

# COMET 実験のためのドリフトチェンバー検出器の フロントエンド回路の開発

大阪大学大学院 理学研究科物理学専攻  
久野研究室 博士前期課程 2 年

林達也

2013 年 2 月 14 日

# 目次

<b>第 1 章</b>	<b>序章</b>	<b>9</b>
1.1	はじめに . . . . .	9
1.2	レプトンフレーバー . . . . .	10
1.2.1	ミューオンについて . . . . .	10
1.3	ミューオン電子転換現象 . . . . .	11
1.3.1	ミューオン電子転換過程 . . . . .	11
1.3.2	ミューオン電子転換現象探索の現状について . . . . .	12
1.4	ミューオン電子転換現象探索過程 COMET . . . . .	13
1.4.1	パルス陽子ビーム . . . . .	14
1.4.2	パイオン捕獲ソレノイド . . . . .	16
1.4.3	輸送ソレノイド . . . . .	18
1.4.4	検出器ソレノイド . . . . .	18
<b>第 2 章</b>	<b>ROESTI</b>	<b>21</b>
2.1	実験要求 . . . . .	21
2.2	要求実現策 . . . . .	22
2.3	ROESTI . . . . .	23
2.3.1	フロントエンド . . . . .	23
2.3.2	Waveform Digitizer . . . . .	23
2.3.3	FPGA . . . . .	23
2.3.4	Ethernet . . . . .	24
2.4	ROESTI プロトタイプ . . . . .	24
2.4.1	フロントエンド . . . . .	24
2.4.2	Waveform Digitizer . . . . .	25
2.4.3	FPGA . . . . .	25
2.4.4	Ethernet . . . . .	26
<b>第 3 章</b>	<b>プロトタイプの評価</b>	<b>29</b>
3.1	パターンコレクション . . . . .	29

3.2	波形評価	30
3.2.1	目的	30
3.2.2	評価方法	30
3.2.3	結果	31
3.2.4	今後の改善点	33
3.3	基本性能評価	34
3.3.1	目的	34
3.3.2	評価方法	34
3.3.3	結果	35
3.4	時間分解能測定	42
3.4.1	目的	42
3.4.2	評価方法	42
3.4.3	結果	43
3.5	通信システムテスト	48
3.5.1	目的	48
3.5.2	評価方法	48
3.5.3	結果	48
<b>第 4 章</b>	<b>考察・今後の課題</b>	<b>51</b>
4.1	電荷増幅に関して	51
4.1.1	S/N 比	51
4.1.2	クロストーク	52
4.2	消費電力に関して	53
4.3	時間分解能に関して	54
4.4	データ転送に関して	54
4.5	まとめ	55
<b>付録 A</b>	<b>ポアソン到着</b>	<b>58</b>
A.1	ポアソン到着	58
A.2	取得率	58
A.3	バッファサイズ	59
<b>付録 B</b>	<b>FPGA ファームウェア</b>	<b>60</b>
B.1	SPI コンフィギュレーション	60
B.1.1	シリアル PROM	60
B.1.2	ファームウェア	60
B.1.3	ポート説明	60
B.1.4	レジスタアドレスマップ	61

---

B.1.5	書き込みプログラム . . . . .	61
B.1.6	再コンフィギュレーション (リポート) . . . . .	62
B.2	ROESTI プロトタイプのリジスタマップ . . . . .	62
B.3	パケットインザーター . . . . .	64
B.4	リセット・ロード . . . . .	66
B.5	DAC 制御モジュール . . . . .	66
B.6	DRS 制御モジュール . . . . .	66
B.7	ADC インターフェース . . . . .	67
B.8	TCP チェーン接続 . . . . .	69
B.9	読み出しモード . . . . .	69
B.9.1	共通部分 . . . . .	69
B.9.2	ベースライン取得モード . . . . .	70
B.9.3	ノーマルトリガーモード . . . . .	71
B.9.4	セルフトリガーモード . . . . .	72
B.9.5	COMET トリガーモード . . . . .	72
B.10	イベントビルダー . . . . .	73
B.11	イベントデータパケット . . . . .	73
付録 C	ソフトウェア . . . . .	76
C.1	UDP 通信 . . . . .	76
C.1.1	プログラム . . . . .	76
C.1.2	使用方法 . . . . .	76
C.2	TCP 通信 . . . . .	77
C.2.1	プログラム . . . . .	78
C.2.2	使用方法 . . . . .	78
付録 D	今後の開発に関して . . . . .	80
参考文献		83

# 目次

1.1	DIO スペクトラム	12
1.2	SINDRUM II 実験	13
1.3	J-PARC レイアウト	14
1.4	COMET 実験レイアウト	15
1.5	陽子ビームタイミング	16
1.6	COMET 実験の加速器配置	16
1.7	Beam buckets 入射タイミングイメージ図	17
1.8	パイオン捕獲部レイアウト	17
1.9	検出器ソレノイド概略図	19
1.10	ストローチェンバーの概略図	19
1.11	ストローチェンバーの試作機	19
1.12	電磁カロリメータ概略図	20
2.1	ROESTI のフロー図	23
2.2	ROESTI プロトタイプ	24
2.3	DRS4 の動作原理のイメージ図	25
2.4	チェーン接続の実際の様子	26
2.5	チェーン接続イメージ図	27
3.1	波形パターン例	29
3.2	パターンコレクション	30
3.3	オシロスコープー DRS4 波形比較 1	31
3.4	オシロスコープー DRS4 波形比較 2	31
3.5	オシロスコープー DRS4 波形比較 3	31
3.6	オシロスコープー DRS4 波形比較 4	31
3.7	オシロスコープー DRS4 波形比較 5	31
3.8	オシロスコープー DRS4 波形比較 6	31
3.9	オシロスコープー DRS4 波形差分 1	32
3.10	オシロスコープー DRS4 波形差分 2	32

3.11	オシロスコープ DRS4 波形差分 3	33
3.12	ノイズサンプル図	35
3.13	ノイズ定義イメージ図	35
3.14	ch1~ch8 のノイズヒストグラム	35
3.15	ch9~ch16 のノイズヒストグラム	36
3.16	ch1~ch8 の線形性	37
3.17	ch10~ch16 の線形性	37
3.18	ch1~ch8 のフィット関数からのばらつき	38
3.19	ch10~ch16 のフィット関数からのばらつき	39
3.20	時間差取得の定義イメージ図	43
3.21	チップ間のタイミングの校正イメージ図	43
3.22	0 nsec 遅延 TDC ヒストグラム	44
3.23	100 nsec 遅延 TDC ヒストグラム	44
3.24	200 nsec 遅延 TDC ヒストグラム	44
3.25	300 nsec 遅延 TDC ヒストグラム	44
3.26	ASD の出力をオシロスコープで見た図	45
3.27	ROI Mode によるベースライン取得の一例	46
3.28	ベースラインプロファイル (Full Mode)	46
3.29	ベースラインプロファイル (ROI Mode)	46
3.30	ベースラインプロファイル 2(Full Mode)	47
3.31	ベースラインプロファイル 2(ROI Mode)	47
3.32	転送スピード vs ピーク値のプロット図 (chip1)	48
3.33	転送スピード vs ピーク値のプロット図 (chip2)	49
3.34	データ転送スピードプロット図 (COMET トリガー)	49
3.35	データ転送スピードプロット図 (ノーマルトリガー)	50
4.1	クロストークイメージ図	53
4.2	データ転送スピードプロット図 2(COMET トリガー)	54
4.3	データ転送スピードプロット図 2(ノーマルトリガー)	54
A.1	バッファがフルになる確率	59
B.1	イーサネットパケット破損のイメージ図	65
B.2	UDP/TCP 通信のイメージ図	65
B.3	DRS4 設定の状態遷移図	68
B.4	データ読み出し時の ADC、DRS4 の状態遷移図	70
B.5	ADC データ転送の状態遷移図	71
B.6	ノーマルトリガーのタイミングイメージ	71

---

B.7	ビームパルスの時間構造 . . . . .	72
B.8	トリガー到達時の動作イメージ図 . . . . .	73
B.9	イベントデータのケットフォーマット . . . . .	74
C.1	RBCP 操作の一例 . . . . .	78
C.2	イベントデータ取得の一例 . . . . .	79

# 表目次

2.1	データ量一覧 . . . . .	28
3.1	ノイズヒストグラムの RMS 値 . . . . .	36
3.2	電荷－電圧変換式 1 . . . . .	37
3.3	電荷－電圧変換式 2 . . . . .	38
3.4	ch1 のクロストーク . . . . .	39
3.5	ch3 のクロストーク . . . . .	39
3.6	ch4 のクロストーク . . . . .	40
3.7	ch5 のクロストーク . . . . .	40
3.8	ch6 のクロストーク . . . . .	40
3.9	ch7 のクロストーク . . . . .	40
3.10	ch8 のクロストーク . . . . .	40
3.11	ch10 のクロストーク . . . . .	40
3.12	ch11 のクロストーク . . . . .	41
3.13	ch12 のクロストーク . . . . .	41
3.14	ch13 のクロストーク . . . . .	41
3.15	ch14 のクロストーク . . . . .	41
3.16	ch15 のクロストーク . . . . .	41
3.17	ch16 のクロストーク . . . . .	41
4.1	S/N 比 . . . . .	52

## 概要

ミューオン電子転換過程探索実験 (COMET) は、荷電レプトン混合現象の探索を通して、素粒子物理の標準模型を超える新しい物理を探索する素粒子実験である。本実験は、大強度陽子加速器施設 (J-PARC) にて 2016 年頃に開始を目指して準備が進められている。ミューオン電子転換現象は、標準模型の枠内ではニュートリノ振動を考慮にいれたとしても分岐比は  $10^{-54}$  と実験の検出レベルより遥かに小さい。しかしながら、標準模型を超える理論として新粒子を導入した理論を考えると、それらの新粒子間の混合が引き金となって、 $\sim 10^{-14}$  と実験で検出可能なレベルに現れる可能性がある。従って、もしミューオン電子転換現象を発見できれば、直ちに新物理の証拠となり、標準模型を超える新物理を理解する上での不可欠な情報をもたらしてくれるものと期待されている。

ミューオン電子転換現象の信号の同定は、ミューオニック原子から放出される特徴的なエネルギーを持った電子を捉えることにより行われる。信号の主な背景事象としては、ミューオニック原子からの崩壊電子があるが、この崩壊電子のエネルギーは信号のエネルギー近傍にまで広がっている。このため、信号事象由来の電子を高精度で計測することにより、背景事象から区別する必要があるが、そのためには、物質量の少ない飛跡検出器を使用し、周りを真空にして多重散乱の影響を低減することで運動量分解能を極限にまで高める必要がある。本実験ではこれらの要求を解決するための検出器としてストロー飛跡検出器を使用する。この検出器の読み出しは、真空内に配置された検出器からの多チャンネルの信号を効率よく処理し、かつ実験のバックグラウンドになりうるパイルアップを低減するため、波形計測機能を備えた専用の読み出しエレクトロニクスを開発する必要がある。

本研究では、ストロー飛跡検出器を読み出すためのフロントエンド回路のプロトタイプの開発・評価を行うことが目的である。このプロトタイプを用いて、線型性やノイズ評価による S/N 比や時間分解能、データ転送などの評価を行い、実験要求を満たしているかについて考察した。S/N 比は想定されている最低電荷量 (-16 fC) に対し 13.5 以上となり、検出器の信号を観測するには十分な値となった。データ転送に関しても実験要求である 360 Mb/1.4s を大幅に上回る 850 Mbps となり、こちらも実験要求を満たす結果となった。時間分解能に関しては、時間差が 0 nsec の場合は 0.16 nsec と要求を満たすことが出来た。しかしながら、時間差を 100 nsec、200 nsec を大きくするとダブルピークやテールが広がる結果となった。原因としては、ベースラインの変動を考慮しており、解析時の補正や原因説明が今後の課題となる。

# 第 1 章

## 序章

### 1.1 はじめに

近年、LHC による Higgs Boson らしき新粒子の発見や、ニュートリノ混合角の計測により、注目されることの多い素粒子物理学であるが、現在観測されているほとんどすべての素粒子の反応は、標準模型により記述されている。

しかしながら、現代の素粒子標準模型は究極の理論ではない。近年、標準模型の枠に収まらない現象があることは、中性レプトンであるニュートリノが別の世代へと移り変わる現象（ニュートリノ振動）の発見により確実となった。このような中、標準模型では説明できない現象を発見し、標準模型を内包するような枠組により、素粒子の振る舞いを統一的に理解することが、現代の素粒子物理学の最重要課題となっている。

標準模型を超える現象というのは、言うまでもなく簡単に発見できるものではなく、標準模型が有効であると考えられているよりも高いエネルギースケールで発見されることが予想される。つまり、標準模型を超える現象の発見を目指す高エネルギーの実験は、加速器を使用して、粒子を高エネルギーまで加速して衝突させ、そこで起こる現象を詳細に調べることにより成立する。しかしより高いエネルギーまで加速するためには、より大規模の加速器が必要となり、費やされる費用も世界各国が共同で予算を投入しなければ実現しないほど莫大なものになってきている。このようなアプローチは大規模な加速器を開発し、未知のエネルギーフロンティアを目指すものであり、Higgs Boson の発見という目覚ましい活躍を魅せる欧州素粒子原子核研究機構 (CERN) の大型ハドロン衝突加速器 (Large Hadron Colider, 以下 LHC) などが例として挙げられる。一方、既存あるいは新規に建設された加速器で、大強度ビームを実現し、非常に稀な現象を探す稀崩壊事象探索というアプローチもある。新たな大規模加速器の開発の費用が莫大になった現代、素粒子物理学の発展の鍵を握る手段として、この強度フロンティアを目指す手法は、非常に有効であり、今後さらに有望視されていくであろう。

さて、稀崩壊とは、低次の効果が様々な要因で制限されているが、高次の効果が存在することによって、稀な頻度でありながら引き起こされる現象のことである。そのため、稀崩壊を調べることによって低次の効果に影響を受けることなく、高次の相互作用を調べることが可能になる。さらに

は低エネルギーであっても、高次の効果を介することにより標準模型を超える物理現象を探索出来る可能性を抱いている。我々は、このような稀崩壊の一種である、 $\mu^- N \rightarrow e^- N$  の探索を通じて、標準模型を超える新たな物理の手掛かりを探り、より高いエネルギースケールの物理に迫ることを目指している。

本章では、本研究の物理的背景について簡単にまとめる。まず、標準模型で記述される粒子の振る舞いについて述べ、標準模型では記述できないレプトンフレーバー保存を破るような崩壊の可能性とその探索方法について述べる。

## 1.2 レプトンフレーバー

### 1.2.1 ミューオンについて

ミューオンは1937年に Seth Neddermeyer と Carl David Anderson によって宇宙線の中に電子より重い新粒子として発見された [5]。その後、 $\mu$  粒子に関しては、様々な精密測定が行われ、素粒子標準模型において、第二世代の荷電レプトンとして位置づけられた。荷電レプトンには、電子 ( $e^-$ ) ミューオン ( $\mu^-$ )、タウオン ( $\tau^-$ ) の3世代とそれぞれの反粒子  $e^+$ 、 $\mu^+$ 、 $\tau^+$  が存在し、それぞれ付随する中性レプトンである3種類のニュートリノと対をなしている。

$\mu^-$  粒子が物質中で停止した場合、ミューオンは物質中で原子核に捕獲されてミューオン原子を形成する。その後ミューオンは X 線を放射しながら励起状態から 1s の規定状態にまで落ちていき、最終的に

$$\mu^- + N \rightarrow e^- + \nu_\mu + \bar{\nu}_e + N \quad (1.1)$$

といったような Michel 崩壊か、

$$\mu^- + (A, Z) \rightarrow \nu_\mu + (A, Z - 1) \quad (1.2)$$

のようなミューオン原子核捕獲反応を引き起こす。ここで  $A$  はミューオンが静止した物質の原子量であり、 $Z$  は原子番号である。これらの反応においては、レプトン数と呼ばれる量子数が保存されている。つまり、素粒子の反応の前後でレプトン数の総和が保存するのである。更に、荷電レプトンでは世代毎のレプトン数である「フレーバー」(香り) と呼ばれる量子数も厳密に保存されている。例えば、ミューオン原子核捕獲反応を例にとると、反応前はミューオンの数  $L_\mu$  が  $+1(\mu)$  であるが反応後は、ミューオンの数  $L_\mu$  は  $+1(\nu_\mu)$  となっており、保存していることが分かる。他の崩壊モードに対しても同様にレプトン数は保存していることが知られている。このフレーバーの保存はニュートリノの質量を 0 としている標準模型では当然の帰結であり、厳密に守らなくてはならない対称性の一つであるが、近年中性レプトンにおけるフレーバーの保存はクォークでのそれと同様に、破れているという実験結果が多数報告されている。スーパーカミオカンデや KEK-PS K2K や、J-PARC T2K の解析結果がそれである [4],[8],[2]。このような状況の中、残る荷電レプトンのフレーバー保存の破れについて調べたいのは、当然であろう。もし荷電レプトンに対しても、レプトンフレーバーの破れが起きているとなると (Charged Lepton-Flavor Violation, cLFV)、

我々 COMET 実験の目標とする  $\mu^- + N \rightarrow e^- + N$  といった反応も起こりえることとなり、標準模型を越える新たな物理への有力な手掛かりになる。

### 1.3 ミューオン電子転換現象

cLFV 探索実験は以上のような物理のモデルの精密な実験的追及が出来るだけでなく、様々な新しい物理に対して重要な事実を示す可能性がある。また、cLFV 探索実験によって得られた分岐比から、間接的に SUSY 粒子の質量スケールを決めることも可能である。

#### 1.3.1 ミューオン電子転換過程

前述のように、標準模型の枠内では、物質中で静止したミューオンは、ミューオニック原子を形成した後、Michel 崩壊を起こすか、ミューオン原子核捕獲反応により原子核に捕獲される。しかし、我々が期待するように標準模型を超える物理が存在し、荷電レプトン間で混合が存在すると、ニュートリノ放出を伴わないミューオン捕獲反応が起こりうる。

$$\mu^- + N \rightarrow e^- + N \quad (1.3)$$

この反応をミューオン電子転換過程 ( $\mu - e$  conversion) と呼ぶ。ミューオン電子転換過程の分岐比は、ミューオン原子核捕獲反応に対する比として

$$B(\mu^- + (A, Z) \rightarrow e^- + (A, Z)) = \frac{\Gamma(\mu^- + (A, Z) \rightarrow e^- + (A, Z))}{\Gamma(\mu^- + (A, Z) \rightarrow \nu_\mu + (A, Z - 1))} \quad (1.4)$$

と定義される。ここで、 $\Gamma$  は崩壊幅である。電子  $(A, Z)$  の反応後の状態は基底状態にも励起状態にも成り得るが、一般的にコヒーレント捕獲と呼ばれる基底状態になる反応に支配される。励起状態への遷移に対するコヒーレント捕獲の反応率の割合は原子核中の核子の数に比例する。コヒーレント捕獲では、すべての核子が捕獲に関与しうるためである。

ミューオン電子転換過程が起こると、単一エネルギーの電子が静止標的から放出され、そのときの電子のエネルギー  $E_{\mu e}$  は

$$E_{\mu e} = m_\mu - B_\mu - E_{rec}^0 \quad (1.5)$$

となる。ここで  $m_\mu$  はミューオンの質量、 $B_\mu$  はミューオニック・アトムの中での 1s 軌道における束縛エネルギー、 $E_{rec}^0$  は原子からの反跳エネルギーである。原子からの反跳エネルギーは原子核の質量  $M_A$  を用いて、

$$E_{rec}^0 \approx (m_\mu - B_\mu)^2 / (2M_A) \quad (1.6)$$

と近似できるが、微小なため実験遂行上は無視できる。 $B_\mu$  は原子核の種類によって異なるため、実験に用いたミューオン停止標的の原子核によって、ミューオン電子転換による電子の運動量ピークの値も異なってくる。例として、アルミニウムを停止標的として用いた場合には、 $E_{\mu e} = 105.0$  MeV となる。このときの放出電子の運動量を測定すると、実際には  $\mu^- \rightarrow e^- \nu_\mu \bar{\nu}_e$  崩壊過

程 (Decay In Orbit 以下 DIO) からの電子が支配的となり、図 1.1 に示す運動量スペクトラムになる。分岐比にして、 $10^{-16}$  以下のレベルでミューオン電子転換過程のシグナルと  $\mu^- \rightarrow e^- \nu_\mu \bar{\nu}_e$  のエンドポイントを識別するためには、電子飛跡検出器には、1 MeV/c 以下の運動量分解能が必要となる。

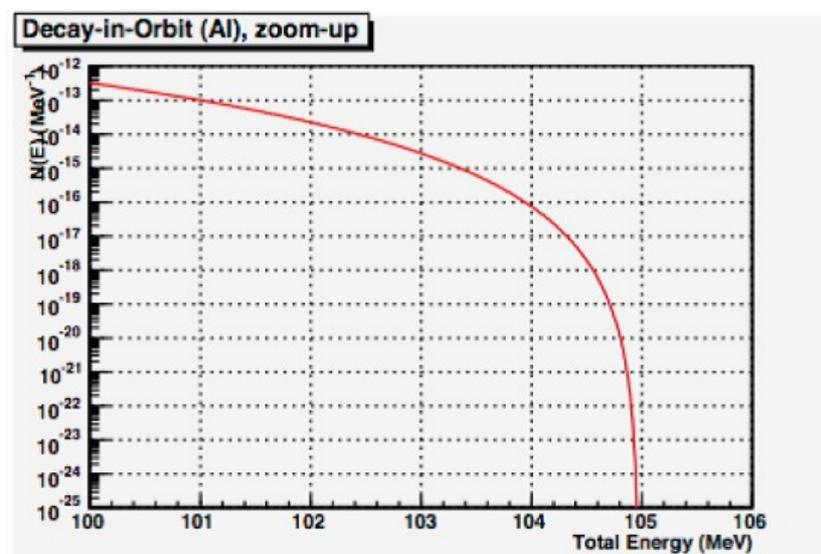


図 1.1 アルミニウム停止標的からの DIO による電子のエネルギースペクトラム

### 1.3.2 ミューオン電子転換現象探索の現状について

ミューオン電子転換現象の探索実験に関しては、現在までに様々な探索実験が行われている。ミューオン電子転換現象の世界記録を保持している実験は PSI で行われた SINDRUM II 実験である。静止標的が金 (Au) のときのデータから、ミューオン電子転換現象の分岐比の上限値が  $B_{\mu e}^{Au} < 7 \times 10^{-13}$  (90% C.L.) と求められている。

図 1.2 に SINDRUM II 実験で得られた電子のエネルギースペクトラムを示す。この図に示されているように実験データから得られたエネルギースペクトラムには、事象信号の領域より高い領域にバックグラウンドと推測される現象が見え始めており、このことが原因となって実験グループは、さらに実験期間を延長して統計をためることを断念した。[3]

今後、新たにミューオン電子転換現象探索実験を行うには、何かしらの手段を講じて SINDRUM II 実験で観測されたような背景事象を減らす方策が必要となってくる。

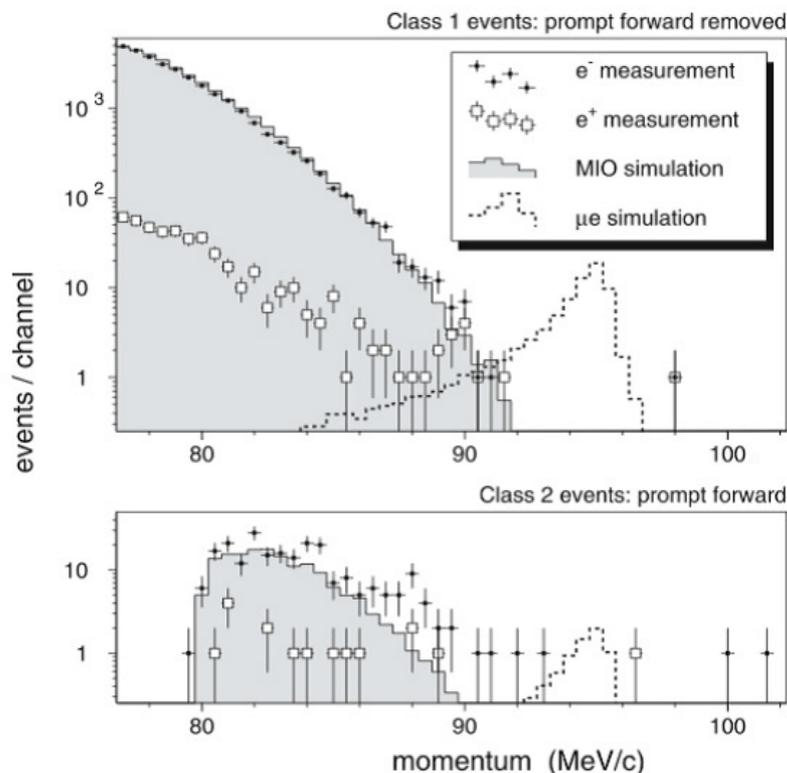


図 1.2 SINDRUM II 実験の静止標的が金 (Au) 測定結果

#### 1.4 ミューオン電子転換現象探索過程 COMET

現在の世界記録よりもさらに分岐比の感度を向上し、ミューオン電子転換現象探索の発見を目標とする COMET 実験が、2016 年の実験開始に向けて準備されている。

COMET 実験は東海村の J-PARC で計画されている。J-PARC は図 1.3 に示すように 3 つの加速器から構成されている。まず、最初にリニアックと呼ばれる線形加速器で加速された陽子は、Rapid Cycling Synchrotron(RCS) で 3 GeV まで加速され、メインリング (MR) に入射される。最大 30 GeV まで加速された陽子は、ハドロンホールに遅い取り出し法により引き出され、COMET 実験が予定されているビームラインへ導かれる。図 1.4 に COMET 実験の概略図を示す。COMET 実験では、陽子ビームを捕獲ソレノイド内の標的に当てて生成したパイオンを高効率で収集し、そのパイオンが崩壊してできるミューオンを利用する。それらのミューオンをアルミニウムの静止標的に止め、そこから崩壊して出てくる電子の運動量を測定する。もしミューオン電子転換現象が存在すれば、前述のように特徴的な運動量の電子が観測されるはずである。COMET 実験のセットアップでは、最上流のパイオン生成標的から最下流の検出器ソレノイドまでがすべて真空に引かれており、二次粒子の多重散乱を低減することができる。(図 1.4)

COMET 実験は Phase-I、Phase-II の 2 段階で計画されている。目標感度は Phase-I では  $10^{-14}$ 、Phase-II では  $10^{-16}$  以下となっており、Phase-I はミューオン電子転換現象の分岐比の上限値を更新することに加え、Phase-II の試験実験も兼ねている。Phase-I ではパイオン捕獲部分と輸送ソレノイドの 90 度分までの装置を建設して実験を行う。メインとなる検出器は Phase-I と Phase-II では異なるが、Phase-I では Phase-II の背景事象になりうるビームバックグラウンドの測定も行われる予定である。以下では、COMET 実験の Phase-II に関し、一次陽子ビームに対する要求、パイオン捕獲部分、輸送ソレノイド部分、検出器ソレノイド部分の特徴に関して述べる。

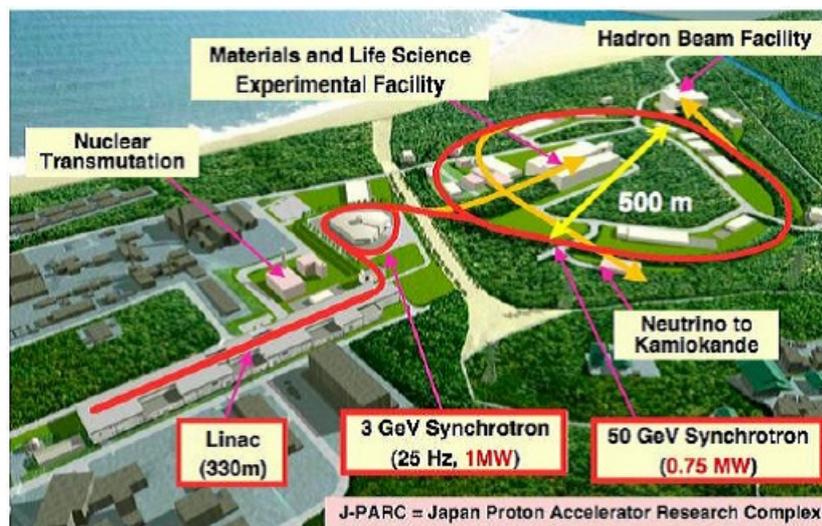


図 1.3 COMET 実験が計画されている東海村の J-PARC のレイアウト

### 1.4.1 パルス陽子ビーム

COMET 実験において、パイオンの生成効率を上げるためには、できるだけ高いエネルギーの陽子ビームを使うほうが良いが、陽子エネルギーが 9~10 GeV 以上になると、背景事象の原因となる反陽子の生成確率が增大するため、入射陽子のエネルギーは 8 GeV に抑えることを予定している。

COMET 実験では、超高感度での探索を行うために、 $10^{18}$  個のミューオンを必要としている。このように大量のミューオンを高効率で生成するため、初期陽子ビームは、強度を出来る限り高める必要がある。現在の設計では、陽子ビームの強度は、 $8\text{GeV} \times 7\mu\text{A}$  ( $4.4 \times 10^{13}$  protons/s) を想定しており、このビーム強度が達成出来れば、COMET 実験は、目標とする物理感度に数年で到達可能である。

ミューオンは、停止標的で静止後、ミューオン原子を形成し、およそ  $1\mu\text{sec}$  程度の寿命で崩壊する。このため、ミューオン電子転換現象転換過程のシグナルは陽子のタイミングから遅れて生成される。一方で、陽子ビームに起因する  $\pi^-$  崩壊によるバックグラウンドは、陽子ビーム後 100

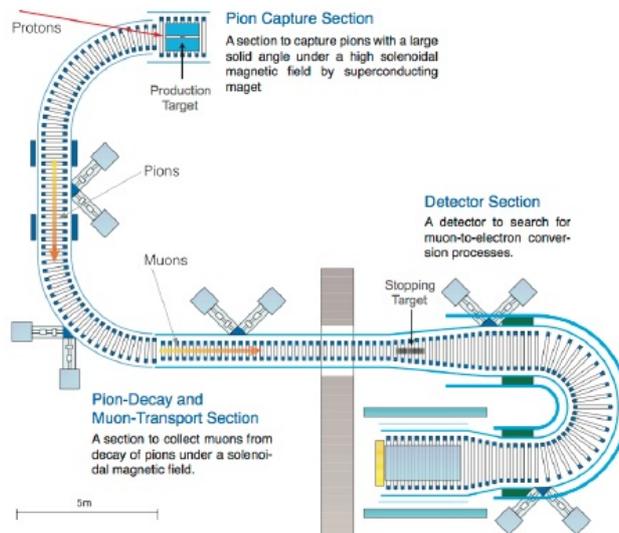


図 1.4 COMET 実験のパイオン捕獲ソレノイド、輸送ソレノイド、検出器ソレノイドの概略図。

nsec 以下の領域に発生する。COMET 実験では、信号事象と背景事象を区別するために、この時間情報を利用するべく、図 1.5 のように、パルス化された陽子ビームを使用する。パルス間の時間間隔は、ミュオニック原子の寿命に合わせて数  $\mu\text{sec}$  オーダーの間隔を持つように加速器のバンチ構造を整える。こうすることで、SINDRUM II 実験<sup>\*1</sup>で見えていたような背景事象を排除できると考えている。

パルス陽子ビームのパルスとパルスとの領域に陽子があると、背景事象を作り出す要因となるため、パルス間の陽子は限りなく少ない必要がある。メインパルスに入っている陽子の数  $N_{main}$  と、パルス間にいる漏れ出し陽子の数  $N_{betweenpulses}$  の比  $R_{Extinction}$  を、Proton Extinction Factor(以下 Extinction) と呼ぶ。COMET 実験の物理感度を考慮した場合に要求される Extinction は  $10^{-9}$  以下である。

$$R_{Extinction} = \frac{N_{betweenpulses}}{N_{main}} \quad (1.7)$$

十分に良い Extinction を実現することが COMET 実験の成功の鍵を握っている。この高い Extinction を達成するために Miss-kick injection と呼ばれる手法が考えられている。J-PARC の MR  $\rightarrow$  beam bucket を入射する際、基本的な入射方法としては一回の入射で RCS の 2 つの beam bucket を kicker で MR  $\rightarrow$  入射する。Miss-kick injection は kicker のタイミングを半位相分、前後にずらして injection を行い、漏れ出し陽子が MR  $\rightarrow$  入射しないようにする手法である(図 1.7)。この手法を用い、MR の運転を考慮して図 1.6 のような時間構造が検討されている。現在の Extinction の測定では速い取り出しによるビーム取り出しではあるが、Miss kick injection

<sup>\*1</sup> SINDRUM II 実験で観測された候補事象は時間情報を用いた解析により、 $\pi$  中間子起源の背景事象である可能性が高いと考えられている。

を用いて、 $R_{Extinction} = 1.5 \times 10^{-11}$  を達成している [11]。今後は MR からの遅い取り出しでの Extinction の測定を行う必要がある。

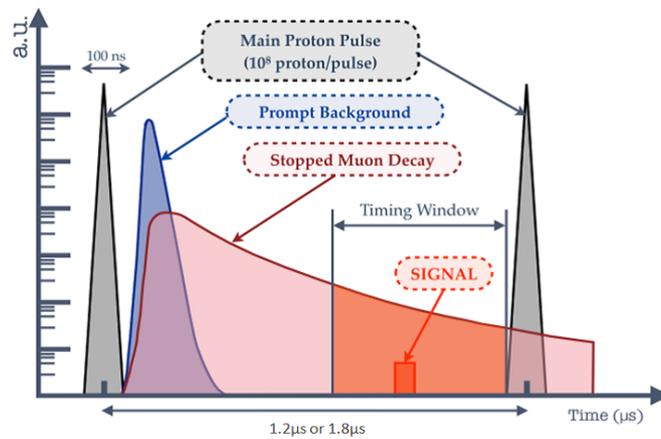


図 1.5 COMET 実験における初期陽子ビームの時間構造の概略図。背景事象除去のために、 $1.1\mu\text{sec} \sim 1.7\mu\text{sec}$  の時間幅を持った陽子パルスビームを使用する。

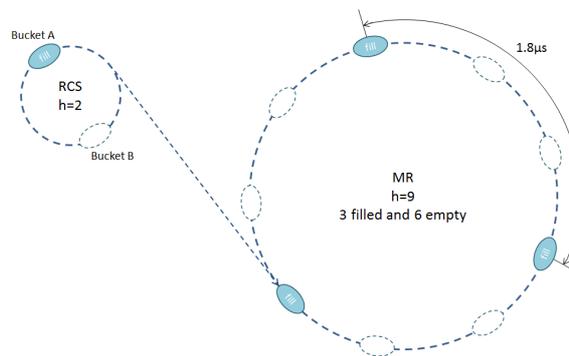


図 1.6 COMET 実験の加速器の配置。この図で示しているのは、MR の  $h=9$ 、3 つのバンチを満たし、6 つバンチを空にしている状態である。配置はまだ検討中であり、MR の  $h=8$ 、4 つのバンチを満たし、4 つバンチを空にする配置も考えられている。

#### 1.4.2 パイオン捕獲ソレノイド

MR から遅い取り出しで引き出した大強度パルス陽子ビームを図 1.8 のように角度をつけてパイオン生成標的に照射する。標的にはより多くのパイオンを生成するために、タングステンのような原子番号の大きな物質を用いる。発生したパイオンは、最大 5 T の強磁場を発生する超電導ソレノイドコイルによって捕獲される。COMET 実験では、ミュオン停止標的において確実にミュオンを静止できるように、低エネルギーのミュオン ( $\sim 40 \text{ MeV}/c$ ) のみを効率よく停止標的に

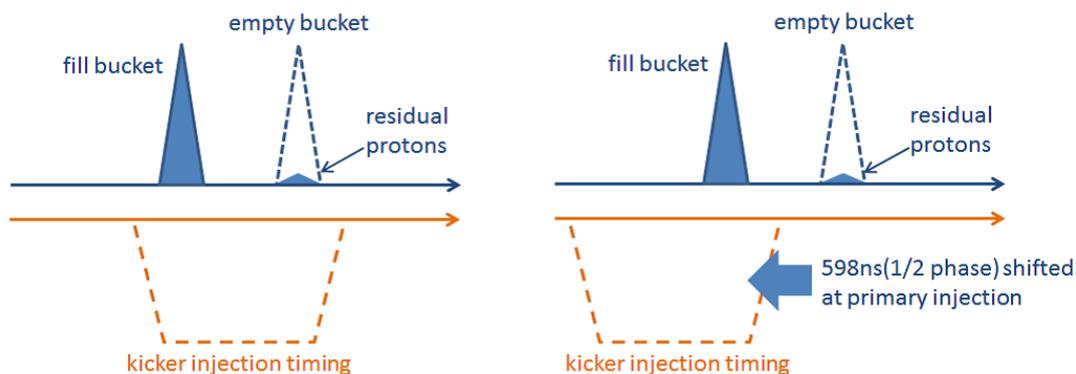


図 1.7 Beam buckets 入射タイミングイメージ図。左側がノーマルオペレーション、右側が Miss kick injection のタイミングとなる。

導く。そのため、ミュオン親粒子であるパイオンのうち、背景事象の原因となる高エネルギーのパイオンを除去する必要がある。高エネルギーパイオンは標的の前方に多く発生するが、我々が必要とするミュオン親粒子である 100 MeV 程度の低エネルギーパイオンは前方・後方であまり強度が変わらない。そのため、COMET 実験では、陽子ビームに対して後方に発生したパイオンのみを捕獲するように磁石システムが設計されている。

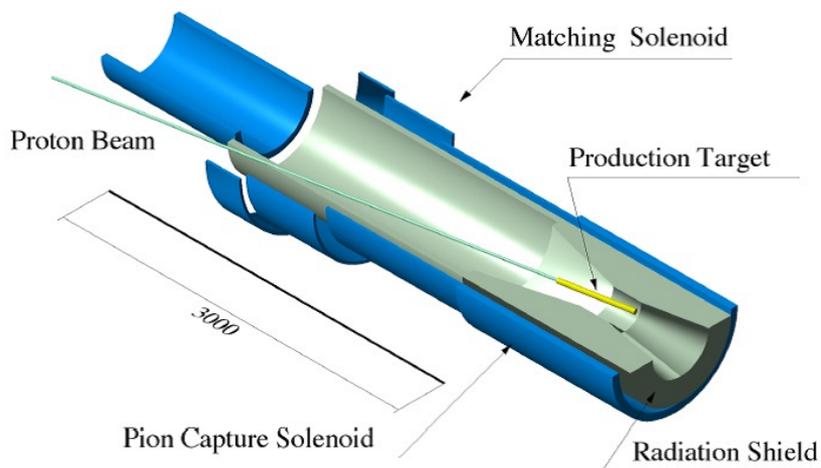


図 1.8 パイオン生成標的とパイオン捕獲ソレノイドの概略図。初期陽子ビームは加速器から角度をつけて生成標的に打ち込まれる。後方に生成したパイオンのみを捕獲し、輸送ソレノイドへと運ばれる。

### 1.4.3 輸送ソレノイド

パイオン捕獲ソレノイドで捕獲されたパイオンは、ミューオンへ崩壊しながら実験室へと導かれるために、輸送ソレノイドへ運ばれる。輸送ソレノイドはミューオンの運動量を選択するために図 1.9 のように、180°カーブしている超電導ソレノイドでできており、2 T の磁場を持つ。曲率をもったソレノイド磁場は、荷電粒子が通過した際、磁場の向きに対して、荷電粒子を垂直方向にドリフトさせる効果を持っている。このとき粒子のドリフト量 [m] は、

$$D = \frac{1}{0.3 \times B} \times \frac{s}{R} \times \frac{p_l^2 + \frac{1}{2}p_t^2}{p_l} \quad (1.8)$$

で与えられ、運動量に依存する。B[T] はソレノイド磁場、 $p_l$  は粒子のビーム軸方向の運動量、 $p_t$  はビーム軸に対して垂直方向の運動量、s[m] はカーブド・ソレノイドに対する粒子軌道距離、そして R [m] はソレノイド半径である。s/R はソレノイドの曲げ角に対応する。この式 1.8 をもとに、ソレノイドの半径を最適化し、コリメータを配置することで、ミューオンを輸送する際、高エネルギーのミューオンを選別し除去する。COMET 実験の輸送ソレノイドはミューオン停止標的で静止する低エネルギーミューオン (40 MeV/c) に対しては高い輸送効率を持ちながら、高エネルギーのミューオンは輸送されないように設計されている。この輸送ソレノイドにより、極めて大強度かつ高純度な低エネルギーミューオンビームが生成できる。

### 1.4.4 検出器ソレノイド

ミューオン停止標的で静止したミューオンは、電子に崩壊したあと電子輸送ソレノイドで測定器へ輸送される。電子輸送ソレノイドもミューオン輸送ソレノイドと同じく、運動量を選択するために、図 1.9 のようにカーブしている。このカーブソレノイドは、電子をドリフトさせることで、バックグラウンドとなる低エネルギー DIO 電子を 1/10000 以下までに落とすとともに、正電荷を持った粒子が検出器に到達することを防止する。一方、ミューオン電子転換過程のシグナル電子 (105 MeV/c) のアクセプタンスは 47% を保っている。これにより、飛跡検出器など検出器の計数率を下げることに役立っている。

#### ストロー飛跡検出器

COMET 実験では信号事象を背景事情から識別するために、電子の運動量を正確に測定することが重要となる。そのために、ソレノイド電磁石の中に 5 つの位置検出器を設置し、磁場中の電子の軌跡を求め電子の運動量を再構成することで、信号事象を同定する。本実験では飛跡検出器としてストロー飛跡検出器 (以下ストローチェンバー) を用いる (図 1.10)。ストロー飛跡検出器を用いる理由は、真空中でも用いることのできる低物質検出器という点がある。現在試作機を製作し性能評価が行われている (図 1.11)。

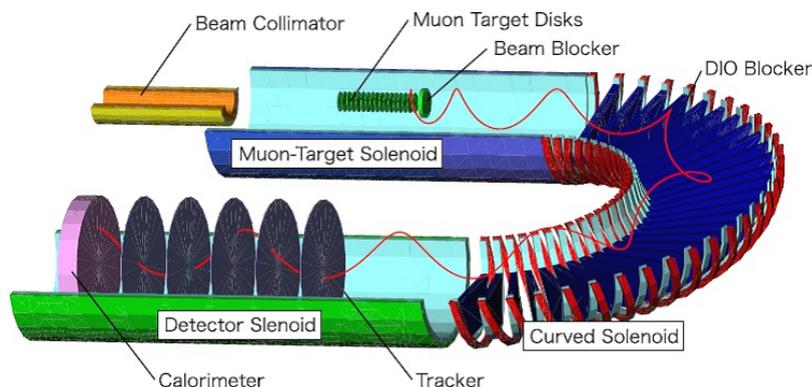


図 1.9 電子輸送ソレノイドと、検出器ソレノイドの概略図。最下流には、ストロー飛跡検出器と電磁カロリメータが設置してある。ミュオン電子転換過程のシグナルの飛跡を赤で示している。

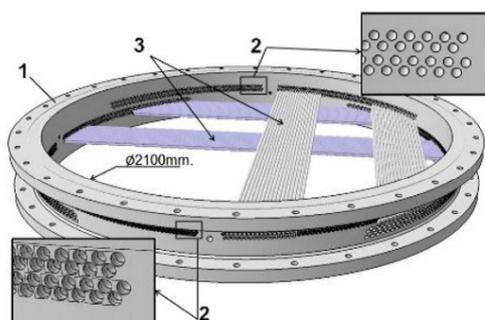


図 1.10 COMET 実験の飛跡検出器としてのストローガスチェンバーの概略図。208 本のストローを板状に並べたものが 4 枚あり、4 枚を 1 組として 1 層と数えてソレノイド電磁石の中に 5 層配置される計画である。



図 1.11 評価に使われたストローチェンバーの試作機

### 電磁カロリメータ

ストローチェンバーだけでは、トラッキングによる運動量の再構成の際に誤再構成が引き起こされる可能性がある。また、トラッカーでは、運動量測定しか行えないため、仮に電子以外の荷電粒子が背景事象として測定器系に混入しても粒子の種類を識別できない。そこで、トラッカーの運動量測定を保証し、背景事象となる他の粒子の種類を識別を行うために、測定器系最下流に図 1.12 のような直径 1 m の電磁カロリメータを設置し、電子の運動エネルギーを測定する。これによりイベント・トリガーの生成も行う。さらに電磁カロリメータをセグメント化することにより、電磁カロリメータでのヒット位置をトラッキングポイントに追加することで、運動量測定の精度

向上が望める。

現在、LYSO、GSOなどの結晶を積層させ、APD(Avalanche Photodiode)で読み出す試作機による研究開発が行われている。

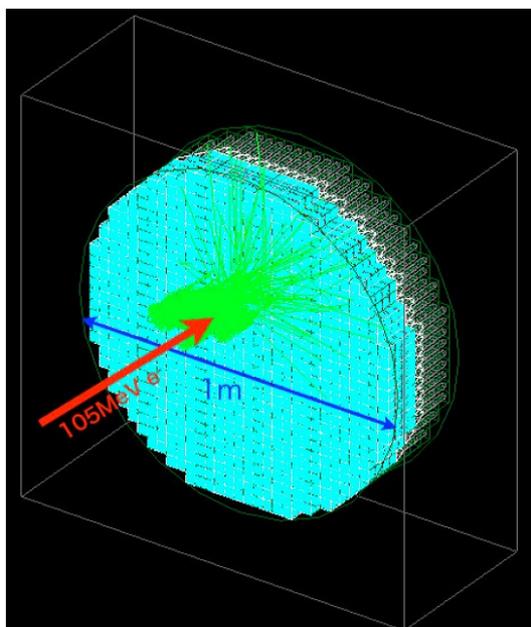


図 1.12 COMET 実験の最下流に設置されるカロリメータの概略図。COMET 実験では、直径 1 m の大きさの電磁カロリメータが計画されている。

## 第 2 章

# ROESTI

ROESTI(Read-Out Electronics of Straw Tube Instrument) とは、COMET 実験に用いられるストローチェンバーの波形読み出しを行うためのフロントエンドデジタイザーボードである。本論文の目的は、ROESTI のプロトタイプを設計・製作し、その性能を評価することである。この章では ROESTI ボードについて説明していく。

### 2.1 実験要求

ストローチェンバーの読み出し回路に対する実験要求としては以下のようなことがあげられる。

#### 時間分解能

COMET 実験では運動量  $1\text{MeV}/c$  の分解能を達成するためストローチェンバーの位置分解能としては  $700\ \mu\text{m}$  を満たす必要がある [1]。このためストローチェンバーのドリフト速度を  $5\ \text{cm}/\mu\text{sec}$  と仮定した場合、時間分解能としては最低でも  $14\ \text{nsec}$  を満たす必要がある。しかしながら、ストローチェンバーからの信号の立ち上がり時間が  $10\ \text{nsec}$  程度である点と、ストローチェンバーの位置分解能 (今までの研究によると  $100 \sim 200\ \mu\text{m}$  程度とされている。参考文献 [12]) を読み出し回路の時間分解能で制限しないようにする点を考慮すると、時間分解能としては  $2\ \text{nsec}$  程度必要となる。

#### パイルアップ事象の選別

カーブソレノイドを採用することによりパイルアップ自体のレートはそこまで高くないと評価されているが、実際にカロリメータで  $105\ \text{MeV}$  のエネルギーが観測されたときに ADC, TDC の情報だけでは 1 つの粒子によるものか、パイルアップの事象であるのかの判断が非常に難しい。したがってパイルアップの事象を比較的容易に行える必要がある。

#### 多チャンネル読み出し

COMET 実験では検出器が真空中に設置される。読み出し回路自体はガスマニホールド内の設

置を考えているが、構造上真空フィードスルーを介してデータは送られる。しかしながら、検出器の読み出しチャンネルは4000以上あり1本1本をフィードスルーで真空壁から外に出すことは現実問題として技術的に非常に難しい。そのため1枚の読み出し回路で多チャンネルの読み出しが可能であったり、チェーン接続するなどして複数枚のボードを最終的に1本のケーブルで読み出すことが出来るような仕組みが必要となる。

### ノイズ低減

一般的にアナログ信号をケーブルで長く伝達すればするほど、信号にノイズがのりやすくなる。ツイストケーブルなどノイズを低減できるケーブルもあるが、他の対策として検出器にできるだけ近い位置でアナログ信号をデジタル化しPCへ転送することが挙げられる。

COMET 実験でも読み出し回路をストローチェンバーに近い位置に実装しアナログ信号を早い段階でデジタル化することでノイズの低減を目指す。そのためガスマニホールド内に実装できるように読み出し回路の大きさにも気を付ける必要がある。検出器からの要求としては8 cm × 20 cm より小さくする必要がある。

### その他

検出器からの信号は小さいため増幅や波形形成などが必要となる。また熱を抑えるために低消費電力化も目指している。

## 2.2 要求実現策

2.1 節で取り上げた要求の実現方法として、読み出し回路で波形をサンプリングしデータを転送する Waveform Digitizer を採用した。波形をサンプリングすることで、電荷情報、時間情報の取得に加えてパイルアップの処理もオフラインで容易に行うことが出来る。時間分解能に関してはサンプリングスピードを500 MSPS 以上にすることで満たすことができ、フィッティングを行うことでさらに時間分解能の向上が期待される。

しかしながら Waveform Digitizer では、波形をサンプリングするため、ADC, TDC を使用する場合と比べてデータ量が非常に多くなる。データ取得のタイムウインドウは1  $\mu\text{sec}$  であるため、例えば500 MSPS、ADC 分解能12bit で波形をサンプリングした場合、1チャンネルの1イベント分のデータ量は  $500 \times 12\text{bit} = 6000\text{bit}$  となる。ストローチェンバーの総チャンネル数は4000以上であるため、全チャンネルを送ると  $6000\text{bit} \times 4000 = 24\text{Mb/event}$  となる。また、カロリメータからのイベントトリガーレートは約1 kHz であると予測されているので、1秒あたりのデータ量は  $24\text{Mb} \times 1\text{kHz} = 24\text{Gbps}$  となる。これだけでも十分にデータ量は多いが、サンプリングスピードが速くなったり、データにヘッダーを付けたりとさらにデータ量が増えてしまう。したがってデータの転送量を減らすような工夫が必要となる。イベントトリガー時にヒットしていないストローチェンバーもあるため、ヒットしたチャンネルのデータのみ送るというゼロサプレッスや、波形が存在していない部分の記録するサンプリングポイントを減らす Rebinning などがデータ量

を小さくするための工夫として挙げられる。

## 2.3 ROESTI

ROESTI とは、前節までで取り上げた実験要求を満たす Waveform Digitizer の機能を実装した ストローチェンバー用読み出し回路である (図 2.1)。以下、ROESTI の主な機能を説明する。



図 2.1 ROESTI の全体フロー図。ストローチェンバーからの信号は左から入力されて、デジタル化され PC へ送られる。

### 2.3.1 フロントエンド

I-V 変換や電荷増幅を行う。予測されているストローチェンバーからの電荷の最低値 (-16 fC) を読み取れるように電荷増幅が必要となる [1]。また、実装するフロントエンドのアンプを交換することで、ストローチェンバーだけでなくカロリメータの読み出しにも使用することも可能である。

### 2.3.2 Waveform Digitizer

波形のサンプリングおよびデジタル化を行う。時間分解能の要求は 2 nsec であるので、サンプリングスピードは 500MSPS 程度あれば十分である。実際は波形のフィッティングを行うことで、500 MSPS より低いサンプリングスピードでも時間分解能 2 nsec を達成することが出来る。しかしながら、ストローチェンバーの信号の立ち上がり時間が 10 nsec 程度であるため、時間分解能を満たしたとしてもサンプリングスピードが低すぎる場合は、立ち上がり部分のサンプリング数が少なくなってしまう。500 MSPS あれば 10 nsec の立ち上がり時間でも 5 サンプル出来て、時間分解能も十分達成できるであろうと見込まれている。

### 2.3.3 FPGA

Waveform Digitizer でデジタル化したデータのサブレスを行う。2.2 節で説明した通り、ヒットしていないチャンネルのデータを送ってしまうと 4000 チャンネル以上のデータ転送を行わなければならない。したがって、転送データ量を減らすためにヒットしているチャンネルのみ波形を記録し、データ転送を行うようにする。ヒットチャンネルの選別は後述する ASD のディスクリミネート出力で判断する。また、今はまだ未実装だがデジタルデータ自体も差分法で圧縮したり、Rebinning を行うことでデータ量をできるだけ小さくできるようにすることも検討している。

また、FPGA 内の処理として転送データの packets フォーマット化も行っている。Packets フォーマットは付録 B に記載してある。

### 2.3.4 Ethernet

FPGA 内に SiTCP と呼ばれるモジュールを実装することでイーサネット通信を実現している。サンプリングデータの転送には TDC 通信を利用する。TDC 通信とは、やりとりを行う送信側と受信側のコネクションを確立してから転送を行う通信である。TDC 通信の転送スピードは最大で 1Gbps となっている。読み出し回路の初期化・設定は UDP 通信を利用する。UDP 通信とは、やりとりを行う送信側と受信側のコネクションを確立せずにデータパケットを送る通信である。したがって UDP 通信を利用する際はデータパケットが正しく届いたか確認を行う必要がある。

## 2.4 ROESTI プロトタイプ

今回、先述のような ROESTI の機能の評価を行うためプロトタイプを製作した (図 2.2)。以下ブロックごとに実際に実装した物や機能を説明する。



図 2.2 ROESTI のプロトタイプの写真。大きさは 8 cm × 18.5 cm であり要求は満たしている。

### 2.4.1 フロントエンド

ASD (Amplified Shaped Discriminator) と呼ばれるアンプを実装する [13]。実装される ASD は Belle II 実験で製作された ASIC である。形状は異なるがガスチェンバーを読み出すために製作されている。出力にはアナログ出力とデジタル出力がある。アナログ出力では ASD へ入力された電荷が I-V 変換・波形形成の過程を経て信号が出力される。増幅率は -500 fC 程度までは 1 mV/fC であるが、-700 fC を超えたあたりから緩やかに電圧は増加していく。ダイナミックレンジは -2.0 ~ 0 pC である。デジタル出力は ASD への入力のディスクリミネート出力である。しきい値を設定することでコントロールすることが出来る。今回のファームウェアでは、デジタル出力をヒットチャンネルの選別に利用している。1 チップあたり 8 チャンネルであり基板上には 2 つ実装されて

いる。

### 2.4.2 Waveform Digitizer

Waveform Digitizer はアナログメモリで波形のサンプリングを行い、Flash ADC でデータのデジタル化が行われる。

アナログメモリには PSI で開発された DRS4 と呼ばれる ASIC を用いる [6]。DRS4 は各チャンネルにスイッチドキャパシタが 1024 個並列に接続されていて、キャパシタのスイッチを順番に切り替えることで入力信号のサンプリングを行う。図 2.3 に原理イメージを示す。サンプリングスピードは 700 MSPS ~ 5 GSPS と可変であり、目安の 500 MSPS を十分上回るスピードが実現されている。今回はテストとして約 1 GSPS で波形を取得している。

デジタル化に使用される ADC の分解能は 12bit である。比較的分解能の高い ADC を選んでいるのは、テストの目的に加えて今後 COMET 実験以外の実験でも使用できるように汎用性も考慮に入れたうえでの選択となっている。

DRS4、ADC とともに 1 チップ当たり 8 チャンネルであり、ボード上にそれぞれ 2 つずつ実装されている。

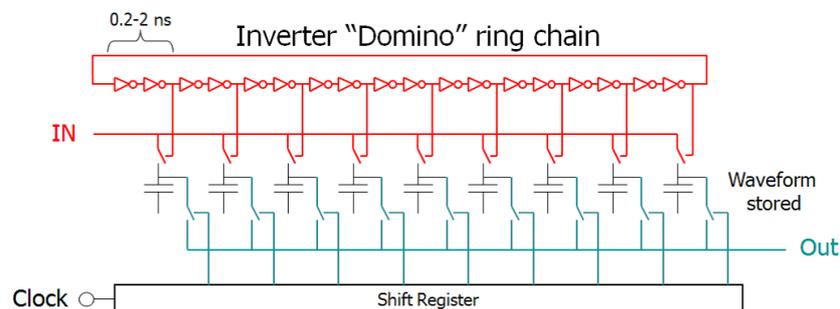


図 2.3 DRS4 の動作原理のイメージ図。図は参考文献 [7]、3 ページ目より引用した。赤線がかかれた部分が Domino Ring + 入力部分、青緑線がかかれた部分が出力部分を表している。入力部分のスイッチは Domino Ring を流れている、サンプリングクロックに同期した信号に従って順番に切り替わっていく。スイッチが入っているキャパシタに入力 (IN) に入っている電荷が保存されていく。出力部分のスイッチは図中の Clock の周期に同期して順番に切り替わっていき、スイッチが入っているキャパシタの電荷が順番に出力される。図中の Clock の周期は外部 ADC のサンプリングクロックと同周期である。

### 2.4.3 FPGA

Xilinx 社製の Spartan6(型番 : XC6SLX150T-2FG(G)676) を用いている。光ケーブルによる通信が使用出来るものを選んでいる。

### 2.4.4 Ethernet

SFP コネクタを実装して光ケーブルで通信する。光ケーブルを用いる理由は、読み出し回路が磁場中に置かれるためである。また、チェーン接続を行うため基板1枚につき2つコネクタを実装している。

#### チェーン接続

今回製作した読み出し回路では、光ケーブルコネクタを2つ実装することによりチェーン接続を行う。ストローチェンバー1層あたりのストローの本数は208本であり、読み出し回路のチャンネル数は16chであるため、1層あたりの読み出し回路の枚数は13枚となる。この13枚の読み出し回路をチェーン状につなげてデータの読み出しを行うことで、真空フィードスルーの数を減らす。チェーン接続は両側からアクセスが出来るようになっていて、光ケーブルコネクタが2箇所壊れない限り、どちら側からでも読み出せるようになっている。

FPGA のファームウェアにはイベントビルダーが実装されており、読み出し回路のつながっている順にストローチェンバーからのイベントデータ packets を PC 側へ送ることが出来る (図 2.5)。また、同時にではないが同じチェーンライン上で各読み出し回路の初期化・設定も行うことが出来る。ROESTI 開発現在のファームウェアでは初期化・設定のみ両側から行うことが出来る。

なお、チェーン接続のイベントビルダーは現在開発中である (図 2.4)。

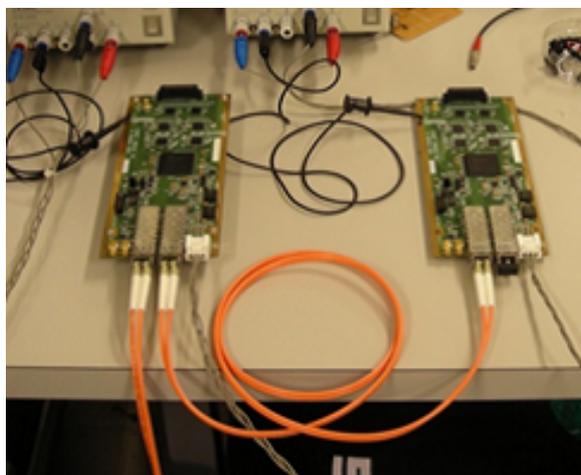


図 2.4 チェーン接続の実際の様子。写真のようにプロトタイプを2枚接続してテストを行っている。

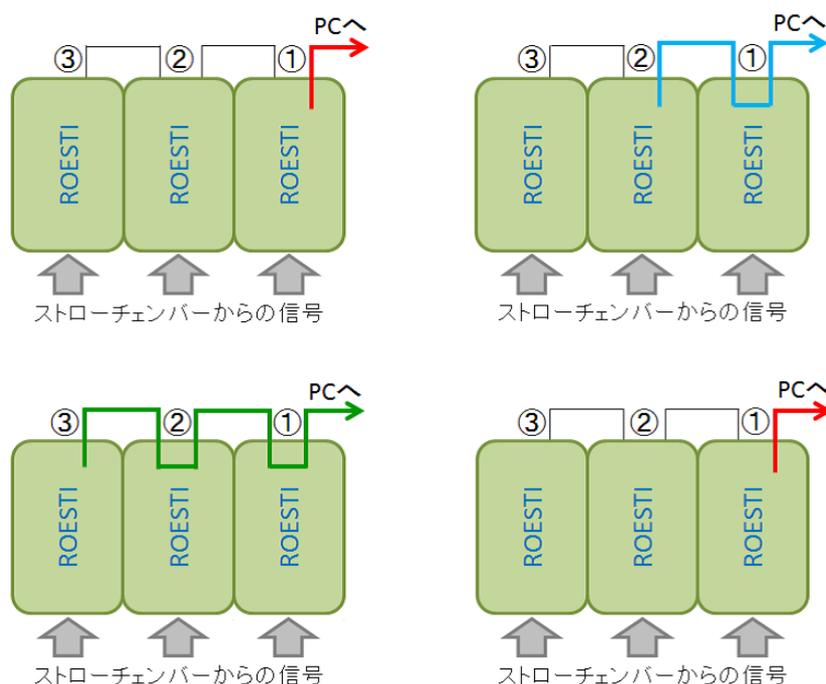


図 2.5 チェーン接続によるイベントデータ転送のイメージ図。データ転送できる状態になると、まず PC 側に一番近い読み出し回路のイベントデータから送り出される (左上図、①)。①の読み出し回路のイベントデータを送り終えた後は、②の読み出し回路のイベントデータを送信する (右上図)。②の読み出し回路のイベントデータを送り終えた後は、③の読み出し回路のイベントデータを送信する (左下図)。この図では③の読み出し回路がチェーン接続の一番端であるので、イベントデータの送信が終わると次のイベントデータを①の読み出し回路から再び送信し始める (右下図)。以上の繰り返しでイベントデータの転送を行う。必ずボードの接続順にデータが転送されるため、PC 側でのイベントビルドが非常に容易に行える。

### イベントレート

COMET 実験の Conceptual Design Report(以下 CDR, 参考文献 [1]) によると、電磁カロリメータからのイベントトリガーレートは約 1 kHz と予想されている。また、1 イベントあたりヒットするストローチェンバーは 540 本と予想されている。ストローチェンバーの層は 20 層あるため、1 層あたりの平均ヒットストローチェンバー数は約 30 本となる。ROESTI のサンプリングスピードを 1GSPS とし、ADC の分解能は 12 ビットであるが転送の都合上 4 ビット追加して波形 1 サンプルポイントあたり 16 ビットとすると、ストロー 1 本あたりのデータ量は約 16700 ビットであるため、1 イベントあたりの 1 層分のデータ量は  $30 \times 16700 = 501000 \text{ bit/trigger}$  となる。イベントレートは約 1 kHz であり、陽子ビームの時間構造 (図 B.7) よりパルスビームが来る時間は 0.7 sec、よって 1 サイクルで来るデータ量平均は  $501000 \times 1000 \times 0.7 = 360 \text{ Mb/1.4sec}$  となる (表 2.1)。1 サイクルは 1.4 sec であるため 1 Gbps のデータ転送性能があれば、1 層ごとにストローチェンバーのデータを 1 本のライン (光ケーブル等) で転送することが理論上可能である。

-	数値	単位
イベントレート	1	kHz
ヒット数	30	ch/(layer $\odot$ trigger)
データサイズ	501000	(bit $\odot$ layer $\odot$ trigger)
データ転送量	360	Mb/1.4s

表 2.1 ストローチェンバー 1 層当たりの想定データ量の一覧。

### デットタイム

今回製作した基板のデットタイムは 45  $\mu$ sec である。この値は DRS4 から波形を読み出すのにかかる時間で決まっている。したがって 1kHz のポアソン到着を考えると式 (A.3) より 95% の取得率となる。また 1 kHz のレートに対し 1 Gbps の処理速度であれば、参考文献 [10] より 6 イベント分のバッファがあれば十分である。この容量分のバッファは基板上 (FPGA 内) に実装可能である。実際の計算は付録 A に記載している。

## 第3章

# プロトタイプの評価

今回製作した ROESTI のプロトタイプ基板で以下の評価を行った。

1. 波形評価
2. 読み出し回路の基本性能評価
3. 時間分解能測定
4. 通信システムテスト

まずは DRS4 のパターンコレクションについて解説し、その後 1 つずつ評価方法とその結果を述べる。

### 3.1 パターンコレクション

DRS4 はチャンネル毎にスイッチドキャパシタが並列に接続されているが、キャパシタごとの特性 (容量など) が異なるために信号を入れていなくても図 3.1 に示すようにチャンネルごとに特有の波形パターンを持っている。パターンコレクションとは、特有の波形パターンを先に取得しておき、実際のデータから引くことで正しい入力波形を取得するためのコレクションである。以下に述べる評価はすべてパターンコレクションを行ったうえでやっている。実際にパターンコレクションを行う前後の波形の比較を図 3.2 に示す。

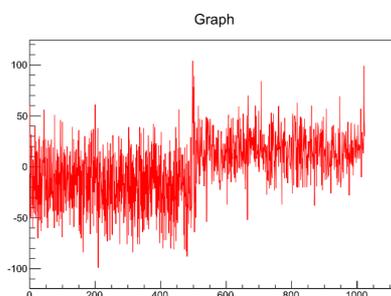


図 3.1 波形パターンの一例。横軸はキャパシタ番号、縦軸は ADC カウントの値。

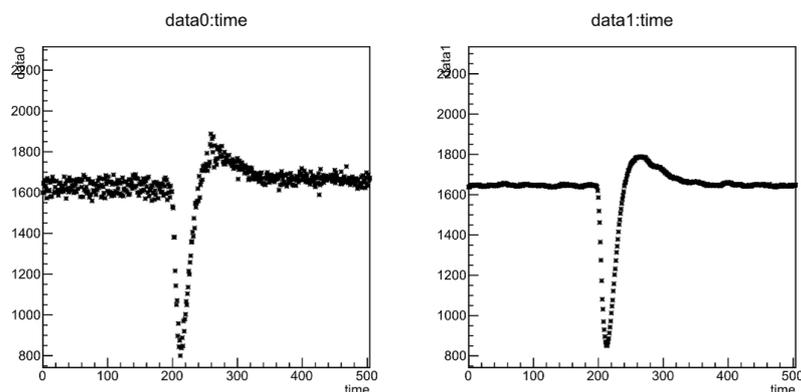


図 3.2 パターンコレクション前後の比較。左の波形がコレクション前、右の波形がコレクション後のものである。グラフの横軸は時間 (nsec)、縦軸の縦軸は ADC カウントの値である。波形がきれいになっていることがわかる。

## 3.2 波形評価

### 3.2.1 目的

波形が正しく読み出せているか確認する。正しくという意味は、DRS4 を介して取得した波形の波高、時間のキャリブレーションが正しく行われているということである。波高、時間のキャリブレーションに関しては別途行うこととし、この節ではまず波形の読み出しの成否を判断するために、読み出し回路上状態で、DRS4 に入力されている波形と DRS4 を介して読み出した波形が相似形になっているかどうかを調べる。DRS4 に入力されている波形はオシロスコープで取得する。相似形でよい理由は、完全に一致していなくても相似形であればパイルアップの処理は問題なく行える上、時間、電圧のキャリブレーションを別途行うことにより、補正できるからである。

### 3.2.2 評価方法

読み出し回路に実装されている ASD の出力を DRS4 による取得に加えてオシロスコープでも取得し、それぞれの波形の平均を比較する。DRS4 で取得した波形がオシロスコープで取得した波形と誤差の範囲内で重なり合えば相似形であると判断する。相似形であればよいので、オシロスコープの波形には係数をかけて波高の調整 (以下、倍率調整と呼ぶ) を行って比較も行う。倍率調整はピーク位置が最も近くなるように調整している。

### 3.2.3 結果

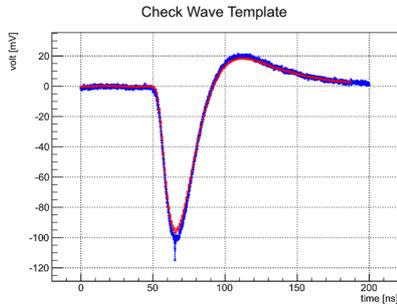


図 3.3 オシロスコープと DRS4 で取得した波形の比較。ピーク位置は約-100 mV で倍率調整なし。

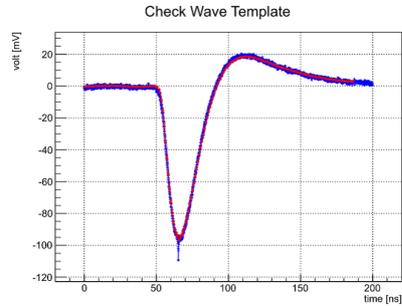


図 3.4 オシロスコープと DRS4 で取得した波形の比較。ピーク位置は約-100 mV で倍率調整あり。

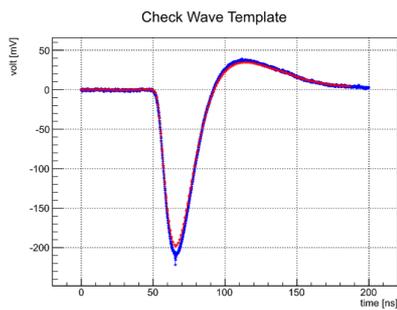


図 3.5 オシロスコープと DRS4 で取得した波形の比較。ピーク位置は約-200 mV で倍率調整なし。

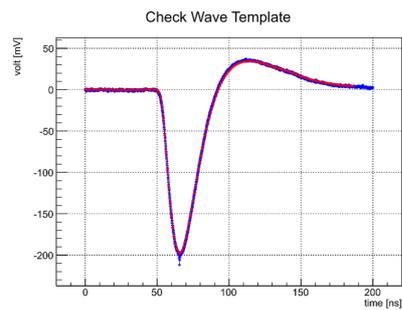


図 3.6 オシロスコープと DRS4 で取得した波形の比較。ピーク位置は約-200 mV で倍率調整あり。

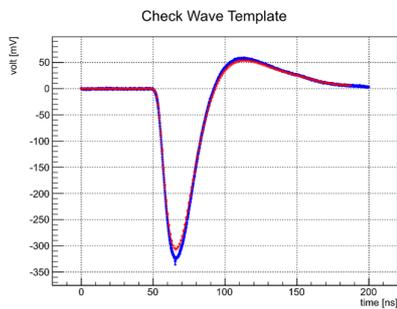


図 3.7 オシロスコープと DRS4 で取得した波形の比較。ピーク位置は約-300 mV で倍率調整なし。

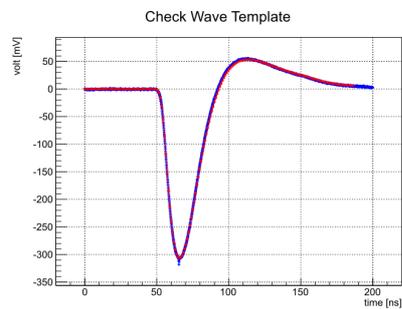


図 3.8 オシロスコープと DRS4 で取得した波形の比較。ピーク位置は約-300 mV で倍率調整あり。

図 3.3 ~ 図 3.8 に 3 つの異なる波高 (-100 mV、-200 mV、-300 mV) の波形を入力した際の比較結果を示す。青線がオシロスコープ、赤線が DRS4 を介して読み出した波形である。定性的な判断では、立ち上がり部分や下方向ピーク位置部分 (50~80nsec) で倍率調整後は誤差の範囲で重なり合っている。しかしながら、波形の後半からオーバーシュート部分 (80 ~ 120 nsec) ではずれが大きいように見受けられる。このずれは 3 つの波形比較のいずれにも見られる。

次に (オシロスコープの波高値 - DRS4 の波高値) のヒストグラムと (オシロスコープの波高値 - DRS4 の波高値) vs オシロスコープの波高のプロット図、時間 vs (オシロスコープの波高値 - DRS4 の波高値) のプロット図を図 3.9 ~ 図 3.11 に示す。実際にはオシロスコープと DRS4 のサンプリングタイミングが一致しているわけではないので、オシロスコープの平均波形から 3 次スプライン補完により得られた関数から求めた波高値と比較を行っている。比較対象は倍率調整後のみとする。

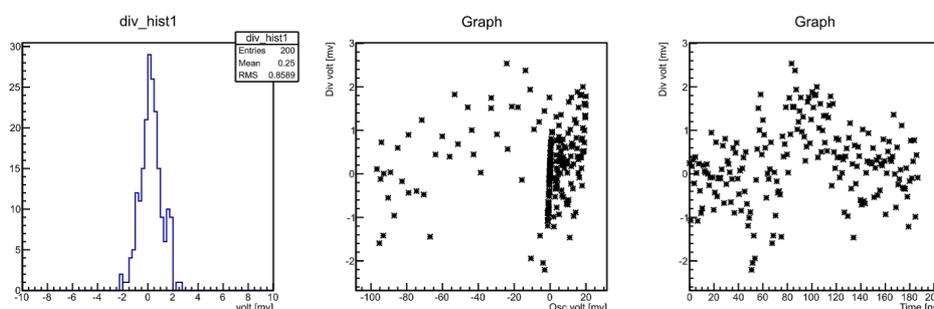


図 3.9 左図が (オシロスコープの波高値 - DRS4 の波高値) のヒストグラム、中央図が (オシロスコープの波高値 - DRS4 の波高値) vs オシロスコープの波高のプロット図、右図が時間 vs (オシロスコープの波高値 - DRS4 の波高値)。図 3.4 で示した波形で差分をとっている。

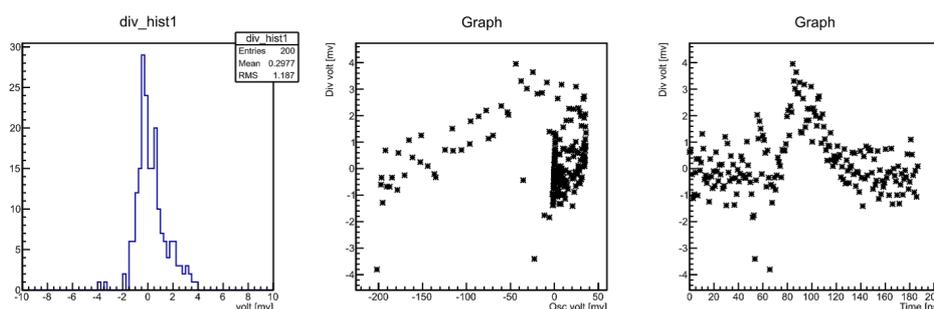


図 3.10 左図が (オシロスコープの波高値 - DRS4 の波高値) のヒストグラム、中央図が (オシロスコープの波高値 - DRS4 の波高値) vs オシロスコープの波高のプロット図、右図が時間 vs (オシロスコープの波高値 - DRS4 の波高値)。図 3.6 で示した波形で差分をとっている。

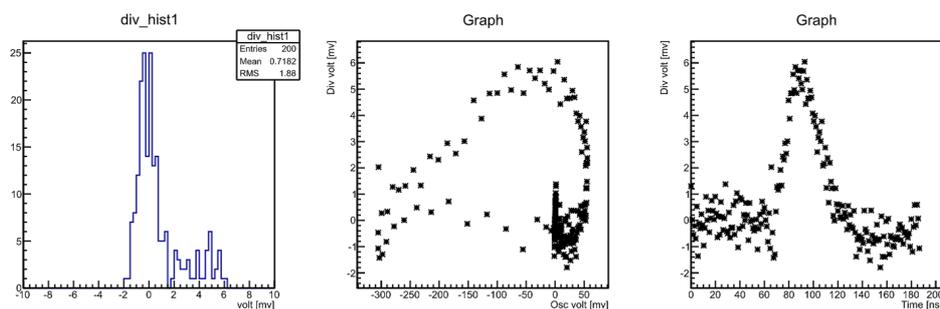


図 3.11 左図が (オシロスコープの波高値 - DRS4 の波高値) のヒストグラム、中央図が (オシロスコープの波高値 - DRS4 の波高値) vs オシロスコープの波高のプロット図、右図が時間 vs (オシロスコープの波高値 - DRS4 の波高値)。図 3.8 で示した波形で差分をとっている。

波高が大きくなるにつれ、差分のばらつきを表すヒストグラムの RMS が大きくなっている。よって、波高が高い方が DRS4 とオシロスコープのずれが大きくなることがわかる。この傾向はプロット図からも読み取ることが出来る。ずれが大きいところは 75 ~ 115 nsec と定性的確認と一致する。

今回の評価では波形が相似形であるということは断言できない結果となった。しかしながら、図 3.3 ~ 図 3.8 に示したグラフからは DRS4 を介して取得した波形には特に歪みなどなく、オシロスコープの波形とも似たような形状であることは見て判断できる。波形の大小による歪みも特に見受けられない。またずれの大きいところはオーバーシュート部分であり、立ち上がり部分は比較的ずれが小さく、正しくサンプリングが出来ているとみなせる。得られたずれが実際のパイルアップ評価にどう影響するかはまだ分からないが、本論文では取得している波形は正しいものとして今後の評価を進めていく。

### 3.2.4 今後の改善点

形状評価の改善点について述べる。図 3.9 ~ 図 3.11 に示したプロット図には誤差を表示していないが、実際にはオシロスコープと DRS4 の差分をプロットしているため両方の誤差が含まれている。また、オシロスコープの波高は 3 次スプライン補完からの関数によって算出している。そのため、得られた関数の誤差も影響してくるはずである。それらすべてを含んだ結果であるため、今後は誤差をうまく反映させることでより深く評価が出来る。

オシロスコープ処理にも改善点が考えられる。本論文での図 3.3 ~ 図 3.8 で示したオシロスコープから取得した波形の平均値 (図中では青線で示されている波形) の求め方は、ピーク位置を基準として波形の平均値を計算するという手法である。このオシロスコープによる波形取得であるが、オシロスコープの性能が帯域 2.5 GHz、サンプリングスピードが 10 GSPS であり、高周波なノイズも同時に取得してしまっている。そのために、ピーク位置を基準にするとノイズの影響で正しく平均値波形が得られていない可能性があり、図のピーク部分でスパイクが発生しているように見

えるのもその影響であると考えられる。よって今後詳しく波形比較を行う際は、オシロスコープのデータ処理についても考えておく必要がある。

## 3.3 基本性能評価

### 3.3.1 目的

ノイズや線型性など基本的な性能を評価する。評価する項目は以下の通りである。

1. ノイズ
2. 線型性
3. クロストーク
4. 消費電力

1~3 までの評価では実際に検出器からの信号を読み取る際の性能評価を行うための材料となる。4 の評価は本番の実験の際に、読み出し回路を置く場所にかかる制限を満たしているか確かめるため調べるが、具体的な制限は決まっていないため今後の目安として調べておく。なお、2ch と 9ch は他の目的のため ASD と DRS4 の間の接続を切り、直接ファンクションジェネレーターの出力をインプットできるようにしてあるため、評価の対象からは外すことにする。

### 3.3.2 評価方法

個々の項目の評価方法を以下に述べる。

#### ノイズ

ベースラインの波形をとり、ベース基準値からの揺れを確認する。正確には 1 イベント毎に全キャパシタの平均をとり平均値を差し引き、波形のベース基準値 (平均値点) を 0 に合わせてから、ベース基準値からの揺らぎをノイズとして評価している (図 3.13)。ベースラインの波形の一例を図 3.12 に示す。ベースライン取得時は電源と光ケーブル以外のコネクタには何も接続せずに評価している。

#### 線型性

1ch 毎に決まった量の電荷を与えて DRS4 で取得した波形の高さ (電圧値) を調べる。与える電荷を変化させて電荷-電圧のプロットをとり、線型性を確認する。電荷の与え方はキャパシターにパルサーで作った矩形波を与えることで行った。以下、電荷の与え方はすべて同じである。

#### クロストーク

1ch ずつ電荷を与え、電荷を与えた ch で取得した波形のピーク点に対して他のチャンネルの波形の最小値の比をとり算出する。

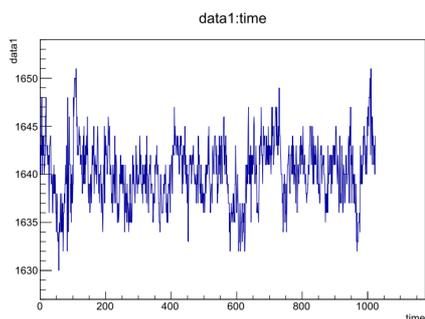


図 3.12 ノイズのサンプル図。横軸はキャパシタ番号 (時間と同等)、縦軸は ADC カウントの値。

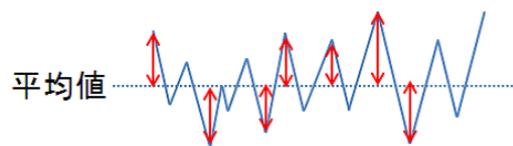


図 3.13 ノイズ定義のイメージ図。平均値からのずれ (図の赤い矢印の部分) をヒストグラムに詰めて、ガウシアンでフィットする。フィット結果の RMS をノイズと定義する。

### 消費電力

電源装置で出力している電流値と設定電圧値から全体消費電力を求める。

### 3.3.3 結果

各項目の評価結果を以下に述べる。

#### ノイズ

各チャンネル毎のノイズ評価のヒストグラムを図 3.14、図 3.15 に、ヒストグラムから得られた RMS 値を表 3.1 に示す。

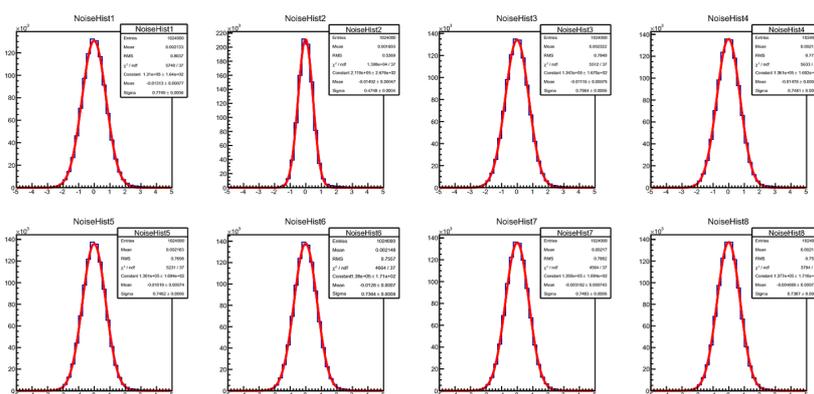


図 3.14 ch1~ch8 ののノイズヒストグラム。横軸は電圧値 (mV)。左上から右に向かって順に ch1 から ch8。

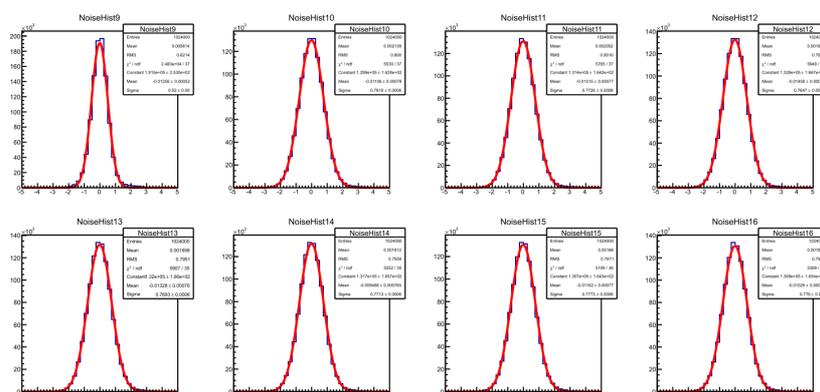


図 3.15 ch9~ch16 のノイズヒストグラム。横軸は電圧値 (mV)。左上から右に向かって順に ch9 から ch16。

ch1	0.77 ± 0.00 mV	ch10	0.78 ± 0.00 mV
ch3	0.76 ± 0.00 mV	ch11	0.77 ± 0.00 mV
ch4	0.75 ± 0.00 mV	ch12	0.76 ± 0.00 mV
ch5	0.75 ± 0.00 mV	ch13	0.77 ± 0.00 mV
ch6	0.74 ± 0.00 mV	ch14	0.77 ± 0.00 mV
ch7	0.75 ± 0.00 mV	ch15	0.78 ± 0.00 mV
ch8	0.75 ± 0.00 mV	ch16	0.78 ± 0.00 mV

表 3.1 ノイズヒストグラムの RMS 値。

表 3.1 よりどのチャンネルもノイズは 0.74 ~ 0.78 mV となっている。チップ 1(ch1~ch8) よりもチップ 2(ch10~ch16) のノイズの方が若干大きいですが、それでも 0.04 mV 程度であるので各チャンネル毎に差はほとんど無いといえる。仕様書によると ASD と DRS4 のノイズはともに 0.35 mV(RMS 値) であるため、そのまま 2 乗和をとると 0.49 mV となる。しかし測定値はどのチャンネルも ~ 0.78 mV であるため、再び 2 乗和を考えると 0.61 mV ほどノイズが大きい。よって、ASD、DRS4 以外のノイズ (例えば電源やレギュレータ) も含まれていると推測できる。仕様書より ASD の入力部に 20 pF の検出器容量が接続されている場合、ノイズが 0.64 mV となり DRS4 のノイズとの 2 乗和をとると、0.73 mV となる。そのため、読み出し回路全体の複合ノイズとして、コネクタに検出器が接続された時に約 20 pF 分の検出器容量がのった場合と同等のノイズが常にあると想定される。

### 線型性

各チャンネルの評価結果を図 3.16、図 3.17、表 3.2、表 3.3 に示す。

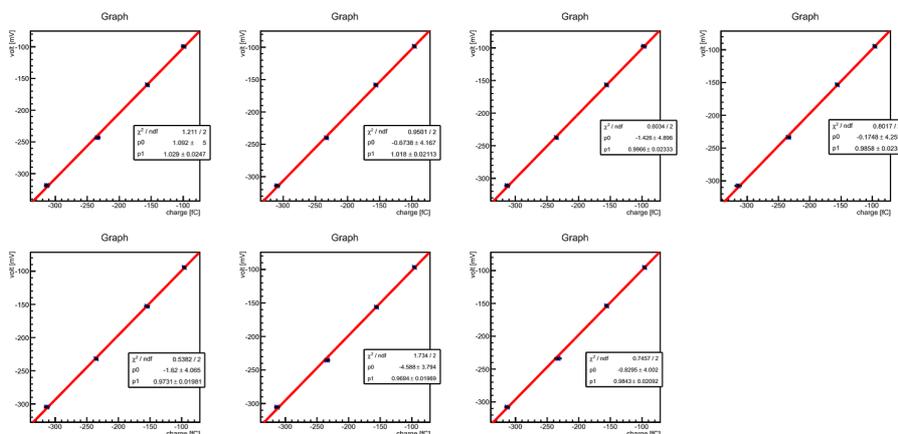


図 3.16 ch1~ch8 の線形性。横軸は与えた電荷量、縦軸は電圧値。左上から右に向かって順に ch1 から ch8(但し ch2 は除く)。

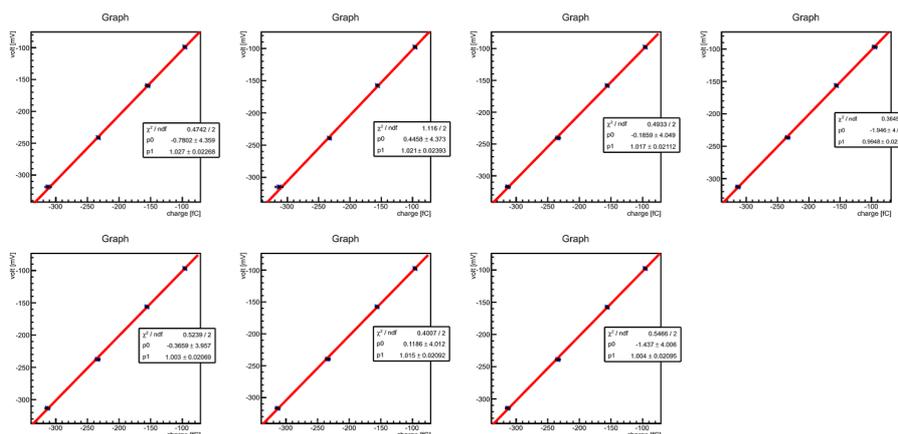


図 3.17 ch10~ch16 の線形性。横軸は与えた電荷量、縦軸は電圧値。左上から右に向かって順に ch10 から ch16。

ch1	$1.03 \pm 0.02 \text{ mV/fC} \times \text{電荷 (fC)} + 1.1 \pm 5.0 \text{ mV}$
ch3	$1.02 \pm 0.02 \text{ mV/fC} \times \text{電荷 (fC)} - 0.7 \pm 4.2 \text{ mV}$
ch4	$1.01 \pm 0.02 \text{ mV/fC} \times \text{電荷 (fC)} - 1.4 \pm 4.9 \text{ mV}$
ch5	$0.99 \pm 0.02 \text{ mV/fC} \times \text{電荷 (fC)} - 0.2 \pm 4.3 \text{ mV}$
ch6	$0.97 \pm 0.02 \text{ mV/fC} \times \text{電荷 (fC)} - 1.6 \pm 4.1 \text{ mV}$
ch7	$0.97 \pm 0.02 \text{ mV/fC} \times \text{電荷 (fC)} - 4.6 \pm 3.8 \text{ mV}$
ch8	$0.98 \pm 0.02 \text{ mV/fC} \times \text{電荷 (fC)} - 0.8 \pm 4.0 \text{ mV}$

表 3.2 図 3.16 で示したグラフフィットの結果。

ch10	$1.03 \pm 0.02 \text{ mV/fC} \times \text{電荷 (fC)} - 0.8 \pm 4.4 \text{ mV}$
ch11	$1.02 \pm 0.02 \text{ mV/fC} \times \text{電荷 (fC)} + 0.4 \pm 4.4 \text{ mV}$
ch12	$1.02 \pm 0.02 \text{ mV/fC} \times \text{電荷 (fC)} - 0.2 \pm 4.0 \text{ mV}$
ch13	$0.99 \pm 0.02 \text{ mV/fC} \times \text{電荷 (fC)} - 1.9 \pm 4.6 \text{ mV}$
ch14	$1.00 \pm 0.02 \text{ mV/fC} \times \text{電荷 (fC)} - 0.4 \pm 4.0 \text{ mV}$
ch15	$1.02 \pm 0.02 \text{ mV/fC} \times \text{電荷 (fC)} + 0.1 \pm 4.0 \text{ mV}$
ch16	$1.00 \pm 0.02 \text{ mV/fC} \times \text{電荷 (fC)} - 1.4 \pm 4.0 \text{ mV}$

表 3.3 図 3.17 で示したグラフフィットの結果。

仕様書より ASD のゲインは  $1 \text{ V/pC}$ 、DRS4 のゲインは  $0.985 \pm 0.03$ 、ADC のゲインは  $0.99$  となっている。よって読み出し回路全体の変換係数は 3 つのゲインをかけて  $0.98 \pm 0.03 \text{ mV/fC}$  となる。結果を見ると全体的に変換係数は高い数値になっている。誤差の範囲で一致はしていないが、ASD の資料に誤差に関しての記述がなく、理論値計算は ASD のゲインの誤差を考慮していないため正確には判断出来ない。しかしながら実験要求からはゲインが理論値に完全に一致している必要はなく、電荷－電圧値の変換が出来ればよいので十分だろう。変換式の切片の誤差が大きいのは、小さい電荷 ( $-100 \text{ fC}$  以下) のプロット点を取ることが出来なかったためと考えられる。 $-100 \text{ fC}$  以下のプロット点をもっと増やせばより正確な値が求められると考えられる。

次に、フィッティングより得られた電荷－電圧変換式に測定時の電荷量を代入して計算した電圧値と実測電圧値の比較を行う。図 3.18、図 3.19 に入力電荷値 vs(計算電圧値－実測電圧値) のプロットを示す。入力電荷値はマイナス表記のままでプロットしているため、グラフは左に向かって電荷量が増えていることになる。縦軸の差分電圧値は絶対値で表している。

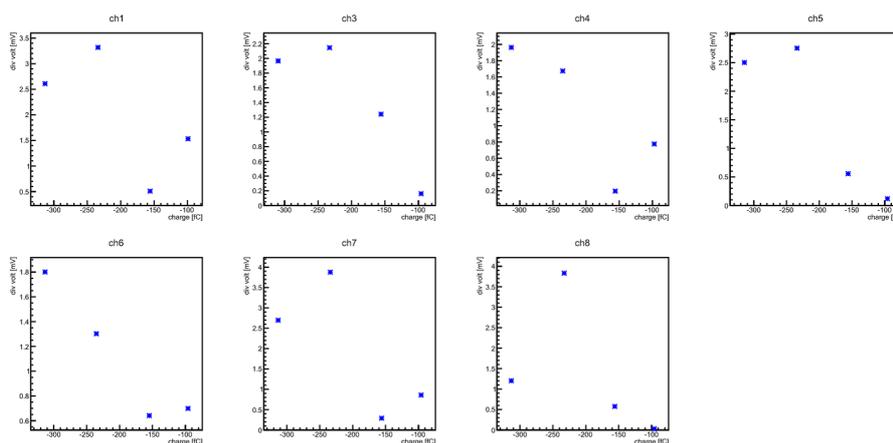


図 3.18 ch1～ch8 のフィット関数からのばらつき。横軸は与えた電荷量、縦軸は (計算電圧値－実測電圧値)。左上から右に向かって順に ch1 から ch8(但し ch2 は除く)。

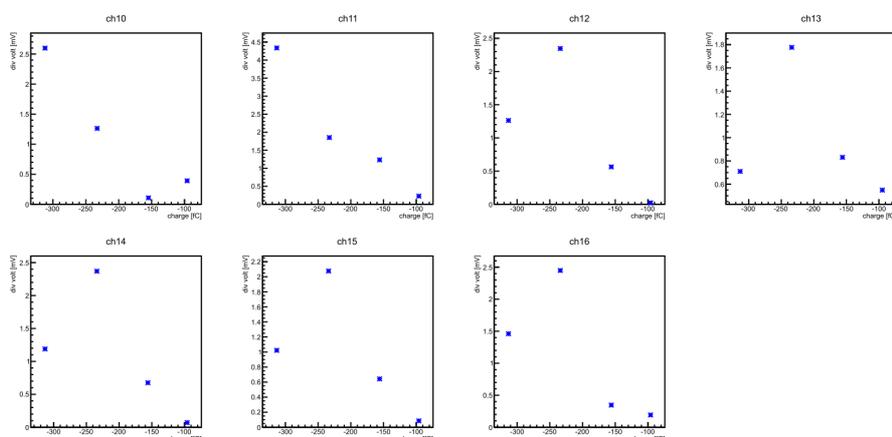


図 3.19 ch10~ch16 のフィット関数からのばらつき。横軸は与えた電荷量、縦軸は (計算電圧値-実測電圧値)。左上から右に向かって順に ch10 から ch16。

先述の通りデータ点が少ないことによるフィッティングエラー増大が影響してしまうため、グラフに誤差を付けていない。傾向としては、チャンネルごとにばらつきがあるものの電荷量が大きくなるにつれて差分電圧値も大きくなっていることがわかる。電荷量-100 fC のプロット点が差分電圧値が一番小さいというわけではないが、-300 fC のプロット点に対しては必ず小さい値をとっている。参考文献 [6] によると DRS4 は入力電圧値が大きくなるほど線型性からのずれが大きくなるので、傾向としては正しいといえる。差分電圧値の評価もプロット点の数を増やせば、より正確に求められると考えられる。

### クロストーク

クロストークはデシベル単位で  $\text{dB} = 10 \times \log(\text{電荷を与えていないチャンネルの最小値電圧} / \text{電荷を与えたチャンネルのピーク値電圧})$  のように求める。結果を表 3.4 から表 3.17 に示す。

ch1	-	ch10	-21.1 ± 0.0
ch3	-20.9 ± 0.0	ch11	-21.2 ± 0.0
ch4	-21.2 ± 0.0	ch12	-21.1 ± 0.0
ch5	-20.9 ± 0.0	ch13	-21.2 ± 0.0
ch6	-21.2 ± 0.0	ch14	-21.2 ± 0.0
ch7	-21.2 ± 0.0	ch15	-21.1 ± 0.0
ch8	-21.3 ± 0.0	ch16	-20.9 ± 0.0

表 3.4 ch1 に電荷を与えたときのクロストーク。単位はデシベル。ch1 のピーク値は-316.8 ± 1.8 mV。

ch1	-20.7 ± 0.0	ch10	-21.1 ± 0.0
ch3	-	ch11	-21.1 ± 0.0
ch4	-20.8 ± 0.0	ch12	-21.1 ± 0.0
ch5	-20.7 ± 0.0	ch13	-21.2 ± 0.0
ch6	-21.0 ± 0.0	ch14	-21.2 ± 0.0
ch7	-21.0 ± 0.0	ch15	-21.0 ± 0.0
ch8	-21.0 ± 0.0	ch16	-21.0 ± 0.0

表 3.5 ch3 に電荷を与えたときのクロストーク。単位はデシベル。ch3 のピーク値は-312.5 ± 1.6 mV。

ch1	-20.5 ± 0.0	ch10	-20.1 ± 0.0
ch3	-20.1 ± 0.0	ch11	-21.0 ± 0.0
ch4	-	ch12	-21.0 ± 0.0
ch5	-20.6 ± 0.0	ch13	-21.0 ± 0.0
ch6	-20.8 ± 0.0	ch14	-21.0 ± 0.0
ch7	-20.8 ± 0.0	ch15	-20.8 ± 0.0
ch8	-20.5 ± 0.0	ch16	-20.7 ± 0.0

表 3.6 ch4に電荷を与えたときのクロストーク。単位はデシベル。ch4のピーク値は-308.1 ± 1.6 mV。

ch1	-20.5 ± 0.0	ch10	-20.9 ± 0.0
ch3	-20.0 ± 0.0	ch11	-21.0 ± 0.0
ch4	-20.3 ± 0.0	ch12	-20.9 ± 0.0
ch5	-	ch13	-20.9 ± 0.0
ch6	-20.3 ± 0.0	ch14	-21.0 ± 0.0
ch7	-20.6 ± 0.0	ch15	-20.7 ± 0.0
ch8	-20.4 ± 0.0	ch16	-20.7 ± 0.0

表 3.7 ch5に電荷を与えたときのクロストーク。単位はデシベル。ch5のピーク値は-300.8 ± 1.9 mV。

ch1	-20.5 ± 0.0	ch10	-20.6 ± 0.0
ch3	-20.2 ± 0.0	ch11	-20.8 ± 0.0
ch4	-20.4 ± 0.0	ch12	-20.8 ± 0.0
ch5	-20.2 ± 0.0	ch13	-20.8 ± 0.0
ch6	-	ch14	-20.8 ± 0.0
ch7	-20.3 ± 0.0	ch15	-20.6 ± 0.0
ch8	-20.4 ± 0.0	ch16	-20.6 ± 0.0

表 3.8 ch6に電荷を与えたときのクロストーク。単位はデシベル。ch6のピーク値は-297.9 ± 1.7 mV。

ch1	-20.3 ± 0.0	ch10	-20.8 ± 0.0
ch3	-20.1 ± 0.0	ch11	-20.9 ± 0.0
ch4	-20.5 ± 0.0	ch12	-20.8 ± 0.0
ch5	-20.2 ± 0.0	ch13	-20.8 ± 0.0
ch6	-20.3 ± 0.0	ch14	-20.9 ± 0.0
ch7	-	ch15	-20.7 ± 0.0
ch8	-20.1 ± 0.0	ch16	-20.6 ± 0.0

表 3.9 ch7に電荷を与えたときのクロストーク。単位はデシベル。ch7のピーク値は-302.0 ± 1.6 mV。

ch1	-20.5 ± 0.0	ch10	-20.7 ± 0.0
ch3	-20.2 ± 0.0	ch11	-21.0 ± 0.0
ch4	-20.6 ± 0.0	ch12	-20.8 ± 0.0
ch5	-20.2 ± 0.0	ch13	-20.9 ± 0.0
ch6	-20.5 ± 0.0	ch14	-20.9 ± 0.0
ch7	-20.0 ± 0.0	ch15	-20.7 ± 0.0
ch8	-	ch16	-20.7 ± 0.0

表 3.10 ch8に電荷を与えたときのクロストーク。単位はデシベル。ch8のピーク値は-300.9 ± 1.6 mV。

ch1	-20.4 ± 0.0	ch10	-
ch3	-21.0 ± 0.0	ch11	-21.2 ± 0.0
ch4	-21.3 ± 0.0	ch12	-20.9 ± 0.0
ch5	-21.0 ± 0.0	ch13	-21.1 ± 0.0
ch6	-21.3 ± 0.0	ch14	-21.1 ± 0.0
ch7	-21.3 ± 0.0	ch15	-20.8 ± 0.0
ch8	-20.5 ± 0.0	ch16	-20.9 ± 0.0

表 3.11 ch10に電荷を与えたときのクロストーク。単位はデシベル。ch10のピーク値は-318.5 ± 1.7 mV。

ch1	-20.8 ± 0.0	ch10	-20.3 ± 0.0
ch3	-20.2 ± 0.0	ch11	-
ch4	-20.7 ± 0.0	ch12	-20.5 ± 0.0
ch5	-20.3 ± 0.0	ch13	-20.7 ± 0.0
ch6	-20.7 ± 0.0	ch14	-20.7 ± 0.0
ch7	-20.7 ± 0.0	ch15	-20.6 ± 0.0
ch8	-20.7 ± 0.0	ch16	-20.5 ± 0.0

表 3.12 ch11 に電荷を与えたときのクロストーク。単位はデシベル。ch11 のピーク値は-308.6 ± 1.6 mV。

ch1	-21.4 ± 0.0	ch10	-20.7 ± 0.0
ch3	-21.0 ± 0.0	ch11	-20.8 ± 0.0
ch4	-21.3 ± 0.0	ch12	-
ch5	-21.2 ± 0.0	ch13	-21.0 ± 0.0
ch6	-21.3 ± 0.0	ch14	-21.0 ± 0.0
ch7	-21.3 ± 0.0	ch15	-21.0 ± 0.0
ch8	-21.2 ± 0.0	ch16	-20.8 ± 0.0

表 3.13 ch12 に電荷を与えたときのクロストーク。単位はデシベル。ch12 のピーク値は-317.2 ± 1.8 mV。

ch1	-21.0 ± 0.0	ch10	-20.8 ± 0.0
ch3	-21.1 ± 0.0	ch11	-20.9 ± 0.0
ch4	-21.3 ± 0.0	ch12	-20.7 ± 0.0
ch5	-21.2 ± 0.0	ch13	-
ch6	-21.3 ± 0.0	ch14	-20.4 ± 0.0
ch7	-21.2 ± 0.0	ch15	-21.1 ± 0.0
ch8	-20.8 ± 0.0	ch16	-20.8 ± 0.0

表 3.14 ch13 に電荷を与えたときのクロストーク。単位はデシベル。ch13 のピーク値は-312.9 ± 1.6 mV。

ch1	-21.0 ± 0.0	ch10	-20.8 ± 0.0
ch3	-21.0 ± 0.0	ch11	-20.9 ± 0.0
ch4	-21.2 ± 0.0	ch12	-20.8 ± 0.0
ch5	-21.2 ± 0.0	ch13	-20.8 ± 0.0
ch6	-21.2 ± 0.0	ch14	-
ch7	-21.2 ± 0.0	ch15	-20.5 ± 0.0
ch8	-21.2 ± 0.0	ch16	-20.5 ± 0.0

表 3.15 ch14 に電荷を与えたときのクロストーク。単位はデシベル。ch14 のピーク値は-313.4 ± 1.7 mV。

ch1	-21.1 ± 0.0	ch10	-20.9 ± 0.0
ch3	-21.0 ± 0.0	ch11	-21.0 ± 0.0
ch4	-21.3 ± 0.0	ch12	-21.1 ± 0.0
ch5	-21.2 ± 0.0	ch13	-21.1 ± 0.0
ch6	-21.2 ± 0.0	ch14	-20.5 ± 0.0
ch7	-21.2 ± 0.0	ch15	-
ch8	-21.2 ± 0.0	ch16	-20.3 ± 0.0

表 3.16 ch15 に電荷を与えたときのクロストーク。単位はデシベル。ch15 のピーク値は-316.7 ± 1.9 mV。

ch1	-20.8 ± 0.0	ch10	-20.9 ± 0.0
ch3	-21.0 ± 0.0	ch11	-20.9 ± 0.0
ch4	-21.2 ± 0.0	ch12	-20.8 ± 0.0
ch5	-21.1 ± 0.0	ch13	-21.0 ± 0.0
ch6	-21.2 ± 0.0	ch14	-20.5 ± 0.0
ch7	-21.2 ± 0.0	ch15	-20.5 ± 0.0
ch8	-21.1 ± 0.0	ch16	-

表 3.17 ch16 に電荷を与えたときのクロストーク。単位はデシベル。ch16 のピーク値は-314.9 ± 2.5 mV。

表 3.4～表 3.17 より、どのチャンネルに信号を入れた時でもほとんどのチャンネルが 20dB よりも良い結果となっている。電荷を与えているチャンネルに一番近いチャンネルのクロストーク値が一番低い値であるといった傾向は特に見られない。しかしながら、チップ間で比較してみると、電

荷を与えているチャンネルがあるチップのクロストーク値の方が、電荷を与えていないチップのクロストーク値よりも若干低い値となっている。この結果よりクロストークの影響が全くないとは言えないが、全体としてクロストークの影響は全チャンネルで大きな差異はなく-20 ~ -21dB の範囲に収まっていると結論できる。

### 消費電力

本読み出し回路では +5.0 V と +2.5 V の電源を用いている。+5.0V 電源に流れた電流値は  $0.79 \pm 0.01$  A、+2.5 V 電源に流れた電流値は  $0.43 \pm 0.01$  A である。これらの値より消費電力は  $5.0 \times 0.79 \pm 0.01 + 2.5 \times 0.43 \pm 0.01 = 5.0 \pm 0.1$  W となる。電力値は DRS4 や ADC リセット時に少し低くなるが、それ以外の場合には特に時間依存性は見られなかった。

## 3.4 時間分解能測定

### 3.4.1 目的

実験要求である時間分解能 2nsec を満たしているかを確認する。

### 3.4.2 評価方法

ある1つのチャンネルを基準とした時に、基準チャンネルと他のチャンネルで測定した波形の時間差を測定する。今回は測定した時間差のヒストグラムを作り、ヒストグラムの RMS を時間分解能として定義する。時間の決定方法は取得した波形のベースラインから-100 mV に到達するタイミングを波形の到達時間とした。異なるチップ間同士の校正は DRS4 の1つのチャンネルに基準となるクロックを与えて、取得したクロック波形の位相差で行う。詳細を図 3.20 と図 3.21 に示す。

今回製作した読み出し回路では基準となるクロックを与えるポートを基板上に作っていなかった。このため、チップごとに1チャンネル(読み出し回路上の ch2 と ch9)をこの用途用に、入力から切り離して入力ポートを製作し直接 DRS4 へクロックを入れている。なお、時間分解能測定では基本性能評価とは DRS4 の異なる読み出しモードを使用している。理由については後述する。

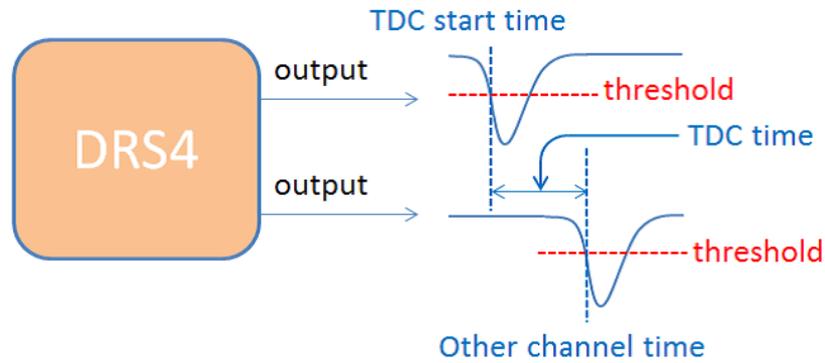


図 3.20 時間差取得の定義イメージ図。DRS4 からの出力波形より決定される TDC start time と Other channel time の差をヒストグラムに詰めて、ガウシアンでフィットする。フィット結果の RMS 値を時間分解能として定義する。

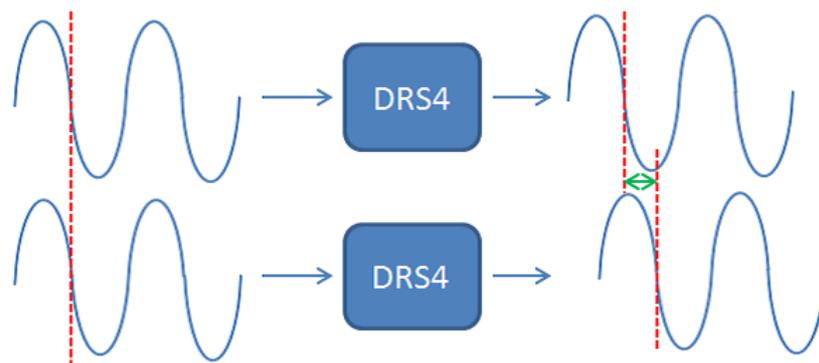


図 3.21 チップ間のタイミングの校正イメージ図。例えばタイミングの揃えた、あるいは固定遅延の正弦波を各 DRS4 チップに入れ出力の位相差 (図の緑矢印で表した部分) を各イベント毎に取得して、チップ間のタイミングのずれを校正する。

### 3.4.3 結果

テストは基準となるチャンネルに対し 0 nsec、100 nsec、200 nsec、300 nsec 遅延させヒストグラムを作成した。基準チャンネルは ch1、遅延チャンネルは ch7 である。結果を図 3.22 から図 3.25 に示す。

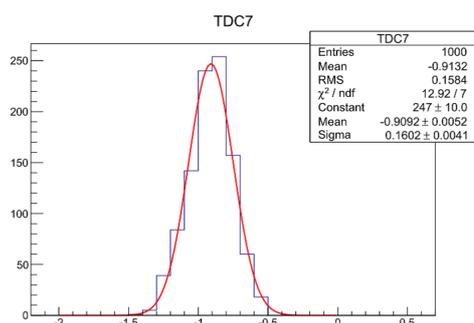


図 3.22 0 nsec 遅延 TDC ヒストグラム。横軸の単位は時間 (nsec)。ヒストグラムはガウシアンでフィットしており、Sigma を時間分解能とした。

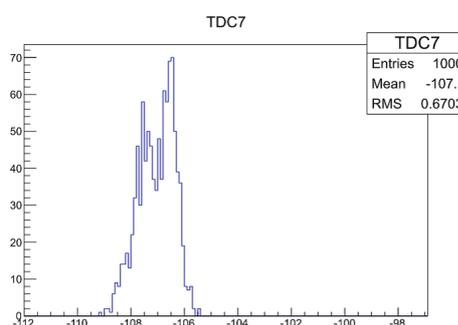


図 3.23 100 nsec 遅延 TDC ヒストグラム。横軸の単位は時間 (nsec)。

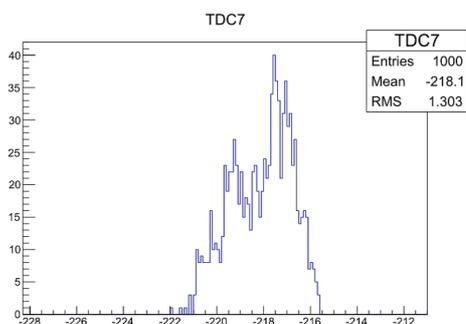


図 3.24 200 nsec 遅延 TDC ヒストグラム。横軸の単位は時間 (nsec)。

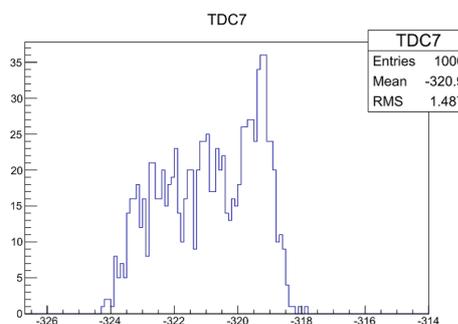


図 3.25 300 nsec 遅延 TDC ヒストグラム。横軸の単位は時間 (nsec)。

遅延なしの時は、きれいにフィットすることが出来て時間分解能  $0.16 \pm 0.00$  nsec を求めることが出来た。しかしながら、100 nsec、200 nsec、300 nsec の遅延の時は、ピークが広がるあるいはダブルピークが見えていて、うまくフィットできなかった。ピークが広がっていく傾向は遅延時間を長くするほど大きくなっている。図 3.26 に ASD 出力のオシロスコープ図を示すが、このタイミングではヒストグラムで見られる広がりを確認できなかったので DRS4 以下で問題が起きていると考えられる。

原因の一つとして考えているのがベースラインの変動である。本論文での解析方法ではベースラインから -100 mV を下回るタイミングを到達時間と定義しているが、パターンコレクションをした後のサンプリングデータをそのまま利用している。波形のフィッティングなどはせずにタイミングを決定しているため、ベースラインを出来るだけ 0 V に合わせて解析は行っているが、影響を完全に取り除くことが出来ていないと考えられる。また、本解析ではベースラインに傾きがないと想定しているが、実際には傾きがあり、その傾き自体が変動している可能性もある。傾き自体が変動している場合、傾く向きは同じと仮定して波形の 0 s 地点側から離れるほど (グラフでいうと右



図 3.26 ASD の出力をオシロスコープで見た図。TDC ヒストグラムで見られるような広がりは見られない。

側に行くほど) 変動による差が大きくなる。この差は到達時間の揺らぎを大きくしてしまう。これが原因である場合、遅延が大きくなるにつれベースラインの傾きの変動が大きくなるため、ヒストグラムの広がりが大きくなることにも説明がつく。ヒストグラムを見るとテールが時間差の大きい方へ広がっているため、基本的にベースラインには傾きがあり、あるタイミングで傾きが小さくなるという現象が起きていると考えられる。そのため解析方法を見直し、より正確に到達時間を決定する必要があるが、この点は、ROESTI 開発における今後の課題である。

チップ間のタイミング校正に関しては、同チップ内の時間分解能をうまく求めることが出来なかったため行わなかったが、解析手法は述べておく。図 3.21 でも書かれているが、チップ間に基準となる正弦波を与え、取得した波形をフィッティングし、チップ間の正弦波の位相差を求める。2つのチップ間には求められた位相差分のずれが生じているので、2つのチップ間の各チャンネルで時間差を求めるときは位相差分を足して(あるいは引いて)補正をかけることで正しい時間差を求めることが出来る。

### ベースラインに関して

現在使用している DRS4 の読み出しモードには Full Read Mode(以下、Full Mode) と Region of Interest Readout Mode(以下 ROI Mode) の 2 種類ある。DRS4 のスイッチドキャパシタにはアドレスが割り当てられていて、サンプリングを止めた後、Full Mode では 0 番目のキャパシタから波形の読み出しを開始する。Full Mode はパターンコレクション用のパターン波形の取得や基本性能評価の時に使用している。一方、ROI Mode ではサンプリングを止めたキャパシタから読み出しを開始する。そのため、DRS4 に入力された波形の時系列を非常に確認しやすい。今回、時間分解能測定では ROI Mode を利用している。

この Full Mode と ROI Mode であるが、2つのモードで電荷を与えていない時の波形(ノイズを測定する際の波形、ベースライン)を取得すると差異があることを確認できる。図 3.27 に ROI で取得したベースラインを示す。

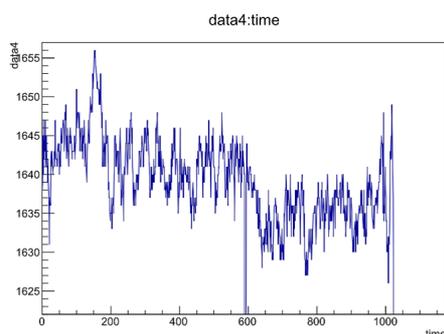


図 3.27 ROI Mode によるベースライン取得の一例。横軸は時間 (nsec)、縦軸は ADC カウントの値

正確な 1 次関数ではないが、時間が経つにつれベースラインが下がっている。図 3.12 と比べてみても違いは明らかである。すべてのベースラインで同じ傾向が見られるわけではないが、ベースラインの変動が Full Mode で取得したベースラインに比べて大きい。変動に関しては原因が不明\*1であるので、今回は Full Mode で取得したデータを線型性やノイズ評価などの際に利用した。

次に 1000 イベント分のプロファイルをとったベースラインのプロット図を Full Mode、ROI Mode で比較する。図 3.28、図 3.29 に結果を示す。

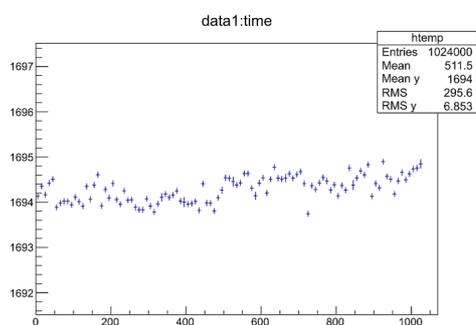


図 3.28 Full Mode で取得したベースラインのプロファイル図。横軸は時間 (nsec)、縦軸は ADC カウント値である。

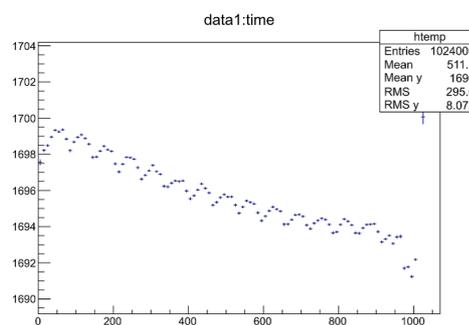


図 3.29 ROI Mode で取得したベースラインのプロファイル図。横軸は時間 (nsec)、縦軸は ADC カウント値である。

それぞれ図 3.28 が Full Mode、図 3.29 が ROI Mode で取得したものである。図 3.28 では右肩上がりになっているが、ADC カウントで 1 の変動である。これは電圧値としては、0.5 mV 程度である。参考文献 [6] の Page 6、Plot3 と Plot4 を見てみるとパターンコレクションを行った後でも、ベースラインには若干の傾きが残っている。傾く向きはパターンコレクションの波形で高い方から低い方、Plot3 では波形の前半部の方が後半部よりも電圧値が高く、その傾向が Plot4 でも残っている。図 3.28 で見られる傾向は参考文献 [6] での傾向と同じであるので (本論文ではパター

\*1 開発の現時点では、波形を読み出す際に DRS4 に与えるクロックのタイミングのずれや、読み出しモードを変更する際の DRS4 の初期化不足などファームウェア上の問題だと考えている。

ンコレクションの波形は後半部分が大きい)、取得は正しいといえる。しかしながら、ROI Mode で取得した波形は図 3.29 を見ると、明らかに右肩下がりとなっている。変動値も ADC カウント値で 6 であり、電圧値では約 3 mV 分となる。ROI Mode ではサンプリングを止めたキャパシタから読み出しているため、止まる位置で傾きに違いが出ることは想定できるが、変動値が Full Mode に比べて約 6 倍と大きくずれている。プロットはプロファイルであるため、実際にはもっと大きく変動している波形もあれば、あまり変動していない波形もあるはずである。大きく傾く理由は今後調べる必要がある。

図 3.28、図 3.29 は Full Mode で取得した波形パターンでコレクションを行っている。ROI Mode で取得した波形パターンからコレクションを行ったベースラインのプロファイルを図 3.30、図 3.31 に示す。この図を見ると Full Mode のプロファイル図も右肩下がりとなっており、ROI Mode で取得する波形には大きめなベースラインの傾きがのっていることが見て取れる。

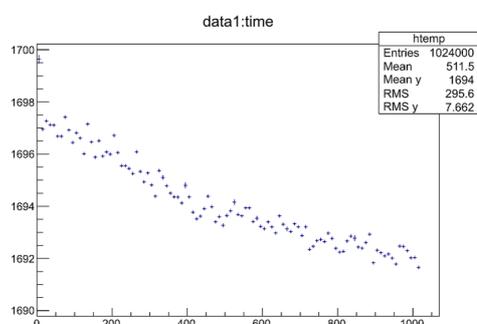


図 3.30 Full Mode で取得したベースラインのプロファイル図。横軸は時間 (nsec)、縦軸は ADC カウント値である。波形パターンは ROI Mode で取得した波形を用いている。

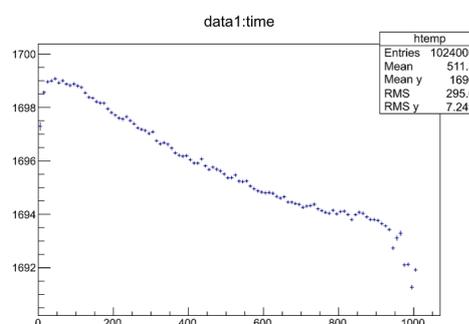


図 3.31 ROI Mode で取得したベースラインのプロファイル図。横軸は時間 (nsec)、縦軸は ADC カウント値である。波形パターンは ROI Mode で取得した波形を用いている。

以上の結果より、ROI Mode のベースラインは Full Mode に比べて大きく傾いていることを確認した。電荷を与えていないときの評価であるが、実際に電荷を与えたときも同じ現象が起きている可能性は大いにある。そのため、解析で傾きを考慮しないと、正しく時間差を測定することはできないと考えられる。また、傾き自体が変動している場合にも、時間分解能に影響を及ぼすことは想像できる。図 3.27 や図 3.29 に示したように 1 次関数で近似できるような傾き方をしていない場合もあるため、傾く原因の調査とともに解析手法の改善が非常に重要となる。

## 3.5 通信システムテスト

### 3.5.1 目的

前章で説明した TCP 通信を利用したデータ転送などは SiTCP\*<sup>2</sup>というモジュールをファームウェアに実装することで行っている。今回はチェーン接続によるデータ転送確認は間に合わなかったため、読み出し回路 1 枚でデータ転送速度を調べた。

### 3.5.2 評価方法

読み出し回路に与えるトリガーレートを上げていき PC 側で受信したビット数と受信にかかった時間より転送スピードを計算した。入力は全チャンネル (2ch, 9ch は除く) に決まった量の電荷を与えた。

### 3.5.3 結果

まずはトリガーレートを上げていきながら、線型性評価と同じように波高をプロットした。与える電荷は一定である。結果を図 3.32、図 3.33 に示す。

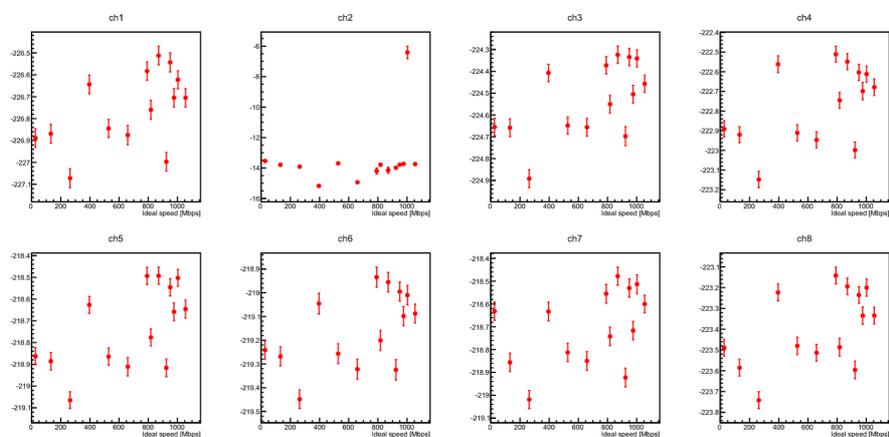


図 3.32 チップ 1 の転送レート当たりのピーク値プロット図。横軸は転送スピード (単位は Mbps)、縦軸の単位は mV。

\*<sup>2</sup> SiTCP は KEK の内田智久氏が製作した FPGA 実装モジュールである (参考文献 [14])

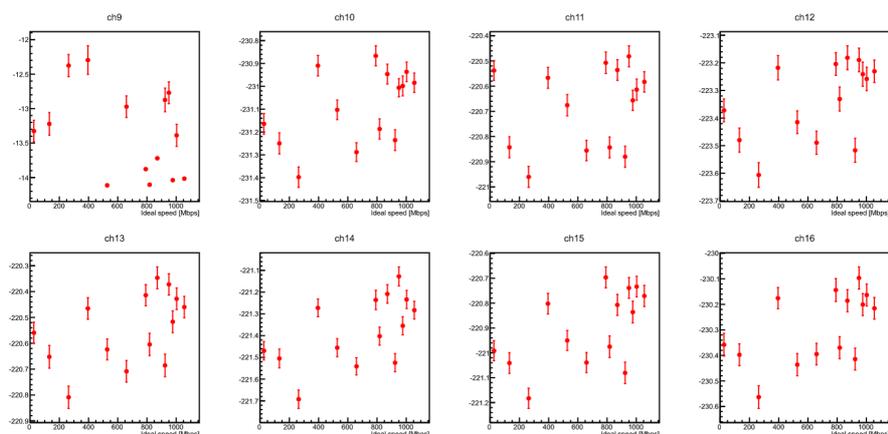


図 3.33 チップ 2 の転送レート当たりのピーク値プロット図。横軸は転送スピード (単位は Mbps)、縦軸の単位は mV。

プロット図を確認すると、ASD 出力から切り離している ch2、ch9 を除いたすべてのチャンネルで出力値の変動は 1 mV 未満に収まっている。波形図を全てのチャンネル、全てのイベントで確認したわけではないが波高の確認により、トリガーレートの増加で波形データに影響を及ぼすことなく正しく取得出来ているとした。

次に転送スピードの想定値と実測値でプロットした結果を図 3.34、図 3.35 に示す。図が 2 つあるのは、トリガーモード 2 つ (COMET トリガーモード、ノーマルトリガーモード) で測定したためである。ノーマルトリガーモードは外部トリガーが読み出し回路に到達直後に波形を読み出すモードであり、COMET トリガーモードは陽子ビームタイミングを意識したトリガーモードである。トリガーモードについての詳細は付録 B に記載している。

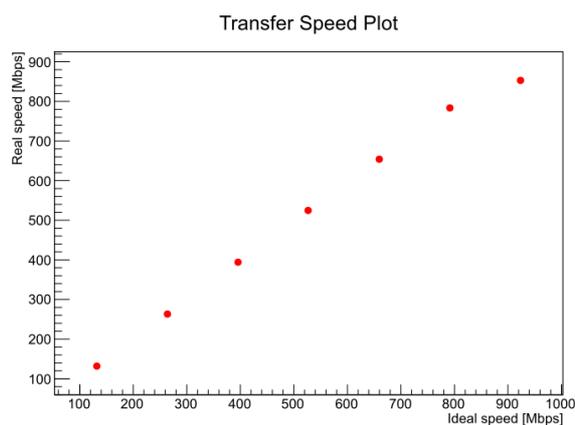


図 3.34 データ転送スピードのプロット図。横軸はトリガーレートから計算した転送スピード、縦軸は受信ビットとかがった時間から計算した転送スピードである。単位は Mbps。トリガーモードは COMET トリガーモード。

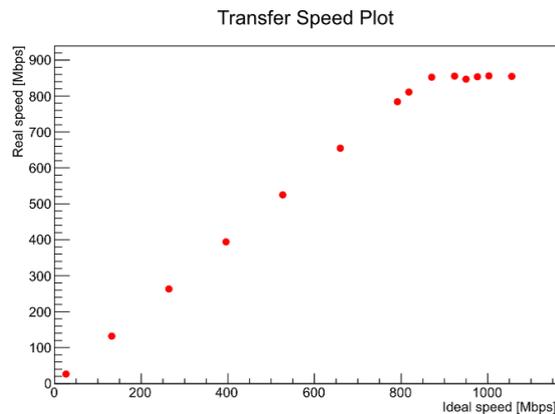


図 3.35 データ転送スピードのプロット図。横軸はトリガーレートから計算した転送スピード、縦軸は受信ビットとかがった時間から計算した転送スピードである。単位は Mbps。トリガーモードはノーマルトリガーモード。

どちらのトリガーモードでも 800 Mbps までは想定通りの転送スピードが実現出来ている。

ノーマルトリガーモードでは想定転送スピードが 850 Mbps を超えたあたりから実測転送スピードが飽和している。よって、実際の最大転送スピードは 850 Mbps であることがわかる。参考文献 [9] によると、950 Mbps まで転送スピードが出るはずだが今回のテストでは出ていない。達していない理由は、FPGA ファームウェアで SiTCP を駆動しているクロックの周波数が低いためと考えている。

COMET トリガーモードでも、900 Mbps を超えたトリガーレートでは 1 つしか取得していないが 850 Mbps 程度となっていて、ノーマルトリガーモードと傾向は同じである。しかしながら、今回のテストではグラフにプロットしていない 900 Mbps を超えるトリガーレートで何回か試したが、データ自体をうまく取得できなかった。ノーマルトリガーモードでは想定転送スピード 900 Mbps 以上でも取得が出来たので、COMET トリガーモードのファームウェアが原因であると考えられる。レート増加によって想定していないタイミングで FPGA の内部回路が動作している可能性があるため、今後専用ツールを使って動作の確認を行い、エラーの詳細を確認する必要がある。

## 第4章

# 考察・今後の課題

この章では今回の評価で得られた結果が実験要求を満たしているかについての議論を進める。また今後性能向上を測る上で必要と思われる事項についての検討を行う。

### 4.1 電荷増幅に関して

#### 4.1.1 S/N 比

ストローチェンバーからの最低電荷量は $-16\text{ fC}$ であると予想されているが、今回の評価では手持ちの道具が足りず $-16\text{ fC}$ を直接読み出し回路に入れて評価をすることが出来なかったため、得られた電荷-電圧変換式とノイズから計算し $-16\text{ fC}$ に対するS/N比を評価していく。ただし先に得られた返還式は入力電荷 $-100\text{ fC} \sim -300\text{ fC}$ でえられたものであるため、 $-16\text{ fC}$ までの外挿には大きなエラーがつきまとわざるを得ない。

例としてch1を取り上げる。電荷-電圧変換式は  $\text{電圧} = 1.03 \pm 0.02 \times \text{電荷} + 1.1 \pm 5.0$  であるので、想定最低電荷 $-16\text{ fC}$ を代入すると、信号電圧値は $-15.4 \pm 5.0\text{ mV}$ となる。この $5.0\text{ mV}$ という誤差はフィッティングのエラーによるものである。先述の通り変換式は $-16\text{ fC}$ よりも大きな電荷量( $-100\text{ fC} \sim -300\text{ fC}$ )を与えて求めているため、外挿時に大きなエラーがついてしまうが今回はこのままで計算を進めることにする。ch1のノイズ電圧値は $0.77 \pm 0.00\text{ mV}$ であるので、信号をノイズで割って絶対値をとると、S/N比は $20.0 \pm 6.5$ となる。誤差を考慮して、最低のS/N比は $20.0 - 6.5 = 13.5$ となる。実験要求としてS/N比が8以上とされているので、クロストークを考慮しなければ信号を観測するには十分なS/N比であるといえる。クロストークに関しては後述する。

以上の計算をすべてのチャンネルに対して行った結果が表4.1である。ばらつきはあるものの、最低S/N比は全チャンネルで10以上であり、 $-16\text{ fC}$ の信号を読み出すには十分であるといえる。

どのチャンネルもS/N比の誤差が大きくなってしまっているのは、フィッティングの際の切片の誤差が大きくなってしまっているので前述した通り線形性を評価する際の低電荷( $-100\text{ fC}$ 以下)のプロット数を増やすことでS/N比の誤差も小さくすることが可能であると考えられる。また読

み出し回路に正しく-16 fC を与えて評価できるような環境があれば一番簡単であるので、環境作りも必要である。

今回の S/N 比評価では最低電荷を-16 fC としているが、最近のストローチェンバーの研究 [12] によるとチェンバーへの印加電圧 1500 V で約-280 fC の電荷がストローチェンバーより発生するという結果が得られている。この値であれば S/N 比は 10 倍以上よくなるので、ストローチェンバーが正常に動作すれば信号を見るには十分である。

ch	S/N 比	最低 S/N 比	ch	S/N 比	最低 S/N 比
ch1	20.0 ± 6.5	13.5	ch10	22.2 ± 5.6	16.6
ch3	22.4 ± 5.5	16.9	ch11	20.7 ± 5.7	15.0
ch4	23.4 ± 6.5	16.9	ch12	21.7 ± 5.3	16.4
ch5	21.4 ± 5.7	15.7	ch13	23.0 ± 6.0	17.0
ch6	23.1 ± 5.5	17.6	ch14	21.3 ± 5.2	16.1
ch7	26.8 ± 5.1	21.7	ch15	21.1 ± 5.1	16.0
ch8	22.0 ± 5.3	16.7	ch16	22.3 ± 5.1	17.2

表 4.1 想定されている最低電荷量-16fC に対する S/N 比。一番低くても 13.5 であるので、測定するには十分といえる。

#### 4.1.2 クロストーク

クロストークの評価は今回 dB 単位で示した。クロストークに関しても実験要求として決まっているわけではないので表 3.4 から表 3.17 で示した程度であるということしかできない。実際のクロストーク評価時のピーク電圧値は最大で-3 mV であったので、仮に ch7 以外のチャンネル (ch2 と ch9 は除く) に-300 fC 程度の信号が同時に到達した場合、ch7 に-42 mV 程度のクロストークが起こると想定できる。この場合であると ch7 に現れた信号がストローチェンバーからの信号かどうか区別がつかなくなるどころか埋もれてしまう可能性もある。しかしながら、十数 nsec の精度で信号のピーク位置が 14 ch 全てで一致するとは考えにくく、実験のイベントレートから考えても 1 ボードあたりの平均ヒットチャンネルが 3~4 チャンネル分程度である。よってクロストークの影響はそこまで及ばないと考えられる。このクロストークの評価も最低電荷を-16 fC と仮定してのことなので、S/N 比でも述べた通りストローチェンバーが正常に動作すれば影響はさらに小さくなる。

現在どのタイミングでクロストークが起きているのか正確に特定できていないので、起きている箇所の特定も必要となる。

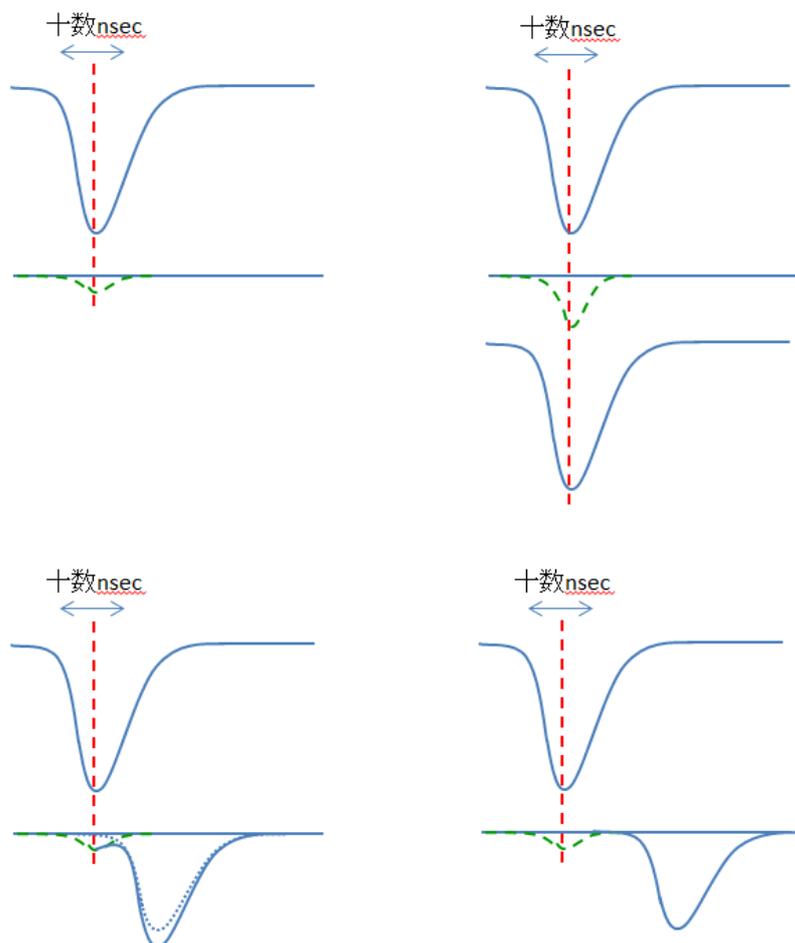


図 4.1 クロストークのイメージ図。あるチャンネルに信号が入力されると、実際は来ていないのに他のチャンネルに小さな信号が入力されたように見えてしまう (左上図、緑の点線がクロストークにより発生した信号を表している)。クロストーク評価により実際に入力されている信号の 20dB 程度の波高である。2 チャンネルにほとんど同時に信号が来るとクロストークは増加すると考えられる (右上図)。クロストークが発生したチャンネルに実際に信号が入力されたとき、信号の到達時間が近いとクロストークの影響を受けて本来の波形よりも波高が高くなってしま (左下図、青の点線波形が実際の波形)。しかし信号の到達時間に差があるときは影響をあまり受けずに波形を読み取れる (右下図)。クロストークが発生しても左下図のように少しでもずれていれば、波形解析で取り除くことも可能だと考えられる。クロストーク発生タイミングはピーク位置付近と考えているので十数 nsec の範囲でピークが重ならない限り影響は取り除くことができるであろう。

## 4.2 消費電力に関して

読み出し回路 1 枚で消費する電力は 5 W である。これはチャンネル当たりで  $5/16 = 312.5$  mW/ch となる。実験要求としては 240 ~ 480 mW/ch と言われているので現状で要求は満たして

いる。

### 4.3 時間分解能に関して

本論文では、まず同じチップ内のチャンネルで基準チャンネルと測定チャンネルを定義し、2つのチャンネル間の時間差を求めることで時間分解能を評価しようとした。しかしながら基準チャンネルと測定チャンネルの時間差を大きくすると時間分解能が悪くなるという現象が見られた。この原因の一つとして、ベースラインに傾きが生じているためと現段階では考えている。時間測定の際のモード (ROI Mode) ではベースラインが右肩下がりであり、右肩下がりの割合もイベント毎に変動していると思えばより信号が後に来れば来るほど右肩下がりの影響を強く受けると考えられる。よって時間取得の際のモードのベースラインの右肩下がりの補正が必要となる。また、基板実装や DRS4 を制御する FPGA のファームウェアなど別の箇所が原因という可能性も完全に捨て切れていないため、そのあたりも含めて考えていく必要がある。

### 4.4 データ転送に関して

転送スピードは 800 Mbps 程度までは想定近くの転送レートをだすことが出来た。実験要求はストロー 1 層あたり 360 Mb/1.4sec であるので、どちらのトリガーモードであってもチェーン接続時にスピードが正しくできれば問題はないと考えられる。図 4.2、図 4.3 に実験要求の赤線を加えたプロット図を示す。チェーン接続に関しては SiTCP 開発者に現在評価を頼んでいる最中であるため、その評価結果次第である。

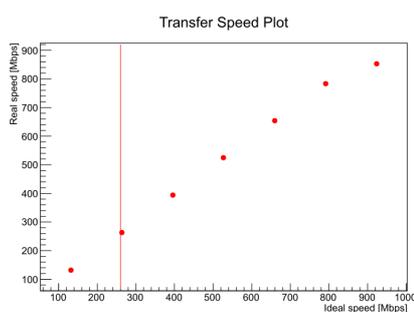


図 4.2 データ転送スピードのプロット図。図 3.34 に線を引いたものである。赤線は実験要求から計算した転送スピードである。赤線部分では転送スピードの線型性は保たれている。

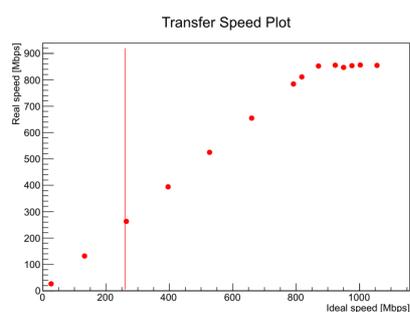


図 4.3 データ転送スピードのプロット図。図 4.3 に線を引いたものである。赤線は実験要求から計算した転送スピードである。赤線部分では転送スピードの線型性は保たれている。

## 4.5 まとめ

本論文では COMET 実験で用いられるストローチェンバーの読み出しエレクトロニクス (ROESTI) のプロトタイプの開発、評価を議論した。評価した項目は以下の通りである。

1. 波形読み出しの正当性
2. ノイズ、線型性などの基本性能
3. 時間分解能
4. 通信評価

正当性評価では、オシロスコープで取得した波形との比較して相似形であるかどうか調べた。差分評価からは完全に相似形であるとは言い切れなかった。しかしながら、波形の歪みは波高を変化させても特に見受けられず、立ち上がり部分も正確に読み出せていると判断できる結果となった。

基本性能評価ではノイズ、線型性、クロストーク、消費電力を調べた。各チャンネルでノイズは  $0.74 \sim 0.78 \text{ mV}$  であり、電荷-電圧変換係数は  $0.97 \sim 1.03 \text{ mV/fC}$  であった。ノイズ、線型性からは S/N 比も評価した。フィッティングエラーが大きかったため最終数値の誤差も大きくなってしまったが、全てのチャンネルで S/N 比が 10 よりも良い結果となり、想定されている最低電荷量であっても読み出すことが出来ることが分かった。クロストークも各チャンネルで 20dB よりも良い結果となり、CDR によるヒットチャンネルレートから計算してもそこまで影響はないと判断した。消費電力は簡単に調べた程度であるが、 $5.0 \pm 0.1 \text{ W}$  となり要求値  $240 \sim 480 \text{ mW/ch}$  を満たすことが出来た。

時間分解能はうまく評価することが出来なかった。原因の1つとして、ベースラインの変動を考えており、解析手法の改善が必要となる。またファームウェアに問題があることも考えられるため、改善が必要である。

通信評価では読み出し回路1枚でデータ転送スピードを調べた。最大転送スピードとして 850 Mbps が得られ、実験要求  $350 \text{ Mb}/1.4\text{s}$  を十分に満たすことが出来た。参考文献によると 950 Mbps まで出るはずであるので、今後ファームウェアの改善でスピードはさらに速くなると考えられる。

全体のまとめとして、今回開発した ROESTI プロトタイプは現状では評価がまだ不十分であるため、実験に使えるかどうかはまだ判断できない。よって、今後もファームウェア開発・解析手法を改良して評価を続けていく必要がある。また、時間キャリブレーション用クロックの入力部など明らかに改善しないといけないこともあるので、次のプロトタイプの開発も必要となる。

## 謝辞

本論文の執筆、引いては本研究を進めるにあたり、多くの方々に助けていただきました。この場を借りて感謝を申し上げます。

久野良孝教授には自由に研究できる環境を与えていただき感謝しています。青木正治准教授には映画、読書など自身の興味のある事をよく話してくださいました。研究だけでなく、これら趣味のお話は大変楽しかったです。佐藤朗助教授には発表資料作成の際に鋭く厳しい意見をいただきました。大変なことも多くありましたが自分の成長につながりました。板橋隆久氏には研究分野に限らず、様々なお話をしてくださいました。知識をより深めることが出来ました。坂本英之氏には、学部4年時から解析において色々と教えていただきました。自分で解析を進める上で過去の多くの助言を参考にさせていただきました。小出義夫先生や長島順清氏には物理に関して詳しく教えていただきました。秘書の戎家美紀さん、駒井順子さんには出張が多い中、いろいろと調べていただき、非常に助かりました。

この2年間の多くを過ごした高エネルギー加速器研究機構の方々にもよく面倒を見ていただきました。機構所属の三原智氏には本論文の執筆を始め、実験に関すること、解析手法を丁寧に教えてくださいました。同機構の西口創氏には飛跡検出器について教えて下さりました。両氏のサポートにより、実験の理解を深めることが出来ました。

同機構のエレキグループの方々にもこの2年本当にお世話になりました。内田智久氏にはファームウェアの書き方を始め、普段の進捗も気にしていただき、よく面倒を見ていただきました。池野氏には経験のない回路設計において色々と教えていただき、また回路図確認作業でも多くの時間を割いていただきました。田中真伸氏には実験、回路開発全般に関して色々と教えていただきました。学会の際にもお世話になりました。

回路開発においては企業の方々にもお世話になりました。GNDの宮沢正和氏には時には無理な注文も聞いていただきました。迷惑を多々おかげしましたが、回路を製作していただきありがとうございました。林栄精器(株)の青木巖氏にはチップ、ケーブル購入の際にお世話になりました。

また、多くの学生の方々にも助けていただきました。久野研究室のTran Nam Hoai氏、Nguyen Duy Thong氏、Izyan Hazwani Hasin氏、Nguyen Minh Truong氏、日野祐子さん、伊藤慎太郎君、松本侑樹君、矢井克忠君、岩見大樹君、相川脩君、宇津木卓君、鷹尾賢三君、去年卒業された仲井裕紀さん、曳田俊介さん、日浅貴啓さん、一昨年卒業された松下絵里さん、高橋諭史さん、伊藤恭之君、いずれの皆さんにも出張が多く時々研究室に帰った時でも声をかけていただき、明るい

雰囲気の研究室にさせていただいたことに感謝しています。高エネルギー加速器研究機構で知り合った学生の方々にも、研究生生活を楽しいものにしていただきました。つくばでおいしいご飯を食べることが出来たのは皆さんのおかげです。他にも学部時代からの友人、部活動の同期等、数えきれない方にお世話になりました。皆様のおかげで有意義な学生生活を送ることができました。

最後となりましたが、両親、祖父母には一番迷惑をかけてきましたが、今まで支えていただき本当にありがとうございました。

## 付録 A

# ポアソン到着

### A.1 ポアソン到着

本論文では取得率やバッファサイズの見積もりのため、イベントヒットがポアソン分布に従うと仮定して計算を行った。ある事象がポアソン分布に従って起こると仮定して計算するとき、ポアソン到着という。ポアソン分布の式は

$$P_a(k, x) = \frac{(\lambda x)^k}{k!} e^{(-\lambda x)} \quad (\text{A.1})$$

と表される。x は到達時間、λ は単位時間内に到着する平均個数 (到着率)、k は到達する個数を表している。例えば、実験に当てはめてみると λ はイベントレートに相当すると考えられ、x 時間以内に k イベントが到着 (ヒット) する確率が  $P_a$  であるということが出来る。

### A.2 取得率

ポアソン到着を仮定した場合、あるイベントが到着してから t 秒以内に次のイベントが到着するまでの確率は

$$P(t) = \int_{t_0}^{t_0+t} \lambda e^{-\lambda t'} dt' = e^{-\lambda t_0} (1 - e^{-\lambda t}) \quad (\text{A.2})$$

となる。ここで  $t_0$  は最初のイベントの到達時間であるが基準時間ということで  $t_0 = 0$  とすれば、

$$P(t) = 1 - e^{-\lambda t} \quad (\text{A.3})$$

となる。ここで 2 章で述べたようにイベントレートが 1 kHz、デッドタイムが 45 μs であると仮定する。デッドタイムの間に到達したイベントは取得することが出来ないのでデッドタイム以内に次のイベントが来る確率を計算し、1 から計算した確率を引けば取得できる確率 (取得率) を求めることが出来る。λ = 1000、t = 45 × 10<sup>-6</sup> を代入すると、取得率は  $1 - (1 - e^{-1000 \times 45 \times 10^{-6}}) = 0.956 > 95\%$  となる。

### A.3 バッファサイズ

前節で述べた取得率はバッファサイズが十分であると仮定して計算したものである。バッファサイズとはどのぐらいイベントを記録しておくことが出来るかである。実際にはどのぐらいの容量があれば十分かを計算していく。

まず、データ転送スピードについて考える。転送スピードは最大で 1 Gbps であると固定する。CDR より考えると、1 イベントで 1 読み出し回路あたり 3ch のヒット平均であるのでデータサイズは 49.6kb となる。1 イベント分を転送するのにかかる時間は  $49.6\text{kb} / 1\text{Gbps} = 49.6\ \mu\text{s}$  となり、理論上では 1 秒あたり  $1/49.6\ \mu\text{s}$  分のイベントを転送することが出来る。ここで  $\rho = 1\text{kHz} / (1/49.6\ \mu\text{s}) = 0.0496$  という量を定義すると、参考文献 [10] より引用したグラフ (図 A.1 を参照) と照らし合わせれば、6 イベント分記録しておくことが出来ればバッファがフルになる確率が非常に小さくなることがわかる。

今回の評価よりデータ転送スピードは 800Mbps 程度までしか安定した動作は確認できていないが、その場合先述の  $\rho$  を 0.8 で割って  $\rho = 0.062$  となる。しかしながらグラフと照らし合わせればやはりバッファがフルになる確率は非常に小さいので十分といえる。

6 イベント分であれば FPGA 内で使用できるメモリで十分であるため、新しくメモリを実装したりする必要はない。

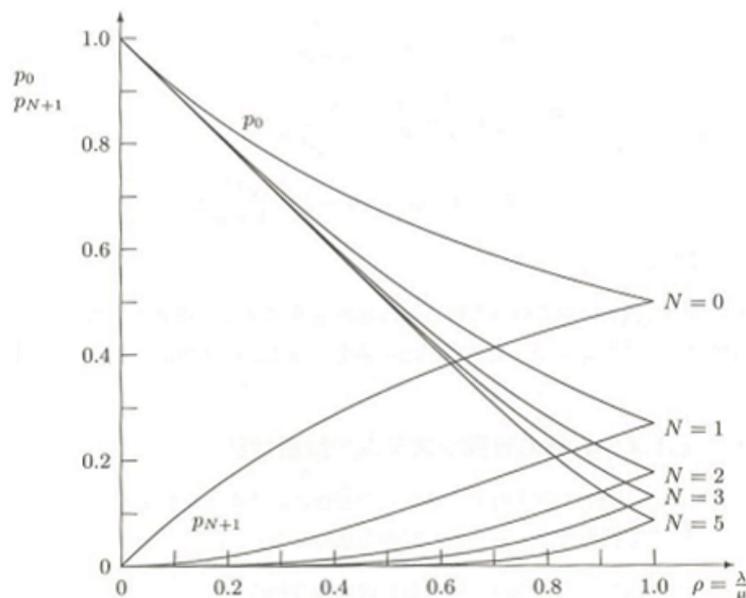


図 A.1 バッファがフルになる確率。縦軸がバッファがフルになる確率、横軸は本文中で定義した  $\rho$  である。グラフ中に書かれている  $N=5$  はバッファサイズが 6 であることを示しており今回  $\rho = 0.062$  であるので、グラフを確認してみるとほとんど 0 である。よってバッファがフルになることはほとんどないと仮定できるため、バッファサイズは十分であるということが証明出来る。グラフは参考文献 [10] の 89 ページ、図 5.3 を引用した。

## 付録 B

# FPGA ファームウェア

以下、FPGA のファームウェアなど開発したもの・読み出し回路の使用方法・動作を記述していく。

### B.1 SPI コンフィギュレーション

FPGA に書き込むためのダウンロードデータは、通常は ROM に保存しておき電源がオンになった時に FPGA に書き込まれるようにしておく。よく使われるものは Xilinx Flash PROM であるが、他社製のシリアル PROM なども使用できる。書き込みは専用のプラットフォームケーブルを用いるが、今回は SiTCP 経由 (UDP 通信) でダウンロードデータをシリアル PROM に書き込み、FPGA をリブートできる機能を追加した。

#### B.1.1 シリアル PROM

読み出し回路に実装したシリアル PROM は、M25P64 という Xilinx 社も使用の保証をしているメモリである。サイズは 64MB である。

#### B.1.2 ファームウェア

FPGA に実装しているシリアル PROM へ書き込むためのモジュールは、KEK の内田智久氏が開発したものに改良して実装している。SiTCP の使用を想定して作成している。モジュールファイルは SPLIF.v。

#### B.1.3 ポート説明

SPLIF モジュールの主なポートを説明する。

##### CLK

システムクロック入力。

**SPI\_ 信号名**

シリアル PROM の対応しているピンに接続する。

**RBCP\_ 信号名**

SiTCP の RBCP ポートにそれぞれ接続する。

**div\_value**

システムクロックの周波数に応じて定数を与える。

~40MHz:2'd1 40MHz~160MHz:2'd0,2'd2 160MHz~:2'd3

**prog\_start**

FPGA リブートのスイッチ信号。

**B.1.4 レジスタアドレスマップ**

UDP 通信でアクセスする。アドレスあたり 8 ビット長である。

0x00-0x03 : FPGA Version ID (RBCP の読み取り) / RAM の書き込み/読み出し開始位置の設定。シリアル PROM に書き込むデータ、またはシリアル PROM からのデータはいったん FPGA 内の RAM に保存されてから送信・受信されるが、RAM の書き込み/読み出し開始アドレスを設定することが出来る。

0x04 : User Input

0x05 : Control command (4bit) / Control command length (4bit)

0x06-0x07 : シリアル PROM に書き込むデータの長さ

0x08 : シリアル PROM のコマンド (WRITE,READ,WRITE ENABLE など)

0x09-0x0b : シリアル PROM の書き出し/読み出し開始のアドレス

0x1F : コンフィグ (リブート) のスタート。このアドレスに 0xFF を書き込むとコンフィグがスタートする。

0x800-0xFFFF : FPGA 内の RAM アドレス。自由に書き込みができる。

**B.1.5 書き込みプログラム**

FPGA との通信は UDP 通信を用いる。プログラムは Python で記述されていてコマンドライン上で実行される。使用方法は以下の手順である。

1. プログラム実行前に mcdfile\_folder に ISE で生成した spi 用の mcs ファイルをコピーする。
2. config\_folder 内の config.txt に、1 行目 : 書き込みたいファイルネーム、2 行目 : 書き込み先の IP アドレス、3 行目 : 書き込み先の UDP ポート番号を記述しておく。

3. fpga\_configuration.py を実行する。
4. mcs ファイル変換をパスするか問われるので、Yes か No を選択する。基本的には No を選択する。
5. mcs ファイルの変換、PROM のイレイス、PROM への書き込み、Verify が行われる。失敗した場合は途中で中断される。
6. 5 がすべて終わると最後に FPGA を再コンフィギュレーション (reboot) するかを問われる。
7. Yes と書き込むと、reboot 用の信号が FPGA に送られてコンフィグを開始する。

### B.1.6 再コンフィギュレーション (リブート)

書き込みプログラムの最後の工程でリブート用の信号が FPGA に送られると、読み出し回路上に実装してある IC にパルスが送られる。その後 IC より  $1\mu\text{sec}$  程度の幅を持った負論理のパルスが FPGA のコンフィギュレーションを開始するピンに送られて、FPGA のコンフィギュレーションが開始される。設計段階では、Xilinx Flash PROM と切り替えて使うことを想定していたのでジャンパピンによってパルスが送られるラインとシリアル PROM からデータを読み出すラインを選択することが出来る。ダウンロードケーブルを用いる際はジャンパを外してラインを切っておく必要がある。

## B.2 ROESTI プロトタイプのレジスタマップ

開発したプロトタイプのレジスタマップを説明する。特定のアドレスに特定の値を書き込むことで初期化・ASD のしきい値設定などを行うことが出来る。UDP 通信でアクセスする。アドレスあたり 8 ビット長である。前節で説明した SPLIF を改良して用いているため、一部異なる部分もある。なお、光ケーブルのポートが 2 つ実装されているが、パケットインザーターの機能をオンにしておくと、どちらのポートにつないでもアクセスが可能となる。(IP アドレス、ポートの指定は必要)

0x00-0x03 : FPGA Version ID (RBCP の読み取り専用)

0x05 : Control command (4bit) / Control command length (4bit)

0x06-0x07 : シリアル PROM に書き込むデータの長さ

0x08 : シリアル PROM のコマンド (WRITE, READ, WRITE ENABLE など)

0x09-0x0b : シリアル PROM の書き出し/読み出し開始のアドレス

0x10 : 0xF0 を書き込むことで FPGA モジュールのリセットを行う。

0x19 : 0x0F を書き込むことで FPGA モジュールのロード (スタート) を行う。

0x11-0x14 : RAM の書き込み/読み出し開始位置の設定。シリアル PROM に書き込むデータ、またはシリアル PROM からのデータはいったん FPGA 内の RAM に保存されてから送信・受信されるが、RAM の書き込み/読み出し開始アドレスを設定することが出来る。

0x15-0x16 : ASD のしきい値を設定する。総ビットは 12 ビットであり、0x15 の下位 4 ビットが 12 ビットの上位 4 ビット、0x16 の 8 ビットが 12 ビットの下位 8 ビットとなる。

0x17 : 通信を行っている読み出し回路の回路番号を設定する。

0x18 : チェーン接続の一番端の読み出し回路番号を設定する (使用方法は後述)。

0x1A : ADC データのレイテンシの調節を行う。具体的にはサンプリングした波形の最初のサンプリングポイントがデータの最初に正しく来るように調整するためのものである。現状は 0x04 を書き込めば正しく動作することが確認されている。

0x0C-0x0F : 読み出しイベント数の設定を行う。これらのアドレスに書き込まれた値分のイベントデータが PC 方向へ送信されると、ファームウェア側でイベントデータ送信の停止し、トリガーの受け付けを停止する。PC 側でデータ転送の TCP 通信接続を切断しても、ファームウェア側が送信を停止しないことがしばしば発生したため、この機能を追加した。

0x1E : テスト用の信号出力。0xF0 を書き込むとパルスが発生する。

0x1F : コンフィグ (リブート) のスタート。0xFF を書き込むとコンフィグがスタートする。

0x1D : 読み出しモードの設定。上位 3 ビットは使用していない。上から数えて 4 ビット目は時間分解能の項目で説明した DRS4 の Readout Mode の設定をする。1 であれば Full Readout Mode、0 であれば ROI Readout Mode となる。下位 4 ビットは以下のビットを書き込むことでモードを切り替えることが出来る。上から順に優先される。

4'b0001 : COMET トリガーモード。

4'b0010 : ベースライン取得モード。

4'b0100 : セルフトリガーモード。

4'b1000 : ノーマルトリガーモード。

その他 : 動作確認はしていない。

読み出しモード・トリガーに関しては後述する。

### B.3 パケットインザーター

本読み出し回路には光ケーブルコネクタが 2 つ実装されてる。これは、2 章で説明したようにチェーン接続を想定して実装したものである。光ケーブルコネクタと SiTCP モジュールは 1 対 1 対応のため、ファームウェアには 2 つの SiTCP モジュールが実装されている。SiTCP モジュールには IP アドレスが 1 つ割り当てられているため、読み出し回路 1 枚につき IP アドレスが 2 つ割り当てられている。例えば、読み出し回路を 2 枚チェーン接続した場合を考えると、IP アドレスは 4 つ存在することとなり正しい送り先にイーサネットパケットを送る必要がある。その他にも、イベントデータ転送用の TCP 通信に関してはデータ量が非常に大きいため伝達経路を制限する必要がある。これらのことを解決し、イーサネットパケット破損が起きないようにパケットインザーターというモジュールをファームウェアに実装した。イーサネットパケット破損のイメージを図 B.1 に示す。また、UDP 通信、TCP 通信のイメージも図 B.2 に示す。

パケットインザーターではイーサネットパケット破損が起きないようにイーサネットパケットを挿入するモジュールである。基本は光ケーブルコネクタと 1 対 1 対応している SiTCP とのイーサネットパケットの間にスペースがあれば、もう一つの SiTCP からのイーサネットパケットを挿入するというものである。挿入したいイーサネットパケットのデータ長をあらかじめ調べておき、挿入するタイミングになった時にデータ長分のスペースがあった場合にのみ挿入する。条件が満たされない場合はイーサネットパケットは破棄される。

現状の開発状況としては、図 B.2 のパケットフィルター、パケットインザーターともに問題は発生せずに動作している。関連 FPGA コードは `Packet_Insert.v`、`Packet_Filter.v`、`packet_inserter.v`、`addr_buffer.v`、`shift_buffer.v` である。`addr_buffer.v`、`shift_buffer.v` は ISE のツールで作製したものである。

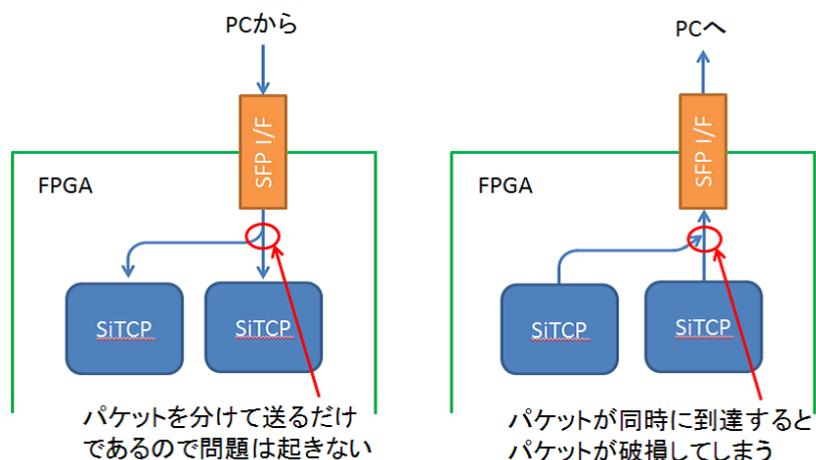


図 B.1 イーサネットパケット破損のイメージ図。PC から SiTCP モジュールへの送信は問題なく行えるが、SiTCP から PC への送信の場合、うまくイーサネットパケットを挿入しないとイーサネットパケットデータの混合が起きてイーサネットパケット破損が起きてしまう。そのため、挿入するタイミングで挿入したいイーサネットパケットのデータ長分のスペースがあった場合にのみイーサネットパケットを挿入する、パケットインザーターが必要となる。

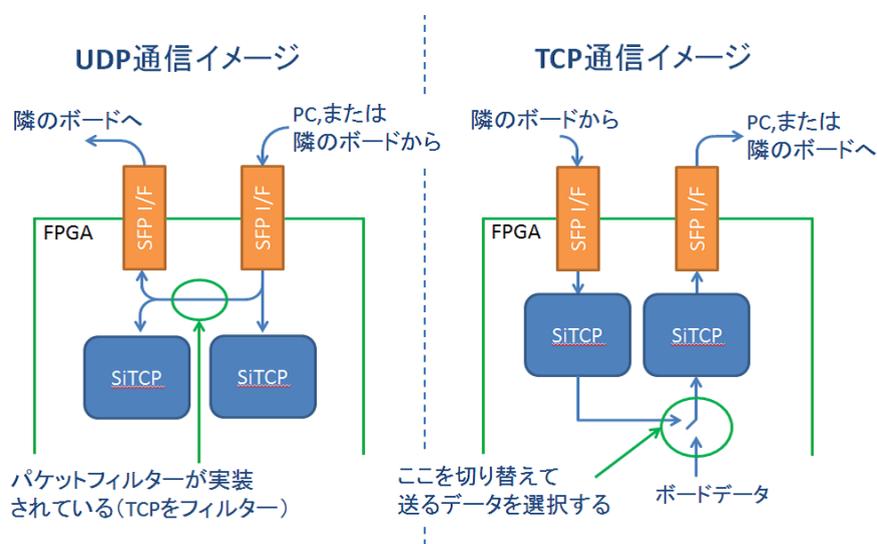


図 B.2 UDP/TCP 通信のイメージ図。左図が UDP 通信、右図が TCP 通信を表している。UDP パケットはチェーン接続されている PC には直接接続されていない読み出し回路にも送信する必要があるため光ケーブルコネクタ (図中の SFP I/F) の 1 つからもう 1 方の光ケーブルコネクタへ直接接続されている。途中にパケットフィルターが挟まれており TCP パケットは通過できない。TCP 通信を用いたイベントデータ転送は SiTCP を介して行われる。SiTCP を介することでイベントビルダーも同時に行っている。イベントビルダーに関しては後述する。

## B.4 リセット・ロード

ファームウェア内のリセット・ロードについて説明する。

PC よりリセット信号が FPGA へ送られると、ファームウェアの各モジュールはほとんどがリセットされる。リセットの意味は、例えばカウンターの値を 0 にする、I/F にリセットがかかった状態にするなどのことであり、モジュール自体がリセット信号をトリガーとして動作を始めるわけではない。ただし、DRS4 をコントロールモジュールに関してはリセット信号をトリガーとして DRS4 の設定を開始する。また、PC との通信を制御しているモジュール (SiTCP や光ケーブルインターフェースなど) やシステムクロックから生成しているクロックイネーブルは信号を送ってもリセットされることはない。DRS4 に関しては Readout Mode を ROI から Full に切り替えるときにも DRS4 の設定が開始される。

実際の動作は PC からロード信号が送られることで開始する。動作を開始するモジュールは DAC 制御モジュールと ADC インターフェースなどである。ASD のしきい値設定用の DAC を制御するモジュールをロードするときは、ロード前に設定したいしきい値を PC から指定しておく必要がある。DAC はもう 1 つ、DRS4 を制御するためのものが実装されているが、出力する値が固定値でよいので PC 側からは設定出来ないようになっている。ADC インターフェースではロードをすることでシリアルパラレル変換モジュールのリセットを解除することになる。

関連 FPGA コードは `sync.signal_generator.v` である。

## B.5 DAC 制御モジュール

読み出し回路に実装されている DAC は全部で 3 つある。1 つは ASD しきい値設定用、残りの 2 つは DRS4 の設定用である。チャンネル数の違いやデータ幅の違いなどはあるが、基本的な動作は同じでシリアルデータを DAC に送ることで DAC が送られたシリアルデータをもとに電圧値を出力する。シリアルデータに同期したクロックやイネーブルも DAC へ送る。先述の通りであるが、DRS4 の設定用 DAC の出力は変更できない。参考文献 [6] によると、DRS4 設定電圧値は入力される信号のレンジで決定されるためである。入力信号のレンジはほとんど固定であるため、設定電圧値も固定となる。関連 FPGA コードは `dac_for_asd.v`、`dac_for_drs.v` である。

## B.6 DRS 制御モジュール

DRS4 は 2 つ実装されていて、リセット信号をトリガーとして設定が始まる。具体的な動作は以下の通りとなる。

まずは DRS4 のリセットがなされる。これは DRS4 にリセット専用のピンがあり、負論理のパルス (最低幅 10nsec) を与えることで初期化される。DRS4 リセットの後に Configuration Register と Write Shift Register の設定を行う。設定値は仕様書に従っている。なお DRS4 リセットをか

けた後であれば、Configuration Register の設定はいらぬようであるが念のためということで今回はファームウェアに記述している。2つの設定が終わった後に DRS4 の読み出しモードが Full Readout Mode であれば Read Shift Register の設定を行う。ROI Readout Mode の場合、この設定は行わない。以上の設定がすべて終了した後に、今度は DRS4 へリファレンスクロックを与える。サンプリングスピードは与えるリファレンスクロックに依存していて与えるクロックの周波数の 2048 倍の速度でサンプリングを行う。リファレンスクロックを与えサンプリングクロックが安定すると DRS4 からの信号である PLLCK が 1 となる。リファレンスクロックを与えるのほとんど同時に、DENABLE、DWRITE と呼ばれる信号も 1 にする。DENABLE とは Domino Ring を動作させるイネーブル信号、DWRITE とはキャパシターに書き込む時のイネーブル信号となる。Domino Ring とは、スイッチドキャパシターのスイッチを切り替える大本の機構である。PLLCK が 1 となり、かつ DENABLE、DWRITE を 1 にしてから  $3\mu\text{sec}$  経過した後に、トリガーが到達した際に波形の読み出しを開始することが出来る。 $3\mu\text{sec}$  待つ理由は不明であるが、DRS4 のサンプルコードに記述があったため真似をしている。

以上で DRS4 の設定部分は終了である。DRS4 の読み出しに関しては読み出しモードの節で説明する。関連 FPGA コードは `drs.control.v` である。制御時の状態遷移図を図 B.3 に示す。

## B.7 ADC インターフェース

ADC には AD9637 というものを使用している。AD9637 はシリアルデータ出力であるため、FPGA 内でパラレル変換する必要がある。今回のファームウェアでは Spartan6 の機能としてすでにある、高速用のシリアル-パラレル変換モジュール (ISERDES と呼ばれている) を使用している。この変換モジュールはリセット信号でリセット状態になり、ロード信号でリセット状態が解除される。基本は自動でシリアル-パラレル変換を行ってくれるのだが、最初から正しくビットがそろって変換されるとは限らない。例えば実装している AD9637 は 12 ビットであるため変換したパラレルデータは最上位ビットが 12 ビット目、最下位ビットが 1 ビット目となるはずである。しかしながら、送られてくるデータはシリアルデータであるため、データだけではどこがビット列の先頭であるか識別できず、正しく変換された時の 10 ビット目が誤って最上位ビットになることも起きる。そのため送られてくるシリアルデータの先頭を識別する機能が必要となる。実際には ADC からシリアルデータに同期したクロックとビット列の先頭を識別するためのフレームクロックも FPGA に送られてくるためこれらを利用する。フレームクロックは 12 ビット長であり上位 6 ビットが 1、下位 6 ビットが 0 と必ずなっていて、フレームクロックとデータは同期している。つまりフレームクロックを正しく変換するように調節できれば、同様の調節で ADC からのシリアルデータもパラレルデータへ変換ができる。調節はシリアル-パラレル変換モジュールの機能としてあるビットをずらす機能を利用している。行われていることは、フレームクロックのパラレルデータを何回かに 1 度確認して、上位 6 ビットが 1、下位 6 ビットが 0 となっているかチェックする。なっていない場合はビットをずらす機能を用いて、1 ビット読み出すタイミングをずらす。再びフレームクロックを確認して正しくなければ、また 1 ビットずらす。以上の作業をフレームクロックが正し

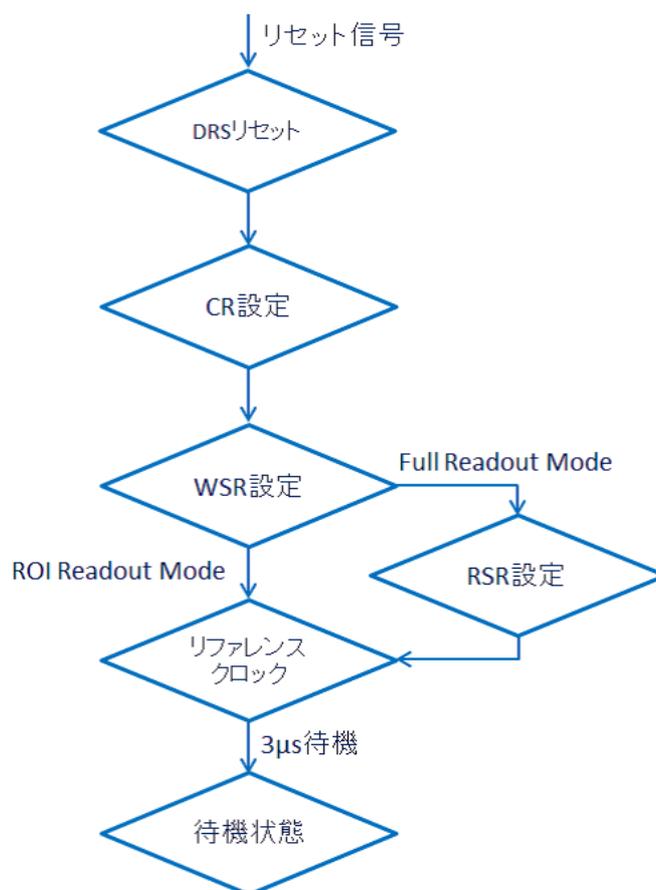


図 B.3 DRS4 設定の状態遷移図。一番下の待機状態とはサンプリングを行っている状態のことを指す。

く読み出せるまで行い、正しく読み出せたら調節を終了する。ビットをずらす作業をシリアルデータにも同様に行っておけば、調節が終了した段階で読み出したパラレルデータも正しく調節されているはずである。実際確認として、ADC の出力を PC から指定して正しく読み出せているか確認を行ったが、どれも正常に読み出すことが出来た。これは本論文の本文中で記載した DRS4 の波形サンプルの一例 (図 3.2) を見ても正しく読み出せていると判断できる。

さらに技術的な話となるが、変換したパラレルデータはいったんメモリに記録され再び読み出される。これはシリアル-パラレル変換を行っているクロックと FPGA 内のシステムクロックのソースが異なるため、ツールのタイミング解析がうまく行われなからである。

関連 FPGA コードは ADC\_SIF\_IF.v、SIF\_IF.v、eras\_fifo.v である。SIF\_IF.v、eras\_fifo.v は ISE のツールで作製したものである。

## B.8 TCP チェーン接続

PC と FPGA 間で TCP 接続が確立されると、イベントデータの転送を行えるようになるが同時にチェーン接続している隣の読み出し回路へも TCP 接続を試みる。特に問題がない限り、チェーン接続されている読み出し回路同士で TCP 接続を確立することが出来るが、何か問題が生じた場合、あるいはそもそも下流側にチェーン接続をしていない場合は接続が行われない。最終的に接続が行われない読み出し回路をチェーン接続の一番端の回路と定義している。一番端の読み出し回路番号はイベントビルダーに使用されるが、TCP 接続を行う際に回路番号を取得する機構をファームウェアにまだ実装出来ていない。現在は 2 枚しかチェーン接続していないため直接 UDP 通信で一番端の回路番号を与えている。関連 FPGA コードは `tcp_request.v` である。

## B.9 読み出しモード

読み出しモードは 4 つある。共通部分を説明してから、1 つずつトリガー固有の部分を説明していく。

### B.9.1 共通部分

#### B.9.1.1 DRS4 読み出し

有効なトリガーが読み出し回路に発生すると、まずは DWRITE が 0 になりサンプリングが停止する。このとき DENABLE は 1 のままであり Domino Ring は動作し続ける。サンプリングを停止させた後は、DRS4 を読み出しモードに切り替える (普段はサンプリングモードである)。モード切り替えを行うと DRS4 出力の差動出力が有効になり出力電圧値が上昇する (設定したコモン電圧値になる)。このモード切替時の電圧上昇が原因で ADC の出力が一時的におかしくなってしまうため、モード切替から読み出しまでに少し時間を置く必要がある。現在は ADC のサンプリングクロック分で 40 周期分あけている。ADC 出力がおかしくなる原因としては AD9637 がシングルエンド入力にうまく対応していなかったためと考えている。40 クロック分時間を置いた後、DRS4 のキャパシタ読み出しを開始する。読み出しは専用のクロックを DRS4 へ送ることで、送ったクロックに同期してデータが ADC へ送られる。現在は Full Readout Mode でも ROI Readout Mode でもキャパシター 1024 個分すべて読み出ししている。読み出しが終わると DWRITE を 1 にしてサンプリングモードに切り替え、3 $\mu$ sec 経過した後に再び読み出しを開始出来る待機状態となる。

#### B.9.1.2 ADC データ操作

DRS4 に読み出し用のクロックが送られるタイミングに合わせてパラレル変換されたデータを記録していく。ADC では入力された信号はすぐにデジタル化されるわけではなく、あるレイテンシののちにシリアルデジタルデータとして出力される。またファームウェア上でもシリアル-パラレ

ル変換やクロック間の受け渡し時でも遅延が起きるため、すべてのレイテンシを考慮してパラレルデータを記録する必要がある。レイテンシは UDP 通信によっても調整出来る。

タイミングを合わせてデータは FPGA 内のメモリへ記録されるが、ゼロサプレス機能が有効なときはヒットがないチャンネルのデータは記録されない。ヒットしたかどうかは ASD のディスクリミネート出力で判断する。現在ゼロサプレス機能は COMET トリガーモードでしか機能していない。1 イベント分のデータの記録が終わると再び待機状態となり、メモリに書き込みができる状態となる。メモリに 6 イベント分たまっているとバッファフルの信号が 1 となり、トリガーを受け付けなくなる。

データが一定分メモリにたまと PC 側へイベントデータ転送を開始する。イベントデータ転送はまずイベントデータパケットのヘッダー (16 バイト) が送られ、ADC データ、最後にイベントデータパケットのフッター (4 バイト) が送られる。1 イベント分送り終わると、次のイベントデータを送り出せる状態になるまで待機状態になる。次のイベントデータは、イベントビルダーからチェーン接続の一番端のイベントデータの送信終了信号が来る、かつ再びデータが一定分メモリにたまることで送信ができる。

関連 FPGA コードは、ADC\_Data\_Read.v、make\_header\_packet.v、ADC\_Read\_Counter\_Module.v、adc\_fifo.v、header\_fifo.v、hit\_ch\_fifo.v である。adc\_fifo.v、header\_fifo.v、hit\_ch\_fifo.v は ISE のツールで作製したものである。DRS4 読み出しと ADC データ操作の状態遷移図を図 B.4 に、ADC データ転送の状態遷移図を図 B.5 にそれぞれ示す。

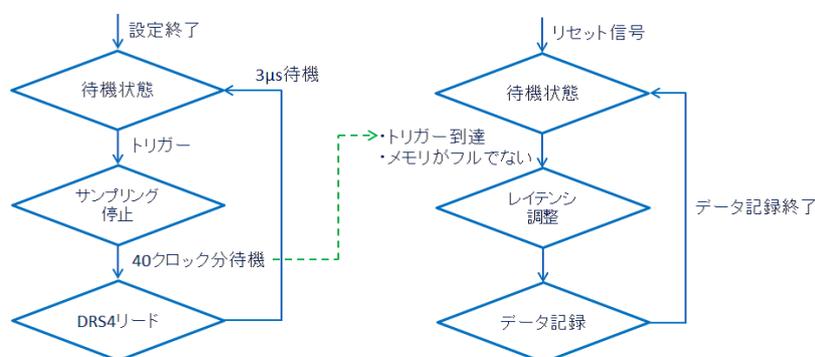


図 B.4 データ読み出し時の ADC、DRS4 の状態遷移図。ADC のトリガーは DRS4 制御で 40 クロック分待機した後に到達する (図中の緑点線)。

## B.9.2 ベースライン取得モード

ベースライン取得モードは図 3.1 で示したパターンコレクションを取得するモードである。DRS4 の開発者によるとパターンコレクションは年に 1 度程度の頻度で十分であるが、現在は開発中であるので定期的にパターンコレクションを取得して波形の解析を行っている。

ベースライン用トリガー生成は FPGA 内のモジュールで行われる。レートは 10Hz 程度である。

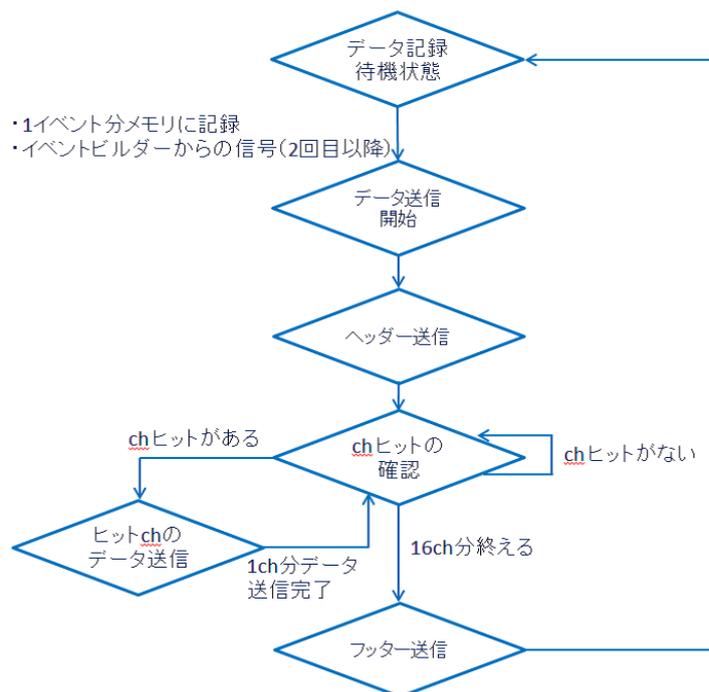


図 B.5 ADC データ転送の状態遷移図。

関連 FPGA コードは baseline\_read.v である。

### B.9.3 ノーマルトリガーモード

読み出し回路に実装されている NIM 入力をトリガーとするモードである。トリガーが入力されるとすぐにサンプリングが停止するため、目的の波形がサンプリングされた後にサンプリングが停止するように、タイミングを読み出し回路外で合わせる必要がある。ノーマルトリガーイメージを図 B.6 に示す。

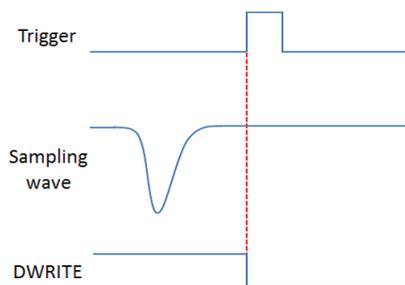


図 B.6 ノーマルトリガーのタイミングイメージ。Roesti プロトタイプでは ASD の波形が 200nsec 程度であるので、波形に対してトリガーは 200nsec~1000nsec 遅延させる。ただし、遅延させすぎても上書きされてしまうので注意が必要となる。

### B.9.4 セルフトリガーモード

ASD からのディスクリミネート出力をトリガーとするモードである。16 チャンネルの OR 論理をとっているため、どれか 1ch でも反応すれば全チャンネルを読み出す。ディスクリミネート出力が FPGA へ到達してすぐに読み出しを開始してしまうと波形をサンプリングする前に読み出してしまうことになるので、FPGA 内で 500nsec 程度遅延させている。

### B.9.5 COMET トリガーモード

COMET トリガーモードとは図 1.5、図 B.7 で示すような陽子ビームタイミングを仮定してのトリガーモードである。外部から陽子ビームタイミングに同期したパルスとカロリメータからのイベントトリガーパルスを読み出し回路に与えることで DRS4 の読み出しを開始する。

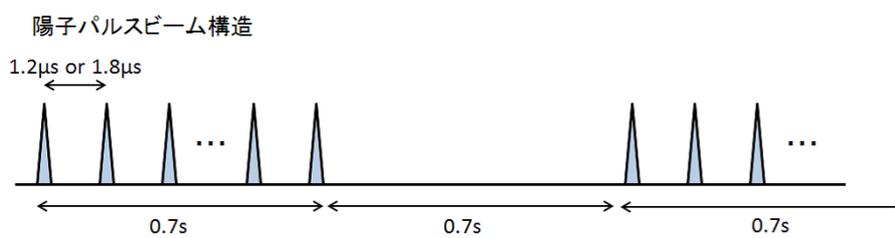


図 B.7 ビームパルスの時間構造。

COMET トリガーモードでは、ファームウェアでは陽子ビーム同期パルスのタイミングで DRS4 を読み出すか判断を行っている。パルス間は 1.2 or 1.8 μsec であるのでイベントトリガーが来ない限り 1.2 or 1.8 μ sec 間隔で常に判断していることになる。イベントトリガーが読み出したいタイムウインドウ (図 1.5 を参照) 内に読み出し回路に送られた場合、直後の陽子ビーム同期パルスのタイミングで読み出しシーケンスを開始する。COMET トリガーモードではゼロサプレッション機能が有効になっているので、ヒットチャンネルのデータも FPGA 内に記録される。記録されたヒット情報をもとにどのチャンネルのデータを書き込むか、あるいは読み出すかを判断する。

ノーマルトリガー、セルフトリガー、COMET トリガーモードの関連 FPGA コードは Trigger\_IF.v である。

最後にトリガーが来た際の動作イメージを図 B.8 に示す。

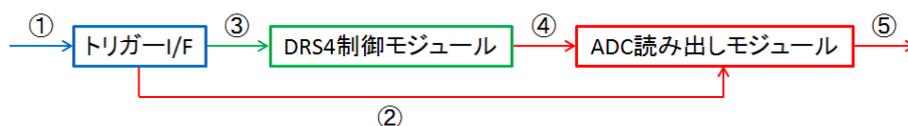


図 B.8 トリガー到達時の動作イメージ図。外部からのトリガーはまずトリガー I/F に送られる (①)。トリガー I/F ではモードによって処理が異なるので詳細は述べないが、どのモードでも ADC 読み出しモジュールにヒットチャンネル情報が送られる (②)。ただし、現時点でのモジュールは COMET トリガーモードでしかゼロサプレス行っていないため、他のモードでは全チャンネルヒットがあったとして処理される。ヒット情報処理後は、DRS4 制御モジュールへ信号を送る (③)。この信号をもとに DRS4 は読み出しを開始する。DRS4 からの読み出しタイミングに合わせて ADC データの記録も開始される (④)。ADC データが一定数書き込まれたら PC 側へ転送を開始する (⑤)。各モジュールの詳細な動作は先述の通りである。

## B.10 イベントビルダー

イベントビルダーとはストローチェンバーからのイベントデータを PC に近い方からボード順に送る機能である。イベントトリガーが来て DRS4 の読み出しを行った後、回路はまず自分のイベントデータを PC 側へ送信する。送り終わったらチェーン接続されているボードのイベントデータの送信を始める。送信を開始するタイミングはデータパケットのヘッダー分 (16 バイト) がメモリに保存されたときである。まずヘッダー分のデータを送る。その時にデータ長、回路番号を読み取っておく。次に読み取ったデータ長分のデータを送る。データ長分送り終わったら、最後にフッター (4 バイト) を送信して 1 パケット分のイベントデータ転送を終える。その後、読み取った回路番号を確認し、チェーン接続の一番端の読み出し回路番号であれば、再び自分のイベントデータの送信を始める。一番端の読み出し回路番号でなければチェーン接続されているボードから送られてくるデータを再び同じ手順で送信する。一番端の読み出し回路であれば、自分のイベントデータを送り続ける。図 2.5 も参照。

以上の工程を繰り返すことで、常に PC 側に近い読み出し回路のイベントデータから順に送信することが出来る。現在の開発状況としては、イベントビルダーの機能を追加することには成功し、転送にも成功している。しかしながら、転送スピードが想定しているよりも非常に遅いため (想定 10 分の 1 程度)、SiTCP の開発者である内田氏に確認していただいている最中である。

関連 FPGA コードは `event_builder3.v`、`event_fifo.v` である。`event_fifo.v` は ISE のツールで作製したものである。

## B.11 イベントデータパケット

イベントデータのフォーマットを図 B.9 に示す。

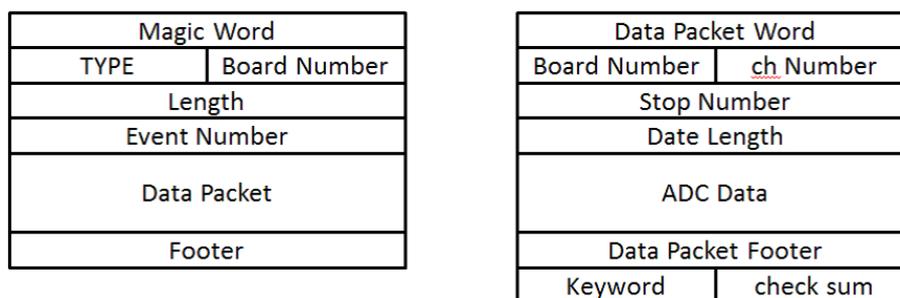


図 B.9 イベントデータの packets フォーマット。左図がイベントデータ packet、右図はイベントデータ packet 内の Data Packet 内の packets フォーマットである。左図のイベント packets フォーマットの幅は 32 ビット、右図の Data Packet の幅は 16 ビットである。Data packet は 1 イベントでのヒットチャンネル分含まれる。

以下、packets フォーマットについて説明していく。

### イベントデータ packets

#### Magic Word

イベント packets の先頭を識別するためのマジックワード。32 ビット。ROESTI プロトタイプ  
のマジックワードは 0x89ABCDEF。

#### TYPE

任意に設定できる箇所。16 ビット。現在は 0xFF0000。

#### Board Number

回路番号。イベントデータ packets がどの読み出し回路から送り出されたものか識別するためのものである。8 ビット。

#### Length

Data Packet の全バイト長。32 ビット。

#### Event Number

イベント番号。32 ビット。

#### Data Packet

ADC データを packets 化したもの。ヒットチャンネル分含まれる。

#### Footer

イベントデータ packets の最後を判別するためのフッター。32 ビット。ROESTI プロトタイプ  
のフッターは 0xFEDCBA98。

### Data Packet

#### Data Packet Word

ADC データをパケット化したもの (Data Packet) の先頭を識別するためのワード。16 ビット。ROESTI プロトタイプでは 0x1234。

**Board Number**

回路番号。イベントデータパケットと同様。8 ビット。

**ch Number**

チャンネル番号。Data Packet がどのチャンネルのものか識別する。8 ビット。

**stop Number**

ROI Readout Mode 時に使用する。サンプリングを止めて読み出しを開始するキャパシタの番号。16 ビット。上位 6 ビットは全て 1。下位 10 ビットがキャパシタ番号となる。

**Data Length**

ADC Data のバイト長。16 ビット。現在は読み出すキャパシタ数および 1 サンプルポイント分のビット幅が固定であるので、Data Length も 2048 と固定長である。

**ADC Data**

ADC からの変換されたパラレルデータ。16 ビット。上位 4 ビットはすべて 0。下位 12 ビットが ADC パラレルデータとなる。

**Data Packet Footer**

ADC データをパケット化したもの (Data Packet) の ADC Data の最後を識別するためのフッター。16 ビット。ROESTI プロトタイプでは 0x5678。

**KeyWord**

任意に設定できる箇所。8 ビット。現在は 0xCC。

**check sum**

Data Packet 内のデータを 8 ビットずつ全て Data Packet Word から順番に AND 論理をとっていき、最後の数値を check sum に入れる。現在は機能していないため 0x00 の固定値となっている。

## 付録 C

# ソフトウェア

コンフィギュレーション用の Python プログラムの使用方法については付録 B で説明したが、付録 C ではその他の UDP、TCP 通信をするためのソフトウェアを説明する。これらはすべて C 言語で記述されている。

### C.1 UDP 通信

アドレスマップにアクセスするには UDP 通信を使用する。今回の評価で用いたプログラムは KEK の内田氏がネットで配布している、UDP 通信用のテストプログラムである。使用の詳細は参考文献 [14] に記載されているが、この章でも簡単に説明する。PC 環境は Linux である。

#### C.1.1 プログラム

実行プログラムは以下の 5 つをコンパイルして作る。トップコードは `rbcp.c` である。

`rbcp.c`(トップコード)

`rbcp.h`

`rbcp-com.c`

`myAtoi.c`

`myScanf.c`

#### C.1.2 使用方法

プログラムの起動はコンパイルした `rbcp.c` の実行ファイルより行う。コマンドライン上より

```
./<実行ファイル名> <IP アドレス> <UDP ポート番号>
```

を打ち込むと RBCP のコマンドラインが起動する。例えば実行ファイル名が `rbcp`、IP アドレスが `192.168.10.15`、UDP ポート番号が `4660` としたときは

```
./rbcp 192.168.10.15 4660
```

と打ち込む。実際の様子を図 C.1 に示す。

FPGA との通信は以下の通りとなる。先頭に 0x がついているものは 16 進数表記である。説明で出てくるアドレスは付録 B で説明したアドレスマップと同じものである。

**wrb** 〈アドレス〉 〈書き込みデータ (1 バイト)〉

書き込みたいデータ (1 バイト) をアドレスに書き込む。

例 : wrb 0x10 0xF0     アドレス 0x10 に 0xF0(1 バイト) を書き込む。

**wrs** 〈アドレス〉 〈書き込みデータ (2 バイト)〉

書き込みたいデータ (2 バイト) をアドレスに書き込む。書き込み順はビッグエンディアン。

例 : wrs 0x15 0x0DAC     アドレス 0x15 より 0x0DAC(2 バイト) を書き込む。この場合、0x15 に 0x0D、0x16 に 0xAC が書き込まれる。

**wrw** 〈アドレス〉 〈書き込みデータ (4 バイト)〉

書き込みたいデータ (4 バイト) をアドレスに書き込む。書き込み順はビッグエンディアン。

例 : wrw 0x0C 1000     アドレス 0x0C より 1000 を書き込む。1000 には 0x がついていないため 10 進数となり、4 バイトとして扱われる。

**rd** 〈アドレス〉 〈読み込みデータ数〉

アドレスから読み込みデータ数分のバイトを読み出す。読み出し順はビッグエンディアン。

例 : rd 0x00 16     アドレス 0x00 より 16 バイト分読み出す。

**load** 〈テキストファイル〉

スクリプトを読み込む。

例 : テキストファイルに

```
wrb 0x10 0xF0
```

```
rd 0x00 16
```

と書いておき、load を行うと上から順に実行される。

**help**

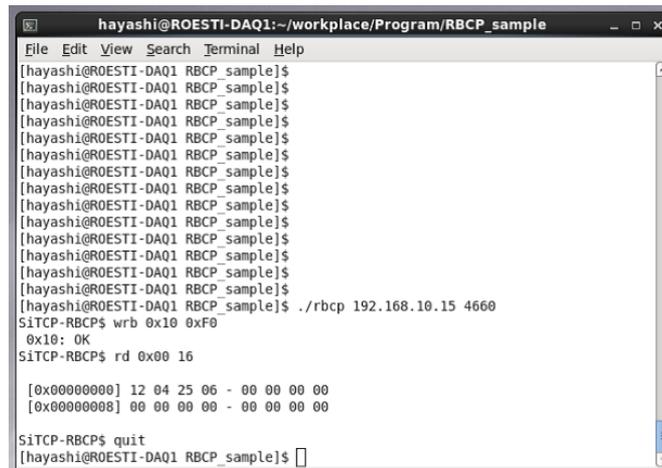
使用可能なコマンドの表示。

**quit**

RBCP コマンドラインの終了。

## C.2 TCP 通信

TCP 通信はイベントデータ転送に用いられる。TCP 通信は接続を確立してから転送を行うため、信頼のある高速通信を行うことが出来、イベントデータ転送に適している。

A terminal window titled 'hayashi@ROESTI-DAQ1:~/workplace/Program/RBCP\_sample'. The window shows a series of shell prompts '[hayashi@ROESTI-DAQ1 RBCP\_sample]\$' followed by several empty lines. The 15th prompt is followed by the command './rbc 192.168.10.15 4660'. The output shows 'SiTCP-RBCP\$ wrb 0x10 0xF0', '0x10: OK', and 'SiTCP-RBCP\$ rd 0x00 16'. Below this, two lines of hex data are shown: '[0x00000000] 12 04 25 06 - 00 00 00 00' and '[0x00000008] 00 00 00 00 - 00 00 00 00'. The prompt then changes to 'SiTCP-RBCP\$ quit' and finally back to '[hayashi@ROESTI-DAQ1 RBCP\_sample]\$' with a cursor.

```
hayashi@ROESTI-DAQ1:~/workplace/Program/RBCP_sample
File Edit View Search Terminal Help
[hayashi@ROESTI-DAQ1 RBCP_sample]$
[hayashi@ROESTI-DAQ1 RBCP_sample]$ ./rbc 192.168.10.15 4660
SiTCP-RBCP$ wrb 0x10 0xF0
0x10: OK
SiTCP-RBCP$ rd 0x00 16

[0x00000000] 12 04 25 06 - 00 00 00 00
[0x00000008] 00 00 00 00 - 00 00 00 00

SiTCP-RBCP$ quit
[hayashi@ROESTI-DAQ1 RBCP_sample]$
```

図 C.1 RBCP 操作の一例。

## C.2.1 プログラム

実行プログラムは以下の 2 つをコンパイルする。トップコードは TestDAQ.c である。

TestDAQ.c

TestDAQ.h

## C.2.2 使用方法

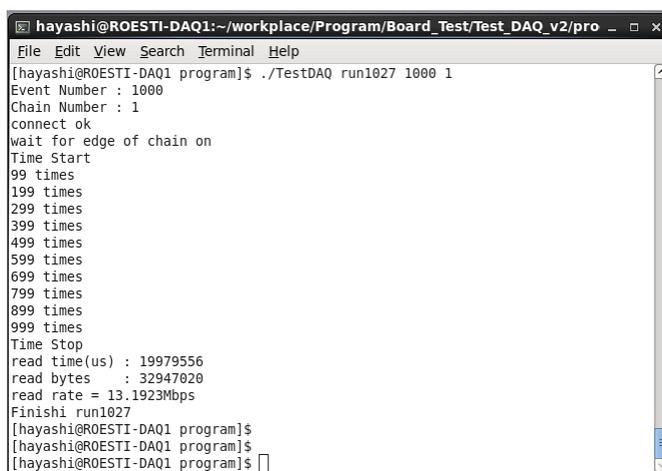
コマンドライン上に

```
./<実行ファイル名> <ラン番号> <イベント数> <読み出し回路数>
```

と打ち込む。例えば実行ファイル名が TestDAQ、ランナンバーが 1027、イベント数が 1000、読み出したい回路の数が 2 つであった場合は、

```
./TestDAQ run1027 1000 2
```

となる。実際の様子を図 C.2 に示す。



```
hayashi@ROESTI-DAQ1:~/workplace/Program/Board_Test/Test_DAQ_v2/pro
File Edit View Search Terminal Help
[hayashi@ROESTI-DAQ1 program]$ ./TestDAQ run1027 1000 1
Event Number : 1000
Chain Number : 1
connect ok
wait for edge of chain on
Time Start
99 times
199 times
299 times
399 times
499 times
599 times
699 times
799 times
899 times
999 times
Time Stop
read time(us) : 19979556
read bytes   : 32947020
read rate = 13.1923Mbps
Finish run1027
[hayashi@ROESTI-DAQ1 program]$
[hayashi@ROESTI-DAQ1 program]$
[hayashi@ROESTI-DAQ1 program]$
```

図 C.2 イベントデータ取得の一例。読み出しが終了すると最後に転送レートが計算される。

## 付録 D

# 今後の開発に関して

今回製作したプロトタイプをもとに、今後製作する時のために改善点などを述べる。

### 検出器接続コネクタ

読み出し回路で使用しているコネクタは 1.27mm ピッチのものを使用している。正確な評価ではないがクロストークの影響はこのコネクタでもでない想定しているが、1 読み出し回路当たり 16 チャンネルであるので無理にピッチ幅の狭いものを使用する必要もない。したがって、次回は 2.54 ピッチのコネクタでも十分であると考えられる。

コネクタに関して、時間校正用のクロックを与えるポートも追加する必要がある。このポートを検出器接続のコネクタと一緒にするのか、別にするのかは次の製作時には決定しておくべきである。

検出器用のコネクタではないが、トリガー用のコネクタも詰めていかなければならない。現在はトリガー用に用いている NIM コネクタが 2 つ FPGA に接続されている。型は MCX を用いている。LEMO よりも流通などがあり若干安く、ノイズも LEMO よりは乗り難いと言われているため、今回は採用した。しかしながら、トリガーがどのような形式で何本のラインで到達するかがまだ決定されていないので、トリガー形式の決定に合わせてコネクタも変える必要がある。

### ADC

時間校正用クロックも DRS4 から ADC で読み出すのだが、その場合チャンネル数が 1 チップにつき 1 チャンネル増えることになる。現在実装している ADC は 8 チャンネルであるため、ADC をもう一つ増やすかあるいは 9 チャンネルの ADC を用いるかどちらかになる。ADC を増やすと問題になる点はクロックである。ADC からのデータは ADC からのクロックに同期して FPGA に送られてくるが、FPGA のクロックを受けることが出来る場所の関係で配線が難しくなる。実装する場所もあまりないので、読み出し回路の大きさ、基板の層数、FPGA のピン配置など業者に任せるところも多いが、考えるべき点が多くなる。9 チャンネルの ADC の問題点は、まずそのようなものが存在するかである。自分では詳しく確認していないが、あまり見かけない。あったとしても、要求を満たしているのか (特に消費電力) も確認しなければならない。

以上 2 つの方針を示したが、正しく時間分解能を求める際には不可欠な事となるので次の製作次時に決定する必要がある。

## FPGA

FPGA は Spartan-6 と呼ばれる、Xilinx 製の FPGA を使用している。これは 1 世前のものであり、現在は種類はまだ少ないが 7 シリーズが販売されている。読み出し回路を設計している当時は、7 シリーズの FPGA より Spartan-6 の方が安く、機能も十分であったため Spartan-6 を選択した。しかしながら、本論文を書いている現在は、比較的 7 シリーズの FPGA も安く販売されている。放射性耐性は 7 シリーズのほうがよいため量産のことも考慮に入れて FPGA の選択が必要となる。

## 光モジュール

ファームウェアが原因かハードウェア自体に問題があるのかわかっていないが、光ケーブルでうまく PC と接続が出来ない事例がしばしば発生した。現在は光モジュールによって接続ができたりできなかつたりという状態である。また、電源を入れた後は一度光ケーブルを抜き差ししないと PC と接続が出来ないという事象も発生している。メディアコンバーターを使用する場合には問題なく通信できるので、PC 側の問題も考えられる。

いずれにせよ、PC と読み出し回路間の接続完了までに問題がまだあり、ファームウェア、ハードウェア、PC のどの部分に問題があるのか分かっていないため、今後詳しく調査する必要がある。

## ファームウェア

光モジュールの項目でもあげたが、ファームウェアの改善も必要となる。現在解決すべき問題は、SiTCP 関連とトリガーインターフェース、エラー処理である。SiTCP 関連では大きな問題としてチェーン接続があげられるが、これは内田氏に依頼中である。他に細かい点として、SiTCP を駆動するクロックも考える必要がある。SiTCP を動作させているクロックは 133.333MHz であるが、内田氏によると最大スペックは 150MHz 程度は与えないと出ないとのことである。なぜ与えていないかということ、DRS4 の仕様にも合わせたいため 66.667MHz のクロックも作り出していることが理由である。クロック間の信号の受け渡しの際に受け渡しをする 2 つのクロックのソース源が同じ場合、クロック同士の最少公倍数で FPGA 内の遅延は計算される。そのため 66.667MHz と 150MHz で受け渡しをすると 600MHz で遅延計算されることとなる。これは 1.67ns の遅延のみ認められるということであらわしており、FPGA 内の遅延計算としては非常に厳しい制約となる。よって、うまく 150MHz 程度で FPGA 内の駆動を合わせる必要がある。トリガーに関しては、本文中にも記述したとおり、COMET トリガーモードでは高レートになると受け付けなくなるので改良が必要となる。また現在のファームウェアではヒットはないがトリガーが来た場合データのないイベントパケットのみ送るということが出来ていない。イベントビルダーの機能も考えるとデータのないイベントパケットを送る必要があるので、改良が必要となる。イベントビルダーでは、回路番号と一番端の回路番号を自動的に取得する機能もまだ実装できていない。エラー処理

に関しては、例えば TCP 通信が途中で途切れた時、ADC データのシリアル-パラレル変換に問題が乗じた時、エラービットをどこに立てるかなどまだ決めていないことが多い。ADC の Data Packet 内の check sum 機能も未実装であるので、エラー処理の開発も必要となる。ADC 関係では、空イベントパケットの転送についても考えなければならない。現在は FPGA 内のメモリに 1 イベント分の ADC データが記録されていれば転送を開始するという構造になっている。そのため、ヒットが無い読み出し回路では ADC データが空のイベントパケットを送ることが出来ない。これはイベントビルダーがうまく動作しないことにつながるので、トリガーは到達したがチャンネルヒットがないときの処理もファームウェアに加える必要がある。

トリガーに関しては本論文で紹介した ROESTI プロトタイプとは別にトリガー回路を作るようであるので、それに向けて開発することも今後考える。なお、データ圧縮、Rebinning なども必要であるが、上記のことを達成した後でも十分である。

## 参考文献

- [1] *Conceptual Design Report for COMET*. 2009,2010.
- [2] K. Abe. First muon-neutrino disappearance study with an off-axis beam. *Phys. Rev. D* 85, 031103(R), 2012.
- [3] W. Bertl et al. *Eur. Phys. J. C* 47, 337-346(2006).
- [4] Y. Ashie. Evidence for an oscillatory signature in atmospheric neutrino oscillations. *PHYSICAL REVIEW LETTERS VOLUME 93, NUMBER 10*, 2004.
- [5] S. H. Neddermeyer and C.D.Anderson. *Phys. Rev.* 51, 884, 1937.
- [6] PSI. *9 Channel, 5 GSPS Switched Capacitor Array DRS4*. 2009.
- [7] Stefan Ritt. Development of high speed waveform sampling asics. 2010.
- [8] M. H. Ahn et al The K2K collaboration. Measurement of neutrino oscillation by the k2k experiment. *Phys. Rev. D* 74, 072003, 2006.
- [9] Tomohisa Uchida. Hardware-based tcp processor for gigabit ethernet. *IEEE TRANSACTION ON NUCLEAR SCIENCE, VOL.55,NO.3*, 2008.
- [10] 吉岡良雄. 待ち行列と確率分布 情報システム解析への応用. 2004.
- [11] 西口創. Proton beam extinction measurement. 2012.
- [12] 仲井裕紀. ミューオン電子転換現象探索実験 COMET のための Straw Gas Chamber 研究開発. 2012.
- [13] 島崎昇一. ドリフトチェンバー用 ASD チップ開発とその性能. 2012.
- [14] 内田智久. Sitcp 説明書. 2012.