

2018 年度 卒業論文

RHIC-sPHENIX 実験における中間飛跡
検出器 INTT 用シリコンセンサーの性
能評価とテストベンチ構築

奈良女子大学 理学部
数物科学科物理学コース 高エネルギー物理学研究室

一色萌衣 呉羽広子 杉野和音

2019 年 3 月 31 日

概要

現在、奈良女子大学は、アメリカニューヨーク州のブルックヘブン国立研究所で行われている大型の重イオン衝突実験を目的とした RHIC 加速器を用いた PHENIX 検出器に導入予定の飛跡検出器の 1 つ、INTT 検出器の開発に参加している。INTT グループには、理化学研究所、立教大学、奈良女子大学、海外の研究機関等、複数の研究機関が参加している。2018 年後期から、奈良女子大学高エネルギー物理学研究室において、シリコンセンサーの性能評価を目的としたテストベンチの導入と設置が始まった。今年度は、奈良女子大学から 4 名の学生が参加し、協力して研究を進めた。無事、INTT のテストベンチの環境構築に成功し、その後 INTT 用シリコンセンサー、及び読み出し回路に対する性能評価を行なった。私たちが測定したデータとともに、性能評価方法について報告する。sPHENIX 実験の先行実験である PHENIX 実験で使用されていた前方後方向飛跡検出器である FVTX で使用されていたシリコンセンサーを用いて、測定結果の再現性を確認し、あらゆる条件で試験したのち、実際に INTT 用シリコンセンサーを用いて実験、解析へ移っていった。測定方法は大きく分けて、キャリブレーションパルスによるノイズ評価、セルフトリガーモードを利用した β 線源による性能評価、外部トリガーモードを利用した宇宙線による性能評価を行い、シリコンセンサーの評価方法の探索を行なった。そしてノイズの影響の有無を、取得したデータを解析し見極めた。定期的に INTT グループ学生ミーティングで報告し、助言をいただきながら実験と改良を進めていった。2023 年の sPHENIX 実験の Run1 に向けての奈良女子大学におけるこの 1 年間の作業内容と成果、進捗状況とともに、今回の研究で浮かび上がった今後の課題について報告する。

目次

第 1 章	序論	1
1.1	素粒子物理学	1
1.2	研究背景	3
第 2 章	INTT(中間飛跡検出器)	7
2.1	sPHENIX における飛跡検出器群	7
2.2	INTT の役割	8
第 3 章	INTT テストベンチの環境構築	15
3.1	INTT テストベンチ構築のための初期設定と環境設定	15
第 4 章	ノイズ評価	20
4.1	キャリブレーションテスト	20
第 5 章	β 線源による INTT 性能評価	28
5.1	β 線源による性能評価概要	28
5.2	セットアップ	29
5.3	シリコンセンサーにおけるエネルギー損失	31
5.4	結果	35
第 6 章	外部トリガーモードを用いた宇宙線測定	38
6.1	宇宙線測定の目的	38
6.2	外部トリガーモード回路とセットアップ	38
6.3	宇宙線測定条件	42
6.4	宇宙線測定結果	45
6.5	シンチレーションカウンタ 2 台を用いたコインシデンス外部トリガーを用いた宇宙線測定	46
6.6	710mV 以下の低い電圧値における DAC 分布の解析	48
第 7 章	イベント判別と解析方法の検討	51
7.1	宇宙線測定におけるのクラスター判別	51

7.2	宇宙線測定における今後の課題	56
第 8 章	結論	58
	謝辞	59
	参考文献	60
付録 A	FPGA トレーニング	62
	FPGA 操作のためのトレーニング内容とその方法	62
A.1	INTT における FPGA の活用状況	62
A.2	FPGA 活用トレーニング	62
A.3	FPGA コード	67

目次

1.1	E>1GeV での大気圏中での宇宙線の垂直方向のフラックス [10]	3
1.2	RHIC 加速器	4
1.3	PHENIX 実験におけるシリコンセンサー (FVTX)	5
1.4	sPHENIX 測定器	6
2.1	sPHENIX 飛跡検出器	8
2.2	INTT 用シリコンセンサー：L2 モジュール	9
2.3	ROC	10
2.4	左:FEM-IB 右:FEM 本体	11
2.5	単位厚さあたりのシリコン中でのミューオン運動エネルギーとエネルギー損失の 関係 [6]	12
2.6	500MeV のパイオンのシリコン中での Stragglng 関数の変動 [6]	13
2.7	ストリップ数別最大エネルギー損失の見積もり方法概念図	14
3.1	NIM モジュールのセットアップ写真	16
3.2	奈良女子大学に設置したテストベンチ全体写真	16
3.3	INTT 読み出し回路ブロック図	17
3.4	キャリブレーションテストにおけるテストベンチブロック図	17
3.5	DAQ 操作のための GUI (Nevis_gui のウィンドウ写真と操作箇所)	18
4.1	1 chip キャリブレーションテスト結果	21
4.2	26chip キャリブレーションテスト結果	21
4.3	1ch の応答	23
4.4	Fit 関数から σ の算出	24
4.5	全 channel の σ 分布	25
4.6	レートの考え方	25
4.7	1 チップのレート	26
4.8	26 チップの 1 チップ毎のレート	27
4.9	26 チップ分のレートヒストグラム	27

5.1	実際に使用した β 線源	29
5.2	セルフトリガーモードでのブロック図	29
5.3	β 線測定時のシリコンセンサーのセットアップ	30
5.4	センサー付近のセットアップ図解	31
5.5	シリコンを通過する際のミュオンのエネルギー損失	32
5.6	エネルギー損失の分布図	32
5.7	シリコンの厚さごとの運動量と MIP	33
5.8	1 ストリップにおける見積もり値	34
5.9	閾値 (DAC 値) の設定	34
5.10	2 分間測定した際のセットアップ	35
5.11	横軸を $chip_d$ とした際の 2 次元ヒストグラム	36
5.12	横軸を DAC とした際の 2 次元ヒストグラム	36
6.1	外部トリガーモードブロック図	39
6.2	オシロスコープでのタイミング操作	40
6.3	外部トリガー宇宙線測定時のセットアップ図	42
6.4	外部トリガーシリコンセンサーとシンチレーションカウンタセットアップ	43
6.5	1 台のシンチレータ を用いた宇宙線測定の chip ごとのエントリーの分布	45
6.6	宇宙線測定時の chip に対するエントリー分布	46
6.7	全 26chip に対する channel のエントリー分布	46
6.8	全 26chip に対する ADC 分布	46
6.9	2 台のシンチレーションカウンタのコインシデンス信号外部トリガーブロック図	47
6.10	コインシデンス波形論理	47
6.11	シンチレーションカウンタとシリコンセンサーの位置関係	47
6.12	2 台のシンチレーションカウンタ宇宙線測定時の chip に対するエントリー分布	48
6.13	DAC 値を低い方で線形的に設定した宇宙線測定時の chip に対するエントリー分布	49
6.14	DAC 値を 710mV 以下でリニアな間隔で設定した宇宙線測定時の ADC 分布	50
7.1	イベント判別方法	52
7.3	イベントごとのストリップ通過数	53
7.2	通過ストリップ数別波高分布	53
7.4	DAC0=10 の時の chip 分布	54
7.5	DAC0=10 の時の ADC 分布	55
7.6	x 軸 : chip y 軸 : channel の hitmap	55
7.7	hotchannel を除いたイベントごとのストリップ通過数	56
A.1	EDX-007 基盤 [12]	64
A.2	LED 基盤回路図 [13]	65

A.3	7SegmentLED 回路図 [13]	65
A.4	LED・BUZZER 回路図 [13]	66
A.5	7segment 点灯パターン 1	67
A.6	7segment 点灯パターン 2	67

表目次

3.1.1	使用した NIM モジュールとその機能	15
4.1.1	閾値設定	20
6.2.1	シンチレーションカウンタの仕様と性能 [9]	39
6.3.1	ストリップ通過数別エネルギー損失見積もり値	43
6.3.2	宇宙線測定における FPHX チップ 3 bitDAC 設定値	44
6.6.1	宇宙線測定における FPHX チップ 3 bitDAC のリニアな設定値	49
7.1.1	通過ストリップ数ごとの期待されるエネルギー損失に対応する FPHX チッププリ アンプ増幅電圧値	52
A.2.1	EDX-007 仕様	64

第 1 章

序論

1.1 素粒子物理学

1.1.1 素粒子の標準模型

素粒子物理学は宇宙線における新しい粒子の発見に伴って 20 世紀前半に生まれた。物質の基本的な構成要素である素粒子は、現在の物理学ではそれ以上分解できない粒子とされている。素粒子には物質を構成する粒子として、スピン $1/2$ のフェルミ粒子と、整数スピンをもつボーズ粒子が存在する。特にフェルミ粒子は、レプトンとクォークの 2 種類あり、レプトンが、電子、ミューオン、ニュートリノの 3 種類とその逆の電荷を持つ対も数えて全 6 種。クォークは、それぞれ電荷が正負それぞれ逆をもつ対の関係で、トップとボトム、アップとダウン、チャームとストレンジの全 6 種類である。それぞれ持ちうる電荷や、色荷が異なり、それらの組み合わせにより、様々な素粒子に分類される。また他にグルーオンと呼ばれるこれらの素粒子を弱い力でつなぎとめる媒介粒子が存在している。

1.1.2 相互作用

標準模型において定義される粒子間での基本的な相互作用は大きく分けて 4 つある。力の大きい順に述べる。

1 つ目が強い相互作用であり、中性子や陽子中にクォークを束縛し、また原子核の中に中性子や陽子を束縛する力である。クォーク間の力は質量を持たない粒子、グルーオンによって仲介されている。2 つ目が電磁相互作用である。これは、光子を媒介して行われる相互作用であり、原子核と原子殻の電子間や、分子間力で仮想的に行われる。3 つ目が、弱い相互作用である。これは、ゆっくりとした原子中での β 崩壊による過程などで働き、ボソン粒子の一種である Z^0 や Z^{\pm} で媒介される。4 つ目が、重力による相互作用であり、媒介粒子は未だ発見されていない、未知の粒子であるグラビトンである。この力は高エネルギー物理学のような取り扱うスケールが $1(fm)$ といったような小さな領域ではほとんど考慮する必要はない力とされている。4 つの相互作用は 1970 年代に形式化された、いわゆる素粒子の標準模型とそれらの相互作用によって説明されている。これら

の素粒子の性質や振る舞いを実験的に研究する学問が高エネルギー物理学である。

1.1.3 クォーク・グルーオンプラズマ

クォーク・グルーオンプラズマ (QGP) は、高温、高密度な状態で核子やハドロンが重なり合って、格子内に閉じ込められているクォークとグルーオンの結合が切れ、ばらばらになつたプラズマ状態を指す。サラサラとした粘性の低い物質で液体のような性質を持っている。QGP は高温、高密度下であったとされる宇宙初期のビックバンから数 μ 秒後に実現していたと考えられ、QGP 現象の理解は初期宇宙発展の解明に繋がる大きな役割を担っている。QGP は、宇宙空間では中性子星内部のような極めて高压で高エネルギーが加わった環境で実現しているとされている。現在、人工的に、大型加速器実験における原子核衝突実験で QGP を生成できることが確認されている。史上初めて、QGP の存在が実験的に確認されたのが RHIC 加速器による重イオン衝突実験であった。現在もその性質解明や、新物理の探求に多くの研究者が参加している大変注目度の高い物理である。

1.1.4 宇宙線

宇宙線とは宇宙空間を高エネルギーで飛び回る極めて小さな粒子のことである。これらの宇宙線は多量に地球に降り注ぎ、大気中の原子核と反応し中間子を生成する。生成された中間子は高速で周りの原子核に衝突することで粒子の数を増やしながらかエネルギーを落としていく。寿命の短い粒子はすぐに崩壊し、エネルギーの低い μ 粒子 (ミューオン) や γ 線となって地上に降り注ぐ。崩壊過程の観測により 1932 年に陽電子が発見され、それに続き π 中間子 (パイオン) と μ 粒子 (ミューオン) が発見された。荷電パイオンは $\pi \rightarrow \mu + \nu_{\mu}$ と崩壊し、ミューオンを生成する。パイオンの平均寿命は短く大気中ではほぼ全てが崩壊する。一方でミューオンは $\mu^{+} \rightarrow e^{+} + \nu_e + \bar{\nu}_{\mu}$ の崩壊を起こすが、平均寿命が 2200(ns) と他の素粒子に比べ長く、2 ~ 3GeV のエネルギーがあればミューオンのまま地表へと降り注ぐ。地表に降り立つミューオンの流量は約 $10^{-2} cm^{-2} s^{-1}$ である。本論文では、宇宙線のエネルギー損失の計算においては、地上でのミューオンの運動エネルギーを用いて見積もり、計算している。

宇宙線には 1 次宇宙線、2 次宇宙線の二種類がある。1 次宇宙線は宇宙空間で、物理的現象、例えば爆発、星同士の衝突、太陽フレアなどが生じたときのエネルギーにより加速された原子核のことである。2 次宇宙線は星間物質と相互作用した粒子のことをいう。ミューオンが大気圏中を通過し地上へ到達するには、地球の大気圏中の粒子との相互作用で損失する以上のエネルギーが必要である。また低エネルギーの宇宙線 (約 10 GeV 以下) は地球の磁気に影響される。よって、宇宙線は、そのロケーションと、粒子の寿命に左右されると言える。

1.1.5 大気中での宇宙線

大気中での宇宙線は荷電メソンが崩壊したのちに生成されたミュオンとニュートリノ、もしくは、中性メソンが崩壊したのち生成された電子と光子である。ただし、地上で観測される宇宙線のほとんどが寿命が比較的長いミュオンである。このミュオンは高度の高い大気圏 (15km) で生成されたもので、生成後地上に届くまでにイオン化し、2GeV の運動エネルギーを損失する。地上で観測されるミュオンの平均エネルギーはおおよそ 4(GeV) であり、エネルギースペクトラムは 1(GeV) とほぼ平坦である。

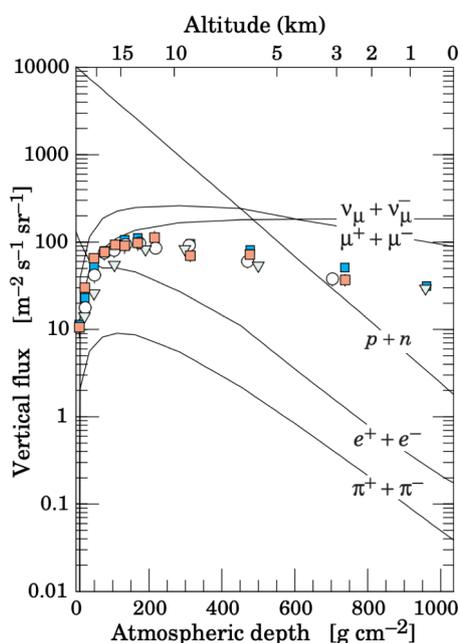


図 1.1 $E > 1\text{GeV}$ での大気圏中での宇宙線の垂直方向のフラックス [10]

図 (1.1) から、垂直方向の宇宙線の流束の、 $\pi_+\pi_-$ と、 e_+e_- の数と大気圏の進行距離の間には負の相関がある。よって、 $\pi_+\pi_-$ と、 e_+e_- の数は大気圏中を進行するにつれて減少する。対して、 $\mu_+\mu_-$ と $\nu_+\nu_-$ の数は進行距離にほぼ影響を受けないことがわかる。

1.2 研究背景

1.2.1 RHIC 加速器

ブルックヘブン国立研究所 (BNL) は、2 粒子衝突型加速器である RHIC 加速器を用いて素粒子物理学を研究している研究所である。QGP を生成しその性質を研究することを目的に建設され

た世界で初めての重イオン衝突加速器であり、RHIC は 2000 年から実際に稼働を開始した。主に金原子核同士の衝突による観測を行っている。他に、ヘリウム原子核や陽子といった様々な原子核衝突実験を実行しており、特に金原子核同士の衝突実験では核子 1 個当たりの重心系エネルギーが最大で 200GeV、陽子・陽子衝突では 510GeV まで実現可能である。RHIC 加速器の全景写真を図 (1.2) で示す。写真に見られるように RHIC には 2 つの環状のビームラインが建設されており、それぞれ時計回りのブルーリングと反時計回りのイエローリングと呼ばれる周長 3.8km の加速リングである。106ns(9.4MHz) の間隔でリングが交差しており両方向に走るビームが衝突する。ビームはバンチ構造を持っており、各リングに 120 のバンチが蓄積されている。また、衝突点は合計 6 箇所設けられ、そのうちの 2 箇所において PHENIX(Pioneering High Energy Nuclear and Ion Experiment) と STAR(Solenoid Tracker at RHIC) の 2 つの実験が稼働していたが、2016 年に PHENIX 実験は終了した。

1.2.2 PHENIX 実験

PHENIX 実験は 2000 年から 2016 年までの 16 年間、RHIC で行われていた高エネルギー重イオン衝突実験の一つで、世界 15 カ国 70 以上の研究機関から約 500 名の研究者が参加していた国際共同実験である。PHENIX 実験では QGP(Quark Gluon Plasma) の生成とその性質の研究の他、陽子スピンの構成要素の解明を目的とした研究が行われていた。BNL では RHIC 加速器を用いて原子核を光速近くまで加速し衝突させることで超高温、高密度下の状態を実現し QGP の再現と検証を行った。PHENIX 実験は観測の結果から QGP 自身の性質を明らかにしていき、高エネルギーでの原子核衝突において QGP が生成できることを確立した。

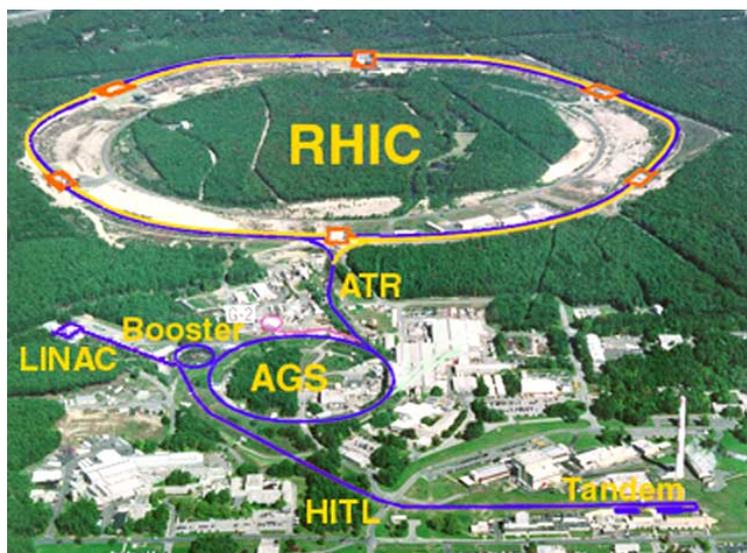


図 1.2 RHIC 加速器

1.2.3 PHENIX 実験におけるシリコンセンサー (FVTX)

また、PHENIX 実験において導入されていたシリコンセンサーは、FVTX(Forward Silicon Vertex Detector) [11] である。バレルのキャップ部分に位置していた関係で、FVTX の形は扇型のようになっており、これらが複数枚円形に並べられ、全方向を網羅し前後方向のすべての方位角方向で、粒子を検出することができる。シリコンの厚さは $320\mu\text{m}$ 、ストリップピッチは $75\mu\text{m}$ で、扇型をしているため、センサーの内側半径は 3.4mm 、センサーの外半径は、 11.57mm と外に向かって広がっていく。検出器の裏面はアルミでメッキしてあり、この裏面のメッキ部分で印加電圧値は決まる。ストリップは接地してある。



図 1.3 PHENIX 実験におけるシリコンセンサー (FVTX)

1.2.4 sPHENIX 実験

sPHENIX 実験はアメリカのニューヨーク州ロングアイランドにあるブルックヘブン国立研究所 (BNL : Brookhaven National Laboratory) の、相対論的重イオン衝突型加速器 (RHIC : Relativistic Heavy Ion Collider) を用いて行われる新たな実験である。この実験は、2023 年から稼働予定であり、2000 年から 2016 年まで同地で行われていた PHENIX 実験を高度化した次期実験にあたる。世界各国の研究者が参加し、従来の PHENIX 検出器では測定することのできなかったハドロジェットや Υ 中間子の測定が計画されており、これらによる QGP のエネルギー密度や色荷識別波長について解明することを目指している。sPHENIX 測定器を以下の図に示す。sPHENIX 検出器群における飛跡検出器群は MVTX、INTT、TPC の 3 つで構成され、詳細は第 2 章で記述している。

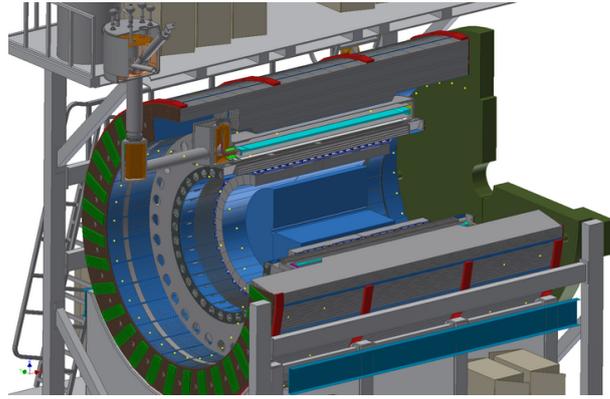


图 1.4 sPHENIX 测定器

第 2 章

INTT(中間飛跡検出器)

2.1 sPHENIX における飛跡検出器群

sPHENIX 実験において導入される、粒子飛跡検出を目的とする検出器は全 3 つ存在する。

2.1.1 MVTX(Monolithic-Active-Pixel-Sensor-based Vertex Detector)

MVTX は、この 3 つの飛跡検出器のうち最も内側に位置し、MAPS(Monolithic Active Pixel Sensor) を用いた $30\mu\text{m}$ のピッチの半導体ピクセルからなる三層構造の半導体検出器である。方位角方向に対しては 2π 、ラピデティ方向に対しては $|\eta| \leq 1$ 、衝突中心からビーム軸方向に $\pm 10\text{m}$ の範囲を検出する。高精度の飛跡検出が可能であり、衝突点と生成された粒子の最近接距離測定を用いて重クォーク特に b イオンクォークの検出を行う。

2.1.2 INTT(INTermediate Tracker) 中間飛跡検出器

INTT は、MVTX と TPC の中間 (ビームパイプから $6\text{m}12\text{cm}$) に位置し、全方位をカバーする 4 層構造の検出器である。方位角方向に 78μ のピッチで半導体ストリップが配備されている。INTT の目的は、衝突後 MVTX を通り、その後 TPC を通過する過程での散乱粒子の飛跡を繋ぎ、運動量分解能を上げると共に、MVTX、TPC のトラックに、より詳細な時間情報を与えることである。

2.1.3 TPC(Time Projection Chamber)

TPC は、3 つの検出器のうち最も外側 (ビームパイプから 20m から 78m) に位置し、衝突中心から $\pm 1\text{m}$ ずつを覆うガス検出器である。読み出しパットからの二次元情報に加えてドリフト時間からビーム軸方向の位置分解能を持つ。希ガスで満たされた有感領域内を通過した荷電粒子の飛跡を、3 次元的に再構成することが可能であり、荷電粒子における運動量測定では中心的な役割を担う。

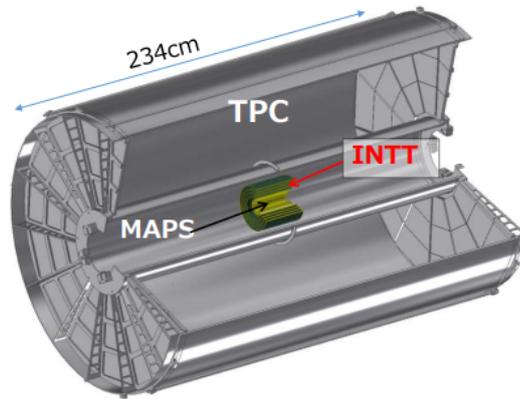


図 2.1 sPHENIX 飛跡検出器

2.2 INTT の役割

INTT は時間分解能に優れ、MVTX、TPC のトラックに対して 1 ビームバンチ幅の時間情報を与えることができ、衝突後の粒子の散乱の進行過程を知る上で重要な役目を持つ。MVTX、TPC は時間分解能が 35~200 ビームバンチ幅あるためそれぞれのヒット情報が一回のデータ読み出しにおいて多重に縮退してしまう。すなわち、どのビームバンチによるイベントなのか、判断し難い。時間分解能の高い INTT によって再構成された飛跡とマッチングを図ることで粒子多重度の高いイベントにおいてもより正確な飛跡を求めることが出来る。

2.2.1 HDI(High Density Interconnect)

FPHX チップへの入出力配線。センサーや FPHX チップへ電源を供給する。ROC とはバスエクステンダーを介して接続され、L0 用には FPHX チップが 10 個、L1,2,3 用には 26 個乗る。シリコンセンサーが乗る部分は、L0 はメッシュになっている。6 層構造になっていて、高速でのデータ送受信を担い、荷重が加わった場合にも耐久性のある素材となっている。

2.2.2 シリコンセンサー

INTTT においては、シリコンストリップセンサーを採用しており、レイヤー番号 L0 と L1,2,3 でセンサー 1 モジュールのサイズが異なる。最もビームラインに近くに位置する最内装から順に L0~L3 の順にナンバリングされており、実際の実験に導入される時には、バレル上にビームラインを中心として複数のセンサーモジュールが取り囲むように配置する。L1,2,3 ではセンサーの位置による粒子のヒット数の違いからストリップ長の異なる 2 種類のセンサーを組み合わせる (Type A,B)。また、ストリップ型シリコンセンサーの両サイドに 13 個ずつ、計 26 個の FPHX チップが配備されている。ただし L0 層は、シリコンセンサーの片サイドにのみ、10 個の FPHX チップが装備されている。ストリップ型のセンサーを 4 層に重ね位置関係を調節することで、3 次

元的情報を獲得できるように補修される。また、シリコンセンサーは n 型半導体と p 型半導体で構成されており、各チャンネルにキャパシタを構成するように設計されている。

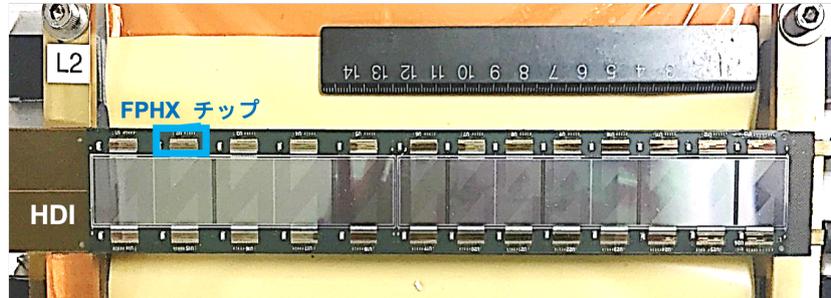


図 2.2 INTT 用シリコンセンサー : L2 モジュール

2.2.3 シリコンセンサー読み出しチップ (FPHX)

FPHX チップは PHENIX 実験でも使用されていたシリコンセンサー用の読み出しチップである。sPHENIX 実験における INTT のシリコンセンサーの読み出しチップとしてもにおいても採用されている。それぞれのチップが 128 個の読み出しチャンネルをもち、各チャンネルで波形整形、3bit の ADC 機能をもつ。ユーザーによって、GUI 上で ADC 閾値、チップ、チャンネルのマスク箇所、チップのゲイン値などの稼働条件が制御でき、タイムスタンプ、ADC 値、ヒットチャンネルを出力する。今回の測定においては、PHENIX の FPHX についての仕様書 [11] から最適ゲイン値を算出し、ゲイン値 300(mV/fC) に設定した。このゲイン値については、全測定において変更していない。入出力は LVDS 信号を介して行われ、センサーと読み出し基盤 (HDI) はワイヤーボンディングで接続されている。

2.2.4 ROC(Read Out Card)

FPHX チップからのデータを同期させて整理したり、ヒット信号にタグ付けを行い、後ろの読み出し回路へ転送する役割を担う。また、稼働条件や閾値などの情報を指定された FPHX チップへ転送することが出来る。他に、キャリブレーションパルスを発生させたり、センサーモジュールへの電源の供給も行う。



図 2.3 ROC

2.2.5 FEM(Front End Module)

VME 規格の読み出しボード。ROC や FPHX からのデータをまとめ、PHENIX 共通のフォーマットに変換する。データは他の検出器の情報との統合を行うモジュールに送られる。FEM1 枚で ROC の出力の半分を賄うため、ROC1 枚の出力を見るためには FEM が 2 枚必要となる。ただし、ROC への命令信号は 1 枚で賄うことが出来る。また、FEM 基盤上には FPGA を搭載しており、トリガーシステムの構築が可能である。

2.2.6 FEM-IB(FEM Interface Board)

FEM 全体を制御するモジュールであり、主にクロック信号やトリガー信号、FEM 制御信号を受け取る。FEM-IB 一つで一個の FEM を統括することができる。今回の実験では、ROC は 1 モジュールしか使用していないため、FEM 1 つ、FEM-IB1 つの使用となっている。

2.2.7 半導体の原理

半導体とは、結晶構造をした、エネルギーバンド構造をとる物質一般のことを指す。半導体には n 型と p 型の 2 種類がある。p 型の方が n 型よりも電子を多く有する。半導体に電圧をかけると、自由電子は正の方向へ移動し、負の方には、それらの電子が離脱したことによって生成される正孔（ホール）ができる。すると、この電荷の偏りからエネルギーギャップが生じる。このエネルギーギャップ間には電圧差が生じる。実質このエネルギーギャップ間には電氣的に中性な、空乏層と呼ばれる領域が生成された状態になるこの空乏層を荷電粒子が通過すると、荷電粒子の殿下に引き寄せられ、通過時に静電エネルギーが発生し、その電流パルスを読み出しチップが広い、通過信号として認識される。

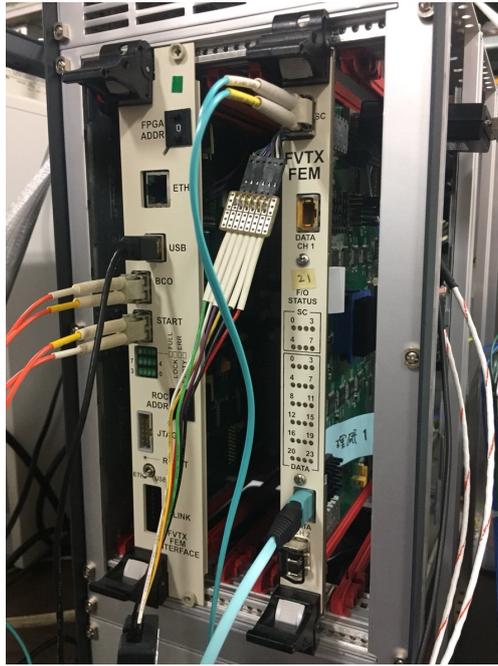


図 2.4 左:FEM-IB 右:FEM 本体

2.2.8 シリコン半導体におけるミュオンのエネルギー損失

荷電粒子が物質中を運動量をもって通過するとき、物質中でエネルギー損失を起こす。このエネルギー損失値は、通過する物質の種類やその厚さ、また入射してきた粒子の持つ運動量に依存している。そのため、高エネルギー加速器実験において、粒子の性質を特定する際などにエネルギー損失を算出することで、時間を遡って親核子を推定したり、衝突時の崩壊反応を再現可能になるなど重要なファクターである。

荷電粒子などの放射線が物質中を通過するとき、その衝突する物質が十分の質量が大きい場合、相互作用を起こし、運動量を失う。この場合のエネルギー移行は運動量移行により表される。一般的に、衝突反応断面積はベーテプロッホの式に従う。ベーテプロッホの式は、Leo, William R. 著 *Techniques for Nuclear and Particle Physics Experiments* [5] により、

$$-\frac{dE}{dx} = 4\pi N_a r_e^2 m_e c^2 \rho \frac{Z}{A} \frac{z^2}{\beta^2} \left[\ln \left(\frac{2m_e \gamma^2 \nu^2 W_{max}}{I^2} - 2\beta^2 \right) \right] \quad (2.1)$$

である。この式に2つの補正、密度効果補正 δ_2 と、殻補正 C が加えられた式が次の式である。

$$-\frac{dE}{dx} = 4\pi N_a r_e^2 m_e c^2 \rho \frac{Z}{A} \frac{z^2}{\beta^2} \left[\ln \left(\frac{2m_e \gamma^2 \nu^2 W_{max}}{I^2} - 2\beta^2 - \delta - 2\frac{C}{Z} \right) \right] \quad (2.2)$$

この時、それぞれのパラメータは以下である。

r_e : 古典的電子半径

- m_e : 電子の質量
- N_a : アボガドロ数
- I : 平均励起ポテンシャル
- Z : 吸収物質の原子番号
- A : 吸収物質の原子量
- ρ : 吸収物質の密度
- z : e の単位下における入射粒子の電荷
- β : 入射粒子の $\frac{v}{c}$
- γ : $\frac{1}{\sqrt{1-\beta^2}}$
- δ : 密度補正
- C : 殻補正
- W_{max} : 単一衝突での最大エネルギー移行

また、 $\frac{dE}{dx}$ は阻止能とも言われ、粒子の物質中での通過を阻止する能力として表現する場合もある。粒子の種類ごとに、かつその粒子の持つ運動エネルギーの大きさによっても異なる。非相対論的なエネルギーにおいて、 $\frac{dE}{dx}$ は $\frac{1}{\beta^2}$ に依存しており、 $\frac{dE}{dx}$ が小さくなると、粒子の速度 $V=0.96C$ =光速まで速度は大きくなっていく。のちに、 $\frac{dE}{dx}$ は最小値をとる。この最小値をとった点の粒子を minimum ionizing という。

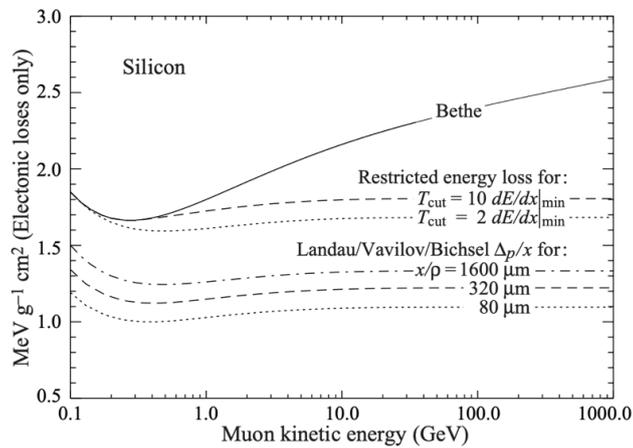


図 2.5 単位厚さあたりのシリコン中でのミューオン運動エネルギーとエネルギー損失の関係 [6]

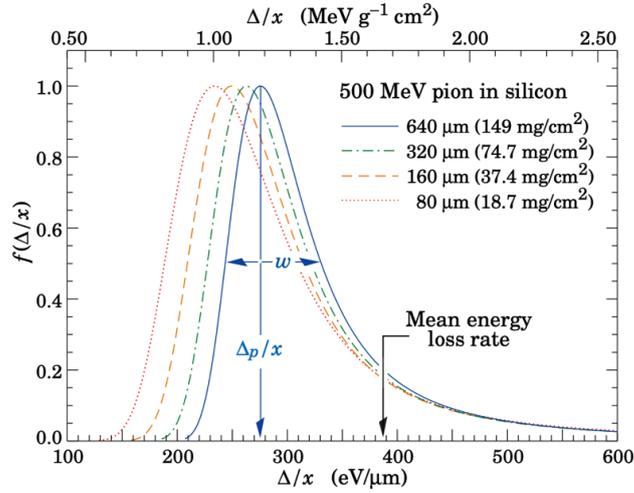


図 2.6 500MeV のパイオンのシリコン中での Straggling 関数の変動 [6]

このグラフは most probable value(最確値)である $\frac{\Delta_p}{x}$ で規格化されている。また W は straggling 関数の最大値の $\frac{2}{1}$ の値に対応する分布曲線の幅の長さである。グラフから、吸収材が薄いほど単位飛程あたりに失う運動エネルギーの値は小さくなり、分布はエネルギー損失の小さい方へ偏っていくことがわかる。また吸収材の十分に厚い場合と比べ、最頻値のエネルギー損失値が低くなる。今回は半導体であるシリコンセンサー通過時のミュオンエネルギー損失の関係グラフを使用した。一般的にエネルギー損失の計算は、式 2.1 に表したように、ベータプロットの式から算出される。ただし、INTT シリコンセンサーで使用するシリコンの厚さは $200\mu m$ すなわち $200 \times 10^{-4}(cm)$ と比較的薄い。この場合、エネルギー損失の確率分布は偏った分布を示す。よって、大きなエネルギーを落とすイベントが起こる確率は極めて低い。このことから、Landou-Vavilov の式の $200\mu m$ の値を採用した。

実際にシリコンセンサーを通過するミュオンのエネルギー損失値を見積もる。今回、地上で観測される宇宙線をミュオンで考え、ミュオンは MIP で入射すると考えた。グラフ??より、MIP の時、即ち最小エネルギー損失値 $0.4(GeV)$ のとき、単位否定あたりのエネルギー損失は $1.1(MeVg^{-1}cm^2)$ である。Si の電子正孔対 1 個の生成に必要なエネルギーは気温 200K で $3.62(eV)$ より、

$$0.0513(MeV) \div 3.62(eV) = 14200(\text{個}) \quad (2.3)$$

14200 個の電子正孔対を電荷量に換算すると、

$$14250 \times 1.60 \times 10^{-16} = 2.27(fC) \quad (2.4)$$

FPHX チップでのゲイン値を配慮して計算する。今回、FPHXchip ゲイン値の増幅率は $300 \frac{mV}{fC}$ 。

ミュオンの MIP の持つ電荷量 $2.27(fC)$ と求めたので、シリコン中を $200\mu m$ 進行したミュオンが損失するエネルギー量は電圧値でおおよそ

$$2.27(fC) \times 60\left(\frac{mV}{fC}\right) \times 5 = 680(mV) \quad (2.5)$$

で読み出されると見積った。

上記で述べた方法で、全ての測定条件においてエネルギー損失値の計算を行った。ミューオンがシリコンセンサーを垂直に 1strip を最短飛程で通過する時は $200\mu\text{m}$ の最短飛程となる。1strip の最大飛程は図??より、入射角度で計算すると、 68.69° の時であり、この過程で 730mV の損失が起こる。同様に 2strip 通過時のエネルギー損失値は 52.05° であり、最大 861mV の損失を起こす。この計算はシリコンストリップを 2 次元的に概算したため、ストリップの奥行き 20mm は配慮していない場合の計算結果である。その概念図が図 2.7 である。

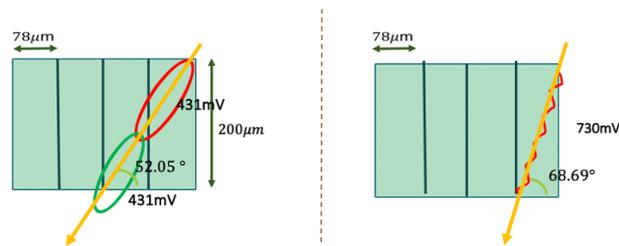


図 2.7 ストリップ数別最大エネルギー損失の見積もり方法概念図

第 3 章

INTT テストベンチの環境構築

3.1 INTT テストベンチ構築のための初期設定と環境設定

奈良女子大学に構築した、INTT テストベンチについてである。図 (3.2) のように、奈良女子大学高エネルギー研究室の実験室内にテストベンチを構築した。High VolutageNIM モジュールは奈良女子大学のものを使用した。使用した NIM モジュールを写真左から順に示す。また、簡単にこれらの NIM 規格のモジュールの機能についてまとめた。

NIM 規格モジュール	機能
TRIPLE 4-FOLD 1-VETO COINCIDENCE	シリコンセンサー 2 台を使用した時に AND 信号を出力する。
QUAD DISCRIMINATOR	任意の閾値以上の電圧パルスの信号のみをアクセプトする。
DUAL GATE GENERATOR	入力されたデジタル信号の width を調節し、delay を与える。
LOGIC LEVEL ADAPTER	NIM 規格のデジタル信号をロジック回路のフォーマットである TTL 信号に変換する。
DUAL HIGH VOLUTAGE POWER SUPPLY (POSITIVE)	シリコンセンサーに電圧を印加する。
100MHz CLOCK GENERATOR	任意の周波数を持った信号を出力することができる。
DUAL HIGH VOLUTAGE POWER SUPPLY (NEGATIVE)	シンチレーションカウンタに電圧を印加する。
8CH VISUAL SCALER	入力端子が 8 ch あり、それぞれエントリー数をカウントすることができる。

表 3.1.1 使用した NIM モジュールとその機能



図 3.1 NIM モジュールのセットアップ写真

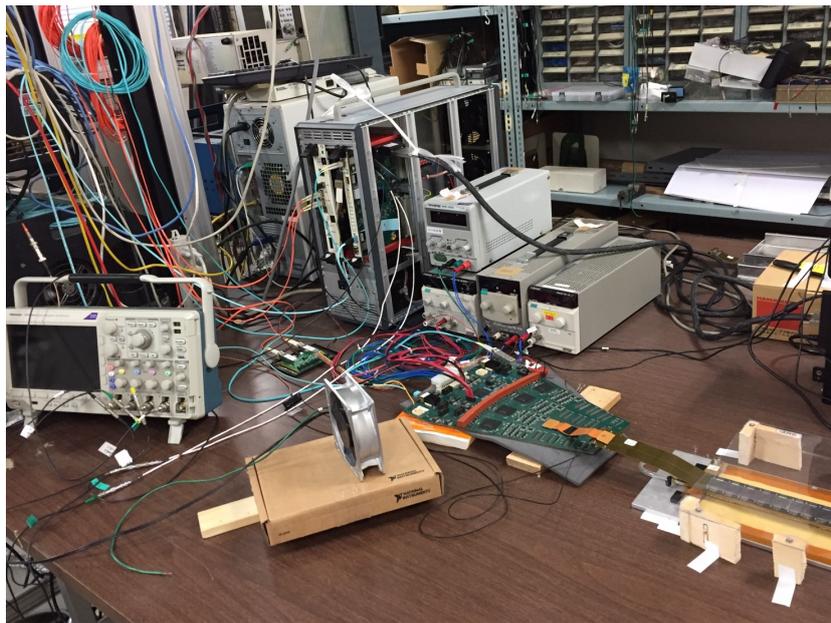


図 3.2 奈良女子大学に設置したテストベンチ全体写真

まず、テストベンチの環境とそのブロック図は簡易的に示すと図 3.3 のようになっている。

今回の性能評価で測定対象とした荷電粒子である宇宙線によるミューオンや、 β 線源から放射した電子は、シリコンセンサーを通過し、半導体にバイアスをかけ、生成される空乏層を通過するときにエネルギーを損失し電荷を発生させる。その電荷によって生じた電圧パルスシリコンセンサーのそれぞれの区画に対応した FPHX チップがアクセプトし、読み出しを行う。FPHX チップの電荷アンプで微弱なエネルギー損失による電荷を増幅し、より識別しやすい大きさのパルスにシ

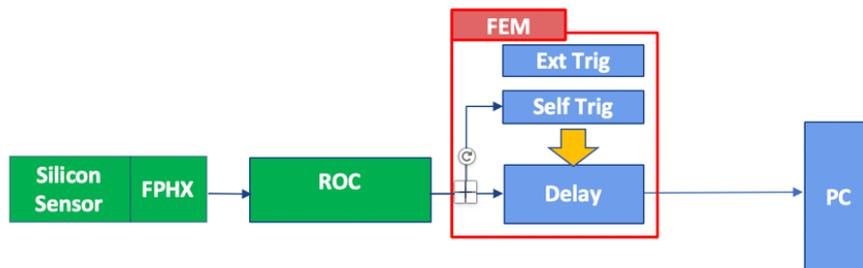


図 3.3 INTT 読み出し回路ブロック図

ンセサイズする。その後、ADC 機能によってアナログ信号からデジタル信号に変換される。そのデジタル信号はバスエクステンダーで接続されている初期段階のデータ処理システムを持つ ROC へ送信される。ROC 内で複数のチップから一斉に入力されたデジタル信号を集積し、同期させる。同期された信号は二次処理システムである FEM へ送られ、ここで、sPHENIX 共通のデータフォーマットに変換され、かつ、FPGA ロジック回路において、トリガー診断され、トリガーが発行される。発行されたトリガーは、USB ケーブルを通して PC へ送られ、このデータを PC 上でファイルに出力し、そのデータをユーザが解析するといった流れになっている。

3.1.1 トリガーモード

シリコンセンサーを性能評価するにあたって、3つのトリガーモードを使用した。まず、キャリブレーションモードである。これは ROC で生成した模擬信号を ROC から FPHX チップに入力し、チップから出力した後に FEM、PC へと信号が送られる。以下にキャリブレーションテスト時のブロック図を簡単に示す。

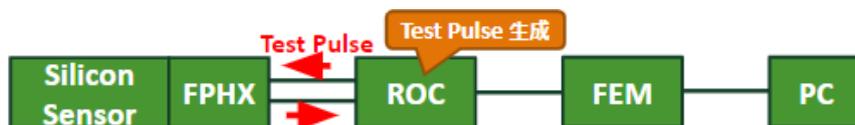


図 3.4 キャリブレーションテストにおけるテストベンチブロック図

キャリブレーションテストではこのキャリブレーションモードを後述の GUI で選択し測定する。次にセルフトリガーモード、外部トリガーモードである。これは線源や宇宙線によるデータ測定の際に利用する。どちらもキャリブレーションモードとは違い FEM から発行するトリガーである。ブロック図は図 3.3 に示す。セルフトリガーモードでは DAC の最低閾値をトリガーとし、閾値以上の電圧値を持つ信号のみを FEM から PC に送る。閾値以上の信号であれば全ての信号を獲得するためノイズが多く、イベント数の少ない宇宙線測定ではこのトリガーモードは使用しない。そこで使用するトリガーモードが外部トリガーモードである。外部トリガーモードではシンチレーショ

ンカウンターを外部トリガーとし、セルフトリガーモードとのコインシデンスをとることでノイズが減少する。したがって、宇宙線由来の信号の選定がより精密なものとなることが期待できる。本研究では線源による性能評価ではセルフトリガーモード、宇宙線測定による性能評価では外部トリガーモードを利用した。

3.1.2 GUI 上での操作方法

DAQ 操作のために, Nevis_gui という名の GUI(Graphical User Interface) を利用した。これは INTT 検出器のためのデータテイキング用独自アプリケーションである。GUI のソフトウェアの window は図 (3.5) のようになっている。図 3.5 に枠や注釈で示しているように、DAC を直接設定し、FEM へ send したり、FPHX チップゲイン値を計算したのち、設定したり、その他細かい設定を任意で変更できることが利点の 1 つである。また、今回ロジック回路である FPGA を搭載した FEM 上とプログラミングケーブルで接続し、ユーザーが制御できる仕組みである。このソフトウェアは Python を使って書かれており、比較的その仕様やオプションをユーザが効率化を求めて、書き換え修正することができる。

この GUI 上では FPHX チップの chip それに付随する channel を各々 mask することができる。この機能により誤作動、その他データ化けやノイズの発生源となる不良ストリップ、チップを避けて測定でき、正常なデータがノイズに埋もれてしまうことを防ぐことができる。さらなる機能として、FPHX チップの Gein 値を変えること、FEM でトリガー処理を行なっている間の信号の delay time、など臨機応変な測定条件を設定できる。また、使用するトリガーモード（キャリブレーションモード、セルフトリガーモード、外部トリガーモード）を選択し、GUI 上のボタンを操作することで、DAQ の各プロセスを実行できる。一度、DAQ を開始すると、スレッドが進行し、次 DAQ の停止命令をユーザが送るまでデータを取り続けるシステムである。ただし、今回 1 つ 1 つボタンで DAQ を操作していると、押し間違いや、順番の入れ違い等、人

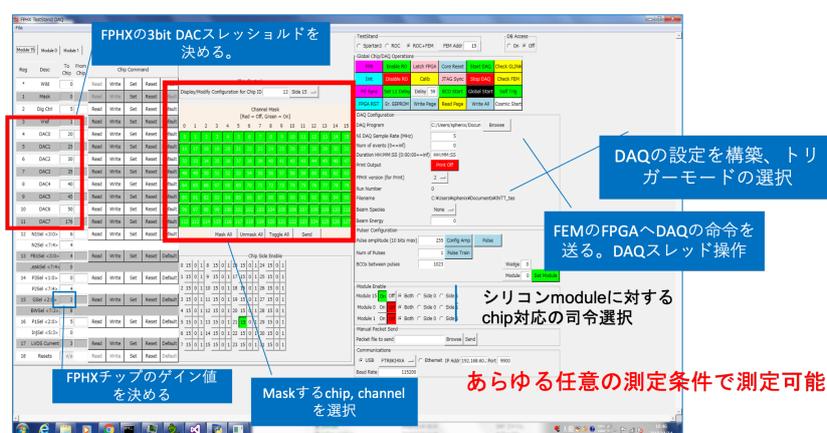


図 3.5 DAQ 操作のための GUI (Nevis_gui のウィンドウ写真と操作箇所

為的なミスが発生した。ミスを避けるため、GUIのpythonによるプログラムに、外部トリガーモードの一連のDAQ操作プロセスをひとまとまりにした関数を定義し、新しくGUI上に外部トリガーモード測定開始ボタン (cosmic start) を作成することに成功し、その正常動作を確認した。このように、私たち自身がテストを行うに当たってより効率よく測定を進められるようにあらゆる工夫も行った。

第 4 章

ノイズ評価

4.1 キャリブレーションテスト

本研究におけるキャリブレーションテストの目的は、キャリブレーションモードで FPHX チップの各チャンネルにダミーパルスを入力し、応答を調べることで、チップの正常動作を確認することである。方法としては、入力パルスの波高を 0 63 の全範囲で、各波高で 20 回測定を等しく繰り返し、応答 (再現性) を検証した。閾値設定は表 (4.1.1) の通りである。

キャリブレーションテスト結果

FPHX1 チップ分のテスト結果を図 (4.1) に示す。26 チップ分の結果を見る前に、まずは 1 チップ分の結果について細かく見ていく。

DAC	閾値	電圧値(mV)
0	20	290
1	25	310
2	30	330
3	35	350
4	40	370
5	45	390

H]

表 4.1.1 閾値設定

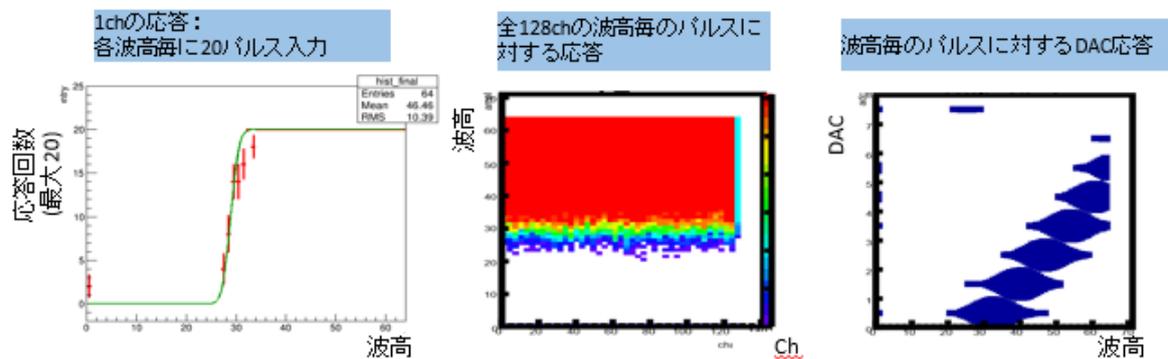


図 4.1 1 chip キャリブレーションテスト結果

FPHX チップ1 チップが持つ 128 c h のうち、1ch 分の応答を示す図から、DAC0 の閾値付近で急激にパルスの応答が始まり、閾値以上の波高のパルスが各 ch に 20 発ずつ入力されていることが確認できた。また、全 128 c h の波高毎のパルスに対する応答を示す図から、先ほどの図から読み取れた傾向が、128ch 全てにおいて同様に見られ、全 ch において期待通りの動作をしていることが確認できた。そして、3 つめの波高毎のパルスに対する DAC 応答を表す図からは、波高と DAC 値の間に正の相関関係があることが確認できた。つまり、設定した 6 つの各 DAC 閾値が波高に対するスレッシュホールドとして機能したと言える。

次に、FPHX26 チップ分のテスト結果を図 (4.2) に示す。

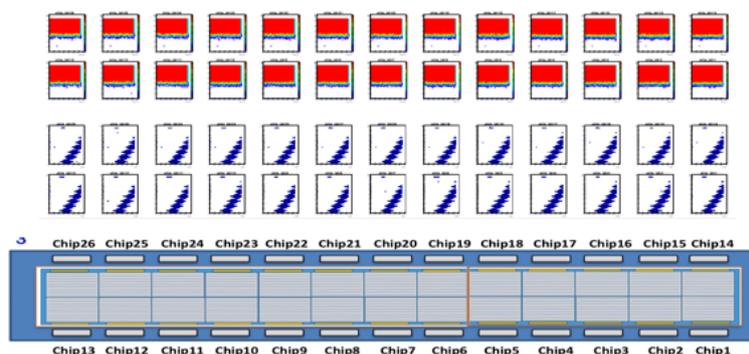


図 4.2 26chip キャリブレーションテスト結果

FPHX チップの全チャンネルにおいて閾値が正常に動作しており、また波高と DAC 値の間に正の相関関係を持つことを確認し、本テストの期待する動作を全てのチップで確認できた。従って、キャリブレーションテストで全チップの正常動作を確認することが出来た。

一番下の図が実際の 26 チップの配置図であり、これと照らし合わせて上の 2 種類の図におけるチップの位置を確認することが出来る。また、1 チップ毎のテスト結果であるため、先ほど 1 チップ分のテスト結果で示した 1ch の応答を表す図はここでは出力していない。

キャリブレーションテストにおいて期待する結果は得られたが、さらなる追求を行うという点で、本テスト結果において未解決に終わった点があるためここに記しておく。図 4.2 の 1 番上の図であるが、全てのチップにおいて一部 DAC 閾値がスレッシュホールドとして働いていないエントリーが見られる。特に、DAC7,ampl<60 にこれらのエントリーが見られる。これらのノイズの正体はわかっていないが、そのエントリー数や入る channel は chip 毎に異なっていることはわかっている。よって、今回はこれらのノイズを今回後述のノイズ評価において評価したいスレッシュホールドにおけるノイズではなく、間違えて自分の入るべきではない DAC に入ってきたノイズであると認識し、DAC7,ampl<60 のエントリーを除いてノイズ評価を行った。

更に、キャリブレーションテストにおいて期待する結果が得られなかった場合の対処法も記しておく。本研究において、シリコンセンサーを用いて測定を行う際には、必ず事前にキャリブレーションテストを行い、26 チップ全てが正常動作することを確認してから測定を行った。その中で、数チップ期待する結果が得られないことがあった。その一例が、閾値での応答が急激ではなくまだらに分布する、波高と DAC 値間に相関が見られない等である。このような問題が生じた際は、テストベンチ全体を見直し、配線や接触その他異常がないか見直す必要がある。特に、バスエクステンダーと ROC・HDI の接続がしっかりできているか、また ROC のポートが破損していないかを注意深く見る必要がある。これらを見直し、問題を解決することで、期待する結果が得られることがほとんどであった。よって、キャリブレーションテストの期待しない結果からすぐにチップの異常性を判断するのではなく、テストベンチを見直し問題を解決してからもう一度テストを行うことを勧める。

4.1.1 ノイズ評価概要

ノイズ評価の目的

本研究におけるノイズ評価の目的は、この後行う β 線源や宇宙線測定による性能評価を行う際、それぞれのデータに対するノイズの影響を配慮すべきかどうか、定量的に判断することである。ノイズのパルスの高さを、 β 線や宇宙線のパルスの高さと比較したいが、まだ β 線や宇宙線の測定は行っていないので、ここではキャリブレーションテストの際の DAC0 閾値の電圧値と、ノイズのパルス高から算出した電圧値と比較して評価する。まず、先述のキャリブレーションテストの 1 チップ分の結果で示した 1ch の応答の図 (図 4.3) を詳しく見ていく。

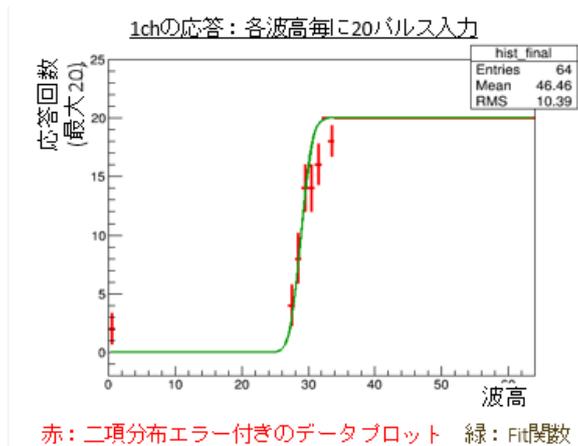


図 4.3 1ch の応答

キャリブレーションテストの際は、この図から DAC0 の閾値付近で急激にパルスに回答が始まると述べた。この DAC0 閾値の立ち上がり部分を注意してみると、理想的には、DAC0 閾値で 20 発の回答が始まり、鋭く立ち上がるはずだが、実際にはノイズの影響で少し緩やかな立ち上がりとなっていることが分かる。

続いて、Fit 関数から σ 値を以下のように算出する。

$$\text{Fit 関数} : f(x) = \text{erf}(b(x - a) + 1) * 10 \quad (4.1)$$

$$\sigma * b = 1 \quad (4.2)$$

$$\sigma = 1/b \quad (4.3)$$

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (4.4)$$

式 (4.1) 中の erf は式 (4.4) の誤差関数 (error function) である。また、a,b はパラメータであり、a は x 方向のゼロ点からの差異、b は誤差関数の σ を調整するパラメータである。図 (4.4) に σ の算出方法をまとめる。

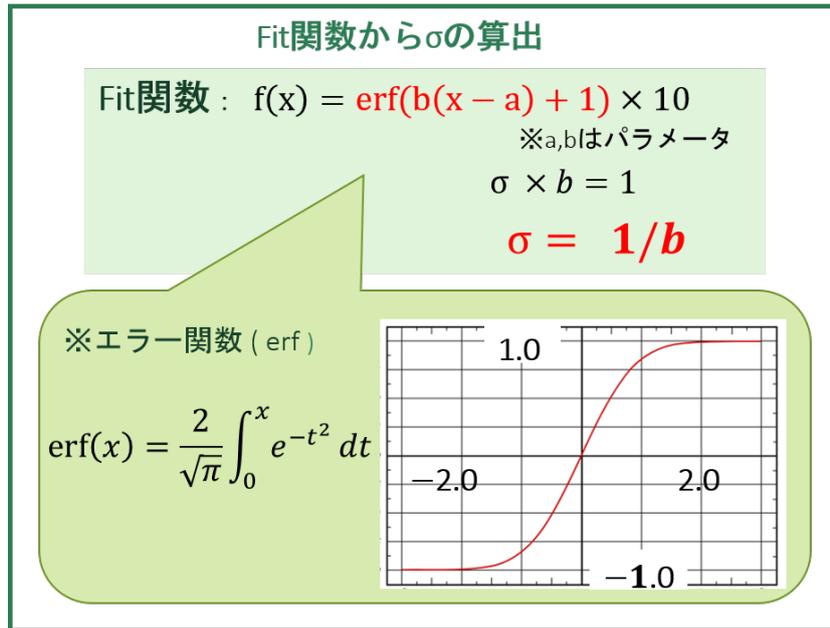


図 4.4 Fit 関数から σ の算出

以上で算出した Fit 関数の σ を全チャンネル分求め、後述のノイズの評価結果で考察した。

ノイズ評価結果

FPHX チップ 1 つがもつ 128 チャンネル \times 26 チップ分、つまり全部で 3328 チャンネル分の σ をプロットしたのが図 (4.5) である。 σ 分布の平均値は 3.276 ± 0.7374 であることが分かる。電圧値 $[\text{mV}] = 2.81 \times$ 波高 $[\text{ampl}]$ より、 1σ の平均値 $3.276[\text{ampl}]$ は

$$2.81 \left[\frac{\text{mV}}{\text{ampl}} \right] \times 3.27 [\text{ampl}] = 9.19 \quad (4.5)$$

式 (4.5) より電圧値は 9.19mV となる。また、DAC0 閾値は 290mV であるから

$$\frac{290[\text{mV}]}{9.19[\text{mV}]} = 31.6 \quad (4.6)$$

式 (4.6) より DAC0 閾値はノイズの波高平均値から約 31.6σ 離れている。従って、DAC0 閾値に対し、ノイズの大きさは十分に小さく、ノイズによる影響は無視できると考えられる。

全 channel の σ 分布

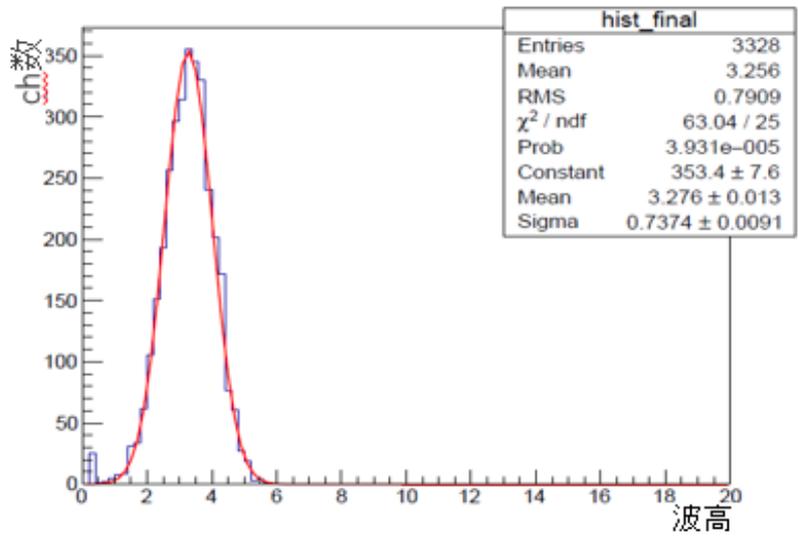


図 4.5 全 channel の σ 分布

ノイズ評価 レート編

ノイズ評価から派生して、FPHX チップの正常性を確かめる指標の一つとして、本解析を行った。

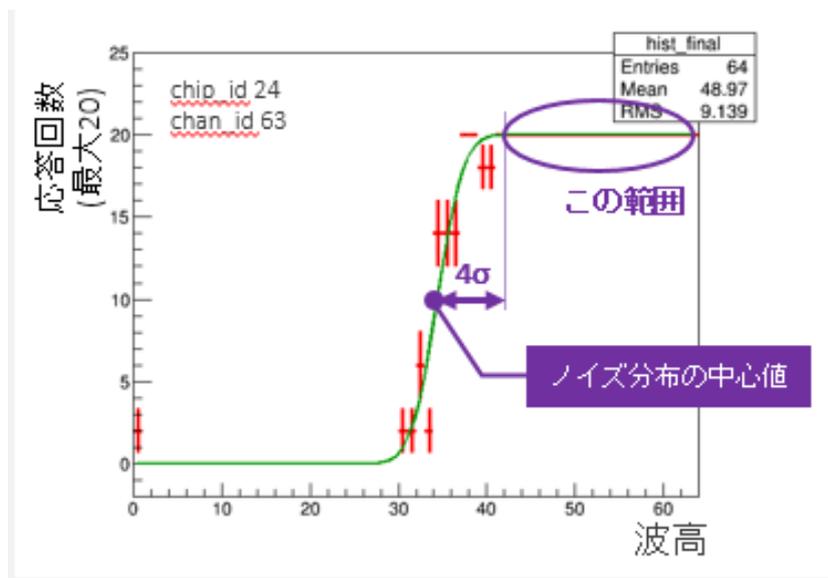


図 4.6 レートの考え方

図 (4.6) のように、ノイズの分布の中心値から 4σ 離れた波高値を始点とし、テストパルスの波高の最大値 64 までの範囲で entry が 20 入っている割合を求めた。この割合をレートと呼ぶ。レー

トの算出式は式 (4.8)。

$$\Delta A = \frac{\sqrt{\sum_n (A_i - \langle A \rangle)^2}}{N} \quad (4.7)$$

$$rate(\%) = (1 - \frac{\Delta A}{\langle A \rangle}) * 100 \quad (4.8)$$

A_i は応答回数、 $\langle A \rangle$ は応答回数の期待値 (20)、 ΔA は応答回数からのズレ、 N はデータ数。このレートをチャンネル毎に求め、1 チップ分出力したグラフが図 (4.7) である。

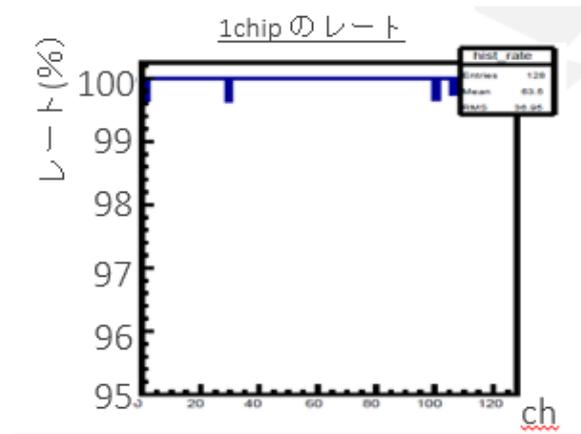


図 4.7 1 チップのレート

結果から、今回のノイズ評価テストにおいて全てのレートが 99% 以上の応答を示していたため、より詳細な情報を得るためにグラフ出力の y 軸スケールを 95% にカットした。続いて、全 26 チップ 1 チップ毎のレートを図 (4.8) に、またこの全ての 26chip のレートをヒストグラムにし、図 (4.9) に示す。

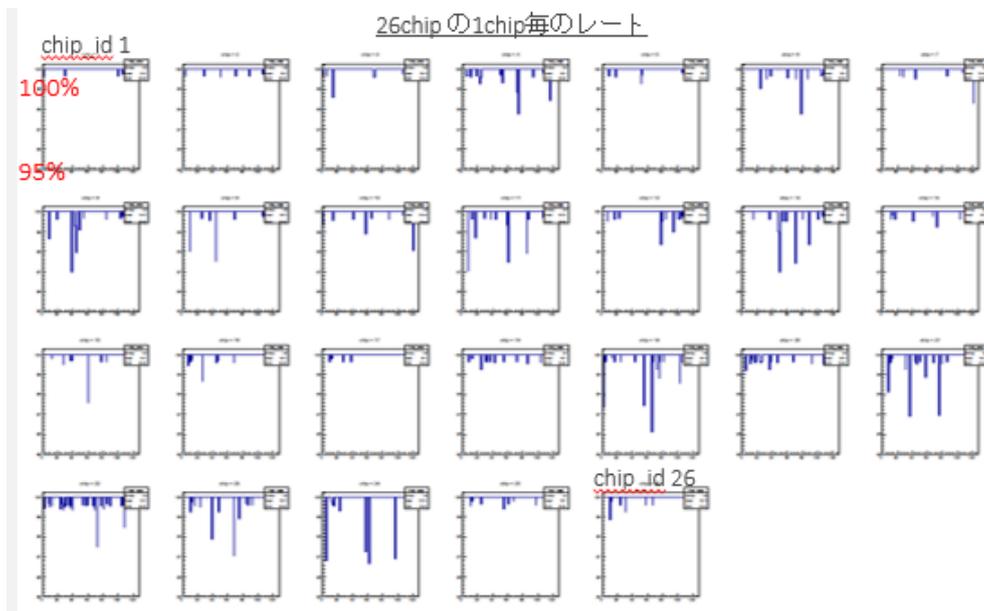


図 4.8 26 チップの 1 チップ毎のレート

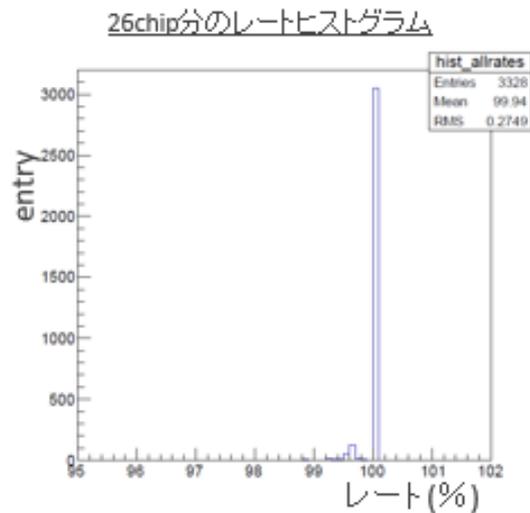


図 4.9 26 チップ分のレートヒストグラム

図 (4.8)、図 (4.9) では、26 チップ分のチップ毎のレート、また総レートを一目で確認することが出来る。今回は全てのレートが 95% 以上であった。このノイズ評価の方法は、このキャリブレーションパルスの応答信号のレートが何パーセント以上であれば FPHX チップは正常か、または何パーセント以下であれば異常と見なし、配慮すべきかという判断基準の指針とすることができる。多くのデータと実際の計測を元にこの基準を決めていく必要がある。

第 5 章

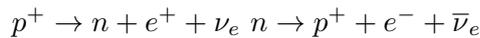
β 線源による INTT 性能評価

5.1 β 線源による性能評価概要

β 線とは放射性核種が崩壊する際に放出される電子のことである。この β 線を放出する β 線源を用いてシリコンセンサーの性能評価を行う。 β 線源は宇宙線等と比較するとシリコンを通過すると予想されるレートは極めて高く、短時間に多くのイベントが期待できる。また、シリコンを通過する際に損失するエネルギーの電圧値は宇宙線と近い値を示すことが分かっている。したがって、本研究では宇宙線による性能評価の前実験として、 β 線源を用いたシリコンセンサーの性能評価を行う。使用する β 線源は ^{90}Sr である。

5.1.1 β 線源

β 線源とは β 崩壊する放射線核種である。 β 崩壊は β^+ 崩壊と β^- 崩壊があり、それぞれの崩壊方式は以下のように表せる。



5.1.2 ^{90}Sr の性質

^{90}Sr は以下に実際に使用した ^{90}Sr を示す。



図 5.1 実際に使用した β 線源

5.2 セットアップ

β 線は前述の通りイベント数が多く、またシンチレーションカウンタを挟んだ場合シリコンセンサーに到達する前にエネルギーを多く損失してしまうことから外部トリガーモードではなくセルフトリガーモードを使用した。トリガーについては第 3 章で説明している。

β 線測定におけるテストベンチの環境とブロック図を以下に示す。

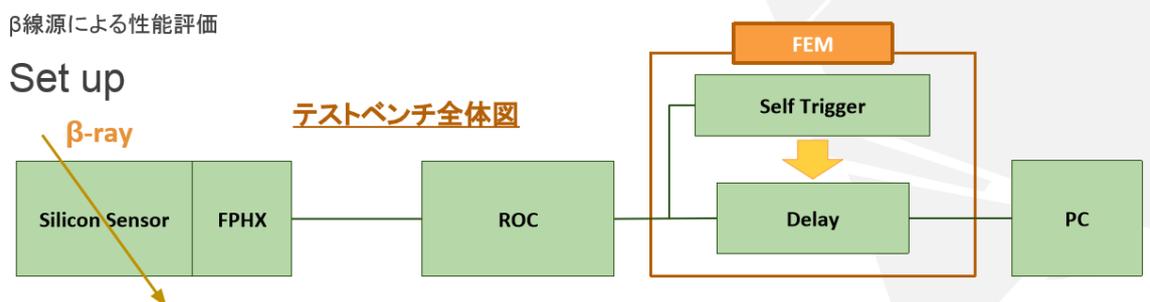


図 5.2 セルフトリガーモードでのブロック図

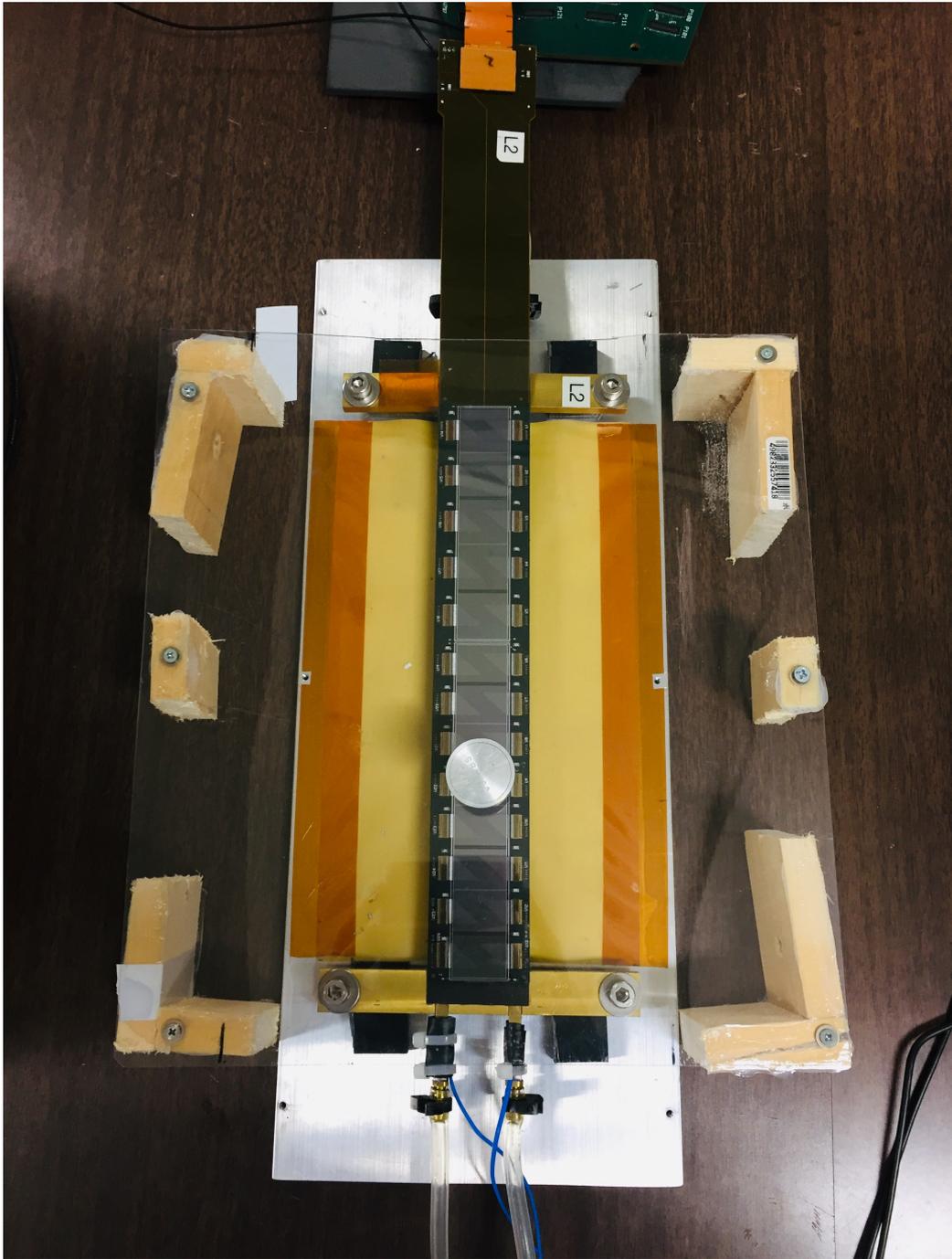


図 5.3 β 線測定時のシリコンセンサーのセットアップ

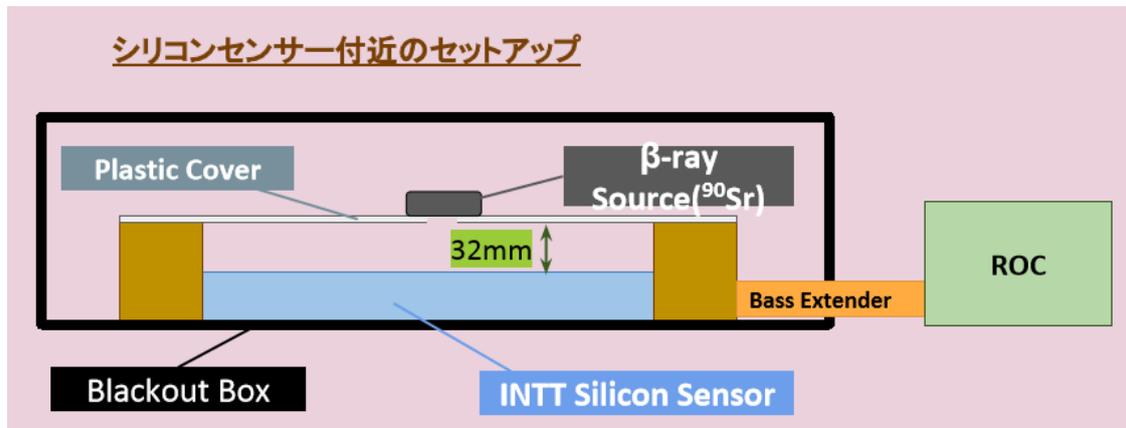


図 5.4 センサー付近のセットアップ図解

図 (5.4) はシリコンセンサーから ROC までの実際のセットアップをより詳しく図式化したものである。厚さ 1mm の薄いプラスチックカバーに β 線源を乗せ、暗幕で覆った箱でシリコンセンサーを覆う。プラスチックカバーには直径 1cm 程の穴が開いており、 β 線がプラスチックを通過せずにそのままシリコンセンサーを通過するようになっている。INTT は光に過剰に反応するため、測定時には箱の上からさらに暗幕を被せ、光が入らないようにしている。

5.3 シリコンセンサーにおけるエネルギー損失

一般的に、エネルギー損失は物質の厚さに依存する。本実験で使用するシリコンセンサーの厚さは約 $200 \mu\text{m}$ であり、非常に薄い。そのため大きなエネルギー損失が起こるとは考え辛い。物質が十分厚い場合、物質を通過する際のミュオンの MIP は平均値である Bethe dE/dx を採用する。しかし本実験では物質が非常に薄いため、最頻値である Landau dE/dx を利用する。

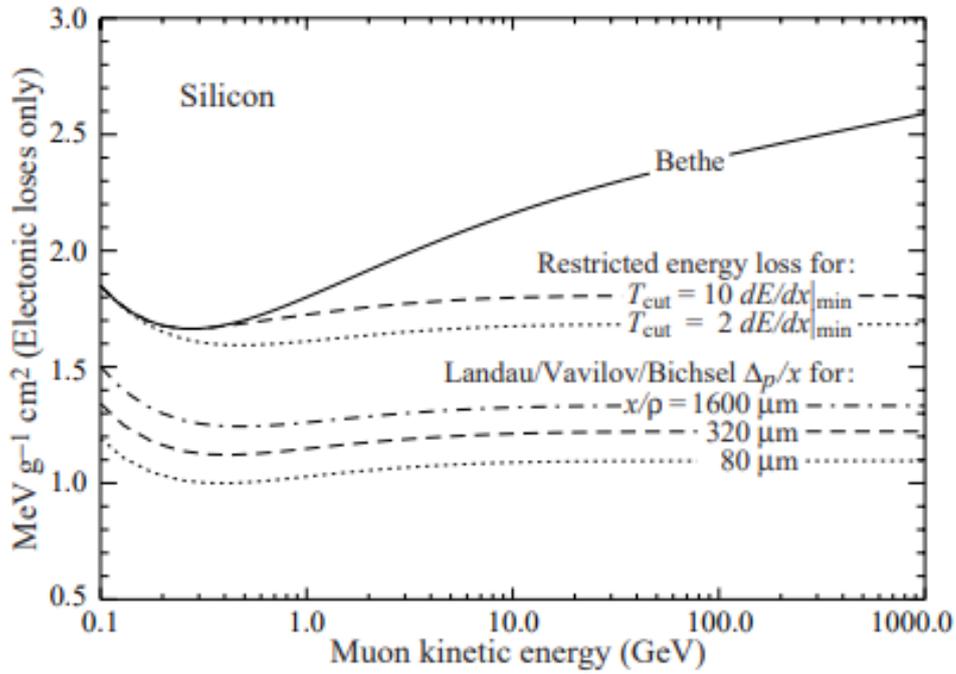


図 5.5 シリコンを通過する際のミューオンのエネルギー損失

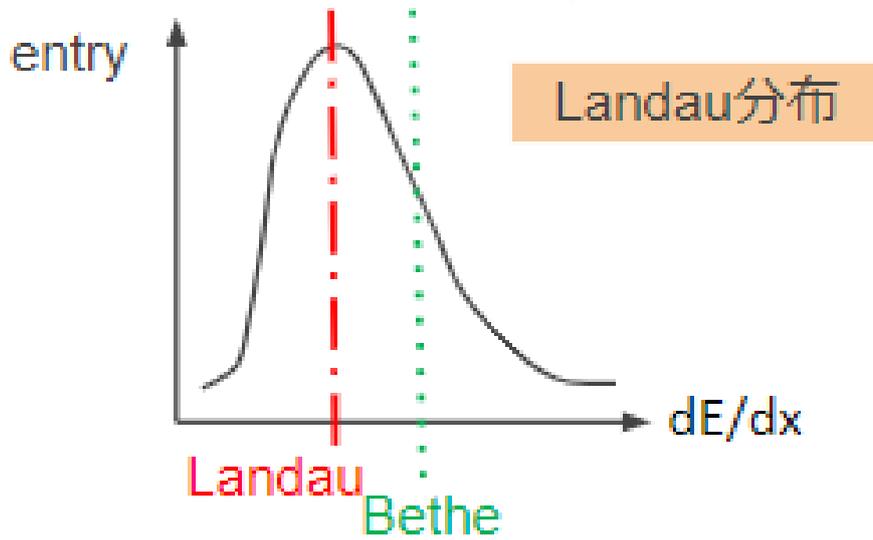


図 5.6 エネルギー損失の分布図

5.3.1 β 線がシリコンを通過する際のエネルギー損失

具体的に β 線、つまり電子がシリコンセンサーを通過する際に損失するエネルギー値を見積もる。電子の平均エネルギーから以下の式を用いて運動量を求める。

$p \rightarrow (E^2 - m^2)^{\frac{1}{2}} \beta\gamma \rightarrow \frac{p}{m}$ これより求めた運動量から以下のグラフより最頻値を利用した際のシリコンのMIPを求める。このグラフは横軸が運動量、縦軸がBetheを採用した際のMIPとの比を表している。

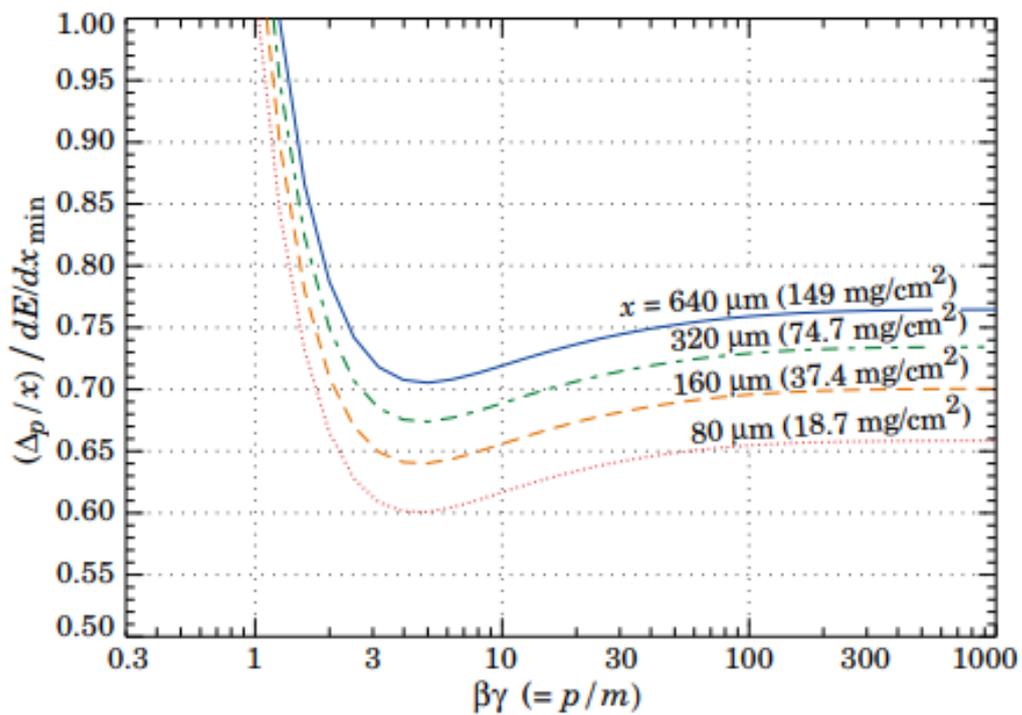


図 5.7 シリコンの厚さごとの運動量と MIP

5.3.2 エネルギー損失値の見積もり

β 線には直進性がある。そのためシリコンセンサーを通過するストリップの数は1ストリップに集中すると見積もることができる。エネルギー損失値を電圧値に置き換えるとストリップを垂直に直進する際は876mV、1ストリップの最大値は941mVと計算できた。

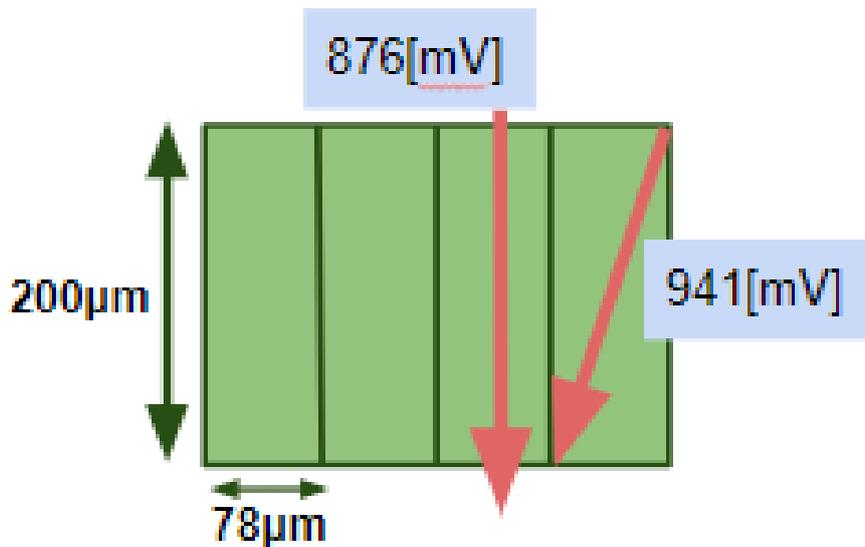


図 5.8 1 ストリップにおける見積もり値

このように 2 ストリップ、3 ストリップと最大値を見積もり DAC 値の設定を行う。DAC 値は最低の閾値を表しており、DAC 値に応じて対応電圧が設定できる計算式は以下の通りである。
 $210[mV] + DAC \text{ 値} \times 4 \rightarrow \text{対応電圧 [mV]}$ 本実験では DAC 値を以下のように定めた。

DAC	DAC 値	対応電圧
DAC 0	20	290 [mV]
DAC 1	35	350 [mV]
DAC 2	50	410 [mV]
DAC 3	65	470 [mV]
DAC 4	80	530 [mV]
DAC 5	95	590 [mV]
DAC 6	110	650 [mV]
DAC 7	125	710 [mV]

図 5.9 閾値 (DAC 値) の設定

また、実際に測定を行った際のセットアップを以下に示す。これにより、chip10、23 付近に entry が集中すると予想できる。この実験では 2 分間測定を行った。

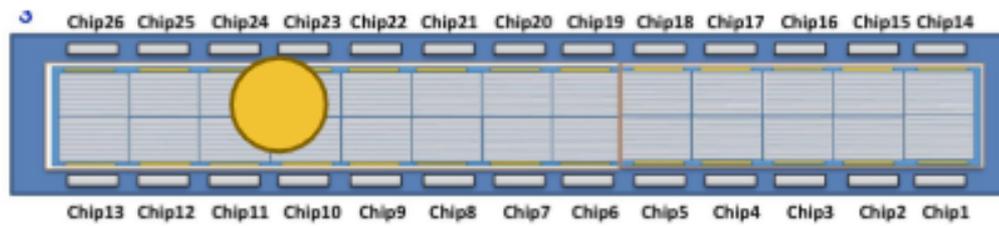


図 5.10 2 分間測定した際のセットアップ

5.4 結果

β 線源による 2 分間の測定結果は以下ようになった。

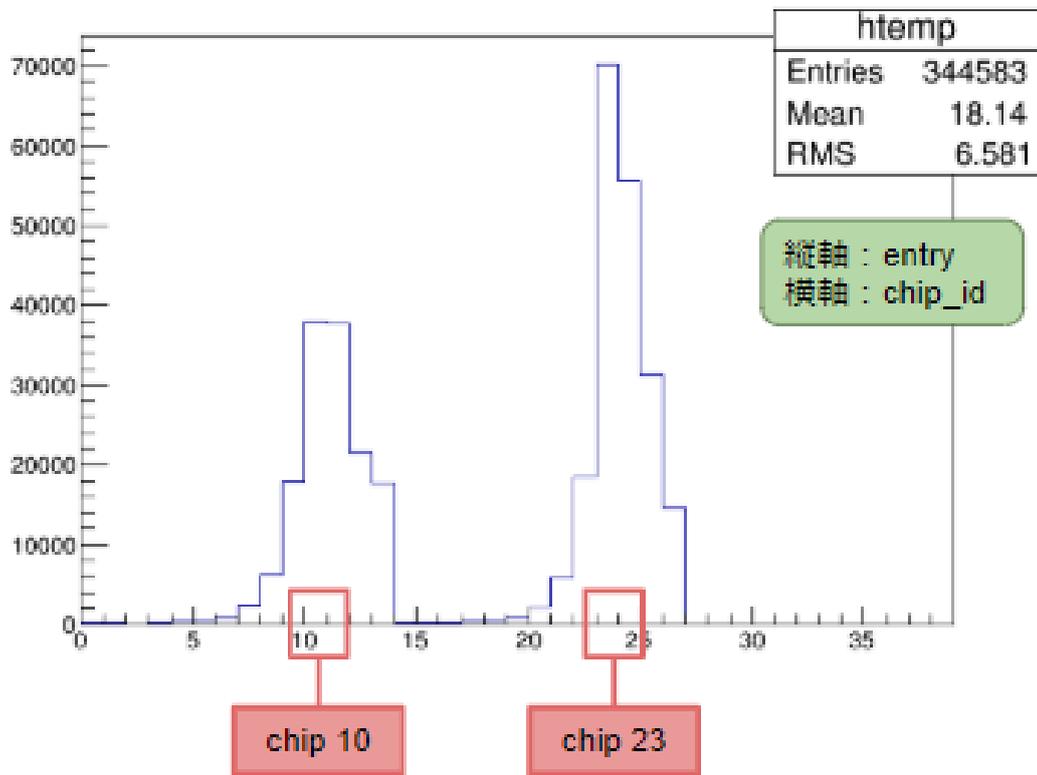


図 5.11 横軸を $chip_id$ とした際の 2 次元ヒストグラム

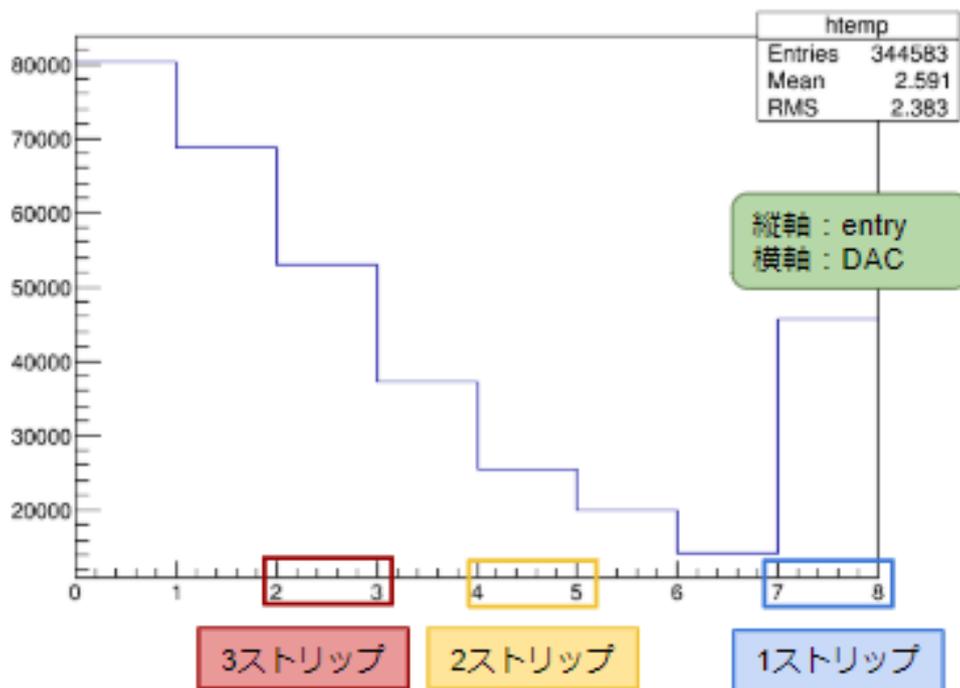


図 5.12 横軸を DAC とした際の 2 次元ヒストグラム

これにより、hit 数が十分に得られていること、また entry が集中している chip が予想していたものと一致していることからシリコンセンサーは β 線源のエネルギーを捉えていることがわかる。しかし、DAC 値は DAC0 の entry が最も多く、予想していたエネルギー損失値よりも低いことが分かる。予想したストリップ別に比べると 1 ストリップと予想した DAC7 には entry が集中しているが 2 ストリップの entry 数が少なく、また DAC7 には閾値以上のエネルギーが全て記録されるため予想通りとは言い難い結果となった。考えられる要因として、シリコンセンサーに到達する前に空気中やシリコンセンサー内で電子のエネルギーを大きく損失している可能性等が挙げられる。

第 6 章

外部トリガーモードを用いた宇宙線測定

6.1 宇宙線測定の目的

衝突時の重心系エネルギーは sPHENIX 実験においては、ゆうに 200GeV を超える。同時に衝突時に放出される崩壊粒子が持つ運動エネルギーも膨大となる。全章で取り扱っていた β 線の最大エネルギーはストロンチウムからイットリウムへの崩壊過程で放出される 2.28(MeV) であった。すなわち、 β 線の崩壊による電子の持つエネルギーと比較すると、加速器実験での衝突時に生成される粒子が持つ運動エネルギーは少なくとも 10^3 倍近くのエネルギーをもつことになる。よって、より、sPHENIX 実験での条件に近づけるために使用すべきプローブは、実験内で再現できる最大のエネルギーをもつ粒子であり、地上付近で MIP が 0.4eV のエネルギーで観測される宇宙線ということになる。よって宇宙線を用いて測定を行い、線源による測定同様、シリコンセンサーの性能評価を行なった。

6.2 外部トリガーモード回路とセットアップ

β 線源を用いた測定では、FPHX チップが送信した信号そのものをセルフトリガーとして使用していた。今回、宇宙線測定の際使用した外部トリガーモードは、宇宙線をトリガーし、シリコンセンサーからのトリガーと、宇宙線からのトリガーのコインシデンス信号を採用するモードである。この測定方法は AND 処理で 2 つの信号が同時になった時のみ採用するため、ノイズを削減することに適している。

宇宙線測定は宇宙線のレートが線源のレートと比較しても、格段に低いため、より長時間の測定が求められ、その分宇宙線由来でないノイズを拾う確率が大きくなる。宇宙線のようにシンチレーターを通過した後も、エネルギーを有することができるほど、エネルギーの大きな測定対象に適した測定方法である。外部トリガーの AND 処理は FEM の基盤上に乗った FPGA で行われ、実験時に FPGA のコードを必要に応じて外部トリガー用のコードにコンパイルし直す必要がある。適

宜、FPGA のコードを対応するように書き換え、測定を進めた。今回使用したシンチレーションカウンタの仕様と性能については下記表 (6.2.1) にまとめた。用いたシンチレータは、シリコンセンサーの一部分のみ覆うサイズで、シリコンセンサー全体を覆っていないことは念頭に置いて評価を進めたい。シンチレーションカウンタの信号レートは、NIM 規格の Visual Scolor を用いて一定時間あたりの信号数をカウントし、概算し求めた数値である。

シンチレーションカウンタ 番号	光電子倍増管品番	陽極-陰極間印加電圧 (kV)	シンチレータ縦 (cm)	シンチレータ横 (cm)	面積 (cm) ²	シンチレータ厚み (cm) ²
A	H3178 HAMAMATU	-1.5(kV)	4.67(cm)	3.69(cm)	17.2(cm) ²	1.00cm
B	H3178 HAMAMATU	-1.5(kV)	4.40(cm)	3.60(cm)	15.8(cm) ²	1.10cm

シンチレーションカウンタ 番号	discriminator でスレッシュホールドを かけた後の信号レート (Hz)	単位面積あたりのレート (Hz)
A	2.961(Hz)	0.1721(Hz)
B	6.155(Hz)	0.390(Hz)

表 6.2.1 シンチレーションカウンタの仕様と性能 [9]

6.2.1 外部トリガーモードブロック図

外部の信号をトリガーするための、回路が追加されたもの。FEM における宇宙線トリガー生成用の外部トリガーモードのコードを FPGA に構築し直した。

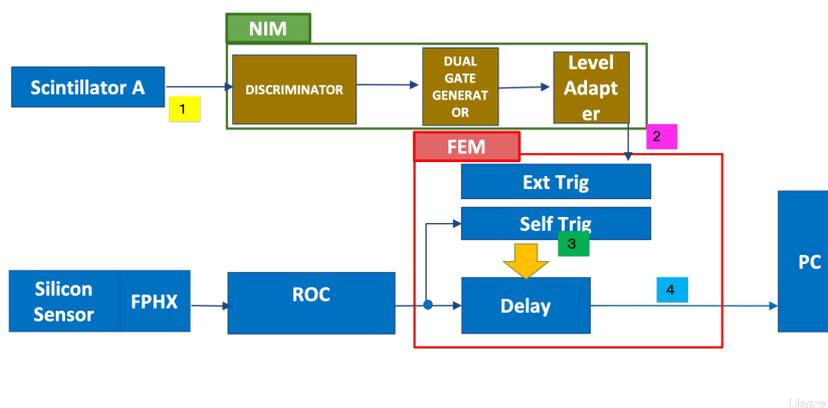


図 6.1 外部トリガーモードブロック図

6.2.2 外部トリガータイミング調節

シリコンセンサーから FEM でセルフトリガー判断が行われる間、FPHX チップにおける信号処理時間、ROC 基盤での信号処理時間分信号は遅延する。この遅延分シンチレーションカウン

タからの信号を FEM のロジック回路へ入るまでに送らせ、AND を取れるタイミングに合わせる必要があった。処理時間は FPHX チップでの信号処理時間 $0.9 \mu\text{s}$ とでの信号処理時間 $1.7 \mu\text{s}$ を足し合わせた $2.6 \mu\text{s}$ である。よって NIM 規格の DUAL GATE GENERATOR モジュールを使用し、delay time と方形波の width を調整した。シンチレーションカウンタのアナログ信号に対するスレッシュホールドは、NIM 規格の DISCRIMINATOR での最低限界である、 -33.4mV まで下げ、シンチレーションカウンタが十分に期待される宇宙線のレートのエントリー数を確保できるようにした。今回スレッシュホールド限界最低値まで下げた理由は、宇宙線として記録された電圧パルスのエントリーの取りこぼしを少なくするためである。

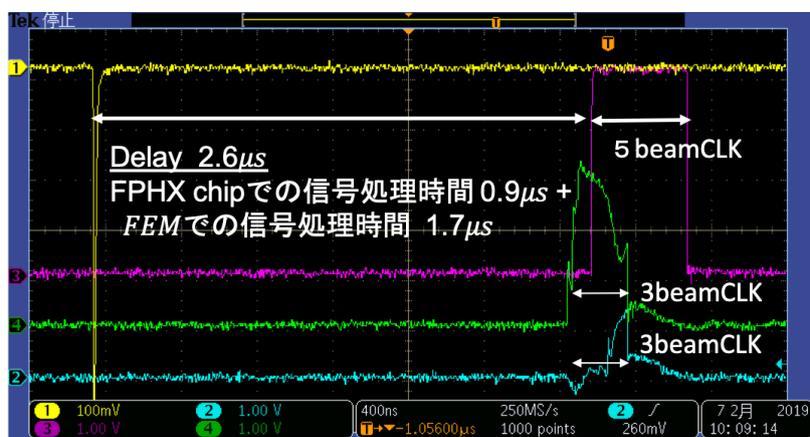


図 6.2 オシロスコープでのタイミング操作

<各オシロスコープ画面出力信号>

1. 黄色 シンチレーションカウンタからのアナログ信号
2. ピンク シンチレーションカウンタからの信号 (黄) が NIM 規格の DISCRIMINATOR DUAL GATE GENERATOR を通過後 LEVEL ADAPTOR で TTL 信号に変換後の信号
3. 緑 FEM テストピンから出力したセルフトリガー発行された Level1 Accept Trigger
4. 青 FEM テストピンから出力した外部トリガー発行後の信号 2 と信号 3 のコインシデンストリガー

ただし、この FEM テストピンはグラウンドを設けておらず、インピーダンスを調整していない信号を取り出しているため、実際は FEM のテストピンからのロジック信号は、3BeamCLK の width の方形波で再現されるはずだが、鋭い波形になっており、幅が安定して出力されない。今回は、コインシデンスタイミングを確認することを目的として使用しているので、オシロスコープ出力波形の不安定さは、気にとめる必要はない。オシロスコープ上で波形を確認し、FEM においての外部トリガ判断が正常に機能しており、シンチレーションカウンタの信号と、シリコンセンサー

からの信号の AND が取れていることがわかった。十分に AND の判断を下すのに最適な delay、width の設定が行えたといえる。

6.3 宇宙線測定条件

β 線源を用いたセルフトリガーモードによる測定とは異なり、宇宙線の場合、地上で観測されるミューオンのエネルギーは十分に大きい。線源を用いた性能評価時に比べ、荷電粒子の空気中、シリコンセンサーでのエネルギー損失は考える必要はない。よって、シリコンセンサーの破損を避けるために、シリコンセンサー本体を、アルミニウム製の保護カバーで覆った状態で測定した。また、この保護カバーは、HDL部分と冷却チューブ接続部分に隙間が空いており、室内灯や太陽光からの光子を遮断するには不十分なため、上から暗幕で厚く覆い測定を行った。また今回は、シリコンセンサーに対し特に冷却は行っていない。

アルミニウム製の保護カバーを付けた状態での外部トリガー宇宙線測定時のセットアップが図(6.3)である。図(6.4)では、後々使用した2台のシンチレーションカウンタを用いた測定のセットアップを示している。シリコンセンサーをシンチレーションカウンタで上部と下部から挟みこむように配置した。1台での宇宙線測定の際は、シリコンセンサーに対し、上部のシンチレータのみ設置した。設置位置は測定の度におよそ一致する位置に設置した。

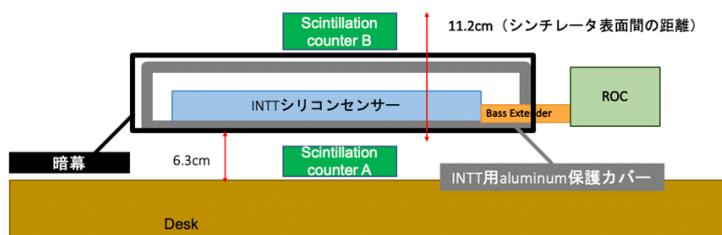


図 6.3 外部トリガー宇宙線測定時のセットアップ図



図 6.4 外部トリガーシリコンセンサーとシンチレーションカウンタセットアップ

地上で測定されるミューオンの MIP のエネルギー損失計算を行い、見積もり値とした。計算過程を示す。

ミューオンの運動エネルギーと、シリコンの厚み $200\mu\text{m}$ に対する反応断面積を示したグラフが第 2 章の図 (2.5) である。このグラフのランダウ分布による most probable energy の MIP を読み取った。グラフより、ミューオンの MIP の運動エネルギーが 0.4GeV の時、シリコン中でのエネルギー損失は、 $1.1\text{MeV}g^{-1}(\text{cm})^2$ (電氣的損失のみの値) である。このことから、シリコンの密度を $2.33(\text{g}/(\text{cm})^3)$ として計算を進めると計算過程は前章までと同様に行った。

$$1.1((\text{MeV})^{-1}(\text{cm})^2) \times 200(\mu)\text{m} \times 2.33(\text{g}/(\text{cm})^3) = 0.0513(\text{MeV}) \quad (6.1)$$

これらの結果から、シングルヒット、ダブルヒットを中心に見積もり値をカバーする範囲で DAC を設定した。それぞれ計算結果を示す。

表 6.3.1 ストリップ通過数別エネルギー損失見積もり値

1event あたりの hit 数	最低損失エネルギー	最大エネルギー損失 (最大飛程で通過時を想定)
1hit	680(mV)	730(mV)
2 hits	730(mV)	862(mV)

表 6.3.2 宇宙線測定における FPHX チップ 3 bitDAC 設定値

DAC 番号	GUI 設定値	FPHX チップ増幅値
DAC0	20	290(mV)
DAC1	25	310(mV)
DAC2	48	402(mV)
DAC3	98	602(mV)
DAC4	148	802(mV)
DAC5	172	898(mV)
DAC6	223	1102(mV)
DAC7	248	1202(mV)

見積もり値をもとに、設定した 3bit の FPHX チップ DAC 値が表 (6.3.2)。

6.4 宇宙線測定結果

表 4.1.1 の DAC 設定で宇宙線測定を行なった結果を示す。様々な条件で測定を行なったため、その条件を示すとともに、結果を示す。

6.4.1 シンチレーションカウンタ 1 台での宇宙線測定結果

約 1 8 時間 (1082 分間) 連続で測定した結果である。図これまでの測定で設定していた DAC 値では、DAC 0~DAC 3 に集中してエントリーが見られた。また、シングルヒットで、電圧パルスの大きさが 300 400mV のエントリーが非常に多い結果となった。実際宇宙線が由来のエントリーは MIP の場合 680mV 付近の DAC6 や DAC7 に入ると見積もっていたが、矛盾した結果となった。

chip ごとのエントリー数を視覚的に確認するためのグラフを図 (6.5) に示す。シンチレータを設置した位置と比較しやすくするため 模式的なシリコンセンサーのイラストに、赤線でシンチレータを設置したおおよその位置関係を示した。このグラフは横軸 channel0 127 ののヒストグラムである。グラフから、シンチレータが重なっていたチップ周辺がアクティブになっていることがわかる。

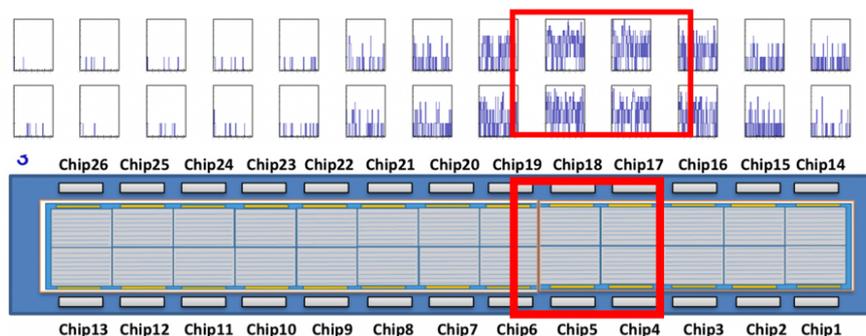


図 6.5 1 台のシンチレータ を用いた宇宙線測定の chip ごとのエントリーの分布

さらにチャンネルごとのエントリーを区別せずチップごとのエントリー数の分布のグラフを図 (6.6) に示す。グラフからシンチレーションカウンタを設置箇所位置するシリコンセンサー区域に対応するチップ両サイド chip6、chip18 を peak に分布していることがわかる。このことから、外部トリガーシステムが正常動作していることが確認できた。同時に、全 26chip に対する channel のエントリー分布がグラフ図 (6.7) である。このグラフ図 (6.7) から、特に著しくノイズを拾う channel は見受けられず、特異的な反応差異は見られなかったことから全 chip、全 channel に対して mask はせず、測定を進めていくことにした。

次に ADC 分布図を分析した。見積もり値から、測定結果としてシングルヒットが期待される DAC 3、ダブルヒットが予想される DAC 0~DAC 3 にヒットが溜まっている結果を期待してい

た。この分布図から判断すると、ADC2 に peak に持ち、見積もり値の範囲内にエントリーが分布していることがわかる。この時点では見積もり値に合致した分布を得られたと判断できる。ただし通過ストリップ数別にイベントを評価していく必要がある。

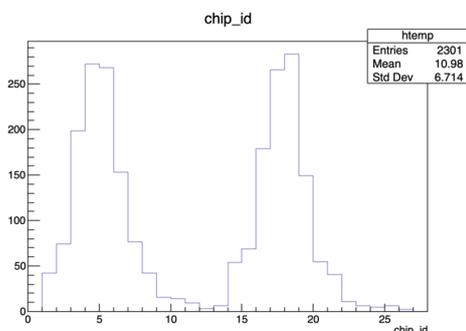


図 6.6 宇宙線測定時の chip に対するエントリー分布

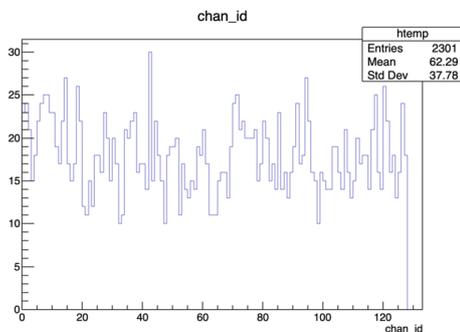


図 6.7 全 26chip に対する channel のエントリー分布

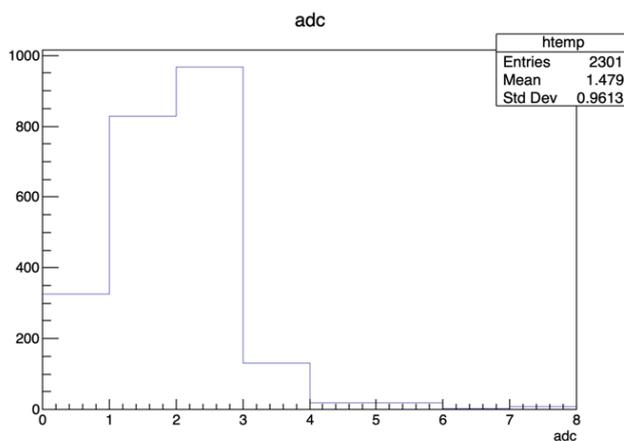


図 6.8 全 26chip に対する ADC 分布

6.5 シンチレーションカウンタ 2 台を用いたコインシデンス外部トリガーを用いた宇宙線測定

今回、ノイズの削減を試みるため、シリコンセンサーの上下にシンチレータ部分が重なり合うように、2 台のシンチレーションカウンタを設置した。NIM モジュールの 1 つである TRIPLE 4-FOLD 1-VETO Coincidence を導入し、この 2 台のシンチレーションカウンタからのシグナルのコインシデンスをとり、このコインシデンス信号をトリガーとして採用した。シンチレーションカウンタを通過後、シリコンセンサーに入射する粒子の分布範囲は、シンチレータの立体角に依存

する。よって、シリコンセンサーとシンチレーションカウンタの距離が大きいほど、分布領域が広がると言える。今回、シンチレーションカウンタ2台でシリコンセンサーを上下で挟むことで、chipのエントリー分布範囲が絞られ、より精度の良い測定が可能であると考えた。

6.5.1 測定セットアップ

シンチレーションカウンタとシリコンセンサーの回路ブロック図と、その位置関係を下に示す。

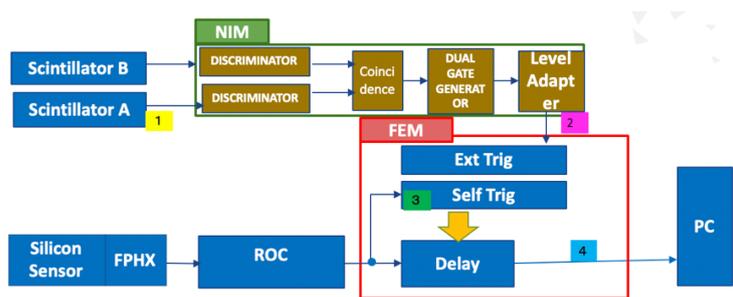


図 6.9 2 台のシンチレーションカウンタのコインシデンス信号外部トリガーブロック図

2 台のシンチレーションカウンタからの信号をそれぞれ Discriminator に入力し、スレッシュホールドをかけ、その後 2 つの信号が 1 つの Coincidence モジュールへ入力される。このモジュールが AND 回路担っており、信号が同時に鳴ったタイミングで信号検出をする。

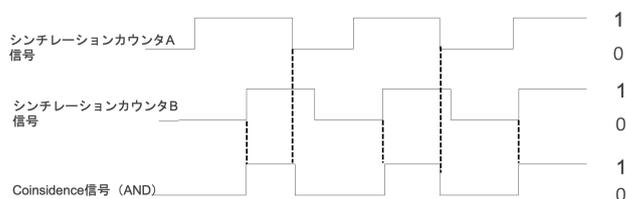


図 6.10 コインシデンス波形論理

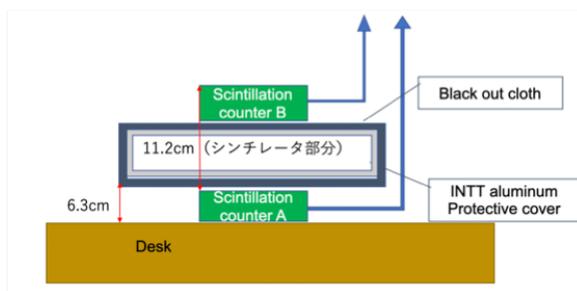


図 6.11 シンチレーションカウンタとシリコンセンサーの位置関係

6.5.2 測定結果

2台のシンチレーションカウンタでシリコンを上下で挟むと、1台のみのシンチレータで宇宙線トリガーをかけていたときと比較して、宇宙線のエン트리分布範囲をシンチレータ部分に絞り込んでいることがわかる。よって結果から、2台シンチレーションカウンタによるコインシデンス外部トリガーは、ノイズをより減らす必要がある DAC 0 最低閾値を低くした測定などで有効なセットアップ手段であるとわかった。chip のエン트리範囲が1台のシンチと比べ、より、シンチレータの設置範囲に絞り込んでいる。この結果から2台のシンチレーションカウンタを用いた外部トリガー測定において、ノイズを減らすための方法として有効的であると言える。

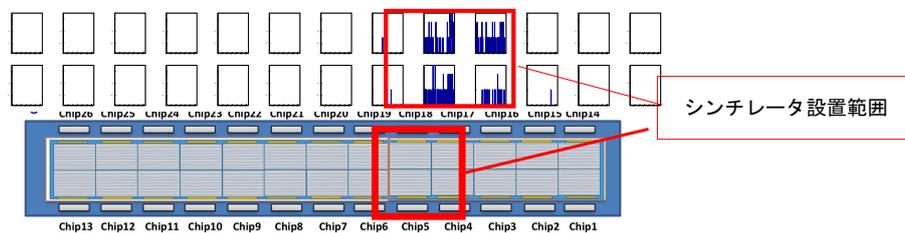


図 6.12 2台のシンチレーションカウンタ宇宙線測定時の chip に対するエン트리分布

6.6 710mV 以下の低い電圧値における DAC 分布の解析

第4章から分かるように、分布は、ほぼ 600mV 以下に分布していることがわかる。どの範囲に分布するかを特定したのち、3bitDAC を 710mV 以下でリニアな値に設定し、より低い電圧値の分布を得ることにした。

6.6.1 測定条件

DAC 番号	GUI 設定値	FPHX プリアンプ増幅値
DAC0	20	290(mV)
DAC1	35	350(mV)
DAC2	50	410(mV)
DAC3	65	470(mV)
DAC4	80	530(mV)
DAC5	95	590(mV)
DAC6	110	650(mV)
DAC7	125	710(mV)

表 6.6.1 宇宙線測定における FPHX チップ 3 bitDAC のリニアな設定値

DAC をで 40mV 間隔で設定した。シンチレーションカウンタ 1 台をトリガーとして使用した。測定時間は 1350minutes(約 22 時間 30 分) である。アルミニウム保護カバー、暗幕等のセットアップは前章までと同様である。

6.6.2 測定結果

宇宙線は 1 分間約 2 エントリーの頻度。chip6、chip20 に 2 つのピークを持って分布しており、これは FPHX チップ両サイドに対応していることがわかる。

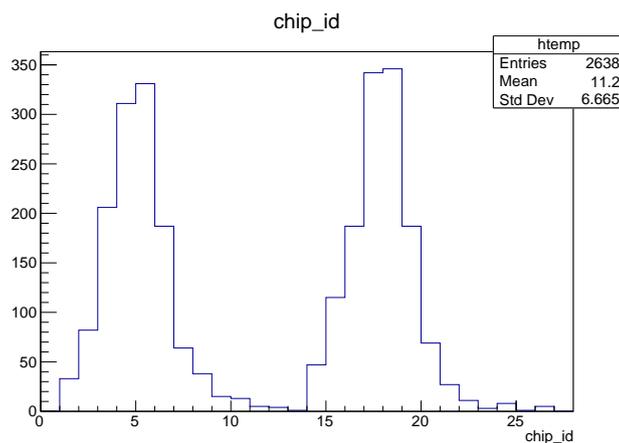


図 6.13 DAC 値を低い方で線形的に設定した宇宙線測定時の chip に対するエントリー分布

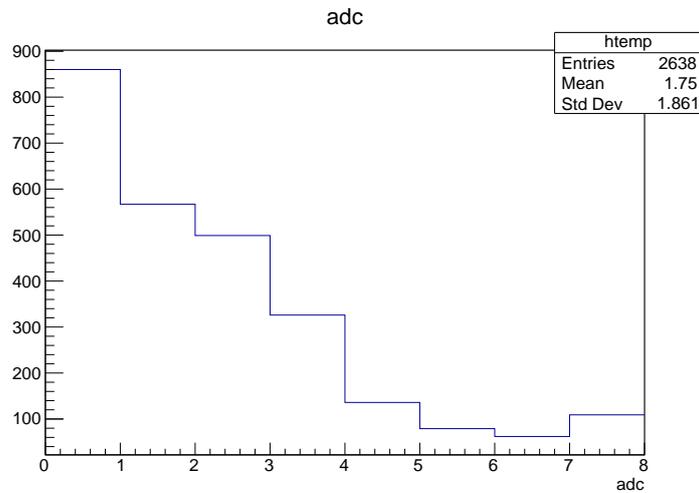


図 6.14 DAC 値を 710mV 以下でリニアな間隔で設定した宇宙線測定時の ADC 分布

結果から、chip のエントリー分布は、シンチレーションカウンタがシリコンセンサーと重なっている付近の両サイドに対応する chip をピークに分布を示している。これから、2 台のシンチレーションカウンタによる宇宙線測定においても、外部トリガーモードが正常に動作していることが確認できた。また entry 数は 2638entry と極めて少なかった。かつ、ADC 分布グラフから見ると、とりわけ DAC 0 にエントリーが集中していることがわかる。これは、290mV 以下の比較的小さなエネルギー損失を起こしたエントリーが多いことを示している。やはり、また、DAC5,DAC6 といった 530mV 650mV のエネルギー損失を起こしたイベントの数は比較したところ、極めて少なかった。DAC7 (710mV 以上) に相当するエネルギー損失を起こしたエントリーが再び増加しているのは、710mV から上限を設けていないため、710mV 以上のエントリーが DAC7 に集約されたためと考えられる。閾値設定をリニアにしたところ、綺麗に階段状にエントリーが減少した。ただしこのグラフからはシングルヒットが溜まったているのか、複数ヒットしたイベントがどの DAC に分布しているのかといったイベントの詳細を把握できない。よって次章第 6 章において宇宙線測定におけるクラスター判別について言及した。

第 7 章

イベント判別と解析方法の検討

7.1 宇宙線測定におけるのクラスター判別

7.1.1 クラスター判別の必要性

シリコンセンサーをミューオンが通過するとき、シリコンセンサー表面に対して垂直方向に入射する場合、1chip の 1strip のみ通過することが多いと考えられるが、実際入射角が垂直でない場合が大半である。この場合、複数の strip にまたがって、entry が記録される。この 1event の情報を得ることで、ミューオンのシリコン中でのエネルギー損失の状態と、シリコンセンサーの動作確認を行うことができる。

7.1.2 クラスター判別方法と解析コード

測定したデータがブランチ構造になっており、同じ beam CLK で記録された entry を、1event ずつ区別し、1event ごとの strip 通過数を得た。その結果をグラフに出力し、分析した。その分析から色々と条件を変え検証を行なった。また、前章で計算したように、エネルギー損失をシリコン中を粒子が通過する飛程から算出し、おおよその分布見積もりをもとに、結果と一致しているかどうかの評価を行なった。

```

Beam CLKタイミング
1961 clk:1 46465 Chip番号 1hit 1
n=1 adc=356
1962 clk:21 26389 : 18 26 : 6 1hit 1
1963 clk:21 26389 : 18 27 : 6 2hit 2
1964 clk:21 26389 : 3 41 : 6 2hit 3
1965 clk:21 26389 : 3 42 : 6 2hit 4
1966 clk:21 26389 : 3 46 : 6 1hit 5
1967 clk:21 26389 : 3 48 : 6 1hit 6
1968 clk:21 26389 : 3 50 : 6 0 7
1969 clk:21 26389 : 3 51 : 6 1hit 8
1970 clk:21 26389 : 3 52 : 6 0 9
n=9 adc=3530
1971 clk:58 33722 : 4 44 : 6 6hit 1
n=1 adc=1152

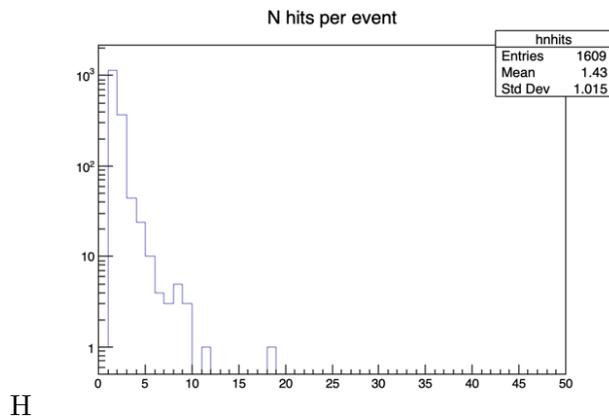
```

図 7.1 イベント判別方法

	1strip	2strip
最低電圧値	680(mV)	730(mV)
最高電圧値	730(mV)	862(mV)

表 7.1.1 通過ストリップ数ごとの期待されるエネルギー損失に対応する FPHX チッププリアンプ増幅電圧値

通過した MIP のミュオンの合計電圧値の分布が 1strip 通過イベントの分布の見積りの範囲に分布していない。見積もり値に一致しない原因は最低閾値である DAC 0 の値が高いため、DAC 0 以下のエントリーを取りこぼしていると考えた。よって、再度、DAC0 の値を引き下げて、再測定を行なった。DAC 0 を引き下げるということは、低い電圧値に対応するノイズをより多量に拾うことになる。懸念されるのはノイズにより、宇宙線由来の正常なデータが埋もれてしまうことである。今回はその最低閾値限界を探りながら、かつシンチレーションカウンタ 2 台のコインシデンス信号を外部トリガーとし、宇宙線由来でないノイズを減らし宇宙線測定を続けた。この時は、シリコンセンサーを挟み込むセッティングではなく、シリコンセンサー上部に 2 つのシンチレーションカウンタが重なるように設置した。



H

図 7.3 イベントごとのストリップ通過数

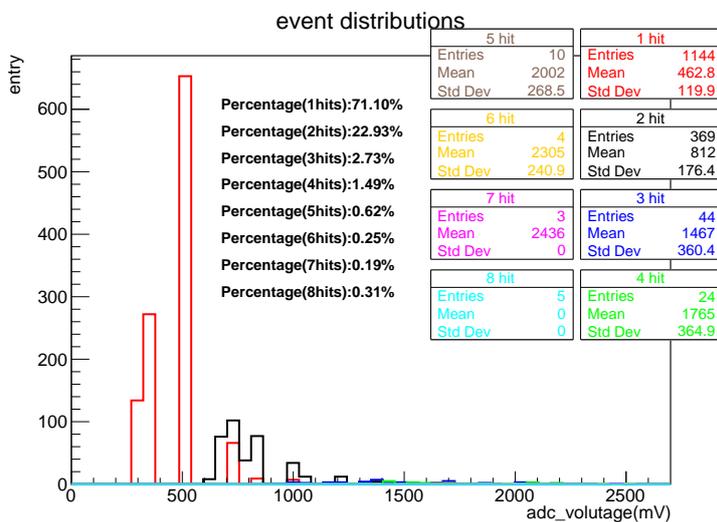


図 7.2 通過ストリップ数別波高分布

通過ストリップ数別エントリー数分布が図 (7.2)。赤色が 1strip 通過したイベント、黒色が 2strip 通過したイベントを示している。今回はエントリー数の多い、2strip を通過したイベントまでを議論することにしたが、実際は 3 ストリップ以上通過する event もある。横軸 1event が全行程で損失したエネルギーに対応する FPHXchip 電圧値である。各 8strip 通過事象までのヒストグラムを重ね書きして表示している。この図 (7.3) から、実際エネルギーがストリップに分配され、DAC0 以上のエネルギーを損失する確率が非常に低い 4 hit 以上のイベントが複数確認できる。

7.1.3 最低閾値 DAC0 を引き下げた測定結果

通過した MIP のミュオン合計電圧値の分布が 1strip 通過イベントの分布の見積りの範囲に分布していない。400(mV) ほど低い位置に多くのエントリーが見られる。DAC 0 以下のエントリーを取りこぼしており、本来 2 strip 以上のイベントとして認識されるべきイベントが、1strip 通過したシングルヒットとして数え上げられてしまっていると考えた。

ここで DAC0 の値を引き下げ、2hit 以上のイベントの割合が変化するか否かの検証を行った。

DAC 0 を 10 まで引き下げた測定を行なった。DAC0 = 10 の場合、絶えずノイズが発生し、長時間の測定で DAQ がフリーズすることを懸念し、2 つのシンチレーションカウンタによるコインシデンス外部トリガーでノイズを減らしつつ測定を行なった結果を示す。

図 (7.1.3) より、DAC0 へのエントリー数の割合が格段に増加したことがわかる。次に chip 対 channel で 2次元 hitmap を作成し、この DAC0 のエントリーが宇宙線由来であるか、ただの機械的なノイズか否かの確認を行った。図 (7.6) より、得意的に channel124 127 付近に hot channel があることがわかる。channel の一部分が過剰に反応する事象は channel 不良などのハードウェア上の問題と考えられるため、実際解析を行う際は、channel123 125 のエントリーは全 chip 分カットし、グラフを出力した。Hotchannel を省いて出力したイベント別エントリー数のグラフが図 (7.7)DAC 0 を引き下げたときの分布を見ると、1strip のエントリーの割合が微小に減少し、2hits のイベントの割合が増加している。ただし、最低閾値を引き下げるにあたりノイズの影響から 2 台のシンチレーションカウンタのコインシデンストリガーを外部トリガーとして用いる設定に変更している。条件を複数変更しているため、今後より条件を淘汰し、測定を行うと、精密なデータを取得できると予想する。

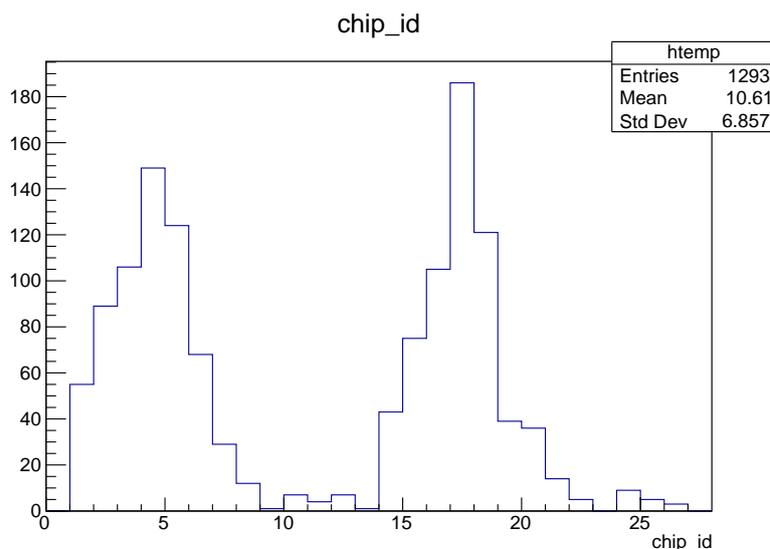


図 7.4 DAC0=10 の時の chip 分布

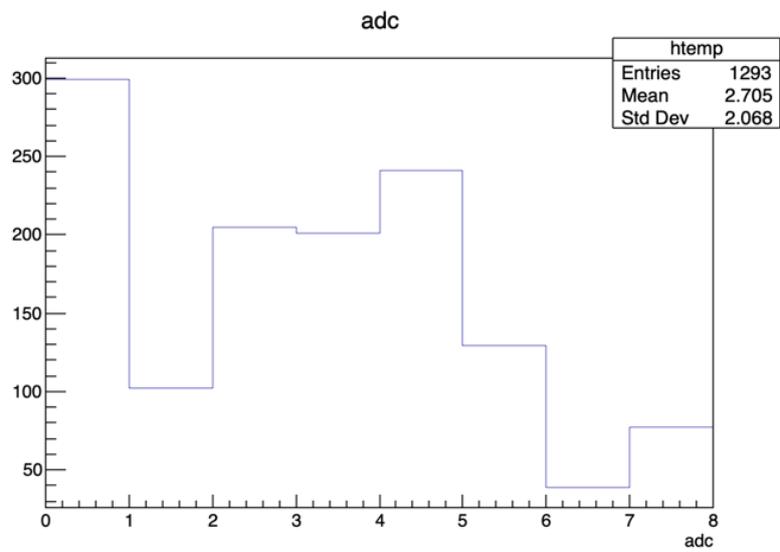


図 7.5 DAC0=10 の時の ADC 分布

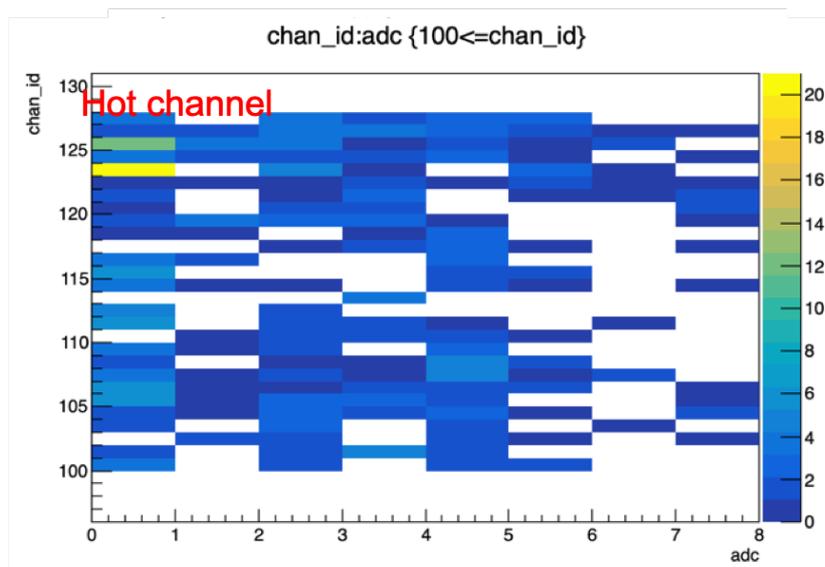


図 7.6 x 軸 : chip y 軸 : channel の hitmap

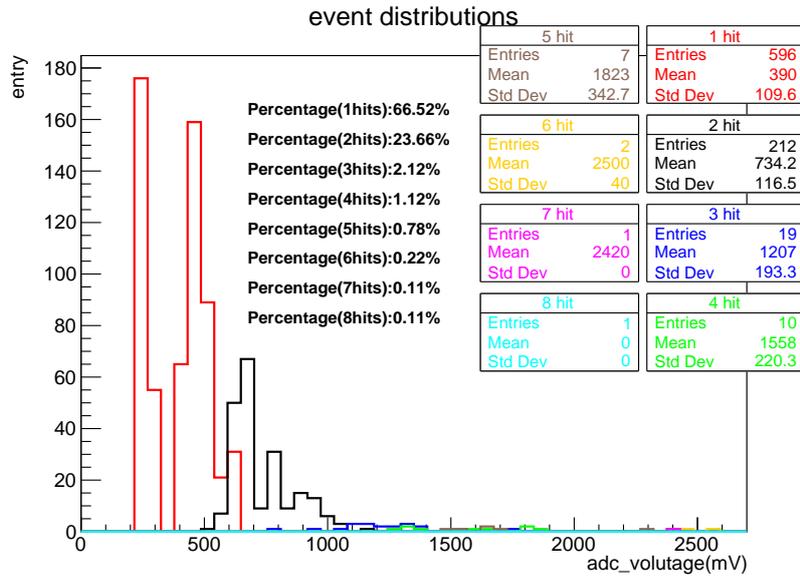


図 7.7 hotchannel を除いたイベントごとのストリップ通過数

7.2 宇宙線測定における今後の課題

今後、外部トリガーモードによる測定効率を上げ、高度な性能評価を行なっていくために、今回研究で明らかになった今後取り組むべき課題をまとめた。

今回は、DAQ の動作や、DAC 値の最適化を目指した。現時点で今後実際に宇宙線のエネルギー損失と、シリコンセンサーの反応動作をより精度を高めて評価するには、現在の統計数では不足している。分布の傾向を結論づけるには、やや説得力に欠ける。測定条件をさだめ、数日の測定データを得る必要がある。

測定を続けていくにあたり、安定して測定できる条件を確認することができたので、その条件のもと実験を行なっていくと良い結果が得られると考えている。現在は、18時間ほどで、約2300entryであり、それぞれの計測条件を維持して3日間連続稼働させた場合、約10000entryの計測である。次のステップとしては、長時間計測を継続出来る環境を整える、もしくは、分割して計測したデータを足し合わせてエントリー数を確保することが求められる。

他、案として測定効率を上げるための複数台のシンチレーションカウンタの OR 回路を外部トリガーとして採用し、宇宙線測定を行い、測定効率を向上するののも一つである。

複数のイベントを同一イベントとして数え上げ、エネルギー損失が非常に大きな event として判別されてしまう問題に対しては判別コードに問題がある。ストリップ数別電圧対応の分布を見るコードは現在同じ Beam CLK で記録されたエントリーを 1 event とする方法を取っている。このコードにイベント識別に隣同士のストリップの呼び出し条件を追加し、クラスターの判別をより正確に行うためのプログラムを作成するべきである。今回の外部トリガーモードにおける宇宙線測定は、手探りで進めたこともあり、性能評価の基準を見出すことが難しかった。また、PC が長

時間測定を継続すると予期せぬシャットダウンを起こす問題も未解決である。また、シリコンセンサーに高電圧を印加する時間や、室温、湿度の変化によりシリコンセンサーの電流値が大きく変動する問題など、不安要素が多い点も課題の一つである。

第 8 章

結論

今回、奈良女子大学における INTT のテストベンチの構築が済み、問題なくデータを得るところまでの環境を整えることができた。データ読み出しシステム、解析までの動作の一連の流れと、DAQ、ハードウェアの理解、シリコンセンサーの特質とエネルギー損失の関係についての知見を広げることができた。ノイズ評価と、 β 線源と宇宙線による性能評価結果から、 β 線と宇宙線両方共に損失するエネルギーの電圧値がノイズの大きさ (9.19mV) に対して十分大きく、シリコンセンサーやチップの信号処理におけるノイズの影響は考慮する必要はないと評価した。

宇宙線測定では、今回の論文中には記載していないが、FVTX 用シリコンセンサーをしようし、その挙動と、最適 DAC 値の見積もりを行い、慎重に測定を進めた。その甲斐もあり、見積もりを元に INTT 用シリコンセンサーの ADC 閾値を適切に設定し DAC 値の最適化を行い、シリコンセンサーの十分な応答を確認することができた。また、シリコンセンサー、テストベンチ回路由来のエラーやノイズの削減に向け、測定条件、解析条件の見直しを行い、実験と結果から日々研究を発展させていくことができた。今後の研究へとつながる課題を見出し、まとめ、繋げることができた。

謝辞

私たち4年生3名、ハードウェアに対する知識がゼロの段階からのスタートでした。手探りでのスタート切り、何から手をつければいいのかわからない中、多くの方のお力をお借りし、卒業研究として成し遂げることができました。sPHENIX 実験で取り扱う物理についてから、シリコンセンサー、また VHDL 言語、解析方法と広い範囲の物理について学ぶことができ、知見を広げることができたと感じています。

テストベンチの構造と、基礎物理、データを取るにあたり、ハードウェアの操作方法や不具合の解決方法を教えてくださった理化学研究所の山口さん、未熟であった私たちに手取り足取り教え、助言してくださった鈴木先輩に心より感謝申し上げます。そして蜂谷先生は、実験の進め方と、課題の相談にのっていただき、私たちを常に後押しし、指針を示してくださいました。林井先生、宮林先生、下村先生には、研究室ミーティングにおいての進捗報告で、図を用いて丁寧に検出器の仕組みを教えていただき、支えていただきました。また、INTT グループの中川さん、轟木さんをはじめ、学生ミーティングにおいて、私たちが悩んだことや、見逃している測定結果等、多くの方々から鋭い意見をいただき、前進することができた実験生活でした。心より、感謝申し上げます。色々と3人で実験、解析し話し合う途中、お互いが衝突し、意見をまとめることが難しく、悩んだ時期もありました。しかし、この経験も今後共同実験に参加する機会において、協力し互いの意見を尊重し、まとめる力につながる貴重な経験であったと思います。ハードウェアと向き合い、制御の難しさとその繊細さ、精巧さを感じ、シリコンセンサーモジュールを自らの手で扱い、性能評価させていただけたこと、大変ありがたく思います。

今後私たち3名それぞれ、異なる道を歩みますが、引き続き精進してまいります。ありがとうございました。

参考文献

- [1] 益田英知 RHC-sPHENIX 実験におけるシリコンストリップ検出器の開発, 立教大学修士学位論文
- [2] 長島徹 RHC-PHENIX 実験 Run15 における FVTX 検出器を用いた高多重度トリガーシステムの開発と陽子+陽子衝突における方位角異方性の検証, 立教大学修士学位論文 (2015)
- [3] 秋葉康之 クォーク・グルーオン・プラズマの物理 実験室で再現する宇宙の始まり (2014)
- [4] Brookhaven National Laboratory The Physics of RHIC
<https://www.bnl.gov/rhic/physics.asp>
- [5] Leo, William R. Techniques for Nuclear and Particle Physics Experiments
- [6] Particle Data Group, 33. Passage of particles through matter 1
<http://pdg.lbl.gov/2018/reviews/rpp2018-rev-passage-particles-matter.pdf>(2015 改訂)
- [7] Sarah Campbell sPHENIX: The next generation heavy ion detector at RHIC,
arXiv:1611.03003v1(submitted on 9 Nov 2016)
- [8] 堀桂太郎 図解 VHDL 実習 [第2版]-ゼロからわかるハードウェア記述言語- (2009)
- [9] 浜松ホトニクスウェブサイト 光電子増倍管アッセンブリ H3178-51
<https://www.hamamatsu.com/jp/ja/product/type/H3178-51/index.html>
- [10] J.J Beatty (Ohio State Univ.), J.Matthews (Louisiana State Univ.), and S.P. Wakely (Univ. of Chicago)
30.Cosmic rays <http://pdg.lbl.gov/2017/reviews/rpp2017-rev-cosmic-rays.pdf>(2017 年 10 月改訂)
- [11] The Forward Silicon Vertex Detector (FVTX) Group Sensor Specifications Document- FVTX Sensor Specifications
<https://www.phenix.bnl.gov/WWW/fvtx/DetectorHardware/>
- [12] HuMANDATA XILINX 社 FPGASpartan6 搭載 教育用ボード
<https://www.hdl.co.jp/EDX-007/index.html> (2018-8-2)
- [13] ザイリンクス FPGA トレーナ EDX-007 カタログ 回路図 (B2)
<https://www.hdl.co.jp/ftpdata/EDX-007/EDX007R1-SCH-B2.pdf>
- [14] XILINX ISEWebPACK 開発ソフトウェア

<https://japan.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.html>
Sencers (2012 年 4 月 27 日更新)

付録 A

FPGA トレーニング

今回 INTT 環境構築と同時進行で、INTT のハードウェア上でも利用されているロジック回路である FPGA の制御方法を理解する目的で、INTT とは別の機材を用いてハードウェアの制御方法の学習を行なった。

A.1 INTT における FPGA の活用状況

s-PHENIX 実験における INTT 検出器では、シリコンでキャッチされた粒子の信号がアナログ信号として、FPHX チップに送信され、DAC 機能により、デジタル信号に変換される。その後、INTT モジュール上の全 26 chip が収集したデータがバスエクステンダー経由で ROC 上の FPGA で集積され一時的に同期し、その後、FEM へ送信され FEM 上の FPGA で遅延時間やデータの同期、フォーマット変換が行われる。実際の加速器実験において一時的に取り扱うデータは非常に膨大であり、これらのハードウェアを制御するプログラムは、効率よく、無駄の少ない言語で高速化されなくてはならない。実際、INTT の読み出し回路の 1 つである FEM には、Xilinx 社の FPGA (Vertex 4) が採用されている。Vertex 4 は、2006 年に発売された FPGA であり、高速処理が可能となったモデルである。

A.2 FPGA 活用トレーニング

A.2.1 FPGA トレーニング教材

FPGA の動作確認と、そのプログラム方法を確認するため、Human Data 社が開発した FPGA 教育ボード EDX-007 [12] を用いて FPGA の制御を試した。

A.2.2 VHDL 言語

VHDL 言語はハードウェア記述言語 HDL の一種である。1980 年代にアメリカで開発された。この言語の特徴は、複数のプロセスを同時処理できることで、一度に多量のデータの処理が必

要なハードウェアの利用目的に理にかなった言語と言える。

A.2.3 開発環境のセットアップ

まず初めに、FPGA の動作、プログラム方法を学ぶ目的で、Human Data 社が開発した FPGA 教育ボード EDX-007 を用いて FPGA の制御のトレーニングを行った。

開発環境は、Xilinx 社が無償で提供している ISE Design Suite14.7 を使用した。

・ ISE Design Suite(バージョン 14.7)

ISE は、Windows XP/7Server Linuxoperating system がサポートされている。今回は、windows7 を使用し、バージョン 14.7 のソフトウェアをダウンロードした。

・ ISE WebPACK [14]

ISE の多くのデザイン機能が収録された ISE のソリューションパッケージ
含まれる機能

機能名	基本機能
Project Navigator	実際にコード編集を行う。
PlanAhead	I/O Pin の割り当てる。 RTL ソースを作成し、作成コードに対応する回路図を展開。
ISE Simulator(Isim)	HDL シミュレーター。デバック機能が付いている。 project navigator で作成したコードと同期することができ、 clock タイミングや、波形の確認を行う。
XST 合成ツール	作成した HDL コードの論理合成を行う。

・ BBC(Bit-Bang-configuration)

FPGA コンフィギュレーションツール。HUMAN DATA 社のホームページから無償でダウンロード可能。BBC は、PC と FPGA ボードを USB ケーブルで接続するだけで、コンフィギュレーションできるアプリケーションであり、特殊なダウンロードケーブルを必要とせず、極めて扱いやすい。ただし、コンフィギュレーションできるファイルの形式は bit ファイルに限られている。

EDX-007 は、FPGA トレーニング用として開発されており、幅広い利用層が取り扱いやすい使用になっている。そのため、FPGA ボード本体と、接続用 USB ケーブルの 2 点がセットになっている。この USB ケーブルを PC 本体と接続するだけで、PC からの電源供給により、稼働する。電圧は所用 PC に依存している。上述したとおり特殊なケーブルが必要なく、初期設定が簡易的である。搭載 FPGA は Xilinx 社の、Spartan-6 である。

機能名	規格
USB コントローラ型	FT232H
速度	480Mbps
LED ライト数	8 個
7セグメント LEDbit 数	4 個
Input スイッチ数	4 個
ブザー	1 個

表 A.2.1 EDX-007 仕様

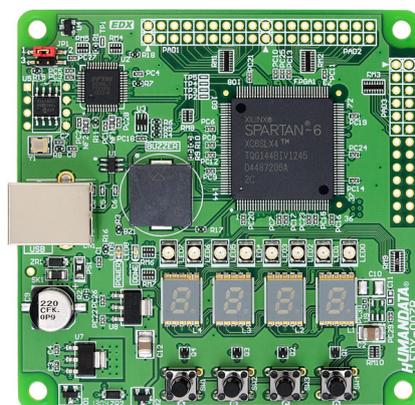


図 A.1 EDX-007 基盤 [12]

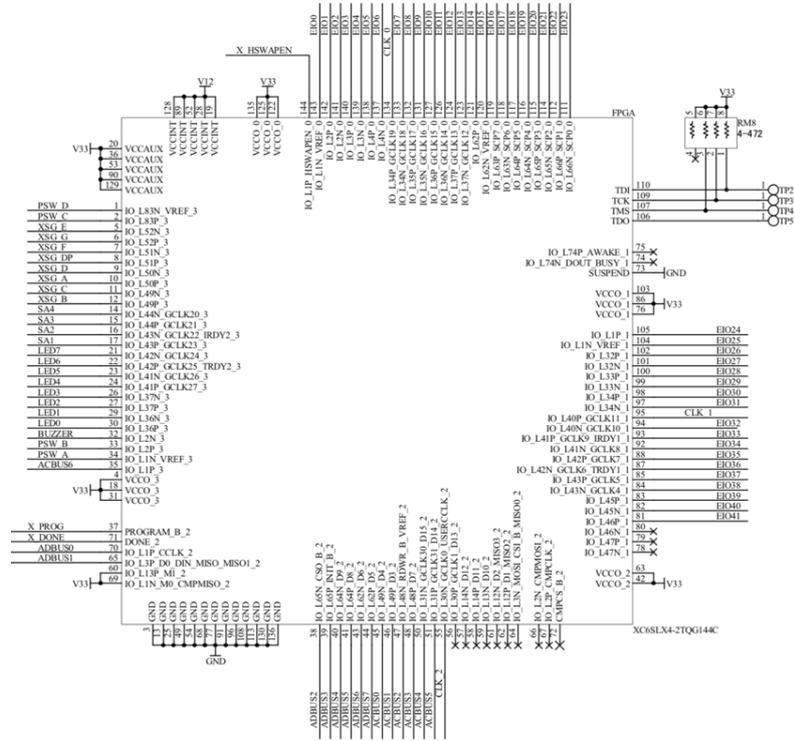
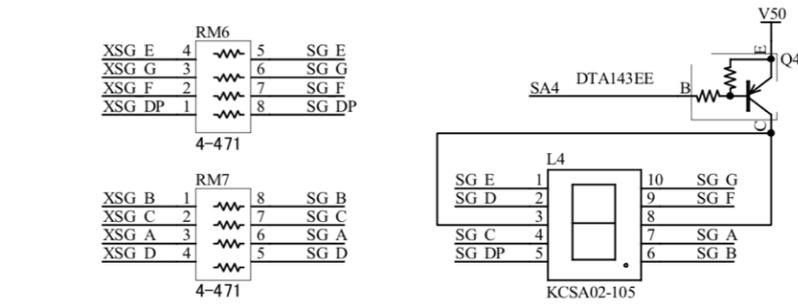


图 A.2 LED 基盤回路图 [13]



7 Segment LED

图 A.3 7SegmentLED 回路图 [13]

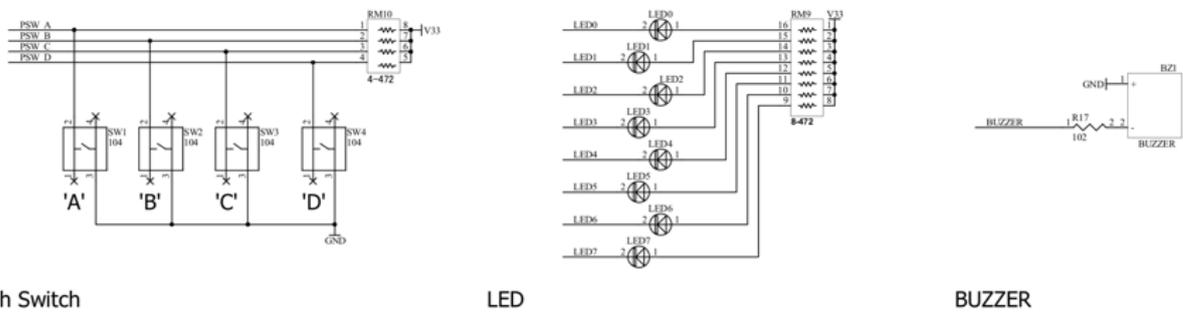


図 A.4 LED・BUZZER 回路図 [13]

A.2.4 トレーニング内容

EDX-007 でのトレーニング内容について述べる。様々なプログラムを作成したがそのうちの複数をここに記す。

1. スイッチを押すとそれぞれに対応した、ブザー音を出力するプログラム
2. スイッチを押した回数をカウントし、そのカウント数を 4 つの 7 セグメント LED に数字を出力させるプログラム。

カウントの数字は 16 進数で行い、1 から F まで出力した後は、1 にもどり、繰り返して行くプログラム。

A.2.5 プログラムコード実行結果

様々なコードを試し、動作を学習したが、その様子を複数示しておく。



図 A.5 7segment 点灯パターン 1



図 A.6 7segment 点灯パターン 2

A.3 FPGA コード

A.3.1 プログラム 1

4 セグメント点灯

```

entity segcounter2_20181025 is
port(A1:in std_logic;
      A2:in std_logic;
      A3:in std_logic;
      A4:in std_logic;
      led7sa:out std_logic_vector(3 downto 0);
      led7seg:out std_logic_vector(7 downto 0)
      --buzzer:out std_logic;
      CLK:in std_logic);
end segcounter2_20181025 ;

architecture Behavioral of segcounter2_20181025 is
signal counter:std_logic_vector(13 downto 0):="00000000000000";
signal chat:std_logic;
shared variable count1:std_logic_vector(4 downto 0):="00000";
shared variable count2:std_logic_vector(4 downto 0):="00000";
shared variable count3:std_logic_vector(4 downto 0):="00000";
shared variable count4:std_logic_vector(4 downto 0):="00000";

begin

```

```

process(CLK)begin
  if (CLK'event and CLK='1') then
    counter <= counter + '1';
  end if;
end process;

process(counter)begin
  if(counter="11111111111111")then
    chat<='1';
  else
    chat<='0';
  end if;
end process;

process(A1,A2,A3,A4,chat)begin

if(chat'event and chat='1')then
  if(A1'event and A1='1') then
    count1:=count1+'1';
  end if;
  if(count1="10000") then
    count1:="00000";
  end if;
end if;

end process;
process(A2,chat)begin

if(chat'event and chat='1')then
  if(A2'event and A2='1') then
    count2:=count2+'1';
  end if;
  if(count2="10000") then
    count2:="00000";
  end if;
end if;

end process;
process(A3,chat)begin

if(chat'event and chat='1')then
  if(A3'event and A3='1') then
    count3:=count3+'1';
  end if;
  if(count3="10000") then
    count3:="00000";
  end if;
end if;

```

```

end process;
process(A4, chat)begin

if(chat'event and chat='1')then
  if(A4'event and A4='1') then
    count4:=count4+'1';
  end if;
  if(count4="10000") then
    count4:="00000";
  end if;
end if;

end process;

process begin
wait for 50 ms;
case count1 is
  when "00000" => led7sa <= "0ZZZ";
                    led7seg <= "11000000";--0
  when "00001" => led7sa <= "0ZZZ";
                    led7seg <= "11111100";--1
                    --buzzer <= counter(12);
  when "00010" => led7sa <= "0ZZZ";
                    led7seg <= "10010010";--2
                    --buzzer <= counter(13);
  when "00011" => led7sa <= "0ZZZ";
                    led7seg <= "10011000";--3
                    --buzzer <= counter(14);
  when "00100" => led7sa <= "0ZZZ";
                    led7seg <= "10101100";--4
                    --buzzer <= counter(15);
  when "00101" => led7sa <= "0ZZZ";
                    led7seg <= "10001001";--5
                    --buzzer <= counter(16);
  when "00110" => led7sa <= "0ZZZ";
                    led7seg <= "10000001";--6
                    --buzzer <= counter(11);
  when "00111" => led7sa <= "0ZZZ";
                    led7seg <= "11001100";--7
                    --buzzer <= counter(12);
  when "01000" => led7sa <= "0ZZZ";
                    led7seg <= "10000000";--8
                    --buzzer <= counter(13);
  when "01001" => led7sa <= "0ZZZ";
                    led7seg <= "10001000";--9
                    --buzzer <= counter(14);
  when "01010" => led7sa <= "0ZZZ";
                    led7seg <= "10000100";--A
                    --buzzer <= counter(15);
  when "01011" => led7sa <= "0ZZZ";

```

```

        led7seg <= "10100001";--B
        --buzzer <= counter(16);
when "01100" => led7sa <= "0ZZZ";
        led7seg <= "11000011";--C
        --buzzer <= counter(11);
when "01101" => led7sa <= "0ZZZ";
        led7seg <= "10110000";--D
        --buzzer <= counter(12);
when "01110" => led7sa <= "0ZZZ";
        led7seg <= "10000011";--E
        --buzzer <= counter(13);
when "01111" => led7sa <= "0ZZZ";
        led7seg <= "10000111";--F
when others => led7sa <= "0000";
        led7seg <= "ZZZZZZZZ";--NULL
        --buzzer <= '0';

end case;

--wait until A2'event and A2='1';

case
when "00000" => led7sa <= "Z0ZZ";
        led7seg <= "11000000";--0
when "00001" => led7sa <= "Z0ZZ";
        led7seg <= "11111100";--1
        --buzzer <= counter(12);
when "00010" => led7sa <= "Z0ZZ";
        led7seg <= "10010010";--2
        --buzzer <= counter(13);
when "00011" => led7sa <= "Z0ZZ";
        led7seg <= "10011000";--3
        --buzzer <= counter(14);
when "00100" => led7sa <= "Z0ZZ";
        led7seg <= "10101100";--4
        --buzzer <= counter(15);
when "00101" => led7sa <= "Z0ZZ";
        led7seg <= "10001001";--5
        --buzzer <= counter(16);
when "00110" => led7sa <= "Z0ZZ";
        led7seg <= "10000001";--6
        --buzzer <= counter(11);
when "00111" => led7sa <= "Z0ZZ";
        led7seg <= "11001100";--7
        --buzzer <= counter(12);
when "01000" => led7sa <= "Z0ZZ";
        led7seg <= "10000000";--8
        --buzzer <= counter(13);
when "01001" => led7sa <= "Z0ZZ";
        led7seg <= "10001000";--9
        --buzzer <= counter(14);
when "01010" => led7sa <= "Z0ZZ";
        led7seg <= "10000100";--A
        --buzzer <= counter(15);

```

```

when "01011" => led7sa <= "ZOZZ";
                led7seg <= "10100001";--B
                --buzzer <= counter(16);
when "01100" => led7sa <= "ZOZZ";
                led7seg <= "11000011";--C
                --buzzer <= counter(11);
when "01101" => led7sa <= "ZOZZ";
                led7seg <= "10110000";--D
                --buzzer <= counter(12);
when "01110" => led7sa <= "ZOZZ";
                led7seg <= "10000011";--E
                --buzzer <= counter(13);
when "01111" => led7sa <= "ZOZZ";
                led7seg <= "10000111";--F
when others => led7sa <= "0000";
                led7seg <= "ZZZZZZZZ";--NULL
                --buzzer <= '0';

end case;

--wait until A3'event and A3='1';

case count3 is
when "00000" => led7sa <= "ZZOZ";
                led7seg <= "11000000";--0
when "00001" => led7sa <= "ZZOZ";
                led7seg <= "11111100";--1
                --buzzer <= counter(12);
when "00010" => led7sa <= "ZZOZ";
                led7seg <= "10010010";--2
                --buzzer <= counter(13);
when "00011" => led7sa <= "ZZOZ";
                led7seg <= "10011000";--3
                --buzzer <= counter(14);
when "00100" => led7sa <= "ZZOZ";
                led7seg <= "10101100";--4
                --buzzer <= counter(15);
when "00101" => led7sa <= "ZZOZ";
                led7seg <= "10001001";--5
                --buzzer <= counter(16);
when "00110" => led7sa <= "ZZOZ";
                led7seg <= "10000001";--6
                --buzzer <= counter(11);
when "00111" => led7sa <= "ZZOZ";
                led7seg <= "11001100";--7
                --buzzer <= counter(12);
when "01000" => led7sa <= "ZZOZ";
                led7seg <= "10000000";--8
                --buzzer <= counter(13);
when "01001" => led7sa <= "ZZOZ";
                led7seg <= "10001000";--9
                --buzzer <= counter(14);
when "01010" => led7sa <= "ZZOZ";

```

```

        led7seg <= "10000100";--A
        --buzzer <= counter(15);
when "01011" => led7sa <= "ZZOZ";
        led7seg <= "10100001";--B
        --buzzer <= counter(16);
when "01100" => led7sa <= "ZZOZ";
        led7seg <= "11000011";--C
        --buzzer <= counter(11);
when "01101" => led7sa <= "ZZOZ";
        led7seg <= "10110000";--D
        --buzzer <= counter(12);
when "01110" => led7sa <= "ZZOZ";
        led7seg <= "10000011";--E
        --buzzer <= counter(13);
when "01111" => led7sa <= "ZZOZ";
        led7seg <= "10000111";--F
when others => led7sa <= "0000";
        led7seg <= "ZZZZZZZZ";--NULL
        --buzzer <= '0';
end case;

--wait until A4'event and A4='1';

case count4 is
when "00000" => led7sa <= "ZZZO";
        led7seg <= "11000000";--0
when "00001" => led7sa <= "ZZZO";
        led7seg <= "11111100";--1
        --buzzer <= counter(12);
when "00010" => led7sa <= "ZZZO";
        led7seg <= "10010010";--2
        --buzzer <= counter(13);
when "00011" => led7sa <= "ZZZO";
        led7seg <= "10011000";--3
        --buzzer <= counter(14);
when "00100" => led7sa <= "ZZZO";
        led7seg <= "10101100";--4
        --buzzer <= counter(15);
when "00101" => led7sa <= "ZZZO";
        led7seg <= "10001001";--5
        --buzzer <= counter(16);
when "00110" => led7sa <= "ZZZO";
        led7seg <= "10000001";--6
        --buzzer <= counter(11);
when "00111" => led7sa <= "ZZZO";
        led7seg <= "11001100";--7
        --buzzer <= counter(12);
when "01000" => led7sa <= "ZZZO";
        led7seg <= "10000000";--8
        --buzzer <= counter(13);
when "01001" => led7sa <= "ZZZO";
        led7seg <= "10001000";--9
        --buzzer <= counter(14);

```

```

when "01010" => led7sa <= "ZZZ0";
                 led7seg <= "10000100";--A
                 --buzzer <= counter(15);
when "01011" => led7sa <= "ZZZ0";
                 led7seg <= "10100001";--B
                 --buzzer <= counter(16);
when "01100" => led7sa <= "ZZZ0";
                 led7seg <= "11000011";--C
                 --buzzer <= counter(11);
when "01101" => led7sa <= "ZZZ0";
                 led7seg <= "10110000";--D
                 --buzzer <= counter(12);
when "01110" => led7sa <= "ZZZ0";
                 led7seg <= "10000011";--E
                 --buzzer <= counter(13);
when "01111" => led7sa <= "ZZZ0";
                 led7seg <= "10000111";--F
when others => led7sa <= "0000";
                 led7seg <= "ZZZZZZZZ";--NULL
                                     --buzzer <= '0';

end case;
end process;

end Behavioral;

```

.0.2 プログラム 2

4 セグ独立カウンター

```
----- low layer module-----
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.std_logic_unsigned.ALL;
use IEEE.std_logic_arith.ALL;

entity bit_counter20181031 is
port(
    CLK:in std_logic;
    SW1:in std_logic;
    SW2:in std_logic;
    SW3:in std_logic;
    SW4: in std_logic;
    LED1:out std_logic_vector(3 downto 0);
    LED2:out std_logic_vector(3 downto 0);
    LED3:out std_logic_vector(3 downto 0);
    LED4:out std_logic_vector(3 downto 0);
    LED7seg:out std_logic_vector(7 downto 0);
    LED7sa:out std_logic_vector(3 downto 0));

end bit_counter20181031;

architecture Behavioral of bit_counter20181031 is

component switch1 is
port(
    SW:in std_logic;
    LED: out std_logic_vector(4 downto 0);
    CLK:in std_logic);
end component;

begin
U1:switch1 port map(SW1,LED1,CLK);
U2:switch1 port map(SW2,LED2,CLK);
U3:switch1 port map(SW3,LED3,CLK);
U4:switch1 port map(SW4,LED4,CLK);

process begin
case LED is
when "00000" => LED7sa<= "0ZZZ";
                LeD7seg <= "11000000";--0
when "00001" => LED7sa<= "0ZZZ";
                LeD7seg<= "11111100";--1
when "00010" => LED7sa<= "0ZZZ";
                LeD7seg<= "10010010";--2
```

```

when "00011" => LED7sa <= "0ZZZ";
                LeD7seg <= "10011000";--3
when "00100" => LED7sa <= "0ZZZ";
                LeD7seg <= "10101100";--4
when "00101" => LED7sa <= "0ZZZ";
                LeD7seg <= "10001001";--5
when "00110" => LED7sa <= "0ZZZ";
                LeD7seg <= "10000001";--6
when "00111" => LED7sa <= "0ZZZ";
                LeD7seg <= "11001100";--7
when "01000" => LED7sa <= "0ZZZ";
                LeD7seg <= "10000000";--8
when "01001" => LED7sa <= "0ZZZ";
                LeD7seg <= "10001000";--9
when "01010" => LED7sa <= "0ZZZ";
                LeD7seg <= "10000100";--A
when "01011" => LED7sa <= "0ZZZ";
                LeD7seg <= "10100001";--B
when "01100" => LED7sa <= "0ZZZ";
                LeD7seg <= "11000011";--C
when "01101" => LED7sa <= "0ZZZ";
                seg1 <= "10110000";--D
when "01110" => LED7sa <= "0ZZZ";
                LeD7seg <= "10000011";--E
when "01111" => LED7sa <= "0ZZZ";
                LeD7seg <= "10000111";--F
when others => LED7sa <= "0000";
                LeD7seg <= "ZZZZZZZZ";--NULL

end case;
end process;

end Behavioral;

```

-----top layer module-----

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.std_logic_unsigned.ALL;
use IEEE.std_logic_arith.ALL;

entity switch1 is
port(SW:in std_logic;
      LED: out std_logic_vector(4 downto 0);
      CLK:in std_logic);

end switch1;

architecture Behavioral of switch1 is

begin

```

```
process(CLK)begin
  if (CLK'event and CLK='1') then
    if(SW'event and SW='1') then
      -- wait for 50ms;
      LED <=LED+'1';

    end if;
  end if;
end process;

end Behavioral;
```

これらの実習において、FPGA の処理能力と、VHDL 言語の構造理解を深めることができ、実際に点灯している様子を確認でき、理解を深めることができた。