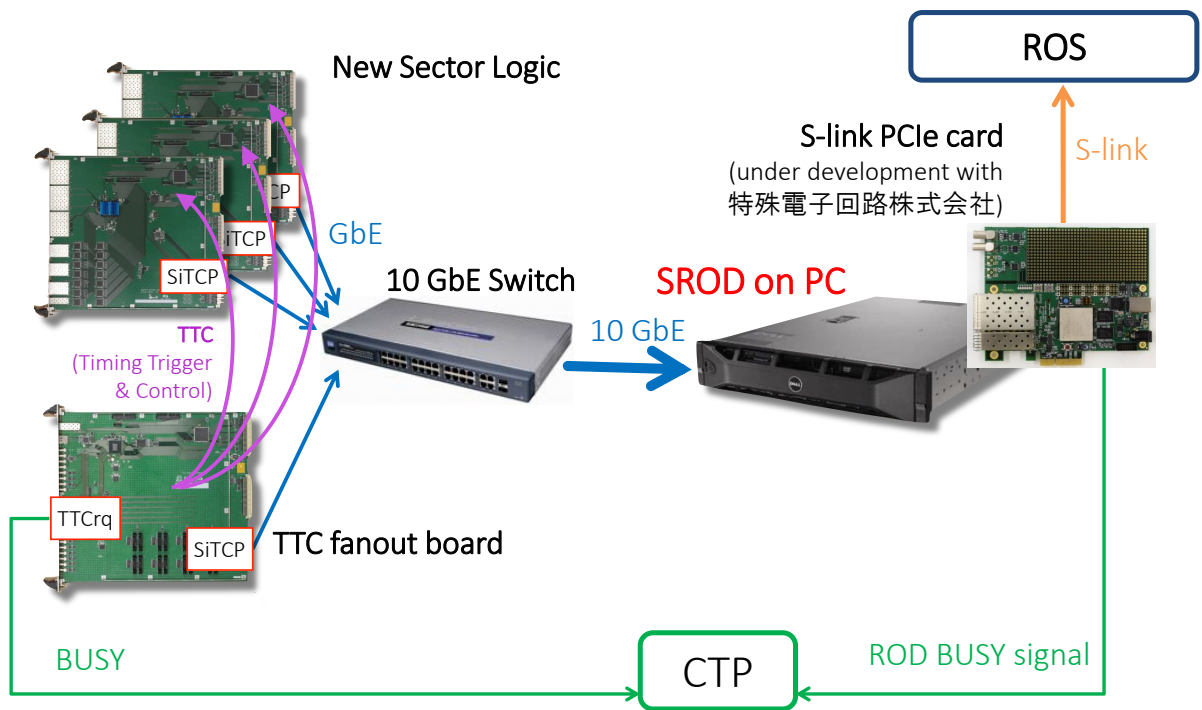


図 5.1: リードアウトシステムの構成

### 5.2.1 10 GbE スイッチ

Run-2 までは専用のモジュール (SSW) を使用していたが、Run-3 ではゼロサプレスの処理を NSL で行い、データ・コンセントレーターには市販のスイッチを用いる。本システムでは高速なデータ収集が求められるため、NSL は GbE を実装するが、SROD とスイッチの間では転送速度を稼ぐため、10 GbE を用いるため、10 Gbit Ethernet に対応したスイッチを選ぶ。

NSL は片サイドにつき 36 枚用いられ、SROD はこれらの NSL からデータを集める必要がある。NSL を搭載する VME クレートには 20 枚しか挿せず、クレートには最低でも Single Board Computer (SBC) と呼ばれる制御用のボードと TTC Fanout Board を搭載しなければならない。その他特別なモジュールをいくつか実装することを考えると、NSL は 1 クレートに 18 枚挿すことは厳しい。そのため SROD は PC1 台につき、12 台の NSL からデータを収集することを目標とする。

SROD PC に 1 つ、NSL 12 台分、そして TTC Fanout board から 1 台、合わせて 14 個のポートがスイッチには必要である。市販のスイッチにはポート数が 12, 24, 48 個のものがああり、予備も必要であることも含め、本システムでは 24 個のポートを持つスイッチを採用する。

### 5.2.2 PC

PC1 台あたりの SL の処理数によって必要な PC の台数は変わってくる。1 台の PC が 12 台の SL を処理すると、SL が全部で  $36 \times 2$  枚あるので 6 台の PC で処理することである。

また高速処理以外に PC に必要な性能は、10 GbE インターフェースを持ち、後述する PCI express card をインストールできる PC であることが求められる。本研究では、ソフトウェアも開発段階であるため、まだ PC の性能は完全に決まっておらず、SROD の性能評価には全て 1 台の PC で試験を行った。表 5.1 に、本研究で使用した PC のスペックを記載する。

表 5.1: 本研究で使用した SROD PC の各パラメータ

項目	種類
OS	SLC 6.7 64-bit
CPU	Intel(R) Xeon(R) CPU E5-2620 v3 , 2.40 GHz
コア数/スレッド数	6 C / 12 T
メモリ	8 GB

### 5.2.3 PCI express card

ROD は収集したデータを規定のフォーマットに変換した後、S-link [23] というインターフェースで ROS というシステムにそのデータを送信する必要がある。さらに ROD の処理が一杯になった際には TTC に NIM 規格信号で ROD BUSY 信号を送らなければならない。ROD が PC ベースになるため、これらを実現するための専用の PCI express カードも新しく開発する。

現在この PCI express card は特殊電子株式会社が開発を進めていて、その写真を図 5.2 に載せる。

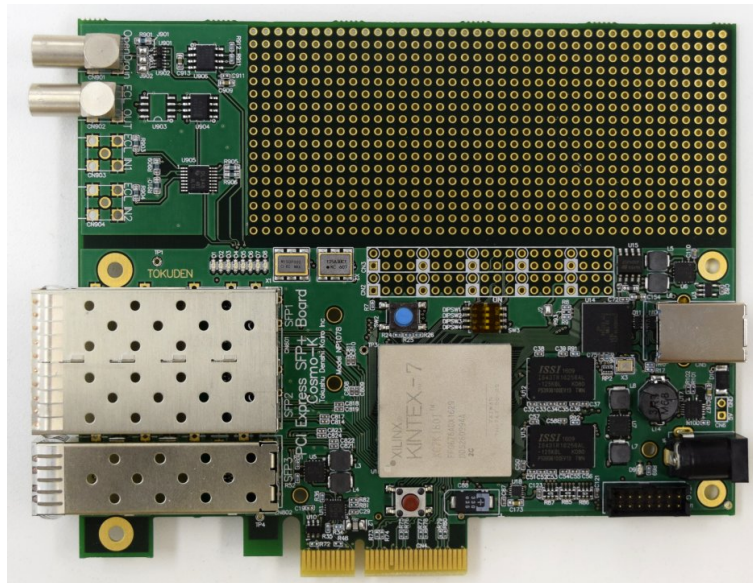


図 5.2: PCI Express Card: CTP に BUSY 信号を送るための NIM と ROS に S-link でデータを送るための光ポートが実装されている。

### 5.3 SROD に要求される性能

NSL は片サイドあたり 36 台設置される予定で、3 台で 1/12 セクターのトリガーを処理する。SROD は 1 台あたり最大 12 台の NSL と TTC Fanout Board からデータを収集する。データサイズの大きさについて、1 word = 4 byte という定義で説明する。TTC から送られるデータサイズは 5 ワードで一定であり、それを L1A 発行頻度である 100 kHz で送るため、TTC から送られるデータレートは

$$5[\text{word}] \times 32[\text{bit/word}] \times 100[\text{kHz}] = 16[\text{Mbps}] \quad (5.1)$$

である。

NSL からのデータは、GTX の受信は 1 レーンあたり 128 bit/BC であるが、16 bit はイベントヘッダであるため、送るデータは 112 bit/BC である。NSL には GTX のレーンが 12 本あり、全て使用した場合は 112×12 bit/BC となる。また NSL の Trigger 判定結果は 1 トリガーセクターあたり 80 bit/BC で、NSL は 1 枚で 2 トリガーセクターを処理するので 160 bit/BC となる。G-link からのデータフォーマットは 2 セクター分合わせて 202 bit で、これらを合わせると 1 バンチあたりのデータ量は

$$(112 \times 12) + 202 + 160 = 1706[\text{bit/BC}] \quad (5.2)$$

となる。NSL はヒットのあったデータと、その前の 1 バンチ分と後の 2 バンチ分のデータを含めて合計 4 バンチ分のデータを送るので

$$1706 \times 4 = 6824[\text{bit}] \quad (5.3)$$

となる。ここからゼロサプレスによる圧縮が行われる。圧縮率を  $10^{-3}$  とすると [24]、平均のデータサイズは

$$6824 \times 10^{-3} = 6.824[\text{bit}] \quad (5.4)$$

となる。これに NSL フォーマットのヘッダ、フッタのサイズを加えると、NSL からのデータのレートは

$$(6.824 + 96)[\text{bit}] \times 100[\text{kHz}] = 10.284 \simeq 10.3[\text{Mbps}] \quad (5.5)$$

である。

そのため SROD に求められる処理速度  $T$  は以下の式で表される。

$$T = 16.0 + 10.3 \times n + 54.4[\text{Mbps}] \quad (5.6)$$

$n$  は NSL の台数で、3 項目は ROB フォーマットへの変換に必要な転送速度である。表 5.2 に片サイドに配置する PC 1 台あたりに必要な転送速度をまとめる。

表 5.2: 片サイドあたりの PC の台数とそのときに必要とされる転送速度

PC の台数	NSL の台数	転送速度 [Mbps]
3	12	194.0
4	9	163.1
6	6	132.2
9	4	111.6
12	3	101.3

また SROD はイベントデータを作成する際に TTC,SL のイベント ID の確認を行う必要がある。TTC と SL は L1A 信号を受け取ると、それに対応するイベントデータを SROD に送信する。このとき担当するセクターにヒット情報がなかった SL からも SROD にデータを送信する。

また SROD には BUSY に対する処理が求められる。BUSY とは大量のデータが送られてきて ROD の処理が追いつかない状態のことで、このとき ROD は CTP に BUSY 信号を送り、L1A 信号の発行を停止させる。

### 5.3.1 TTC, SL からの入力フォーマット

表 5.3: TTC からの入力フォーマット

Words (32 bit)	31-24	23-16	15-8	7-0
Word-0	Header (0xb0d0)		(0x0000)	
Word-1	(0x0)	L1ID (lower 12 bits)	(0x0)	L1ID (lower 12 bits)
Word-2	(0x00)	ECRID (8 bits)	(0x00)	BCID (8 bits)
Word-3	(0x0)	Orbit ID (lower 12 bits)	(0x0)	Orbit ID (lower 12 bits)
Word-4	(0x00)	Trigger Type (8 bits)	Footer (0xe0d0)	

表 5.3 に、TTC からの入力フォーマットを示す。TTC から送られてくるデータフォーマットは一定の長さであり、その中に格納されているデータについて順に説明する。

- Level-1 ID (L1ID)

L1ID は L1A の発行によってカウントアップするイベント ID である。L1ID は TTC から ECR 信号を受け取るとリセットされる。SL の ID との識別には下位 12 bit のみを使用する。

- ECR ID  
L1ID のカウントアップのリセットを行った回数を示すイベント ID である。
- Bunch Crossing ID (BCID)  
BC Clock に同期してカウントアップするイベント ID で、L1ID と同様に BCR 信号を受け取るたびにリセットされる。
- Orbit ID  
LHC 一周分のバンチ衝突の回数をカウントしたイベント ID である。
- Trigger Type  
L1A が発行されたときに、どのような条件で発行されたかを示すパラメータである。

表 5.4: SL からの入力フォーマット

Words (32 bit)	31-24	23-16	15-8	7-0
Word-0	Header (0xb0d0)		(0x0)	L1ID (12 bits)
Word-1	(0x0)	BCID (12 bits)	(0x0)	SLID (12 bits)
Word-2– Word-(N–1)	zero-suppressed variable size data (4 bunches of Trigger Data)			
Word-N	(0x0000)		Footer (0xe0d0)	

また表 5.4 に、NSL からの入力フォーマットを示す。NSL からのフォーマットは、データ部分がゼロサプレスによって大きさが変わるため、データサイズは一定ではない。

- L1ID、BCID  
L1ID、BCID は TTC と別で、NSL 毎にそれぞれ数えられており、この L1ID、BCID が一致した情報が ROD に送られる。そのためこの L1ID、BCID は TTC のものと必ず一致しなければならない。
- SLID  
どの SL からのデータであるかを判別するための ID である。
- Data  
MDT、TGC からのトリガー情報が格納されている。このデータ部分はゼロサプレスという圧縮処理を行っているため可変長である。SL のデータサイズはゼロサプレスがない状態だと 6824 bit である。

NSL から送られるデータサイズは、ヘッダが 2 ワード、フッタが 1 ワードで、データブロックには最大 101 ワード入るため、データサイズの最大は  $2+101+1=104$  ワードとなる。

### 5.3.2 SROD で作成する出力フォーマット

SROD から出力するフォーマットは図 5.3 に示すようにヘッダ、フッタのブロックは ATLAS グループで決められており、各検出器で決めるのはデータブロックとステータスブロックである。

図 5.4 に、SROD が作成する出力フォーマットを載せる。



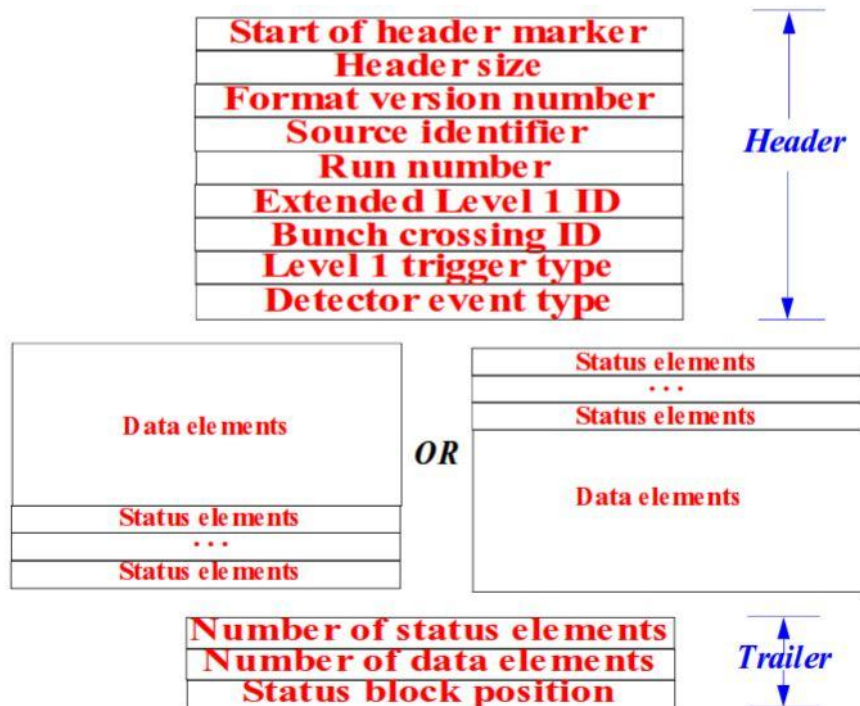


図 5.3: ATLAS で定められている ROB フォーマット:ヘッダ、フッタは格納する値は ATLAS で決められているが、Data ブロックと Status ブロックは各検出器で決められる。

ROB Format				
Words (32-bit)	31-24	23-16	15-8	7-0
Frame	frame word (0xb0f01234)			
Header-0	Header (0xee1234ee)			
Header-1	(0x0000)	header size = 9		
Header-2	ATLAS format version = 3.1		TGC format version = 4.0	
Header-3	(0x00)	0x67 or 0x68 (A/C side)	(0x00)	application ID (1-3)
Header-4	Run Type	Run number		
Header-5	Extended Level-1 ID			
Header-6	(0x0000)	Bunch Crossing ID		
Header-7	(0x000000)			Trigger Type
Header-8	Detector event Type			
Data	(0x0000)	SLID		
	SL data word-0			
	SL data word-1			
	⋮			
	SL data word-N			
Data	(0x0000)	SLID		
	SL data word-0			
	SL data word-1			
	⋮			
	SL data word-N			
Status	first status word; specific/generic			
Status	SL socket disable bits		SL socket timeout or dropped status	
Status	orbit count			
Trail-0	number of status elements = 3			
Trail-1	number of data elements			
Trail-2	status block position = 1, i.e. status follows data			
Frame	frame word (0xe0f01234)			

図 5.4: SROD が作成する出力フォーマット

## ヘッダ

- Start of header marker  
データの先頭を示す word で、TGC ROD の場合は 0xee1234ee で決められている。
- Header size  
ROB フォーマットでは、header size は 9 word で固定である。
- Format version number  
ROB フォーマットのバージョン情報を示す。
- Source Identifier  
上位 16 bit には A side か C side かを判別するパラメータで、A side なら 0x0067、C side なら 0x0068 である。また下位 16 bit には複数の ROD PC のうちどの PC であるかを示す application ID が格納される。
- Run Number  
実験の回数を示すパラメータで、上位 8 bit に Run の種類 (pp 衝突、PbPb 衝突など) を示す Run Type、下位 24 bit に Run Number をそれぞれ格納する。
- Extended Level-1 ID  
上位 8 bit には ECR ID を、下位 24 bit には L1ID を格納する。どちらも TTC から受け取った値を格納する。
- Bunch Crossing ID  
下位 12 bit のみ使用するパラメータで、TTC から受け取った BCID を格納する。
- Level 1 Trigger Type  
トリガーの種類を示す。これも L1ID、BCID 同様、TTC から供給されたパラメータを格納する。
- Detector Event Type  
現在は使用されていない。

## ステータス

現在は 3 word で構成されている。今後、開発の段階で変更される可能性がある。

- first status word  
そのイベントに対する情報を表す。図 5.5 に、このブロックに入るパラメータを載せる。TTC と SL での BCID のミスマッチが生じた場合、0 bit 目の値を 1 にする。同様に L1ID のミスマッチの場合は 1 bit 目の値を 1 にする。
- SL socket disable links / SL socket timeout or dropped status  
SL とのソケット接続の状態を表す。接続が切れた SL が存在する場合はこのブロックの中で対応する部分を 1 にする。
- orbit count  
TTC から供給される orbit ID を格納する。

first status word		
Bit	31-16 specific	15-0 generic
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.
4		An overflow in one of the ROD internal buffers has occurred. The fragment is incomplete.
7-5		reserved
15-8		reserved

図 5.5: First Status Word

## データ

データ部には NSL からの情報のうち、どの SL からの情報かを示す SLID と、SL からのデータを格納する。

## トレイラ

- Number of status elements  
Status ブロックの word 数を格納する。TGC ROD の場合はこの値は 3 である。
- Number of data elements  
データブロックの word 数を格納する。データブロックは可変長なので、イベントビルディングの際に ROD が数える必要がある。
- Status Block Position  
ROB format の中で、Data ブロックと Status ブロックのどちらが先に来るのかを表すパラメータである。この値が 0 ならば Status ブロックが、1 ならば Data ブロックが先に来る。TGC ROD の場合は Data ブロックが先にしたのでこの値は常に 1 である。

## 5.4 システムの構成

SROD の主な役割は TTC、SL からのデータを高速、正確に受け取り、ROB フォーマットへ変換した後、S-link で ROS に送信することである。これを実現するためのソフトウェアを開発する必要がある。そのため SROD を図 5.6 に示すような構成にする。SROD は高速なデータ処理を実現するため、プロセス間通信を用いたマルチプロセスで処理を並列にし、NSL、TTC データを受け取る Collector、データをまとめる EventBuilder、各プロセスにコマンドを送る ReadoutMaster、各プロセスのログ情報を処理する MessageLogger と処理を分けて実装した。またプロセス間のデータの受け渡しには RingBuffer という共有メモリを用いて高速なデータの受け渡しを行い、その他のコマンド等共有すべき情報については RCMemory という共有メモリを作成した。処理の並列化にはマルチスレッドという選択肢もあったが、こちらはマスターのスレッドが強制的に落ちてしまった際に他の処理中のスレッドもすべて落ちてしまうという危険性があり、1つのプロセスが落ちた時でも他のプロセスが生き残るマルチプロセスよりも安全面で劣ると考えた。

またソフトウェアを動作させるためには、ステートマシンを定義する必要がある。図 5.7 に、ROD におけるステートマシン図を示す。はじめに boot コマンドより各プロセスが立ち上がった

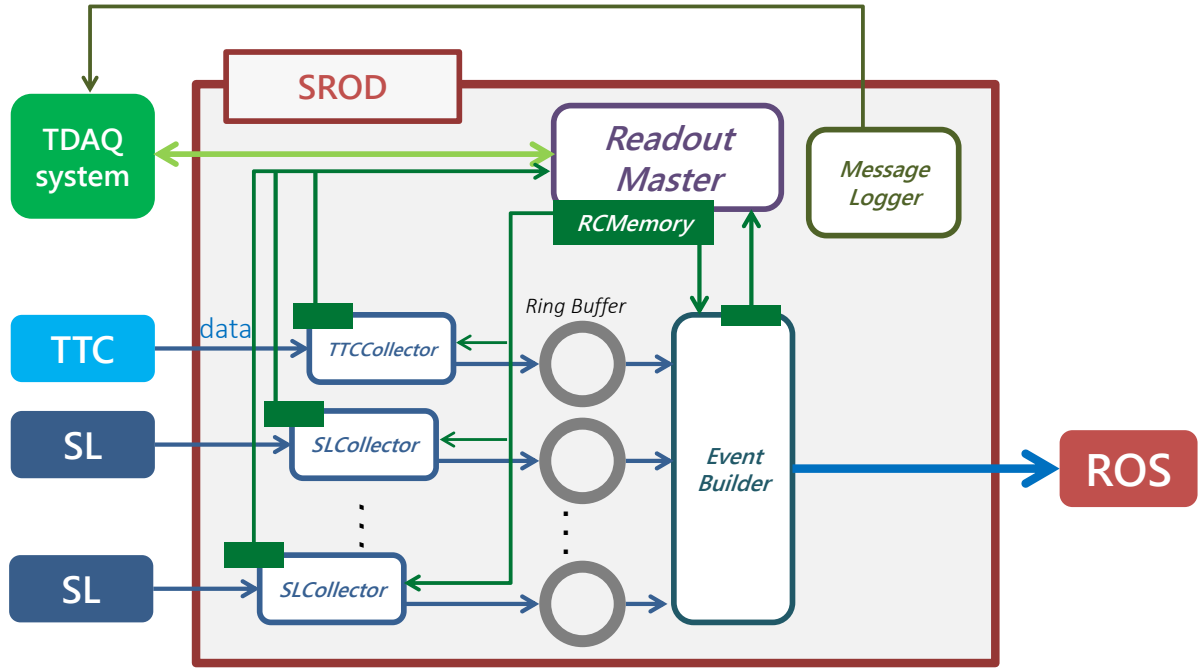


図 5.6: SROD の構成: 図中白い四角で表されているものが SROD のプロセスである。

た INITIAL の状態になり、configure コマンドを受け取ることでバッファの初期化、RingBuffer の設定などを行い、CONFIGURED の状態になる。次の connect コマンドでは、TTC や SL とのソケット接続を行い、それが完了すると CONNECTED の状態になる。その後、start コマンドを受け取り RUNNING の状態となり、この状態の間、ROD は DAQ を行う。停止の際にはこれと逆の手順を踏むことになる。stop コマンドで DAQ を停止し、disconnect コマンドでソケット接続を解除し、unload コマンドで共有メモリの解除を行い、shutdown コマンドでプロセスを終了させる。

これらをふまえて、SROD をオブジェクト指向言語 (C++) で開発を行った。図 5.8 に、メインとなるプロセスである Collector、EventBuilder のクラス図を載せる。各プロセスは ComponentBase というスーパークラスを継承して作ってある。これは ComponentBase でステートに関する処理を行い、システムの動作を統一させるためである。サブクラスの関数に各ステートでのロジックを書き込むだけにするだけでロジック開発を容易にできる点でも効果的である。

### 5.4.1 TTC/SL Collector

TTC、SL から TCP/IP でデータを受け取、そのデータを RingBuffer へ書き込むプロセスである。図 5.9 に、Collector のユースケース図を載せる。

Collector は TDAQ Application によって起動し、OKS や IS からサーバの IP アドレスやポート番号といった必要なパラメータを受け取る。その後 SiTCP サーバとの接続を確立させ、RingBuffer へ書き込み可能な状態にしておく。

図 5.10 に、Collector のメインフローのシーケンス図を載せる。メインフローではサーバからデータを受け取り、そのデータをそのまま RingBuffer へ書き込むという単純な動作をする。RingBuffer への書き込みの際、EventBuilder による読み出しが間に合わず、RingBuffer がいつ

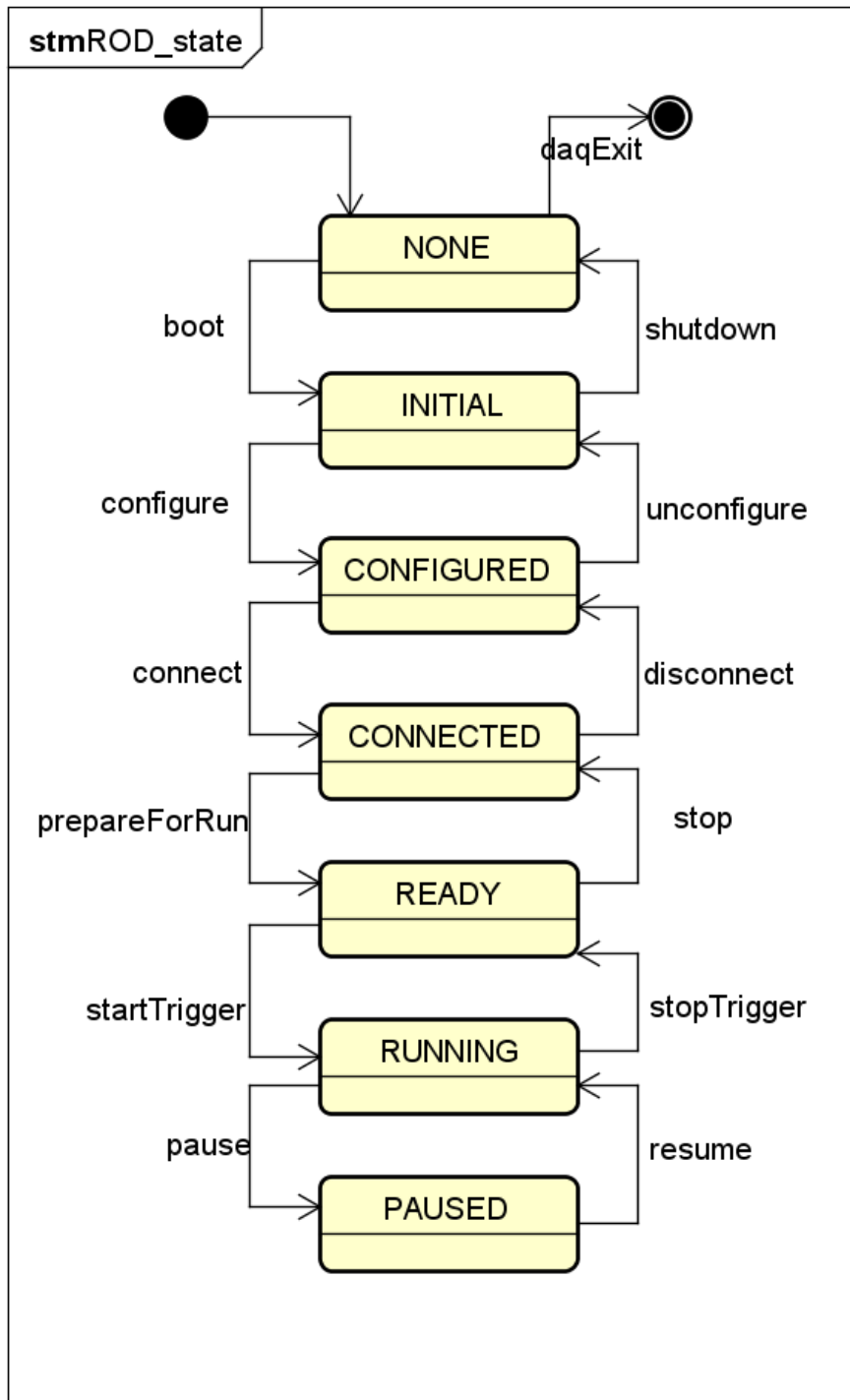


図 5.7: SROD のステートマシン図

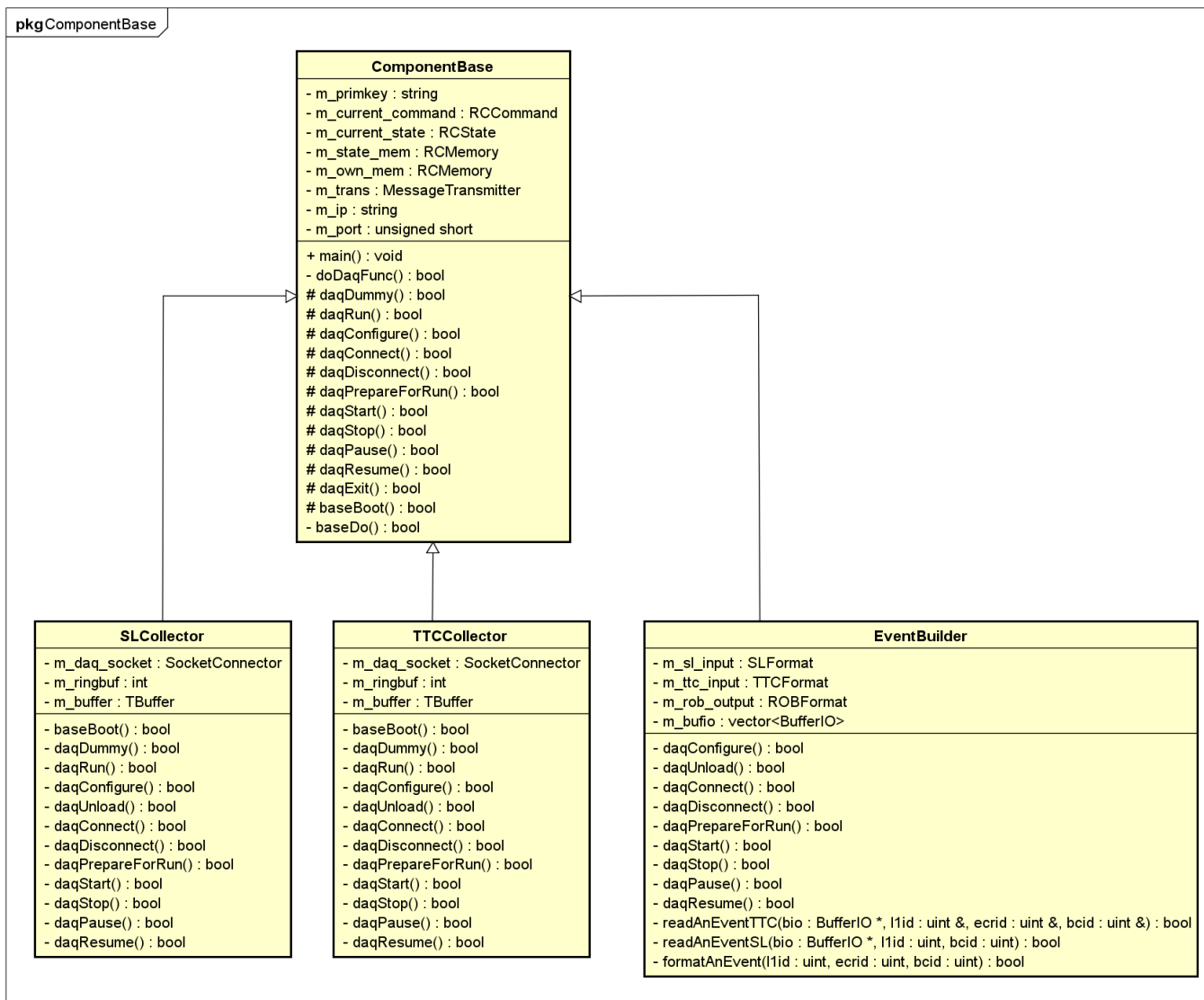


図 5.8: メインプロセスのクラス図

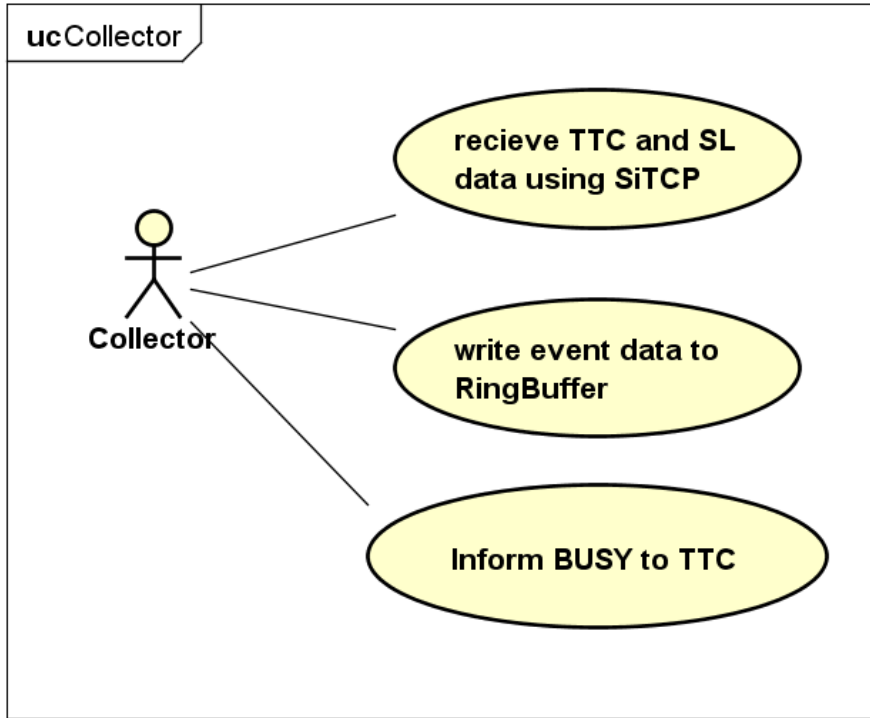


図 5.9: Collector のユースケース図

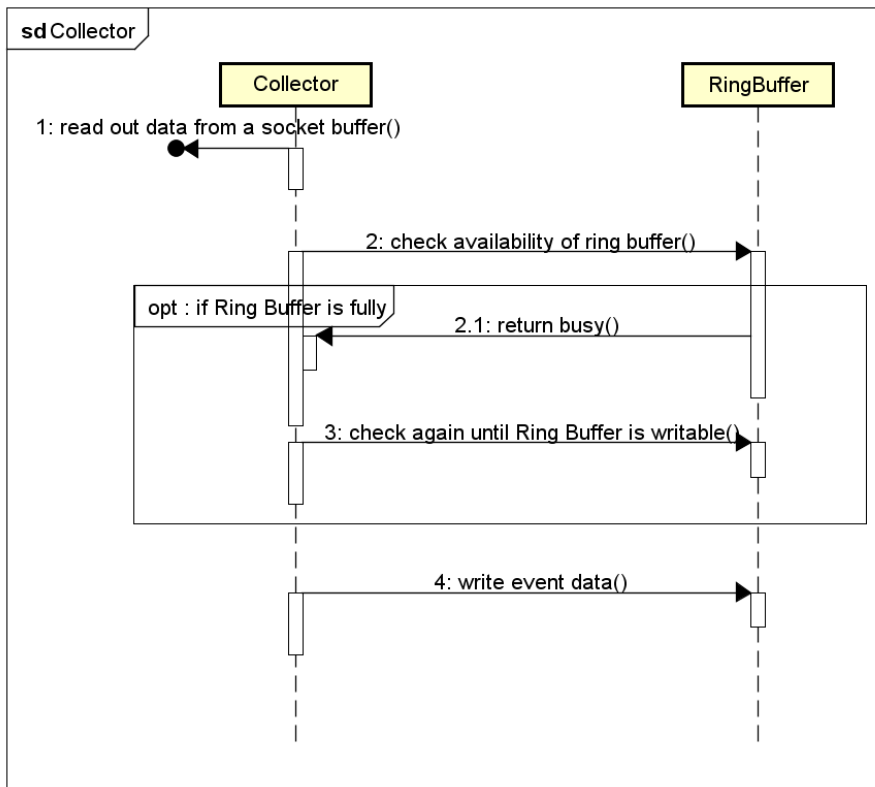


図 5.10: Collector のシーケンス図



ばいになったとき、Collector は RingBuffer へ書き込まず、書き込むための十分なスペースができるまで待機する。

また Collector のもう一つの役割として ReadoutMaster に BUSY を知らせる役割がある。SROD においては、データが一杯になり処理が追いつかない BUSY の状態を、RingBuffer が一杯になる状態と定義している。現在は BUSY のときは RingBuffer に書き込めるようになるまで Collector が待つという仕様になっているが、今後 BUSY を CTP に知らせる機構を作成する必要がある。

## 5.4.2 EventBuilder

TTC、SL それぞれの RingBuffer からデータを集め、ATLAS で決められているフォーマットに変換するプロセスである。図 5.11、図 5.12 に EventBuilder のユースケース図とシーケンス図をそれぞれ載せる。

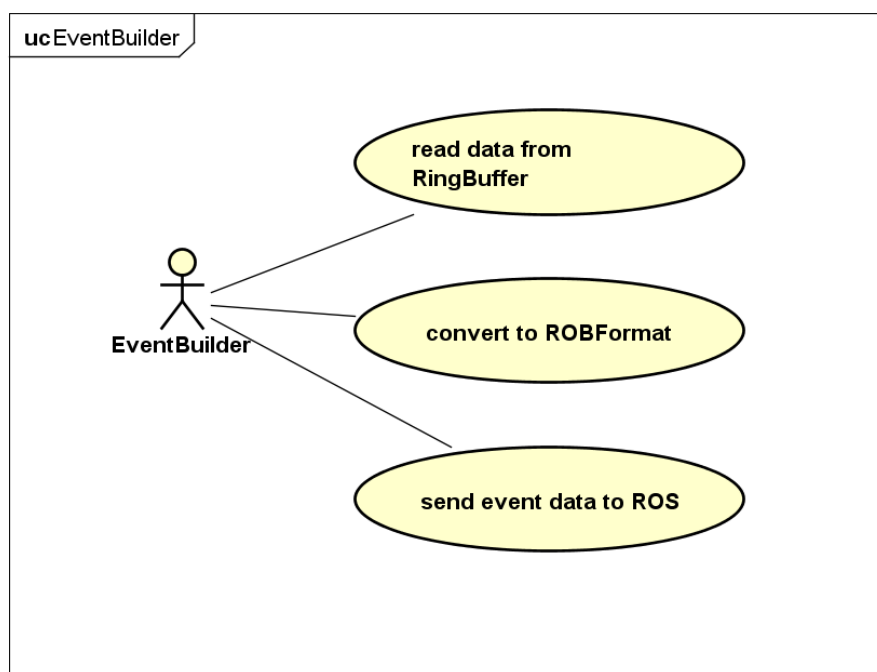


図 5.11: EventBuilder のユースケース図

フォーマット変換の際に、TTC とそれぞれの SL のイベント ID (L1ID、BCID) が一致するかを確認し、もし一致しなければ ID mismatch のエラーを出力する。またフォーマットしたイベントデータを ROS に送るという役割も持つ。これは S-link 用 PCI Express カードがまだ開発段階であったため現在は実装できていないが、今後実装する予定である。

RingBuffer からデータを受け取る際、TTC からの 1 イベント分のデータサイズは必ず一定であるが、SL からのデータサイズは可変長である。このため SL から事前に何 word 分のデータが送られてくるか特定することができない。そのため EventBuilder では RingBuffer からの読み出し用の Buffer を用意しておき、その Buffer で読み出せる分まで読み出す方式をとっている。そしてイベントデータが途中で途切れてしまったイベントは次の読み出しにつなげるように工夫してある。このようにすることで可変長のイベントデータを集めることを可能にしている。

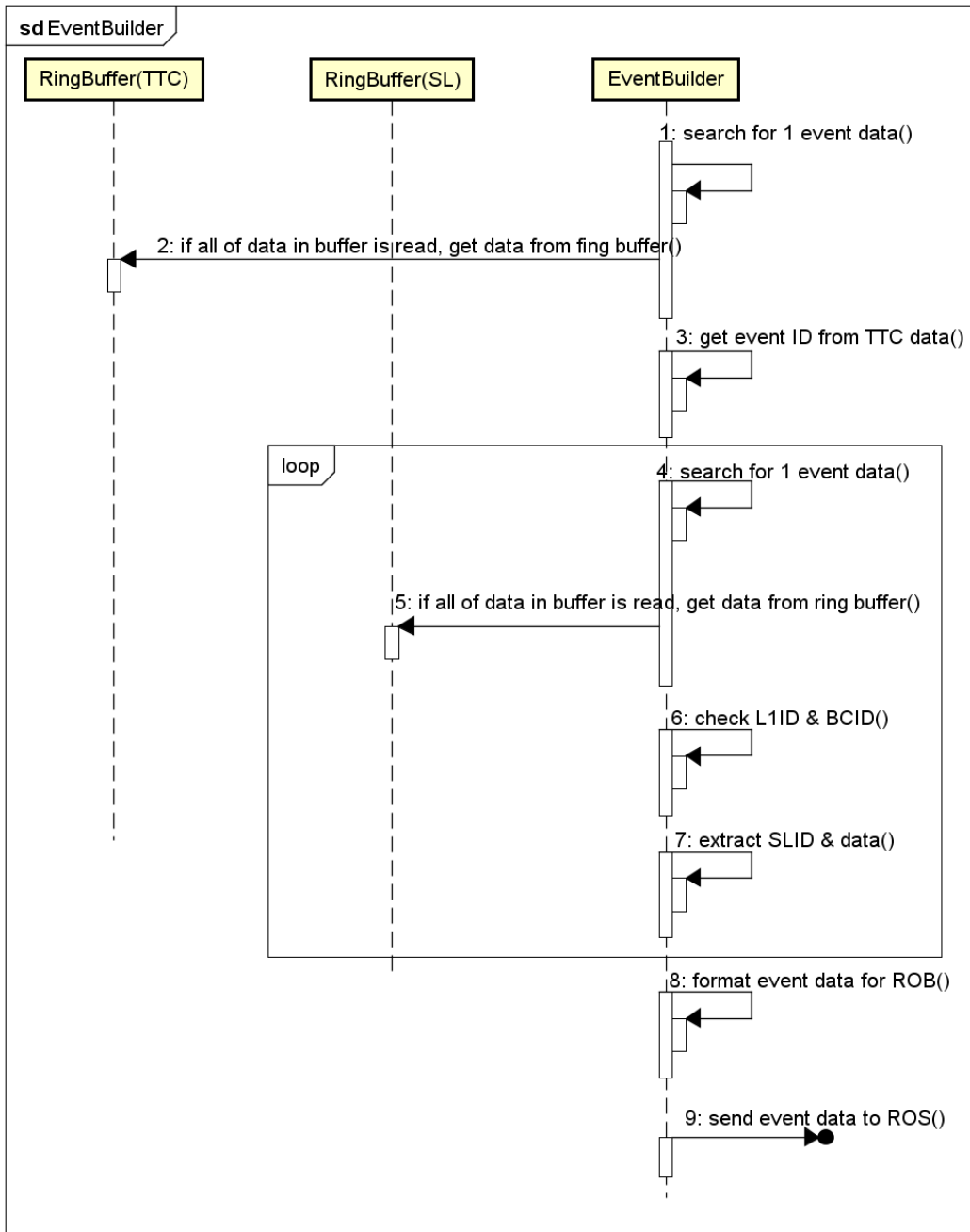


図 5.12: EventBuilder のシーケンス図

### 5.4.3 ReadoutMaster

TTC、SLのCollectorやEventBuilder等、各プロセスにTDAQ Applicationから受け取ったコマンドを送信するプロセスである。また各プロセスが正常に動作しているかを監視するのもこのプロセスの役割で、プロセスの監視やコマンドの伝達はRCMemoryという特別な共有メモリを用いて行う。RCMemoryは後節で詳しく説明する。図5.13に、ReadoutMasterのユースケース図を載せる。ReadoutMasterは他のプロセスと違い、TDAQのステートマシンに従って動作しなければならない。ReadoutMasterはTDAQからコマンドを受け取ると、そのコマンド、ステートに対応したROD独自のコマンド、ステートを各プロセスに伝搬させる。

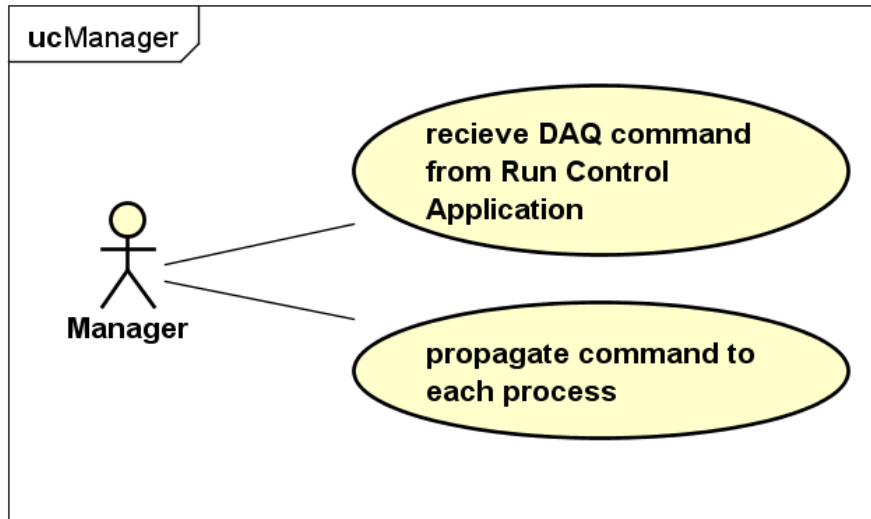


図 5.13: ReadoutMaster のユースケース図

図5.14に、ReadoutMasterから各プロセスへのコマンドの伝達に関するシーケンス図を示す。ReadoutMasterは、TDAQ Applicationからコマンドを受け取ると、自分のRCMemoryにそれに対応したコマンドを書き込む。他のプロセスは、ReadoutMasterのRCMemoryを参照し、新しいコマンドを受け取るとそれに対応した処理を行い、次のステートに移行する。ReadoutMasterは他の全てのプロセスがステートを変更するまで待つ。

### 5.4.4 MessageLogger

それぞれのプロセスの中で問題が発生したりログ情報を出力するためには、TDAQ system上のError Reporting System (ERS)に送らなければならない。RODではそのERSへメッセージを投げるために、MessageLoggerというプロセスを設け、各プロセスはこのMessageLoggerにログやエラーメッセージを送る。

SRODのプロセスがMessageLoggerを介してERSにメッセージを送る理由は、ERSサーバが別マシンで動作するため、各プロセスが外部マシンにメッセージを送り、その応答を待つと時間がかかってしまうためである。そのためSRODと同一のマシンで走り、ERSへメッセージを送るためだけのこのプロセスを用意することで各プロセスの本来のDAQの性能の低下を防ぐためである。

MessageLoggerにメッセージを送る際にはTCPソケットを使用する。各プロセスがMessageLoggerに送るためのフォーマットは図5.15のようにになっている。Severityにはメッセージの重要度、Processにはメッセージを送ったプロセス名が格納される。Severityはそのメッセージの重要

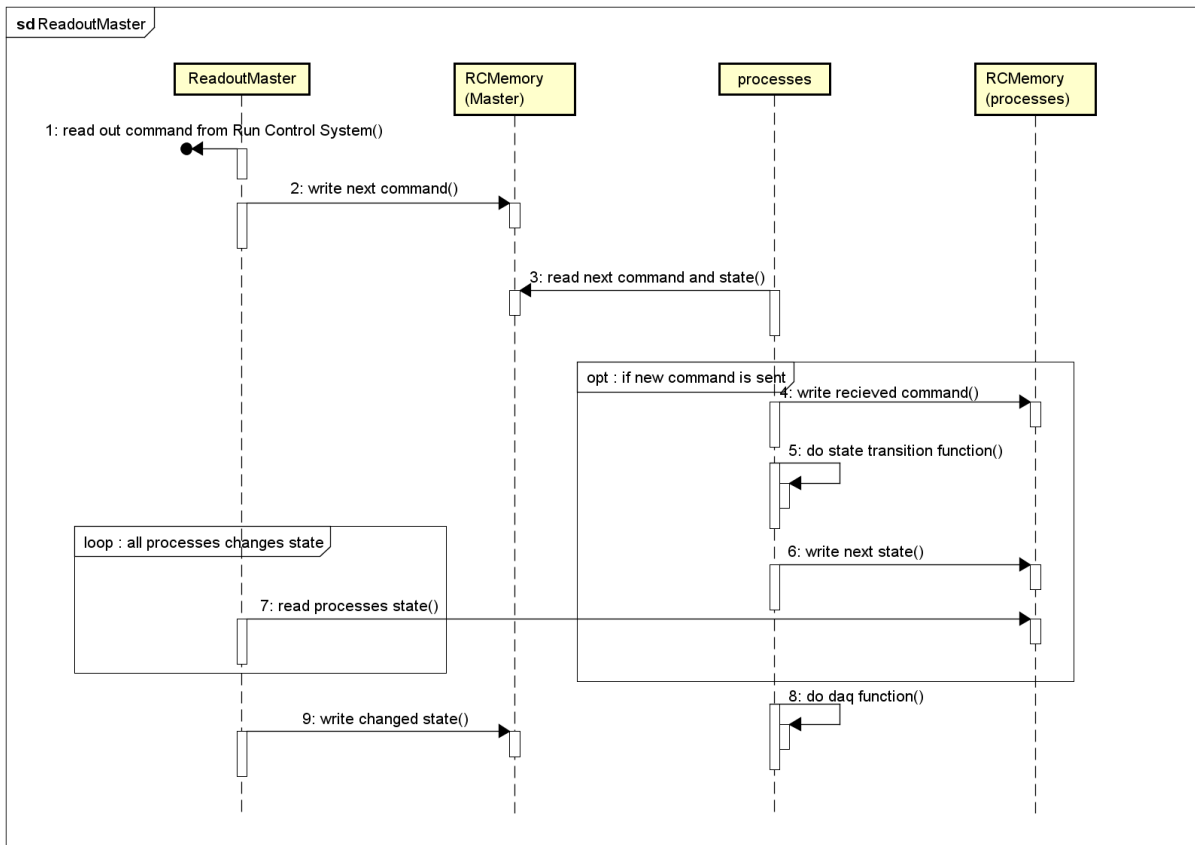


図 5.14: ReadoutMaster からのコマンド伝達のシーケンス図

性に応じて6種類存在しており、開発用のDEBUG、ERSへの出力はないが情報保存用のLOG、ERSへシステムの情報を送信するINFO、システムの危険を警告するWARNING、エラーを表示するERROR、システムの続行が厳しい重大なエラーを示すFATALがある。Severityのレベルに応じてログファイルの書き取りやERSへの表示等、メッセージの処理が決められる。

またfile、lineはデバッグ用のパラメータでメッセージを送ったファイルとその行番号が格納される。CommentにはSeverityのレベルによってフォーマットが微妙に異なり、通常はERSに表示させるメッセージのみだが、ERROR、FATAL等DAQに支障をきたすような場合にはそのときのエラーコードを追加する。

このエラーコードはSRODで独自に決めていて、プログラム上では列挙体で定義していて、その中でも主なエラーコードの説明を表5.5に載せる。

表 5.5: エラーコードの一例

ERROR	エラー番号	内容
ERROR_CANNOT_CONNECT	0x004	ソケット、リングバッファの接続ができなかった
ERROR_CONNECTION_LOST	0x005	ソケット、リングバッファの接続が切断された
ERROR_BUFFER_OVERFLOW	0x101	バッファサイズを超えた不正なメモリアクセス
ERROR_CANNOT_OPEN	0x105	共有メモリを開くことができない
ERROR_ROD_BUSY	0x200	RODの処理が追いつかないBUSYの状態
ERROR_BAD_HEADER	0x201	受け取ったデータのヘッダが正しくない
ERROR_BAD_FOOTER	0x202	受け取ったデータのフッタが正しくない
ERROR_ID_MISMATCH	0x203	TTCとSLとのLIIDまたはBCIDが一致しない

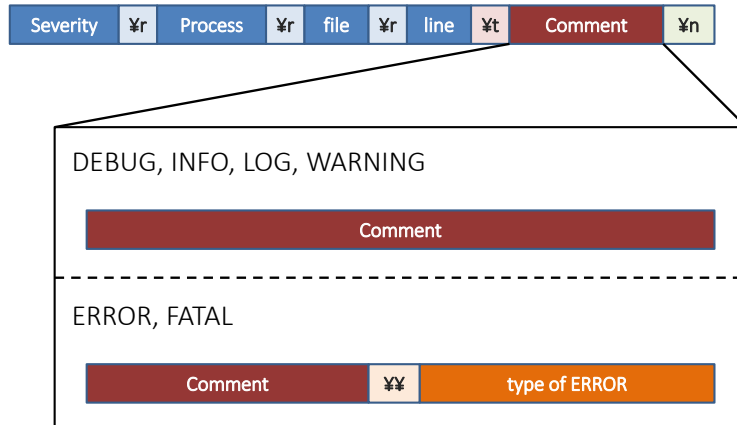


図 5.15: MessageLogger への送信データフォーマット: 各パラメータは特殊文字で区切られており、メッセージのレベルが ERROR または FATAL の場合にはコメントに加えてエラーコードが追加される。

## 5.5 システムの実装

前節で説明したプロセスを、ユースケース図に示した要請を満たすように実装する。そのために必要な技術について本節で説明していく。

### 5.5.1 ネットワークを経由したプロセス間通信

SiTCP サーバとのネットワーク通信のために、Collector ではソケット通信を用いている。ソケットは一般的な OS に標準で備わっている機能で、現在のネットワーク通信において欠かせない存在となっている。ソケット通信は処理を要求するクライアントと、それに応答するサーバの接続によって成り立ち、SROD の場合は SROD(の中の Collector) がクライアント、TTC Fanout と NSL がサーバに当てはまる。

図 5.16 に、ソケット接続の手順を載せる。まずはじめに、クライアント側、サーバ側それぞれでソケットを生成する。次にサーバ側でソケット接続の準備をし、クライアントからの接続の要求が来たらそれを許可することで接続を確立させる。接続したらクライアント、サーバ間でデータのやりとりを行い、通信が終わったら各自ソケット接続を解除する。

### 5.5.2 同一マシン内のプロセス間通信

SROD のプロセスは全て同一のマシンで動作する。これらのプロセス間の通信には共有メモリを使用できる。共有メモリとは複数のプロセスが参照できるメモリで、プロセス間通信の中で最速である。しかし共有メモリの問題として、複数のプロセスが同時にアクセスすると、データ破損の危険性が高いことである。

SROD では 2 種類の共有メモリを使用している。

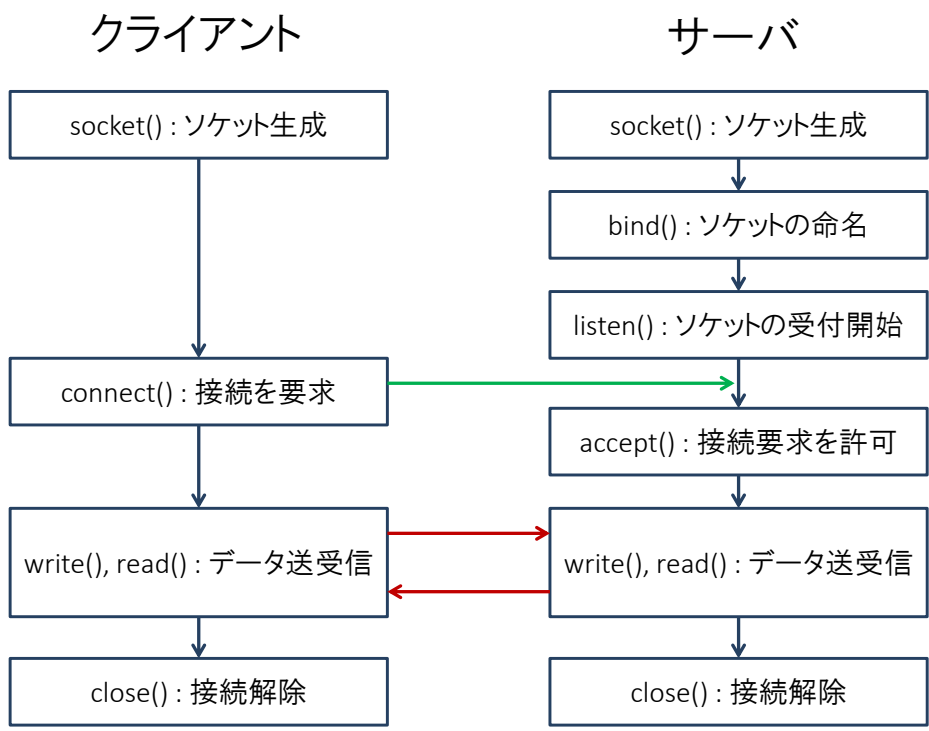


図 5.16: ソケット通信

## RingBuffer

RingBuffer とは、両端を論理的につなげて輪状にしたバッファで、輪状にすることで、永続的な FIFO を実現することができる。

SROD における RingBuffer は図 5.17 に示す通り 2 つの領域に分けられ、それぞれコントロール部とデータ部に分けられる。コントロール部には後述する Mutex と条件変数、そして RingBuffer の読み書きに必要なパラメータが格納されている。以下にその説明を載せる。

- connection\_lost  
RingBuffer の接続が切れたかどうかを示す論理型変数で、接続が切れる際に 1 になる。Run の最中にこのフラグが立ったときにはエラーを返し、エラーハンドリングに移行する。
- Nwriter / Nreader  
書き込み及び読み込みを待つプロセスの数で、RingBuffer が 1 対 1 接続であるため取りうる値は 0 か 1 のみである。この値が 0 の時にはデータ部にそのままアクセスすることができるが、1 の時には条件変数に wait 操作がかけられ、0 になるまで、つまり相手側のプロセスがデータ部への書き込みまたは読み込みが終わるまで待機する。
- Count\_in / Count\_out  
データ部に対して書き込んだ回数及び読み込んだ回数を表す。
- Nword\_in / Nword\_out  
データ部に対して書き込んだ word 数及び読み込んだ word 数の累計を表す。この値をデータ部のバッファサイズで割ることで、どこまで書き込みまたは読み出しをしたかを示すオフセット値を算出できる。

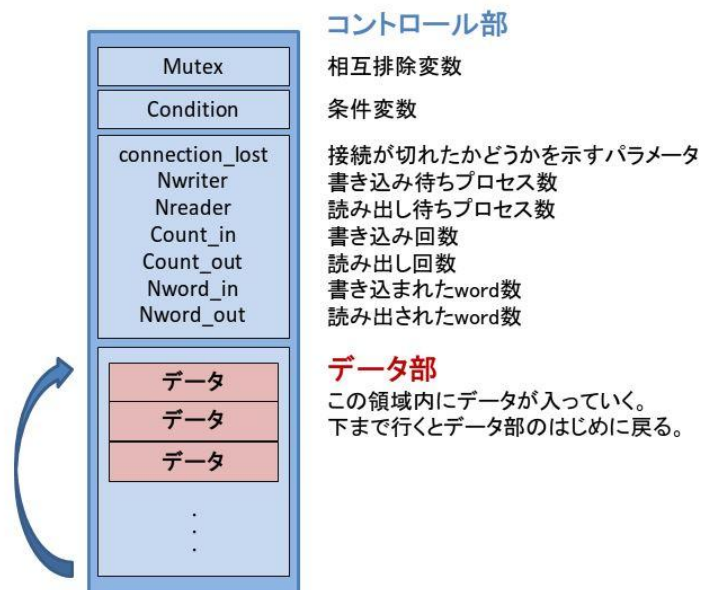


図 5.17: RingBuffer の構成

データ部にデータを書き込む、または読み出す際には、Nword\_in と Nword\_out から書き込み及び読み出しの offset を算出する。書き込みの際には、図 5.18 の左に示すように書き込みの offset に書き込みたいデータを全て書き込んだ場合に、読み出しの offset を超えなければデータを書きこむ。しかし右図のように、書き込みたいデータを全て書き込むと読み出しの offset を超える場合には、書き込み側 (Collector 側) はデータを書きこまずに、読み出しが進むまで待機する。

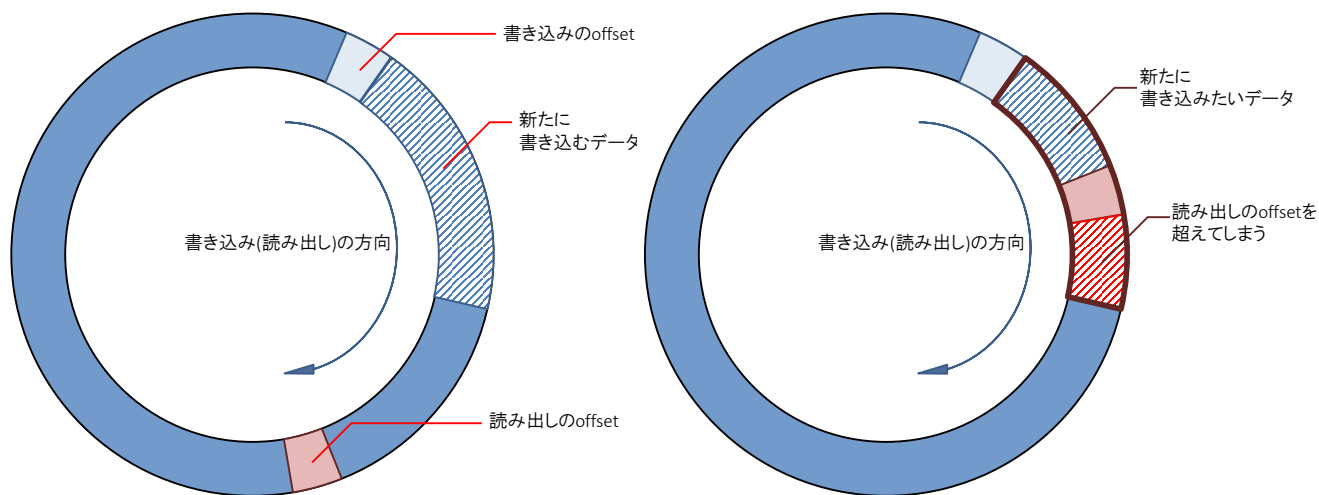


図 5.18: RingBuffer のデータ部:左のように、データの書き込みが読み出しに追いつかない場合はデータを書きこむが、右のように読み出しに追いつく場合にはデータを書きこまず待機する。

## Run Control Memory (RCMemory)

RCMemory は ReadoutMaster から各プロセスにコマンドを送信したり、各プロセスから DAQ の様子を ReadoutMaster に送信するために実装した共有メモリで、RingBuffer と違い、1つの



プロセスのみが書き込めるようにし、他のプロセスは参照のみ可能という設計になっている。データの書き込み操作を1つのプロセスしか行わないため、ロックを用いる必要がなく、コマンドの伝達をスムーズに行える。SRODにおいては図5.6に示すとおり、各プロセスが1つずつRCMemoryを所有しており、ReadoutMasterは他のプロセスのRCMemoryを、CollectorとEventBuilderはReadoutMasterのRCMemoryをそれぞれ読み出し専用でアクセスしている。RCMemoryの構成を図5.19に示す。

前半にはプロセスのコマンド、ステートを格納する。またRCMemoryの後半には、データを書き込むためのスペースを追加できるようにしてある。これは各プロセスからReadoutMasterにパラメータを書き込み、Runの様子をモニターするためのもので、今後の開発及び実際のRunの際に必要なパラメータを適宜変更することができる。

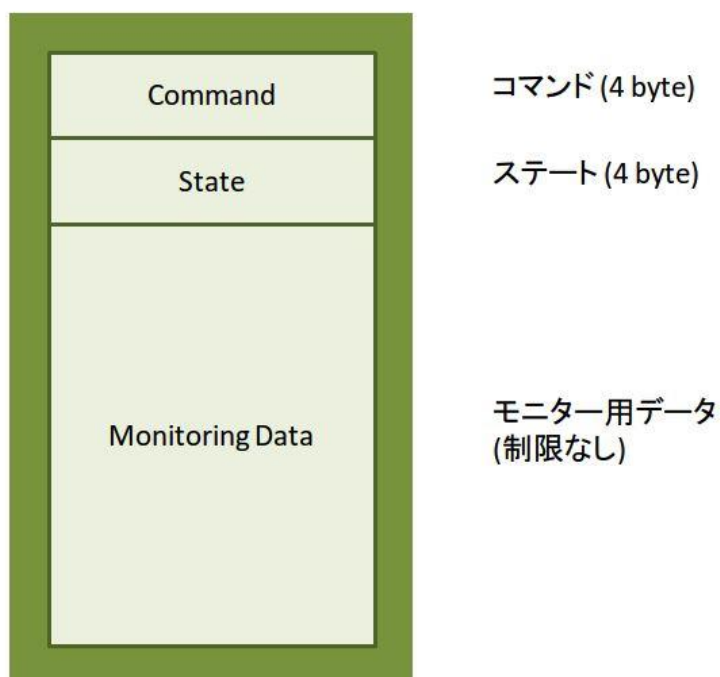


図 5.19: RCMemory の構成

### 5.5.3 ロック機構

Collector と EventBuilder は、RingBuffer を共有し、そこからデータの書き込み及び読み込みを行う。共有メモリはプロセス間通信の中では最速であるが、メモリへのアクセスが競合すると、データが破損するなどの危険を伴う。RingBuffer ではロック機構を使用して複数のメモリが同時にメモリにアクセスするのを防ぐ。

#### Mutex

Mutex とは相互排除変数とも呼ばれ、1つのクリティカルセクションに対して、一つのプロセスだけがその領域にアクセスすることができる機構で、ロック機構の中では至ってシンプルな同期形態である。

## 条件変数

条件変数はデータがある値に達するか、ある特定のイベントが発生するまでプロセスを待機させる機構である。条件変数は通常、Mutex と共に使用され、SROD においてもこの2つのロックを使用する。

# 第6章 ReadoutDriver の性能評価

本研究では TDAQ application に載せた SROD の性能評価を行った。

## 6.1 Test Beam による性能評価

2016年10月に、HL-LHCのための電気回路のアップグレードの試験も兼ねて、テストビームを用いたDAQシステムの動作試験を行った。試験はCERNのPreessinにあるSuper Proton Synchrotron (SPS)のH8ビームラインにおいてミュオンビームを使用して行った。ミュオンビームはspill構造を持っていて、1 spillあたり2000個のミュオンが照射される。

今回のビームテストではNSL、TTC Fanout BoardのファームウェアやSRODとNSL、TTC Fanout Board間の接続の試験、そしてこれらをTDAQシステムにのせてOKSやISからDAQに必要な情報を自動で取得し、実際のRunで使用されているようにRun Controllerから自動でDAQを行う試験を初めて行った。

### 6.1.1 Test Beam の概要、セットアップ

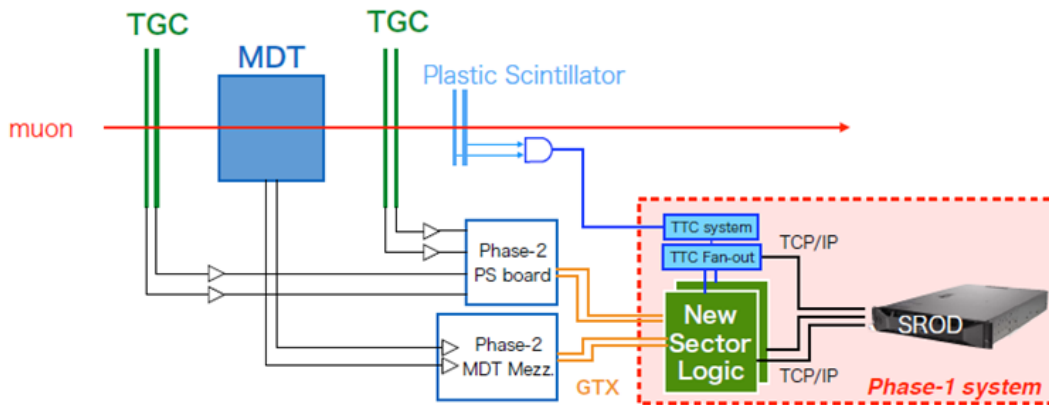


図 6.1: TestBeam のセットアップ : Phase-2 グループは TGC、MDT のデータを光通信で NSL に送るためのシステムの開発を行っていて、Phase-1 はそれらのリードアウトエレクトロニクスの試験を行う。

図 6.1 と図 6.2 に TestBeam 全体のセットアップとその写真を載せる。今回用いた検出器は Doublet TGC 2 組と MDT で、また今回トリガーを得るためにプラスチックシンチレータを用いている。

Phase-1 のシステムの概要について、図 6.3 を用いて説明する。今回、NSL は 2 台用意し、それぞれ TGC、MDT からのデータを受け取って処理を行う。今回のビームテストでは TGC からデータを受け取る NSL は 16 バンチ分のデータを読み出し、MDT からのデータを受け取る

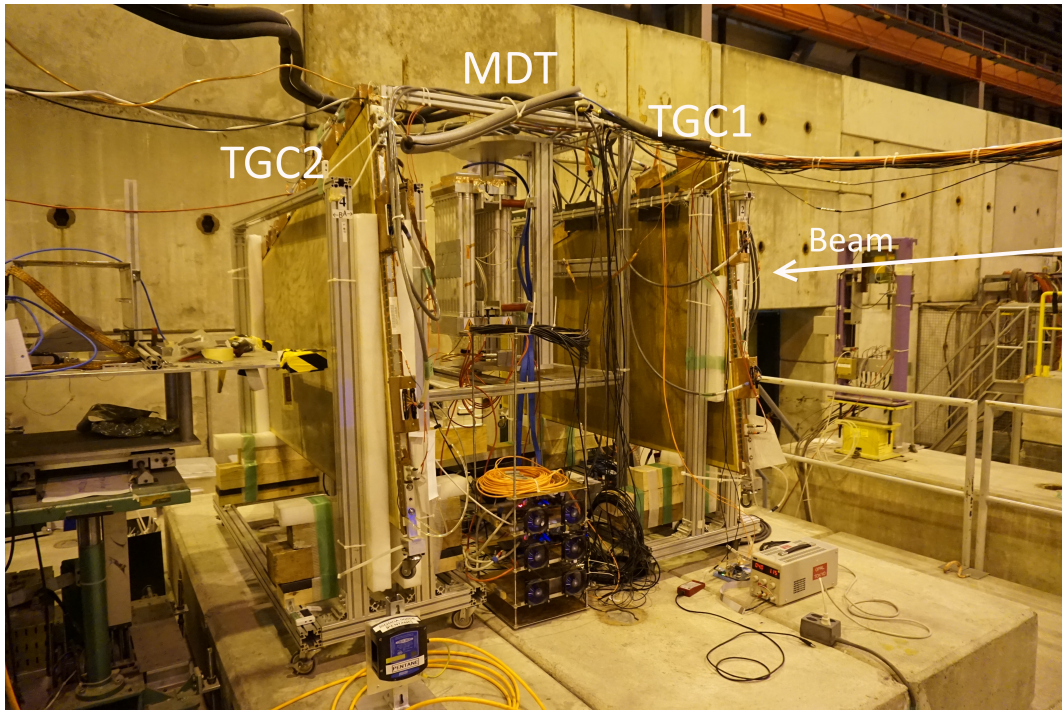


図 6.2: TestBeam のセットアップの写真

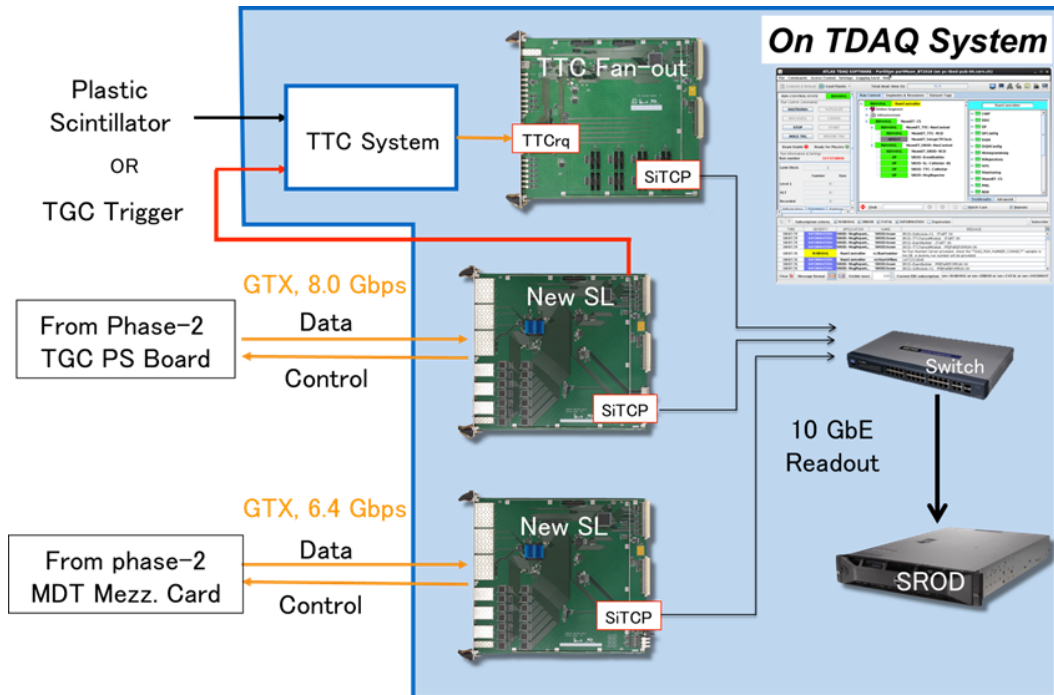


図 6.3: TestBeam における Phase-1 グループのセットアップ

NSL は 64 バンチ分のデータを読み出した。またトリガーにはプラスチックシンチレータからの情報を用い、TTC system から TTC Fanout Board に TTC 信号を送る。TTC Fanout Board は TTC 信号を受け取るとそれを NSL に分配し、TTC Fanout Board と 2 台の NSL は Switch を経由して SROD PC にデータを送る。

今回、Phase-1 のエレクトロニクスを全て TDAQ のシステムから制御を行った。図 6.4 に、エレクトロニクスを制御するための TGC partition のツリー構造を示す。

- MuonBT\_TTC

TTC のシステムを制御するための segment である。図 6.5 に今回の TestBeam における TTC システムのブロック図を示す。TTC の各モジュールについては 3.2.4 節を参照してもらいたい。

また TTC segment はトリガーの種類を決める機能を持つ。トリガーの種類として以下の 3 種類を用意した。

- External L1A (ExtL1A)

プラスチックシンチレータからの信号をトリガーに使用する。ビームのデータを取得する際に使用する。

- Periodic

あらかじめ設定しておいたパラメータに従って、L1A 信号を発行する。ノイズやベダスタルの測定を行ったり、高レートでの試験の際に用いる。

- TestPulse

BCR が発行されてから数クロック後にテストパルス信号を発信する。この信号は TGC 側面に設置された ASD に送信され、ASD からトリガー信号が送られる。この ASD からのトリガー信号を L1A 信号として受け取る。

システム全体の接続試験、クロックの調整に用いる。

- MuonBT\_9U-Segment

TTC Fanout Board と NSL を制御する segment で、VME 経由でエレクトロニクスの制御を行う。これらの制御には、9U クレートにマウントされている SBC から行う。

- MuonBT\_SROD

SROD の各プロセスを制御する segment である。ReadoutMaster は RCD として実装し、それ以外のプロセスは Resource Application として実装した。ReadoutMasterTDAQ のコマンドを受け取り、それに合わせて各プロセスに SROD で定義したコマンドを送信する。

今回の Test Beam における SROD の構成を図 6.6 に示す。今回取得したデータはバイナリファイルにして HDD に保存した。

また今回は SROD が BUSY を LTP に知らせる手段が実装されていない。そのためこの Test Beam では以下のような処理を行う。

1. Collector が RingBuffer に書き込めない場合、書き込めるようになるまで待機する。
2. Collector にデータを送ることができず、NSL 内の L1 Buffer にデータがたまる。
3. 結果、NSL から TTC に BUSY 信号を送り、L1A の発行を止める。

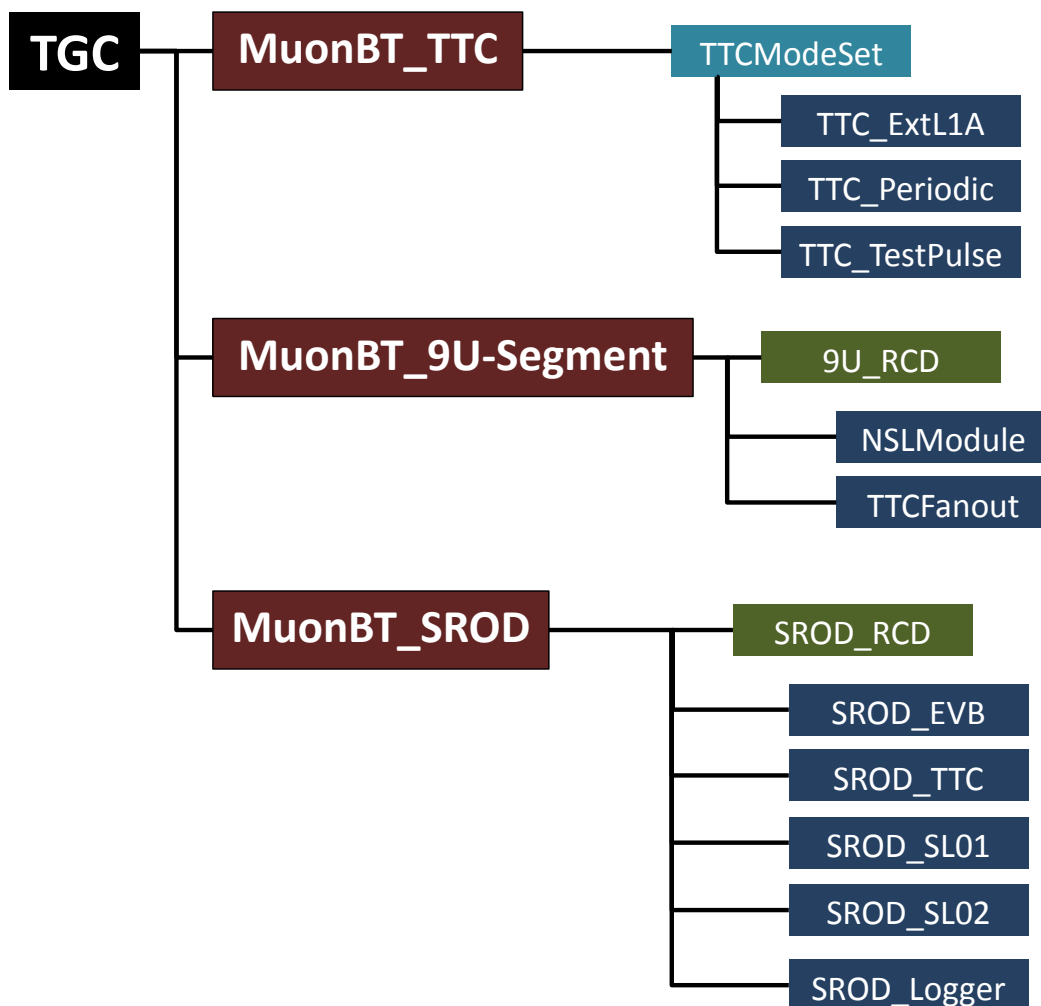


図 6.4: TestBeam の partition 内での segment の構成: 赤が segment, 青が resource, 緑が RCD を示す。また TTC\_ModeSet は、3つの resource のうちどれか1つを選択する resourceset というものである。

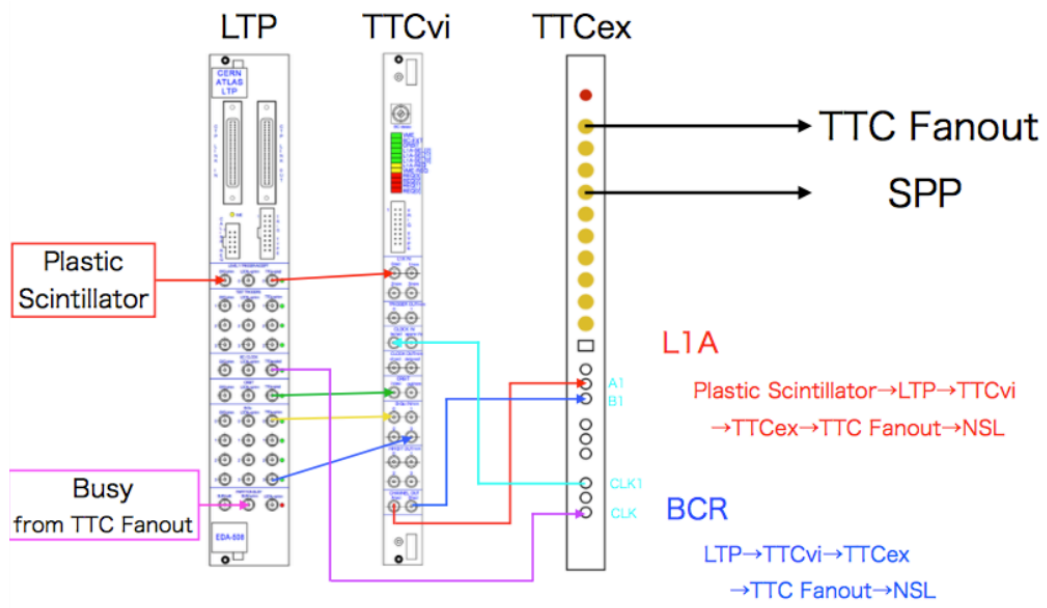


図 6.5: Test Beam における TTC システムの構成

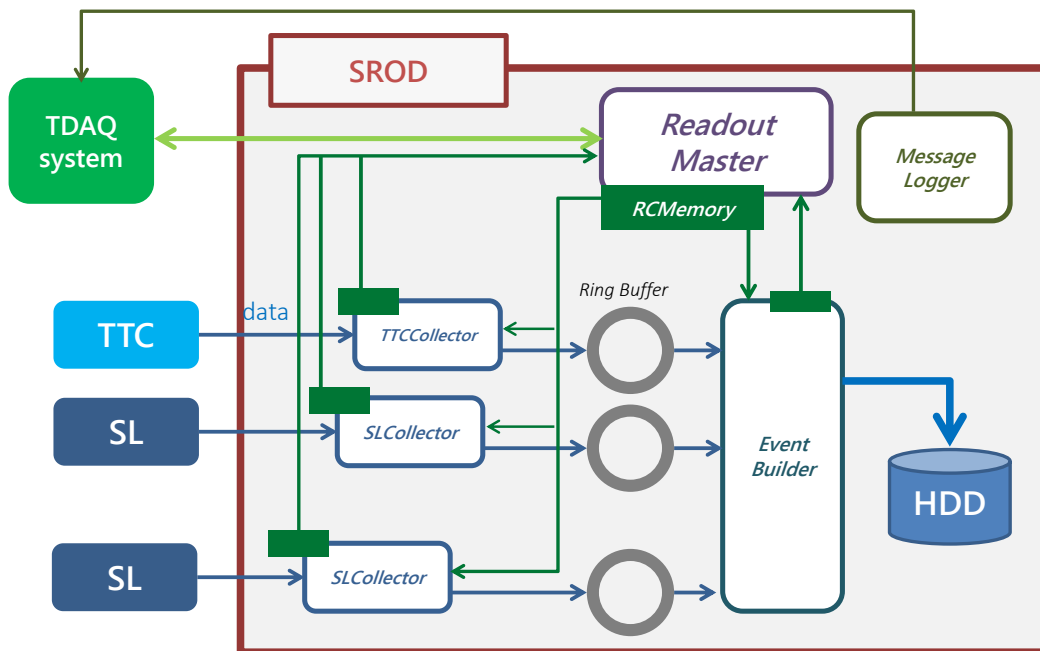


図 6.6: Test Beam における SROD の構成



TTC 信号の基本的な流れは実際の Run とは大きな違いはないが、今回のビームテストでは、トリガーを受け取ってから 2  $\mu\text{sec}$  の間、L1A を発行しない “simple dead time” を設けた。これは、MDT のドリフト時間を待つため、今回はシステムの基本的な性能を調べる試験であったため、MDT のドリフト時間測定の間トリガーが複数入った場合の処理を避けるために設けられた。

さらに今回、ReadoutMaster に Run を終了した際にその Run の情報を IS から読み出して MySQL データベースに記録する機能を実装した。このデータベースに web からアクセスし、後の解析のときに参照する “Run Summary” というシステムも作成した。図 6.7 に Run Summary の web ページ画面を載せる。

Run ID	Run Type	ILB	Start and end time (local)	#Events	Comments
100692	TestBeam	4	2016-11-12 03:16:23 - 2016-11-12 03:20:02	28189 (128.71 Hz)	TGC threshold scan run. Threshold value is 500-500 mV/MDT is masked. Used L1A from P.S.
100691	TestBeam	4	2016-11-12 03:11:01 - 2016-11-12 03:14:32	27566 (132.44 Hz)	TGC threshold scan run. Threshold value is 300-300 mV/MDT is masked. Used L1A from P.S.
100690	TestBeam	2	2016-11-12 03:08:23 - 2016-11-12 03:09:57	12592 (133.42 Hz)	TGC threshold scan run. Threshold value is 300-300 mV/MDT is masked. Used L1A from P.S. Stopped by ID mismatch burst.
100689	TestBeam	4	2016-11-12 03:08:37 - 2016-11-12 03:04:16	27157 (134.95 Hz)	TGC threshold scan run. Threshold value is 200-200 mV/MDT is masked. Used L1A from P.S.
100688	TestBeam	1	2016-11-12 02:58:19 - 2016-11-12 02:56:44	2420 (98.42 Hz)	TGC threshold scan run. Threshold value is 100-100 mV/MDT is masked. Used L1A from P.S. Stopped by ID mismatch burst.
100687	TestBeam	1	2016-11-12 02:58:38 - 2016-11-12 02:55:51	1026 (88.16 Hz)	TGC threshold scan run. Threshold value is 100-100 mV/MDT is masked. Used L1A from P.S. Stopped by ID mismatch burst.
100686	TestBeam	1	2016-11-12 02:53:09 - 2016-11-12 02:53:28	2302 (131.88 Hz)	TGC threshold scan run. Threshold value is 100-100 mV/MDT is masked. Used L1A from P.S. Stopped by ID mismatch burst.
100685	TestBeam	2	2016-11-12 02:58:23 - 2016-11-12 02:52:28	11723 (123.45 Hz)	TGC threshold scan run. Threshold value is 100-100 mV/MDT is masked. Used L1A from P.S. Stopped by ID mismatch burst.
100684	TestBeam	4	2016-11-12 02:43:48 - 2016-11-12 02:47:07	28232 (141.98 Hz)	TGC threshold scan run. Threshold value is 100-100 mV/MDT is masked. Used L1A from P.S.
100683	TestBeam	12	2016-11-12 02:58:12 - 2016-11-12 02:49:33	10336 (120.81 Hz)	TGC threshold scan run. Threshold value is 50-50 mV/MDT is masked. Used L1A from P.S.
100682	TestBeam	4	2016-11-12 02:23:38 - 2016-11-12 02:27:46	8412 (131.84 Hz)	MDT setting modification run.
100681	TestBeam	2	2016-11-12 02:21:52 - 2016-11-12 02:23:07	2624 (124.88 Hz)	MDT setting modification run. Stopped by ID mismatch burst.
100680	TestBeam	8	2016-11-12 02:13:07 - 2016-11-12 02:21:04	16454 (144.48 Hz)	TGC threshold scan run. Threshold value is 50-50 mV/MDT is masked. Used L1A from P.S.
100679	TestBeam	2	2016-11-12 02:03:05 - 2016-11-12 02:04:49	3795 (128.49 Hz)	
100678	TestBeam	3	2016-11-12 01:55:40 - 2016-11-12 01:58:09	19510 (133.83 Hz)	
100677	TestBeam	3	2016-11-12 01:50:09 - 2016-11-12 01:52:34	22481 (132.94 Hz)	MDTMEZK7 miss setting.
100676	TestBeam	16	2016-11-12 01:32:51 - 2016-11-12 01:48:33	65066 (89.89 Hz)	TGC threshold scan run. Threshold value is 10-10 mV. MDT is masked. Used L1A from P.S.

図 6.7: Run Summary

## 6.1.2 Test Beam の試験結果

システム全体が動作してデータを取得することに成功した。図 6.8 に TGC のヒット位置のヒストグラムを載せる。この結果からシステムが全て正しく動作し、ビームの位置情報を取得することができたことが確認できる。

SROD では、TTC Fanout Board と NSL からデータを受け取り、正しくイベントビルディングが行われた。図 6.9 に SROD で作成したイベントデータの一例を載せる。図 5.4 で示した ROB フォーマット通りにデータを作成することができていた。

しかし、データ取得の時に TTC Fanout Board からのデータと NSL からのデータで、L1ID と BCID が一致しないイベントがあった。ビームレートが 50 Hz のときには 245 分で 3 回起き、発生率は

$$3/(50[\text{Hz}] \times (245 \times 60)[\text{sec}]) = 4.08 \times 10^{-6} \quad (6.1)$$

であった。また 100 Hz でビームを照射しているときには 63 分で 2 回発生し、

$$2/(100[\text{Hz}] \times (63 \times 60)[\text{sec}]) = 5.29 \times 10^{-6} \quad (6.2)$$

の発生率であった。

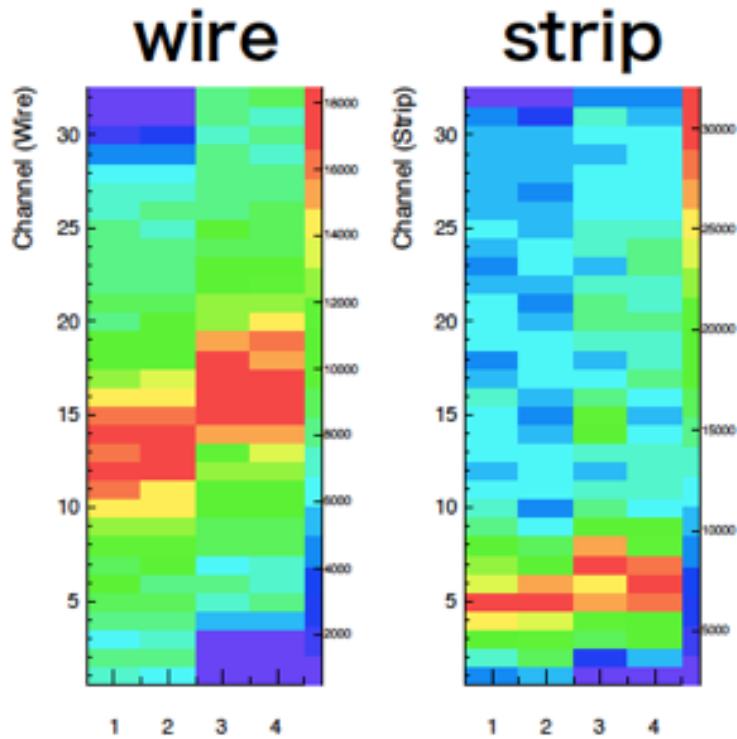


図 6.8: TGCでのヒットマップ図: ヒストグラムの横軸は1,2はTGC1のワイヤ、3,4はTGC2のワイヤを示す。(ストリップも同様) 縦軸にはヒットしたワイヤ(ストリップ)のチャンネルを表している。

```

b0f01234          ee1234ee
00000009          00300030          00670003          12000000
0b0004c5          000001dc          000000ff          00000000

00000022          f1026fa9          f10353b7          f11cbccc
f100bbdd          f11eff00          f124bccc          f1250100
f11d0100          f300bbdd          f3026fa9          f30353b8
f126ff11          f31d0100          f31eff00          f324bccd
f31cbccd          f326ff11          f700bbdd          f7026fa9
f3250100          f71cbcce          f71d0100          f71eff00
f70353b9          f7250100          f726ff11

00000024          f102ccd0          f10380b0          f11cbccc
f100bbdd          f11eff00          f300bbdd          f302ccd0
f11d0100          f31cbccd          f31d0100          f31eff00
f30380b1          f702ccd0          f70380b2          f71cbcce
f700bbdd          f71eff00

00000000          00000000          00000000
00000003          0000002f          00000001
e0f01234

```

図 6.9: エミュレータを用いた試験での作成データの一例

また NSL のデータが 16 bit ずれることでデータ取得が続行できなくなるイベントも発生した。発生率は

$$2/(50[\text{Hz}] \times (245 \times 60)[\text{sec}]) = 2.72 \times 10^{-6} \quad (6.3)$$

であった。

DAQ のエラーが発生するレートは  $O(10^{-6})$  であった。実際の環境のイベントレートは 100 kHz であるため、よりエラーの少ないシステムに、特にデータ取得が続行できなくなることがなくなるように改良することが今後の課題である。

### 6.1.3 High Rate の試験

TDAQ システムにのせた Phase-1 のシステムについて、Periodic モードで測定を行った。Periodic モードは、LTP から周期的に L1A 信号を生成するモードで、TDAQ システムにのせた本システムの性能評価を行うために設けた。

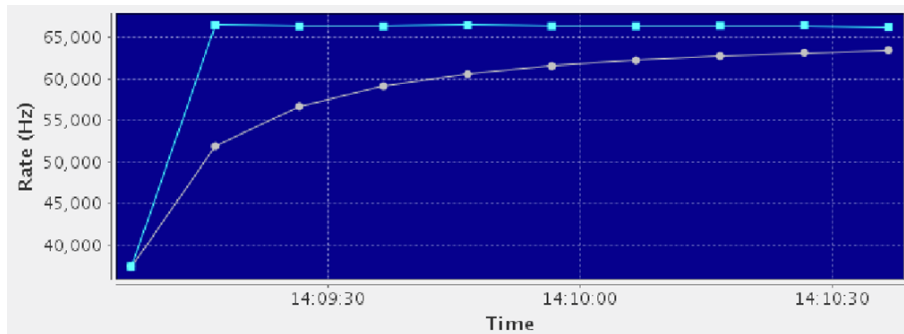


図 6.10: Periodic モードによる試験での L1A 発行頻度の時間変化：横軸は時刻を、縦軸は L1A 発行頻度をそれぞれ表す。図中青の点はその時刻での瞬間レートを、灰色の点は Run 中の平均レートを示す。Periodic では L1A を 100 kHz で発行していたが、67 kHz で処理が頭打ちになっている。

図 6.10 に、Periodic モードで試験を行ったときの TTC の L1A 発行頻度の時間変化を示す。Periodic モードによる試験を行ったところ、SROD の処理速度が 67 kHz までしか出ていないことがわかった。この結果を見ると、SROD の処理が追いつかず、BUSY 信号が発行されていたために、L1A が 100 kHz でていなかった。

## 6.2 処理速度の最適化

SROD の処理速度を改善させるために、SROD のみで試験を行った。SROD 単体の試験には TTC、SL モジュールの代わりとなるエミュレータを用いた。図 6.11 にエミュレータを用いた性能評価の構成を載せる。なおエミュレータは SROD と異なり、TDAQ application で動かさず、外部から単体のプロセスとして実行する。

### 6.2.1 エミュレータ

エミュレータは TTC、SL のデータを擬似的に作成し、一定の時間間隔で SROD に送信する単純なもので、ハードウェアに依存するボトルネックを除き、ソフトウェア単体の性能について検

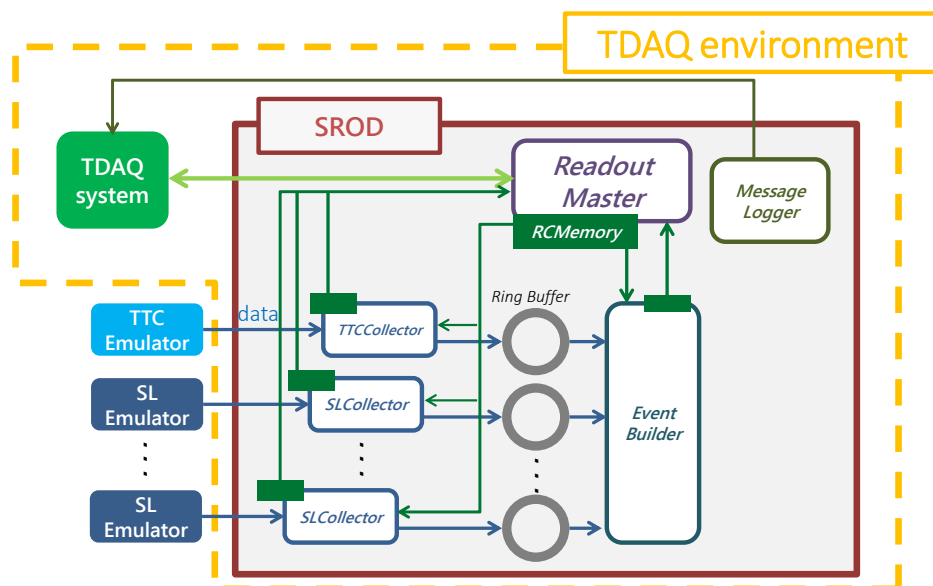


図 6.11: エミュレータを用いた試験環境：エミュレータと SROD は全て同一のマシンで走らせる。

証するためのものである。エミュレータのシーケンス図を図 6.12 に示す。エミュレータは sleep 関数を用いて一定時間待機し、タイマーによるシグナルを受け取るとデータを作成し、TCP 通信で Collector に 1 イベント分のデータを送る。sleep 関数による待機時間は任意に変更することができ、例えば 100 kHz でデータを送る場合には、10  $\mu$ sec 毎にシグナルを発行するように設定する。

- TTCEmulator

TTC のイベントデータを作成し、1 イベント毎に TCP ソケットで送信する。L1ID、BCID 共に 1 イベント毎にカウントアップするようにし、ECR ID は L1ID が 0x00ffffff カウントに達する度にカウントアップするようにしている。また Orbit ID、Trigger Type は今回の試験ではデータが正しく入っているかを試すだけのものなので、適当な一定値に設定して試験を行った。

- SLEmulator

TTCEmulator 同様、SL のデータを作成して TCP ソケットで送信するプログラムである。SL の L1ID、BCID は TTC と同様に 1 イベント毎にカウントアップするようにし、TTC と同じ値になるようにしてある。SL のデータサイズは一定にして走らせたり、実際に合わせてランダムに設定できるようにしてあり、様々な条件で試験を行うことができるようにしている。

これらのエミュレータについて、処理速度を測定した。測定の際には、図 6.13 のように、エミュレータからデータを受け取るだけの TCP Check というプロセスを設け、実際の SROD での試験とほとんど同じプロセス数で行った。

この状態でエミュレータの速度を測定した結果が図 6.14 である。この結果から、エミュレータによる試験で測定できる処理速度は 100 kHz までという結果が得られた。

エミュレータの試験で処理速度が制限される原因として、1 台の PC で処理するプロセスの数が非常に多いため、SROD が十分な性能があげられていないということが考えられる。今回の測定に使用した PC のコア数が 6 つであることに對し、26 個のプロセスを走らせるため、各

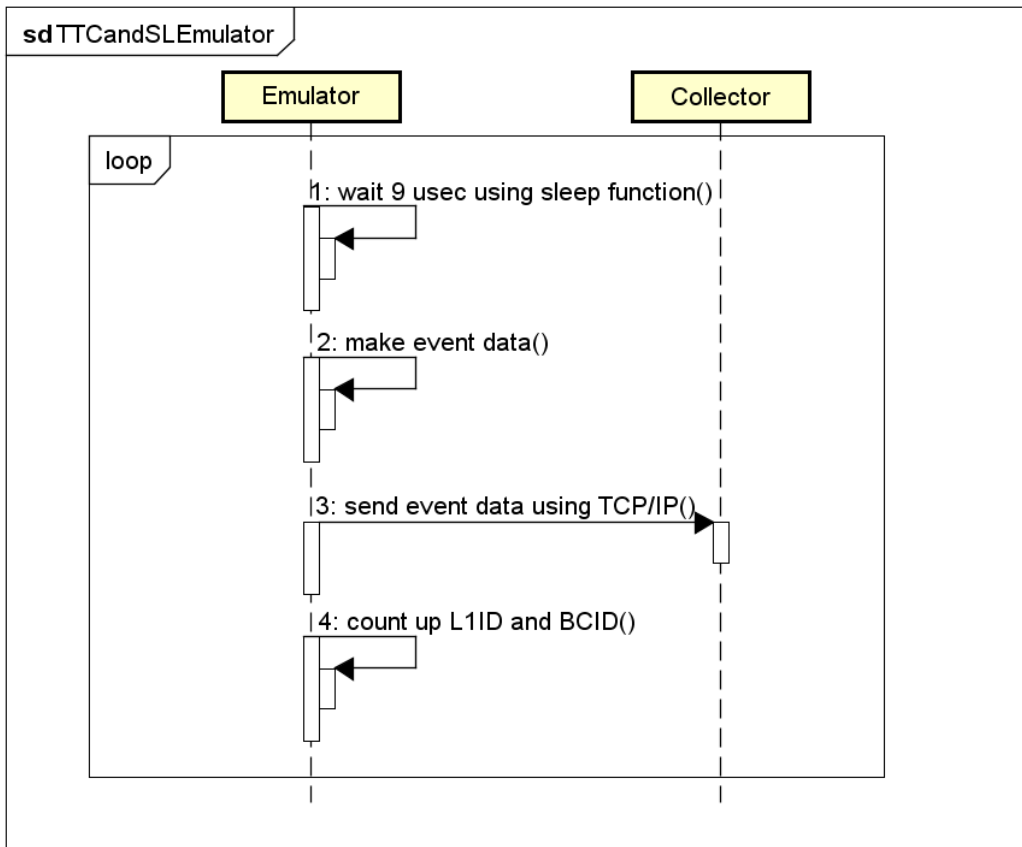


図 6.12: エミュレータのシーケンス図

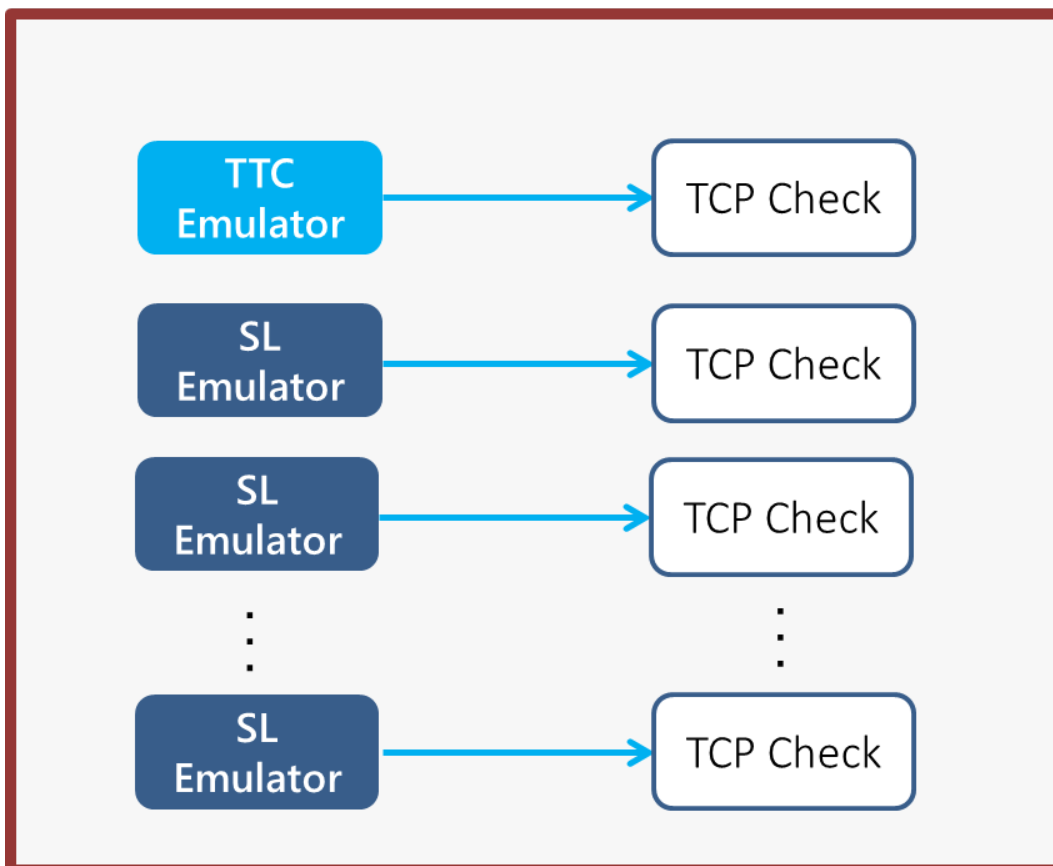


図 6.13: エミュレータの速度測定概要図

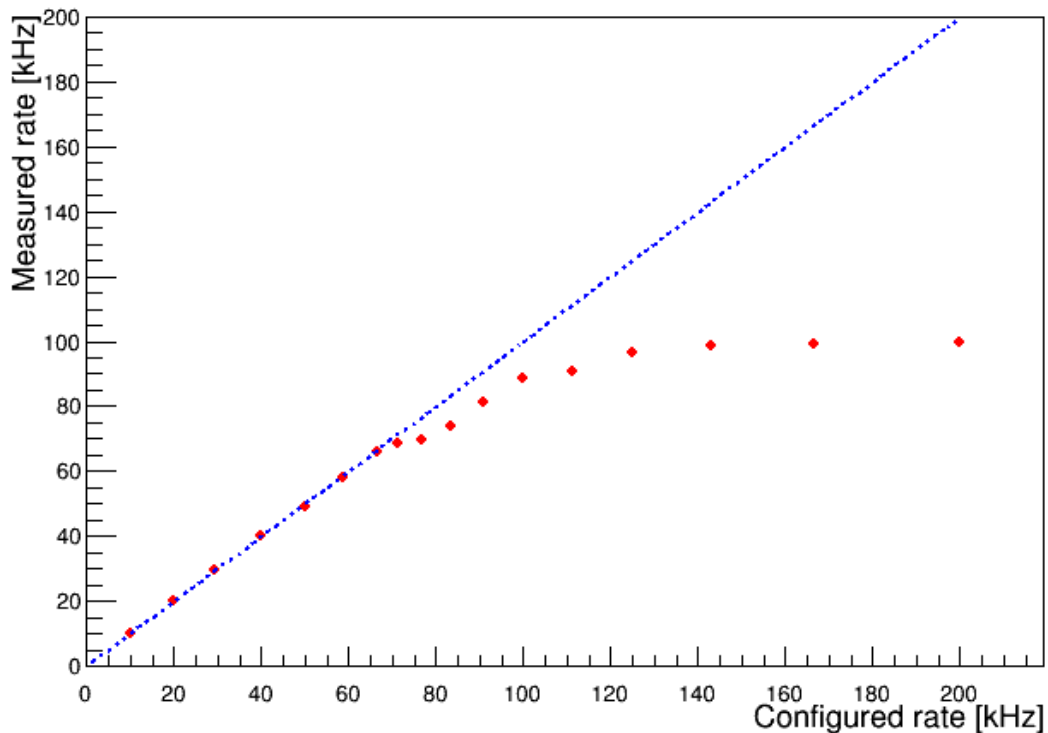


図 6.14: エミュレータの処理速度：横軸は設定したレートを、縦軸にはエミュレータが実際に送信したレートをそれぞれ表す。理想は設定したレートと実際のレートが等しい状態 (青線) であるが、70 kHz を超えたあたりからレートが下がり、最大でも 100 kHz 程度までしか出ていないことが確認できる。

プロセスがリソースを取り合い、全体として処理速度が低下した可能性がある。図 6.15 に、エミュレータと TCP Check の組の数による処理速度の変化を示す。

プロセス数によってエミュレータの処理速度には制限があるが、100 kHz でデータを送信することは可能であるため、エミュレータを用いて SROD の試験を行えることがわかった。

## 6.2.2 バッファサイズの最適化

6.1.3 節で述べた、SROD の処理速度の問題についてエミュレータを用いて原因を調べたところ、SROD の各プロセスで使用するバッファの大きさが処理速度に影響することがわかった。以下に SROD が使用するバッファについて述べる。

- TTCBuffer  
TTCCollector が TCP/IP で受信したデータを格納するバッファである。
- SLBuffer  
SLCollector が TCP/IP で受信したデータを格納するバッファである。TTC Fanout Board と NSL でデータフォーマットのサイズが異なるため、TTCBuffer と別でバッファサイズを定義する必要がある。
- EBBuffer  
EventBuilder が各 Collector との RingBuffer から読み出したデータを格納するバッファである。EventBuilder はデータをこのバッファに格納した後、データの確認を行うため、1 イベント分のデータが入る十分な長さを確保しなければならない。

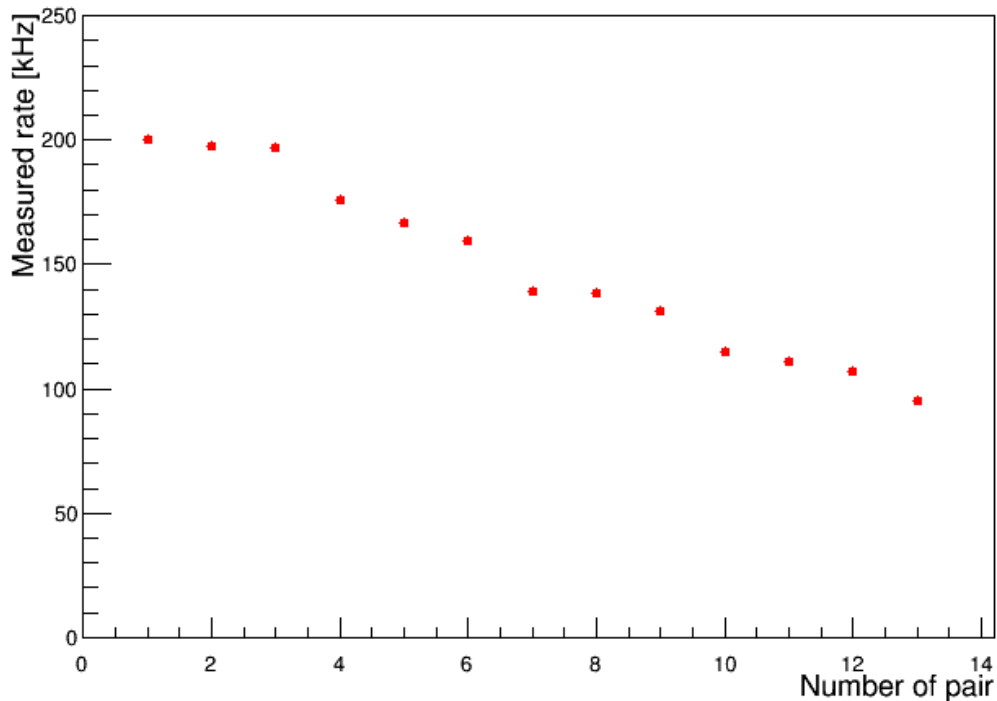


図 6.15: エミュレータ-TCP Check ペアの数による処理性能の変化:エミュレータは 200 kHz でデータを送っている。横軸はペアの数を表し、1つのペアあたりのプロセス数は2である。グラフを見ると、ペア数が3(プロセス数が6)以降、コア数よりプロセス数が多くなり処理速度が下がっていきることが確認できる。

- ROBBuffer

EventBuilder 内で ROB フォーマットに従ったイベントデータの作成のために用意されたバッファで、データ部が可変長なため、ある程度の大きさを確保する必要がある。

- RingBuffer (データ部)

RingBuffer の、データを保存する領域のバッファサイズもその長さを指定できる。RingBuffer については 5.5.2 節で説明したとおりである。

これらのバッファサイズによる SROD の処理速度の変化を調べた。試験の際には、図 6.15 よりプロセス数が少ない場合には 200 kHz で試験が行えることがわかっているため、エミュレータは TTCEmulator と SLEmulator1 つを使用し、送信レートを 200 kHz にして測定した。

まず TTCBuffer と SLBuffer について測定した結果が図 6.16 である。TTCBuffer のバッファサイズは処理速度に影響しなかったが、SLBuffer のバッファサイズを大きくすると徐々に処理速度が低下していることがわかった。

次に EBBuffer について測定した結果を図 6.17 に示す。EBBuffer も TTCBuffer 同様、大きな依存性は見られなかった。

そして ROBBuffer について測定した結果が図 6.18 である。ROBBuffer の大きさは、処理速度に大きく依存していることが分かった。

最後に RingBuffer のバッファサイズの依存性を図 6.19 に示す。RingBuffer の大きさは処理速度には影響していなかった。

結果として、SLBuffer、ROBBuffer が大きいと処理速度が低下することが判明した。またそれ以外のバッファサイズについて、バッファサイズを大きくしても処理速度は関係ないということが分かった。バッファサイズが大きいと処理速度が低下する原因としては、バッファ領域



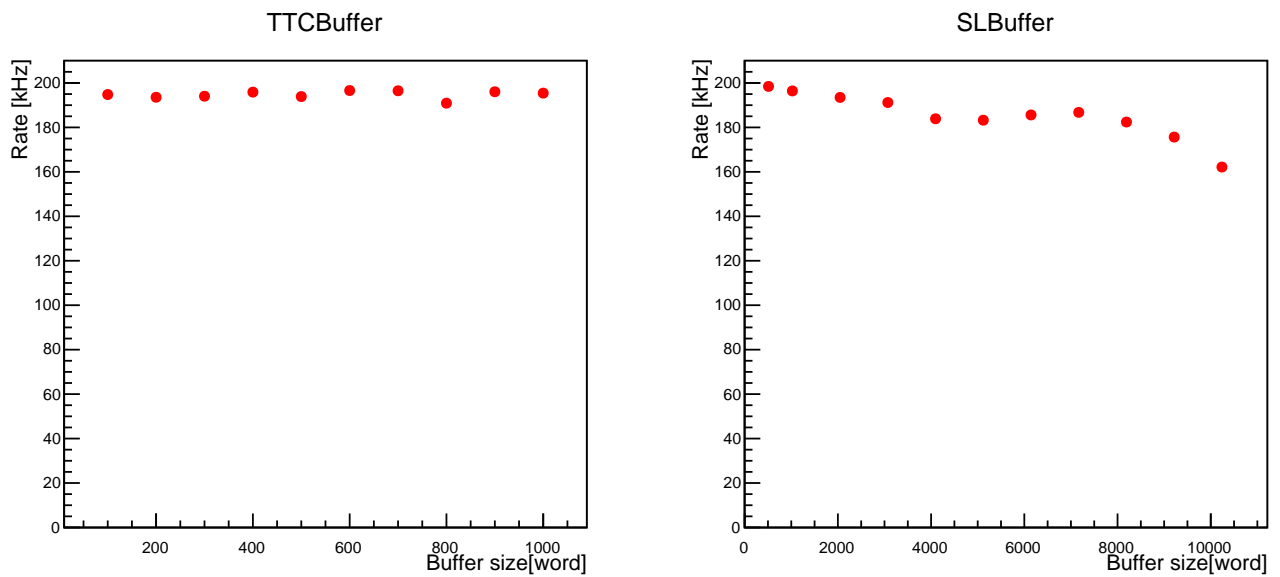


図 6.16: TTC/SLBuffer のサイズによる処理速度の変化: 左図が TTCBuffer、右図が SLBuffer の変化を示す。測定時には、他のバッファサイズは表 6.1 で示す大きさに最適化してある。TTCBuffer にはバッファサイズと処理速度の間に相関は見られなかった。一方 SLBuffer はバッファサイズ 4000 ワード付近で処理速度が低下した。

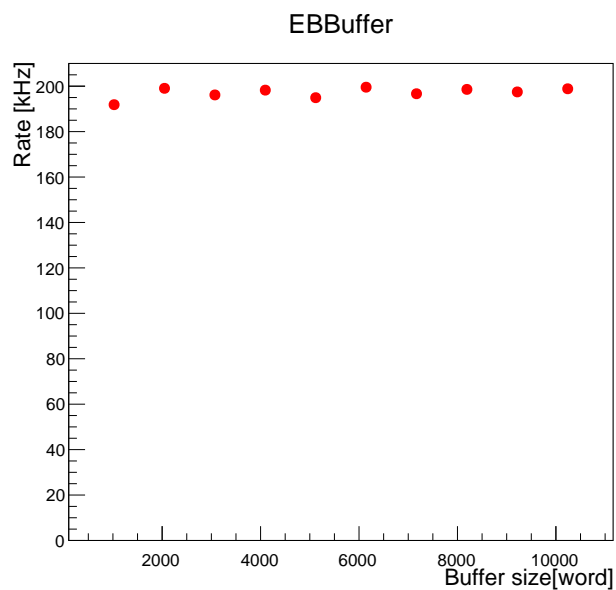


図 6.17: EBBuffer のサイズによる処理速度の変化: EBBuffer

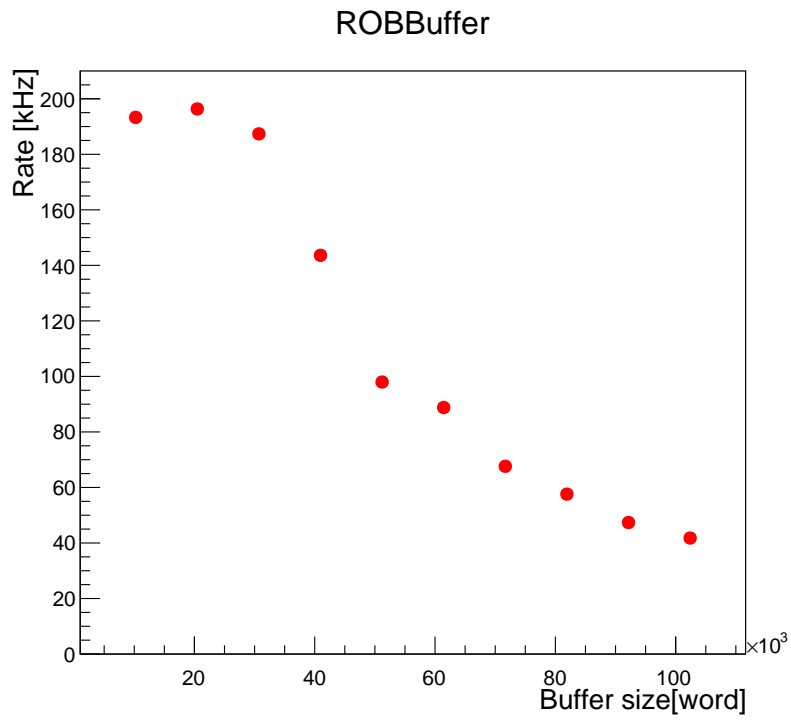


図 6.18: ROBBuffer のサイズによる処理速度の変化

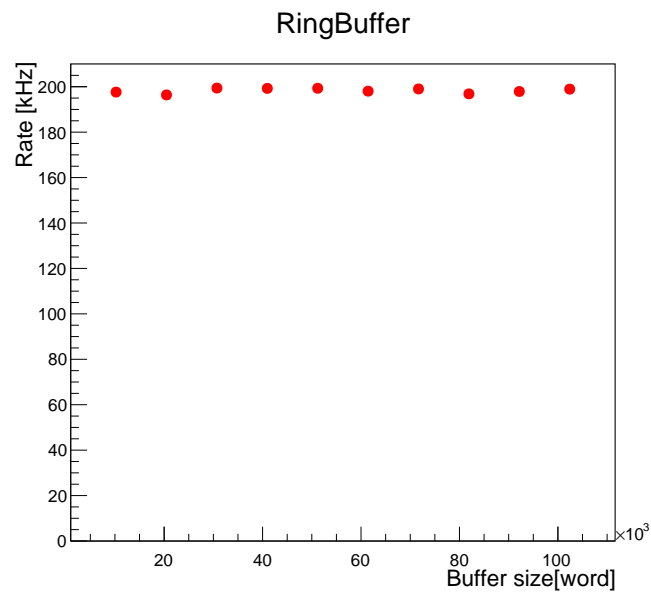


図 6.19: RingBuffer のデータ部のサイズによる処理速度の変化

を確保する領域が大きくなり、RAMを全て使用し、ハードディスク上の領域を仮想メモリとして使用したことが考える。ハードディスク上への仮想メモリへのアクセスは本来のメモリへのアクセスより時間がかかるため、処理速度が低下したと考えられる。

以上のことから、バッファサイズは全体的に小さくすることにした。表 6.1 に、各バッファのビームテスト時のサイズと変更後のバッファサイズを載せる。

表 6.1: SROD で用いるバッファとそのサイズの変更: 単位は全て word (=32 bit) である。

バッファ	変更前	変更後
TTCBuffer	320	5
SLBuffer	6400	512
EBBuffer	64000	1024
ROBBuffer	64000	6144
RingBuffer	1024000	5120

### 6.2.3 処理速度の測定

次に、エミュレータから SROD にデータを送る際の処理速度の測定を行ったところ、図 6.20 のような結果が得られた。SROD の処理速度は約 100 kHz までしか出なかったが、これは 6.2.1 節でエミュレータの処理速度の測定結果を考慮すると、本試験環境における上限値であったことが考えられる。

また本測定では、設定レートが 100 kHz を超えたあたりから、測定レートにばらつきが発生した。図 6.21 に、処理時間のヒストグラムを載せる。イベントレートが大きくなるにつれ、処理時間にばらつきが生じた。

測定結果としては、SROD の処理速度はエミュレータの処理速度と大きな変化は見られず、十分な処理速度を持つと考えられる。今回の測定では、PC を 1 台しか用意できず、エミュレータと SROD を同一のマシンで動作させる必要があったが、エミュレータ用のマシンを別に用意して同様の試験を行うことで、SROD の処理速度をより正確に測定できると考えられる。

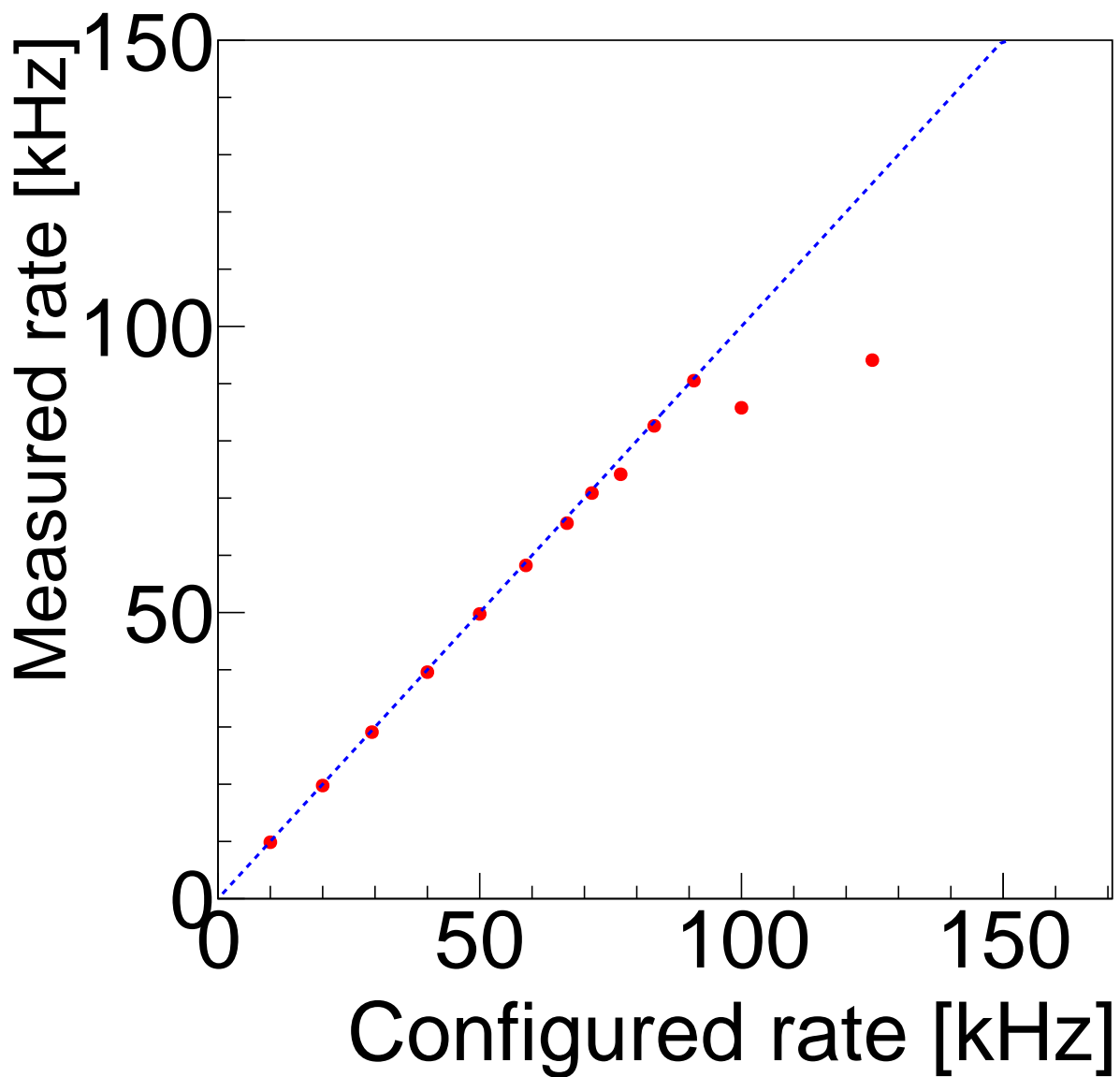


図 6.20: エミュレータを用いた SROD の処理速度の測定結果：横軸は設定した送信速度を、縦軸は SROD の処理速度を示す。エミュレータの転送速度と SROD の処理速度の間には処理速度の差は見られなかった。これは SROD が 100 kHz 以下のレートでは十分な性能を持つことを実証している。

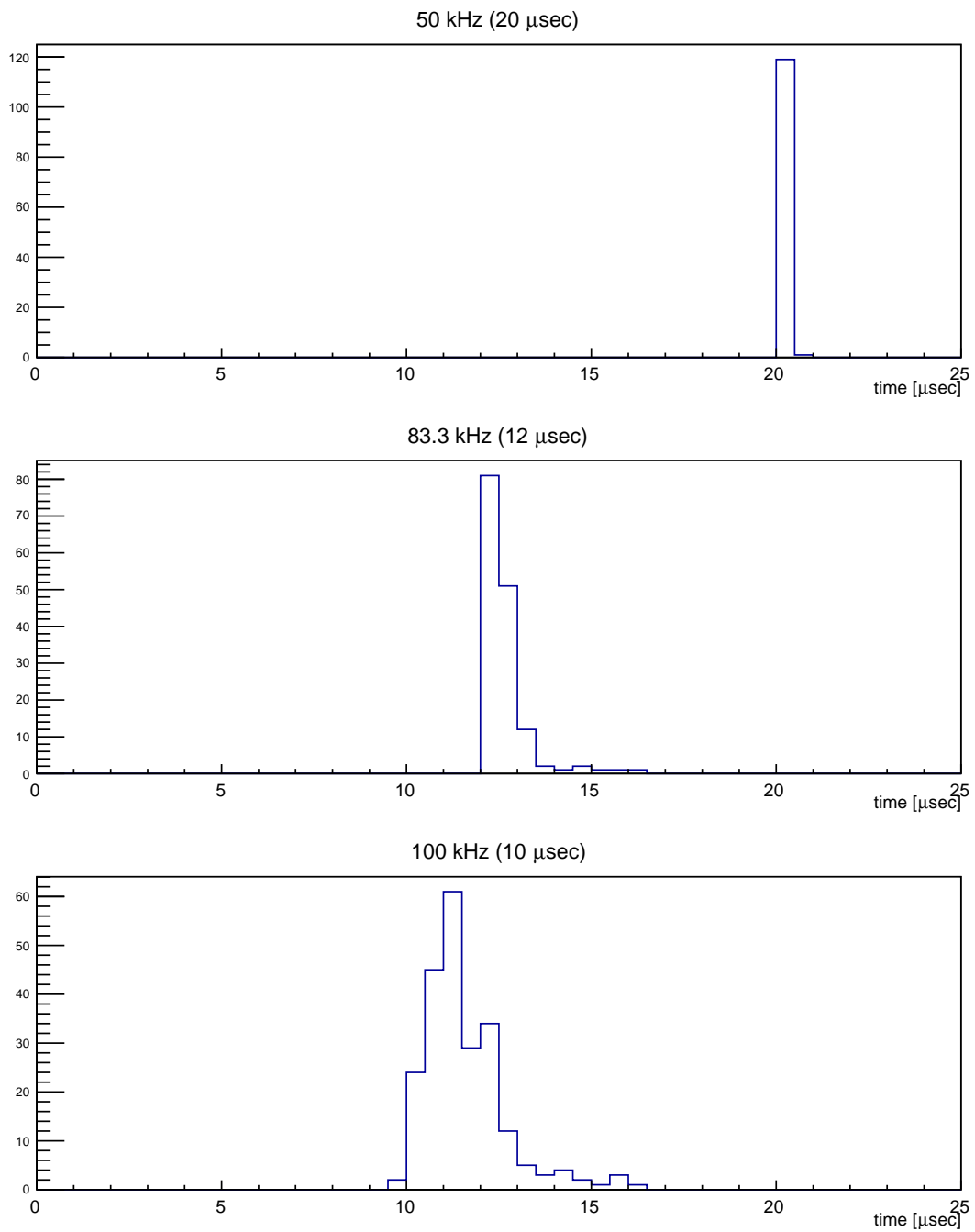


図 6.21: SROD の 1 イベントあたりの平均の処理時間の分布: 上段が 50 kHz (20  $\mu\text{sec}$ ) のとき、中断が 83.3 kHz (12  $\mu\text{sec}$ ) のとき、下段が 100 kHz (10  $\mu\text{sec}$ ) のときのヒストグラムである。イベントレートが高くなるにつれ、処理速度のばらつきが大きくなる。

## 第7章 まとめ

LHC では HL-LHC のための段階的なアップグレードを行い、イベントレートを徐々に増やしている。ATLAS 実験ではイベントレートの増加に合わせてトリガーシステムの改良を行っている。

その中でもミューオントリガーでは、2020 年からの Run-3 から新たなるミューオン検出器、NSW を導入し、そこからの情報を用いた新たなるトリガーシステムを導入するために、トリガー判定を行うエレクトロニクスである NSL を開発する。

本研究では、NSL のトリガー情報を収集するためのソフトウェア、SROD の開発を行い、SROD を TDAQ のシステムに組み込み、現行の ATLAS 実験でも用いられている GUI アプリケーションからソフトウェアの制御を行えるようにした。

SROD はビームテストにおいて、GUI を用いた制御と、実際の New Sector Logic からのデータの収集ができることを実証したほか、実際の Run と同様のシステムを組み、ビームからのデータを取得することに成功した。このビームテストでは Readout Driver における様々な問題点を見つけ出し、改善することができた点でも大きい。

そしてエミュレータを用いた試験において、本番の運転において NSL からデータが送られる頻度である 100 kHz 前後でデータを収集することができた。

処理速度の測定においては 1 台の PC におけるプロセス数による制限が考えられ、外部 PC にエミュレータを走らせれば十分な速度が出せる可能性を示している。

今後はエミュレータ用のマシンを用いた SROD の処理速度の測定のほかに、SROD の今後の課題は SROD から BUSY のシステムを導入することと、SROD から ROB フォーマットに変換したデータを ROS に送信する機能を追加することが課題として挙げられる。また PCI express を用いたデータ通信の試験、実機を複数用いたスピードテスト、そしてエラーから回復するロジックの追加も今後の課題である。

本研究では、モジュールに SiTCP を実装し、スイッチでデータ収集し、PC で読み出しを行うシステムを開発した。これは汎用的が高く、応用が期待される。

# 目 次

1.1	LHC の運転スケジュール	1
2.1	LHC 加速器の全体像	4
2.2	CMS 検出器	5
2.3	ALICE 検出器	5
2.4	LHCb 検出器	5
2.5	ATLAS 検出器の座標系	6
2.6	擬ラピディティ	6
2.7	ATLAS 検出器の全体像	6
2.8	素粒子に対する検出概念図	7
2.9	ミューオンスペクトロメータの全体像	8
2.10	ミューオンスペクトロメータの配置	8
2.11	標準理論の粒子	9
2.12	ヒッグス粒子の生成過程	9
2.13	ヒッグス粒子の生成断面積の陽子・陽子衝突の重心系エネルギーによる変化	10
2.14	ヒッグス粒子の崩壊チャンネルとそのシグナル強度 $\mu$ 値	11
3.1	ATLAS 実験のトリガーシステム	13
3.2	Isolated muon の模式図	14
3.3	$t \rightarrow Wb$ の模式図	15
3.4	TGC 検出器の構造	16
3.5	TGC 検出器の検出原理	16
3.6	TGC Doublet と Triplet の構造	17
3.7	TGC の $R-z$ 平面における配置図	18
3.8	TGC Big Wheel の写真	18
3.9	BW の 1/12 セクター	19
3.10	トリガーセクター及び RoI	19
3.11	TGC における $p_T$ 算出の概念図	19
3.12	コインシデンスウィンドウ	20
3.13	TGC エレクトロニクスの全体像	21
3.14	TGC の段階に分けられたコインシデンス処理	21
3.15	ASD Board	22
3.16	Sector Logic Board	23
3.17	ゼロサプレスロジックの概略図	24
3.18	TGC ROD モジュール	24
3.19	TTC システム	25
3.20	フェイクトリガー	27
3.21	EI または FI でのヒットを課したときの BW のシミュレーションによるヒット分布	27



3.22	Tile カロリメータの配置図	28
3.23	NSW	29
3.24	MicroMegas 検出器の断面図	29
3.25	MicroMegas 検出器のストリップの配置	30
3.26	sTGC の断面図	31
3.27	NSW を含めたトリガーロジック	32
3.28	New Sector Logic	33
3.29	NSL のトリガー及びリードアウトのロジック図	34
3.30	TTC Fanout Board の写真	35
3.31	TTC Fanout Board のリードアウトスキーム	35
4.1	Run Control Application	36
4.2	TGC segment	37
4.3	TDAQ のステートマシン図	39
4.4	OKS データファイルの一例	41
5.1	リードアウトシステムの構成	43
5.2	PCI Express Card	45
5.3	ATLAS で定められている ROB フォーマット	48
5.4	SROD が作成する出力フォーマット	48
5.5	First Status Word	50
5.6	SROD の構成	51
5.7	SROD のステートマシン図	52
5.8	メインプロセスのクラス図	53
5.9	Collector のユースケース図	54
5.10	Collector のシーケンス図	54
5.11	EventBuilder のユースケース図	55
5.12	EventBuilder のシーケンス図	56
5.13	ReadoutMaster のユースケース図	57
5.14	ReadoutMaster からコマンド伝達のシーケンス図	58
5.15	MessageLogger への送信データフォーマット	59
5.16	ソケット通信	60
5.17	RingBuffer の構成	61
5.18	RingBuffer のデータ部	61
5.19	RCMemory の構成	62
6.1	TestBeam のセットアップ	64
6.2	TestBeam のセットアップの写真	65
6.3	TestBeam における Phase-1 グループのセットアップ	65
6.4	TestBeam の partition 内での segment の構成	67
6.5	Test Beam における TTC システムの構成	68
6.6	Test Beam における SROD の構成	68
6.7	RunSummary	69
6.8	TGC でのヒットマップ図	70
6.9	エミュレータを用いた試験での作成データの一例	70

6.10	Periodic モードによる試験での L1A 発行頻度の時間変化 . . . . .	71
6.11	エミュレータを用いた試験環境 . . . . .	72
6.12	エミュレータのシーケンス図 . . . . .	73
6.13	エミュレータの速度測定の概要図 . . . . .	73
6.14	エミュレータの処理速度 . . . . .	74
6.15	エミュレータ-TCP Check ペアの数による処理性能の変化 . . . . .	75
6.16	TTC/SLBuffer のサイズによる処理速度の変化 . . . . .	76
6.17	EBBuffer のサイズによる処理速度の変化 . . . . .	76
6.18	ROBBuffer のサイズによる処理速度の変化 . . . . .	77
6.19	RingBuffer のデータ部のサイズによる処理速度の変化 . . . . .	77
6.20	エミュレータを用いた SROD の処理速度の測定結果 . . . . .	79
6.21	SROD の 1 イベントあたりの平均の処理時間の分布 . . . . .	80

# 表 目 次

1.1	LHC の各 Run に対する重心エネルギーと最大ルミノシティ	2
2.1	現在の LHC の主要パラメータ	5
3.1	$p_T$ の閾値とレベル 1 ミューオントリガメニューの対応関係	13
3.2	現在使用されているトリガメニューの一例	14
3.3	TGC の各パラメータと設計値	15
3.4	Run-3 におけるコインシデンス処理による L1_MU20 のトリガーレート	28
3.5	sTGC の各パラメータと設計値	31
4.1	SROD における Severity 毎の処理の一覧	40
5.1	本研究で使用した SROD PC の各パラメータ	44
5.3	TTC からの入力フォーマット	46
5.4	SL からの入力フォーマット	47
5.5	エラーコードの一例	58
6.1	SROD で用いるバッファとそのサイズの変更	78

# 参考文献

- [1] Xabler Cid Vidal and Ramon Cid Manzano, LHC TIMELINE Taking a closer look at LHC, [http://www.lhc-closer.es/taking\\_a\\_closer\\_look\\_at\\_lhc/0.lhc\\_timeline](http://www.lhc-closer.es/taking_a_closer_look_at_lhc/0.lhc_timeline)
- [2] Dan Green, At the Leading Edge The ATLAS and CMS LHC Experiments, February 2010.
- [3] T.Uchida, Hardware-Based TCP Processor for Gigabit Ethernet, IEEE Transactions on Nuclear Science. vol. 55, no. 3, 2008.6, pp. 1631-1637, <http://hdl.handle.net/2261/15490>
- [4] 浦野祐作, 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 修士論文 2015.
- [5] ATLAS Experiment, <http://atlas.web.cern.ch>
- [6] CMS Collaboration, The CMS Experiment at the CERN LHC, JINST 3 (2008) S08004.
- [7] ALICE Collaboration, K. Aamodt et al. , The ALICE experiment at the CERN LHC , JINST 3 , S08002 (2008), doi:10.1088/1748-0221/3/08/S08002.
- [8] LHCb collaboration A Large Hadron Collider beauty experiment, Technical Proposal, CERN-LHCC-98-004, <http://cdsweb.cern.ch/record/622031>.
- [9] ATLAS collaboration, The ATLAS experiment at the CERN Large Hadron Collider, 2008 JINST 3 S08003.
- [10] 高エネルギー加速器研究機構, ノーベル物理学賞でたどる標準理論 100 年の歴史, <https://www.kek.jp/ja/NewsRoom/Highlights/20120727150000/>.
- [11] Ulrik Egede, Higgs Production, August 1998, <http://www.hep.lu.se/atlas/thesis/egede/thesis-node10.html>
- [12] LHC Higgs Cross Section Working Group, <https://twiki.cern.ch/twiki/bin/view/LHCPhysics/LHCHXSWG>
- [13] ATLAS Collaboration, Technical Design Report for the Phase-I Upgrade of the ATLAS TDAQ System, CERN-LHCC-2013-018. ATLAS-TDR-023, September 2013, <https://cds.cern.ch/record/1602235>
- [14] 佐々木修、池野正弘, アトラスミューオントリガーシステムのためのエレクトロニクス開発と建設, 高エネルギーニュース, 26 巻 3 号, 2007 年, <http://www.jahep.org/hepnews/2007/Vol26No3-2007.10.11.12sasakiikeno.pdf>

- [15] O.Sasaki et al. , ASD IC for the thin gap chambers in the LHC ATLAS experiment, 10.1109/NSSMIC.1998.775179
- [16] 野本裕史 Star Switch Spec, 2005.
- [17] M. Bianco, on behalf of the ATLAS Muon Collaboration, Micromegas detectors for the muon spectrometer upgrade of the ATLAS experiment, Nuclear Instruments and Methods in Physics Research A 824 (2016) 496-500.
- [18] 長谷川誠, ATLAS 実験 RUN2 に向けたレベル 1 ミューオントリガープロセッサの改良, 神戸大学 修士論文, 2015.
- [19] Cerqueira, A.S. , Tile Calorimeter Upgrade Program for the Luminosity Increasing at the LHC, arXiv:1509.08994, September 2015, <http://cds.cern.ch/record/2056477>
- [20] Kataoka, Y. and Leontsinis, S. and Ntekas, K. , Performance Studies of a Micromegas Chamber in the ATLAS Environment, JINST 9 (2014) C03016.
- [21] Itamar Roth, Amit Klier, Ehud Duchovni, Testing sTGC with small angle wire edges for the ATLAS New Small Wheel Muon Detector Upgrade, IEEE Nucl.Sci.Symp.Conf.Rec.
- [22] 佐々木修: Muon Endcap Trigger in Phase-1, New Sector Logic Board, 2013.
- [23] S-link Specifications <http://hsi.web.cern.ch/HSI/s-link/spec/#s-link>
- [24] 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, 東京大学 修士論文, 2015.

# 謝辞

この2年間の研究生活において、多くの方に支えていただき、本論文をまとめることができたことを、この場を借りて深くお礼申し上げます。

まず指導教員である藏重久弥先生にはソフトウェアの開発に必要な数々の知識をはじめ、研究において大切な知識や考え方をご教授頂きました。各発表の準備のときも毎回添削いただいたり、CERN出張の際には今まで海外に行ったことがなかった僕にヨーロッパと日本での生活の違いを細かく教えていただいたりもしました。この2年間、僕に素晴らしい研究の機会を与えて頂きましたこと、あらゆる場面で支えて頂きましたことを、心から深く感謝致します。

前田順平先生にも大変お世話になりました。ソフトウェア開発に関する基本的な技術から発展的な知識まで教えて頂いた他、デバッグ・試験において必要な考え方をご教授頂きました。さらにCERNへの出張の際には海外での生活をしたことがなかった僕に研究だけでなく海外の様々な文化に触れる機会を与えていただきました。日本でもCERNでも、手取り足取り僕の面倒を見てくださったこと、心より感謝申し上げます。

神戸大学では、毎週のミーティングでは山崎祐司先生、越智敦彦先生、清水志真先生、Yuan Li先生にはたくさんご助言いただけましたことに御礼申しあげます。

また研究室全体のコロキウムでは竹内康雄先生、身内賢太郎先生、鈴木州先生、矢野孝臣先生、中野佑樹先生には僕の理解が不十分だったところにしっかりとご指摘を頂いたおかげで、正しい知識を身に着けることができましたことに感謝致します。

秘書の横山有美さんにはCERNへの出張手続きをはじめ、様々な事務手続きで支えていただきました。誠にありがとうございます。

Phase-1グループでは佐々木修先生、坂本宏先生、石野雅也先生、隅田土詞先生、青木雅人先生、奥村恭幸先生、岸本巴氏、鈴木翔太氏にはわからないことを丁寧に教えて頂き、様々な助言で何度も助けて頂きました。心から御礼申し上げます。また戸本誠先生、堀井泰之先生の周到な計画・指導の下、チームテスト無事に成功に納められたことを感謝致します。

CERNへの出張の際には川出健太郎先生、野辺拓也氏、田代拓也氏、救仁郷拓人氏、小野木宏太氏、佐野裕太氏、早川大樹氏、田中雅大氏には生活に役立つ知識を教えて頂いたり、車に乗せて頂いたり、一緒にご飯に誘って頂いたり、部屋を借して頂いたりといった、生活面で支えて頂きましたこと厚く御礼申し上げます。

ATLASの先輩である陳叶氏、木戸将吾氏には親切に接して頂きました。日本でもCERNでも、わからないことを教えて頂き、この2年間数々のアドバイスを頂いたことに大変感謝申し上げます。

他先輩方の岡直哉氏、長谷川誠氏、矢ヶ部遼太氏、山根史弥氏、橋本隆氏には学部4年生のときから物理の内容やプログラミングのことまで教えていただけましたことに御礼申し上げます。後輩の小川圭将君、中澤美季さん、又吉康平君、若宮光太郎君にはM2をはじめ研究室のみんなを支え、励まし、和ませて頂き、感謝しています。特に同じATLAS Phase-1グループである竹田康亮君には、先輩として役に立つことは全然できませんでしたが、誠実で勤勉なその姿に何度も支えて頂き、心から感謝しています。その素晴らしい姿勢を忘れずにこれからも是

非、研究生活に励んでください。

そして何より、神戸大学の同級生の皆様のおかげで学生生活を楽しく過ごせましたことに深く感謝致します。修士から2年間その独特な雰囲気場で場を和ませてくれた長坂憲子さん、学部4年生からの研究室配属以降、毎日一緒にコンビニまで散歩する仲になった齊藤大起君、長谷川大晃君、そして神戸大学での6年間、授業のある日から休みの日まで、苦楽を共に過ごしながら同じ坂道を歩んできた池田智法君、帝釋稜介君、阿部圭吾君。かけがえのない仲間達に、ここに感謝の意を表します。

共に Phase-1 グループとして一緒に頑張ってきた赤塚駿一君、水上淳君、岡崎佑太君には迷惑をかけたことも度々ありましたが、彼らと一緒に研究できたことで、充実した研究生活を過ごすことができました。心から感謝の意を表します。

夜シフトなど、辛いときでも一緒に頑張ってきた水越健太君、宿谷琴子さん、川口智美さんにも深く感謝致します。

最後に、僕に教育を受ける機会を与え、あらゆる面でサポートして頂いた家族に御礼申し上げます。