



Firmware integration of Level-0 muon trigger for HL-LHC

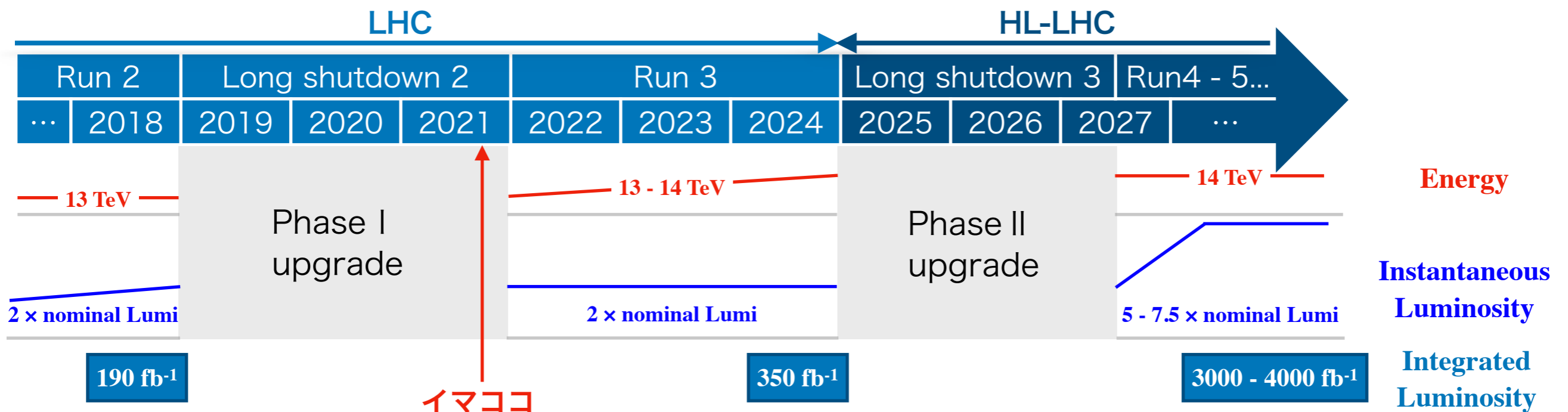
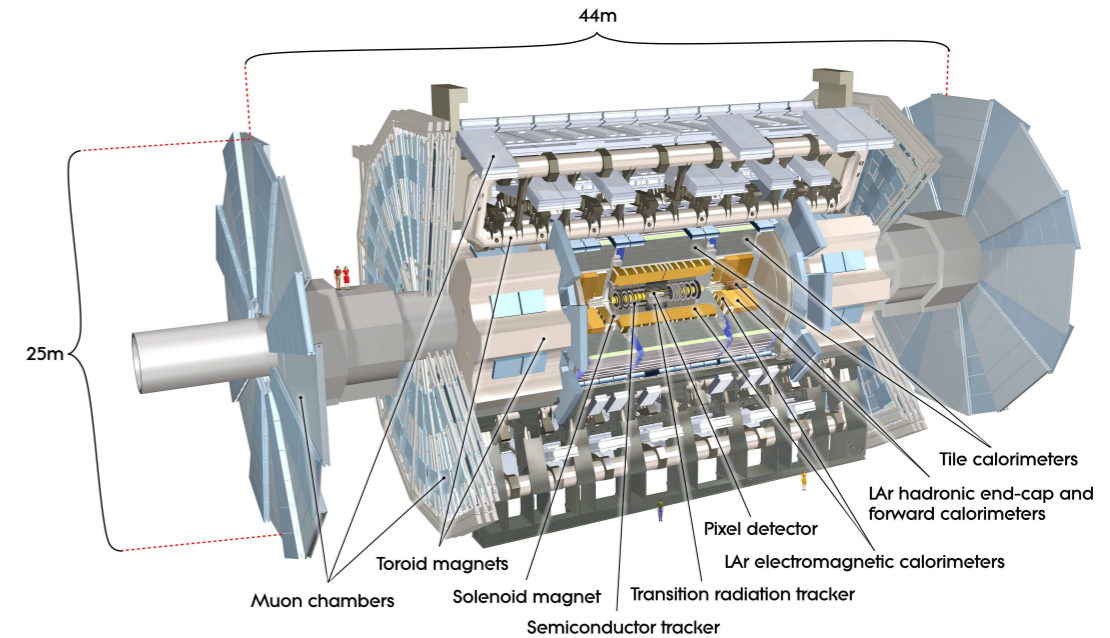
三野裕哉 (京都大学)

ATLAS 日本トリガーグループ

2021/10/28 計測システム研究会

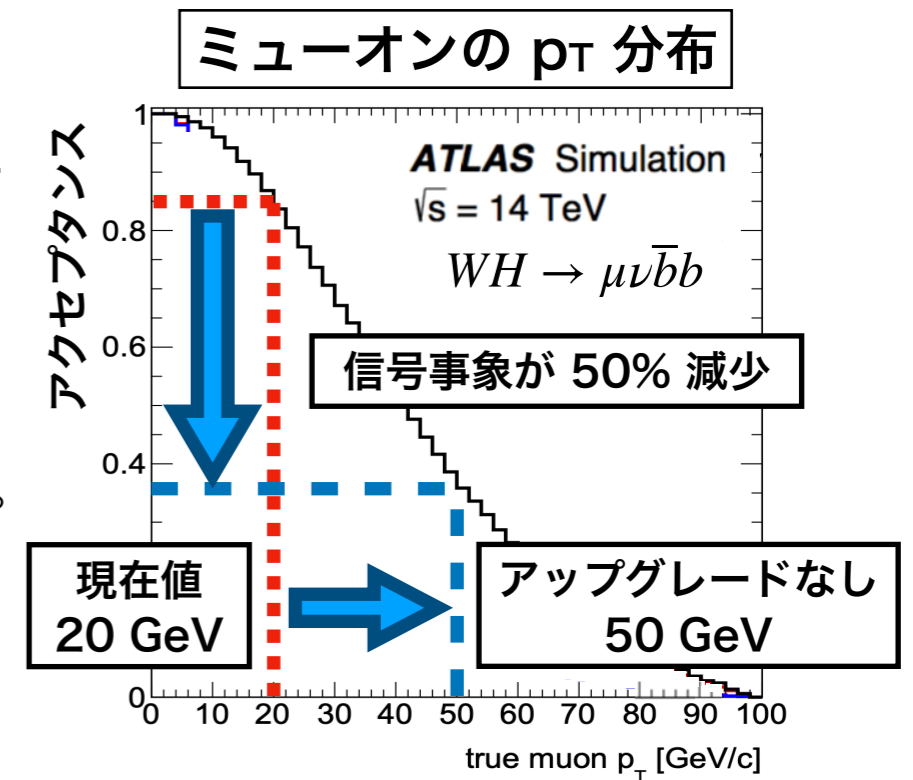
High Luminosity LHC

- Large Hadron Collider (LHC)
 - ▶ CERN の陽子陽子衝突型円形加速器
 - ▶ 衝突頻度 40 MHz、重心系エネルギー 13 TeV
- ATLAS 実験
 - ▶ 大型汎用検出器 (ATLAS) を用いて陽子衝突事象を観測。
 - ▶ 目的：未知の粒子の探索やヒッグス粒子の精密測定
- High Luminosity LHC (HL-LHC)
 - ▶ 高い統計量での標準模型の精密測定や新物理探索を目的として 2027 年から開始予定。
 - ▶ 瞬間ルミノシティを nominal luminosity の 5 - 7.5 倍に増強する。

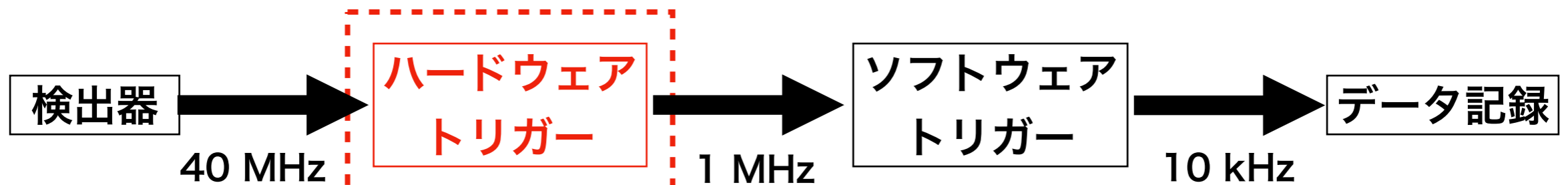


HL-LHC に向けたトリガーシステムのアップグレード

- 瞬間ルミノシティの増加に伴い、背景事象によるトリガーレートが増加する。
 - ▶ トリガーの閾値を上げることでレートを抑制できるが、信号事象が減少する。(例)ヒッグス粒子の W ボソン随伴生成
 - ▶ 物理に対する感度を維持するにはトリガーシステムのアップグレードが必要。



- ATLAS は初段のハードウェアと後段のソフトウェアで実装されたトリガーの 2 段階構成。



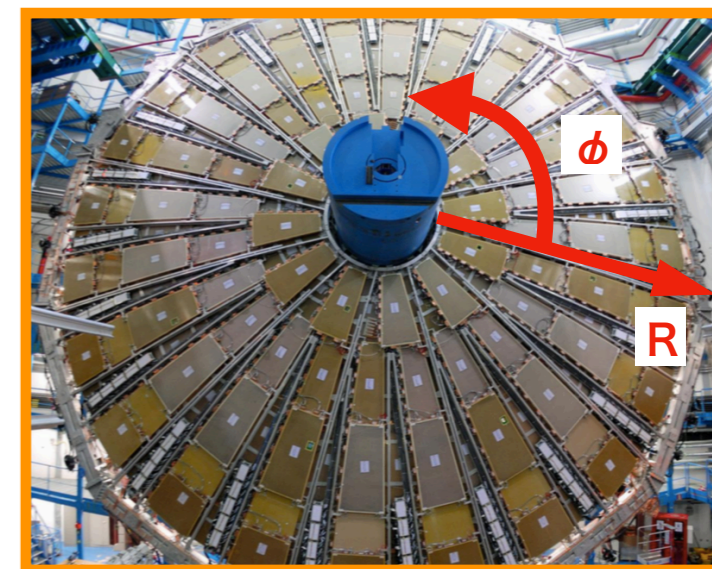
トリガー読み出しシステムを刷新し、
トリガーレート・判定時間を
を増強して HL-LHC に対応する。

	LHC	HL-LHC
トリガーレート [kHz]	100	1000
判定時間 [μ s]	2.5	10

ミュオン検出器

- Thin Gap Chamber (TGC)

- ▶ 多線式比例計数管(MWPC) の一種。
- ▶ エンドキャップ部でメインとなるミュオントリガー検出器。
- ▶ Big Wheel (BW) と呼ばれる円盤に 3 つのステーション (全 7 層) で構成されている。
- ▶ ワイヤーストリップによる R - ϕ の2次元読み出しを行う。



- 磁場内部の検出器

- ▶ $1.05 < |\eta| < 1.3$

- TGC EI & RPC BIS78

Resistive plate chamber (RPC) は TGC と同様に R - ϕ の2次元読み出しを行う。

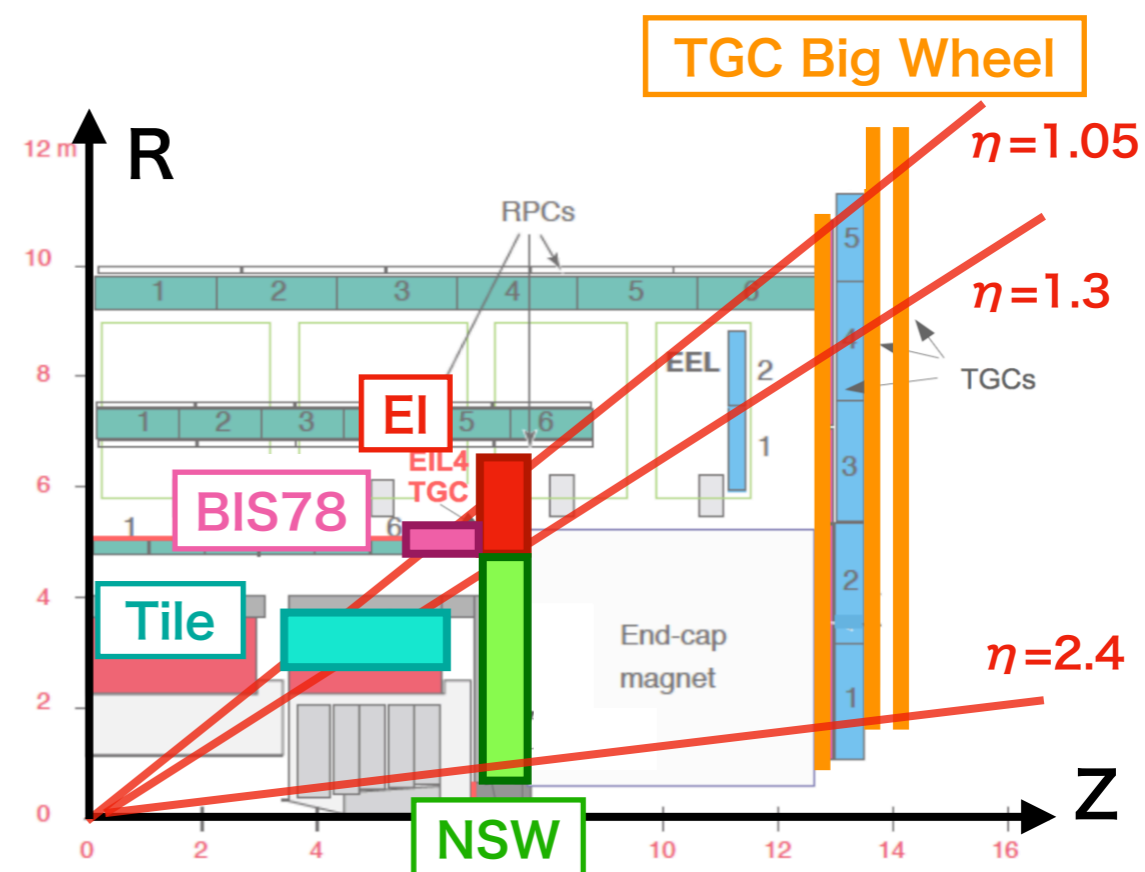
- Tile calorimeter (ハドロンカロリメータ)

EI or BIS78 でカバーできない領域で用いる。

- ▶ $1.3 < |\eta| < 2.4$

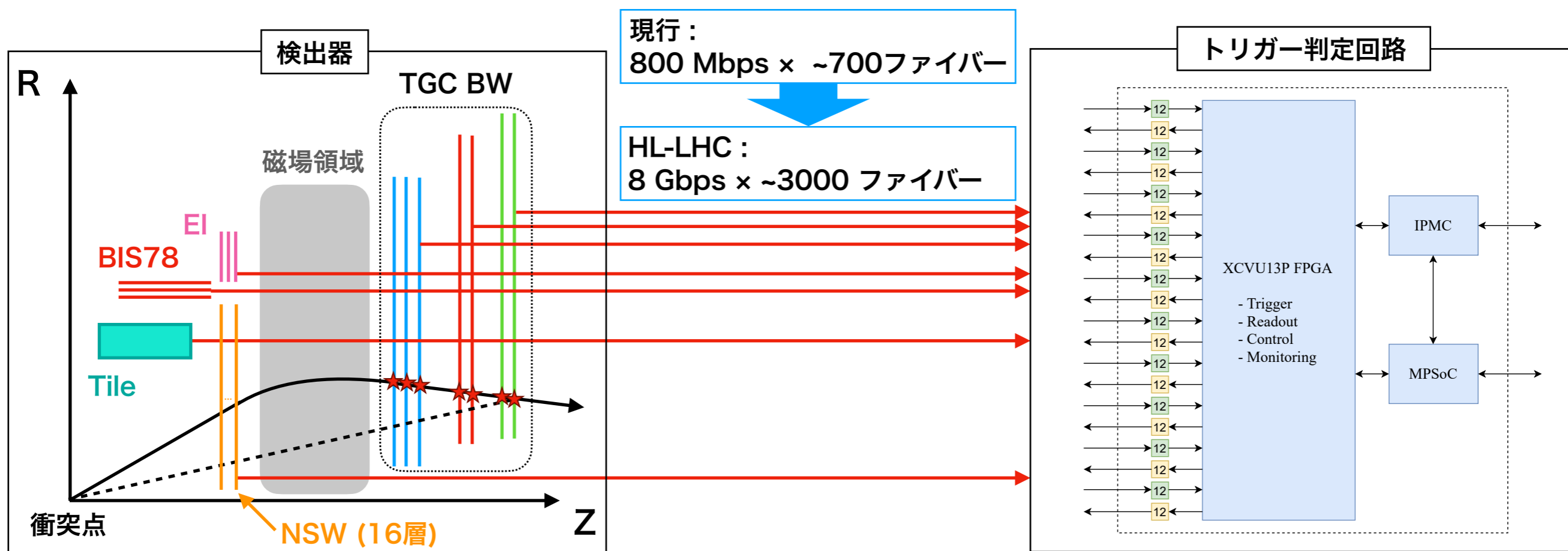
- New Small Wheel (NSW)

飛跡の精密な角度測定が可能な検出器。



HL-LHC でのミュオントリガーアルゴリズム

- 現行のシステムでは前段回路で TGC のヒット情報を用いてコインシデンスを取り、コインシデンス結果をトリガー判定回路に転送する。
 - ▶ HL-LHC では高速データ転送技術を活用し、TGC のヒット情報を全てトリガー判定回路に転送する。



● HL-LHC でのトリガーアルゴリズム

① TGC BW の全ヒット情報を用いて飛跡を再構成。

➡ TGC BW での検出効率は 96 % (現行のトリガーシステムと比較して +4 %)

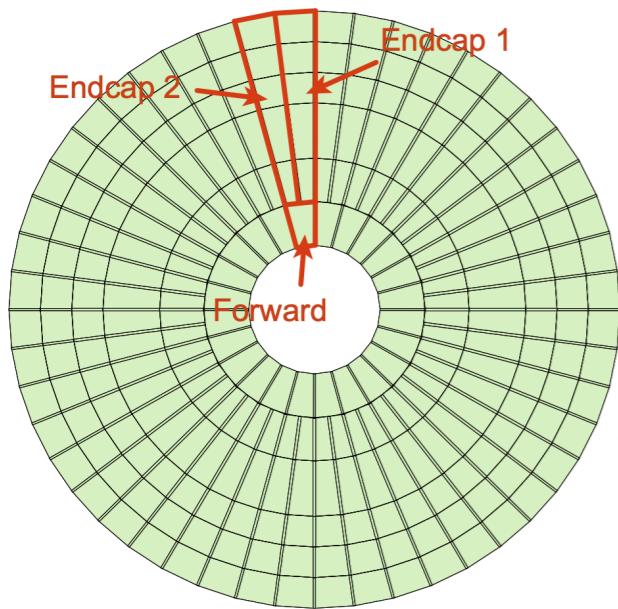
➡ 全層の情報を用いることで p_T 分解能が向上する。

② 磁場内部の検出器で測定した位置・角度情報を用いることでトリガーレートを大幅にカット。

トリガー判定回路 (Sector Logic : SL)

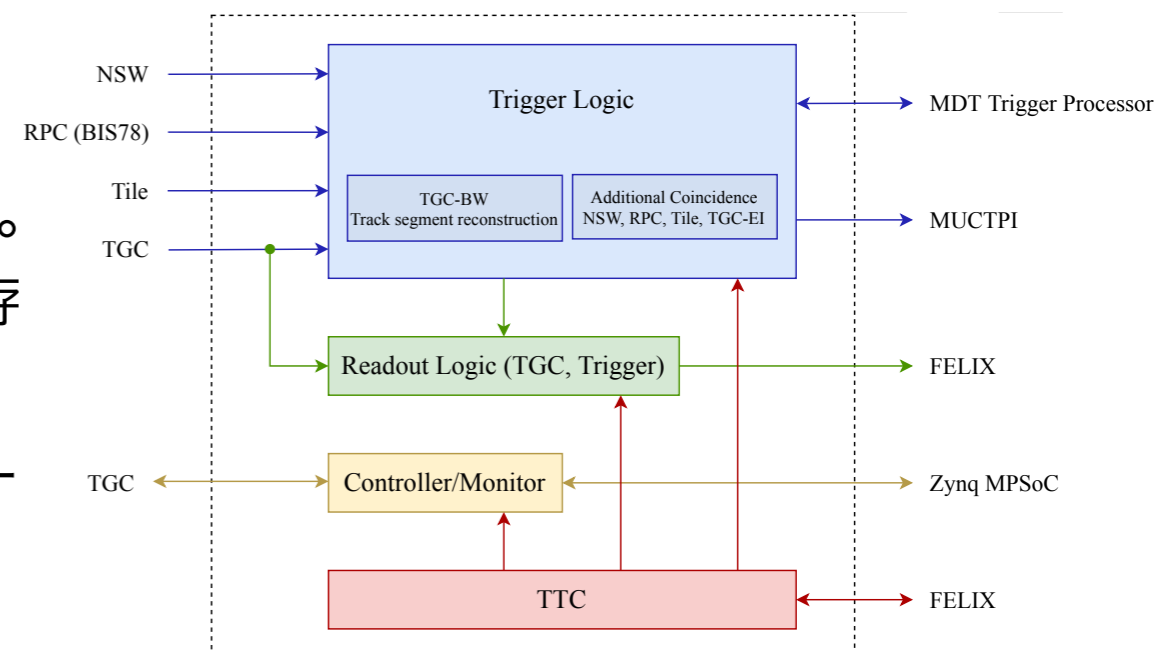
- 事象選別を行うトリガー判定回路 (“Sector Logic”, SL) には Virtex UltraScale+ FPGA (XCVU13P) を搭載予定。(Run 3 で使用する Kintex-7 の ~10 倍規模)

TGC Big Wheel



- XCVU13P のリソース量
 - ▶ ロジックセル: 3.8 M 個
 - ▶ RAM: 455 Mb
 - ▶ MGT (GTY): 128 個
- 1つの SL は 3つの「トリガーセクター」から情報を受け取る。(Endcap 2つ、Forward 1つ)
 - ▶ 左の絵の赤線領域全てをカバーする。

- SL には三つの機能が実装されている。
 - ▶ **トリガー** : ミューオントリガーの候補を再構成する。
 - ▶ **読み出し** : TGC のヒット情報とトリガー情報を保存し、後段でトリガーが発行された際に出力する。
 - ▶ **コントロール/モニタリング** : 前段回路のコントロール・モニタリングを行う。



FPGA (XCVU13P)

① XCVU13P には以下の2種類の容量のメモリが搭載されている。

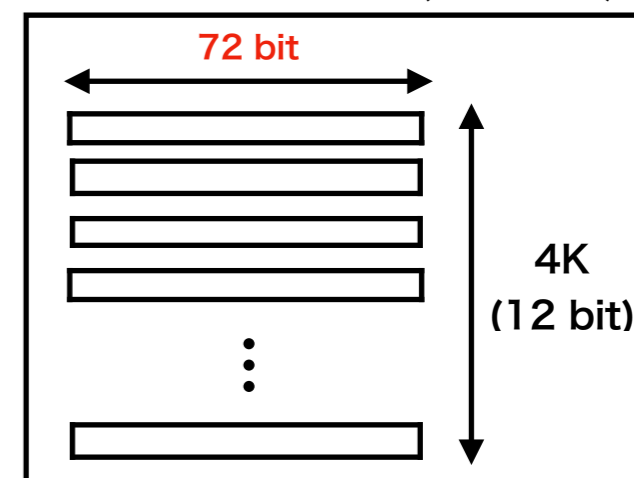
a) BRAM : 36 Kb / block

- データ幅は 1, 2, 4, 9, 18, 36 bit に設定可能。

b) UltraRAM : 4 K × 72 bit = 295 Kb / block

- データ幅は 72 bit で固定。

URAM 1 block (295 Kb)



② XCVU13P は 4 つのダイ (Super Logic Region : SLR) を連結して構成されている。

a) SLR間の信号の送受信には制約がある。

▶ レイテンシ (0(ns))

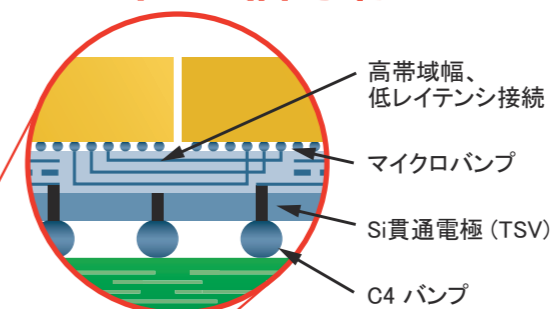
▶ 信号線数が有限 (計23 Kb分)

b) 各SLR上には以下の制約がある。

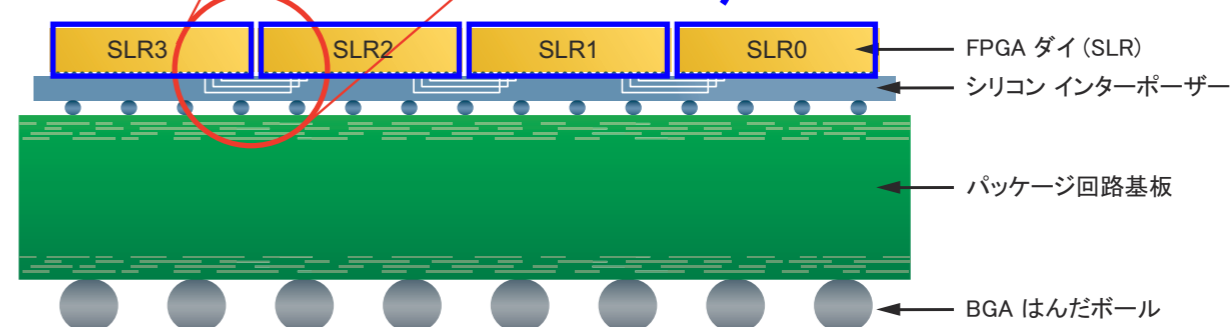
▶ FPGA素子の量

▶ 光通信機(通信容量)

a) SLR間の信号線

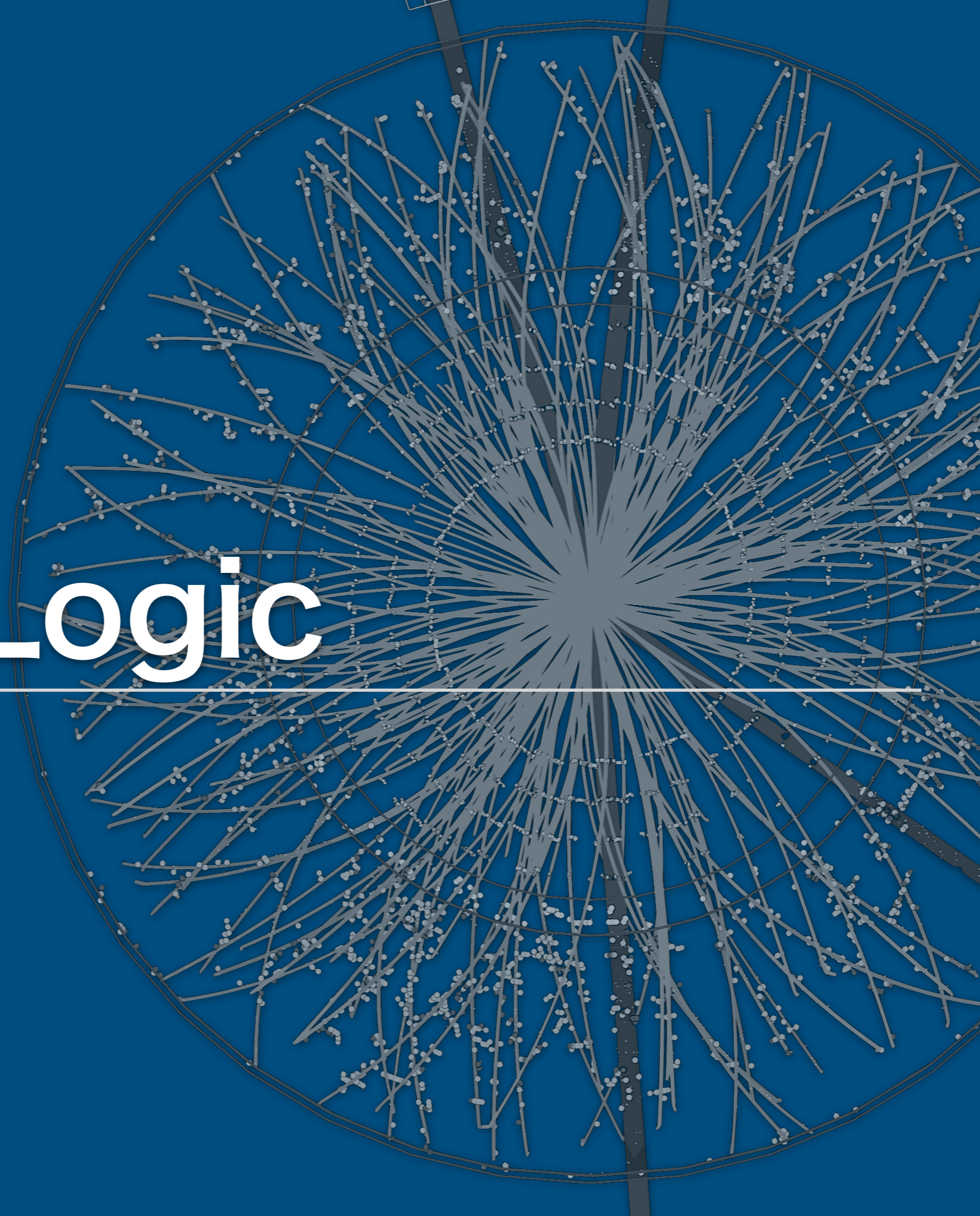


b) SLR



● 以上の二つの特徴 (UltraRAM, SLR) を踏まえてファームウェアのデザイン作成が必要。

Trigger Logic



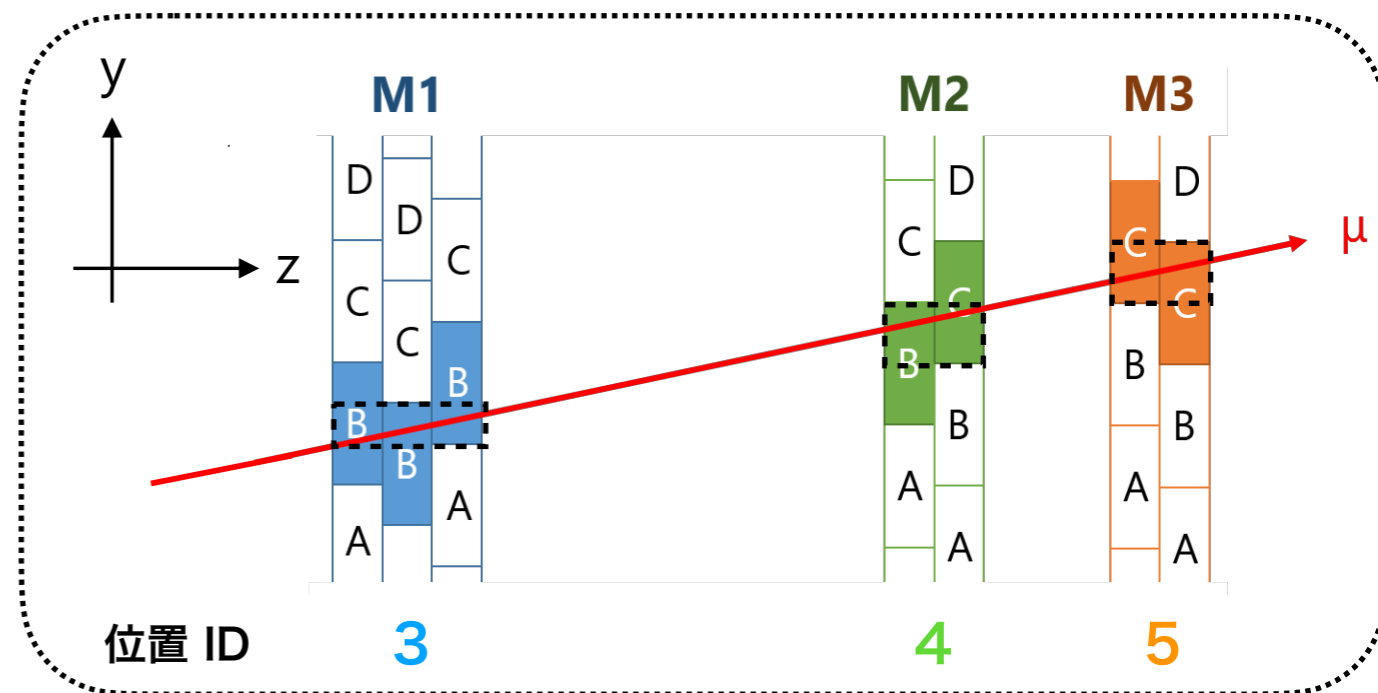
飛跡再構成

- TGC BW のワイヤー(ストリップ)のヒット情報を全て用いて $\eta(\phi)$ 方向の飛跡を再構成。
- あらかじめ TGC のヒットに対応したミュオンの飛跡情報を持ったパターンリストを作成し、UltraRAM に格納する。

① 各 station (M1, M2, M3) で TGC のヒットからコインシデンスをとり、位置 ID を定義。

② パターンリストからヒットの組み合わせに対応する飛跡情報を抽出。

➔ 複雑な計算を必要とせず、高速な処理が可能。



①ステーションコインシデンス

M1 Coincidence

Hit ch: B B B → M1 Position ID: 3

M2 Coincidence

Hit ch: B C → M2 Position ID: 4

M3 Coincidence

Hit ch: C C → M3 Position ID: 5

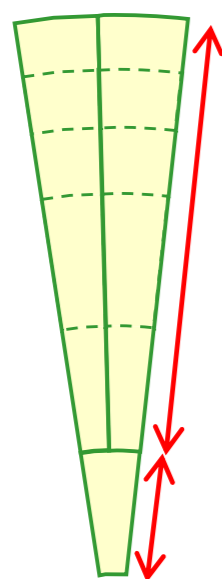
②飛跡情報の抽出

Input (Position ID, 16 bit)			Output (Track segment, 18 bit)
3	4	4	Position η_a , Angle $\Delta\theta_a$
3	4	5	Position η_b, Angle $\Delta\theta_b$
3	5	5	Position η_c , Angle $\Delta\theta_c$

飛跡再構成

- パターンマッチングに必要な RAM 使用量を最小限にするため、検出器領域を 90 個の“ユニット”に分割する。

3つのtrigger sectorで独立にユニットを定義している

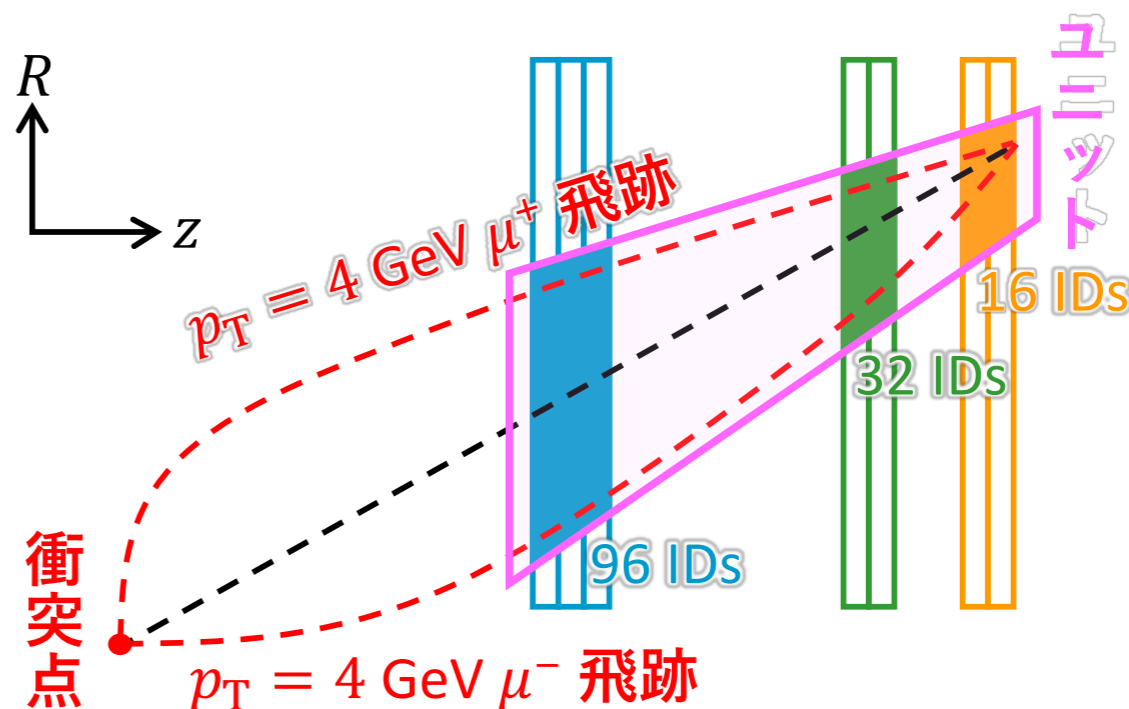


Endcap trigger sector

- $\eta = 1.05 - 1.92$
- 37ユニットずつ
- 2つの領域を分割

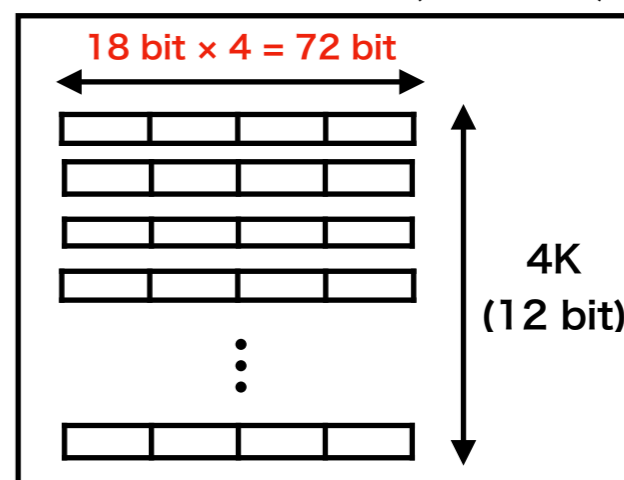
Forward trigger sector

- $\eta = 1.92 - 2.4$
- 16ユニットに分割



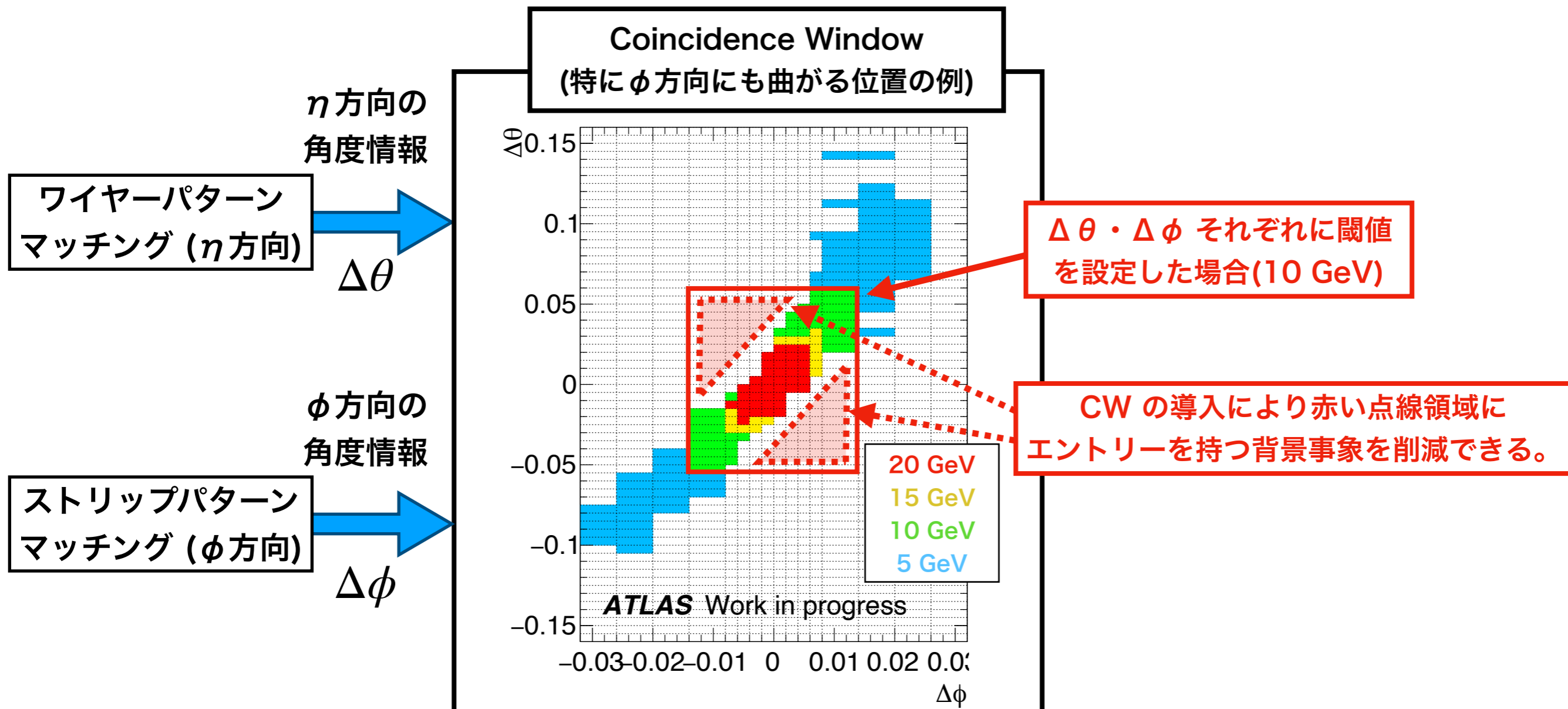
- UltraRAM の扱いにくい点：出力データ幅が 72 bit と固定。
 - ▶ 飛跡情報(18 bit) を各アドレスに 4 セット格納することで UltraRAM のリソース量を最大限に活用する。
 - ▶ 72 bit の飛跡情報を抽出した後に M1 の位置情報をもとに対応する 18 bit を出力する。

URAM 1 block (295 Kb)



飛跡情報を用いた p_T の算出

- ミューオンはトロイド磁場の非一様性により、 η と ϕ 方向のいずれにも曲がるため、ワイヤーとストリップでそれぞれの方向の飛跡をパターンマッチングによって得る。
 - ▶ $\Delta\theta \cdot \Delta\phi$ の2次元の情報から p_T を返す LUT を作成し、BRAM 上に Coincidence Window (CW) として実装する。

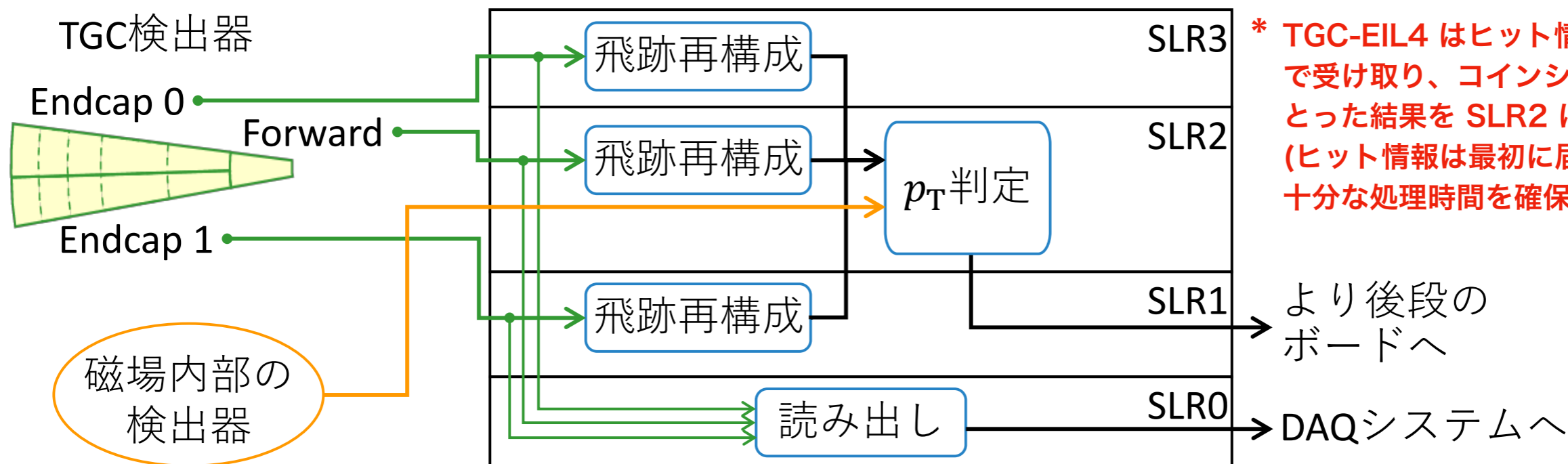
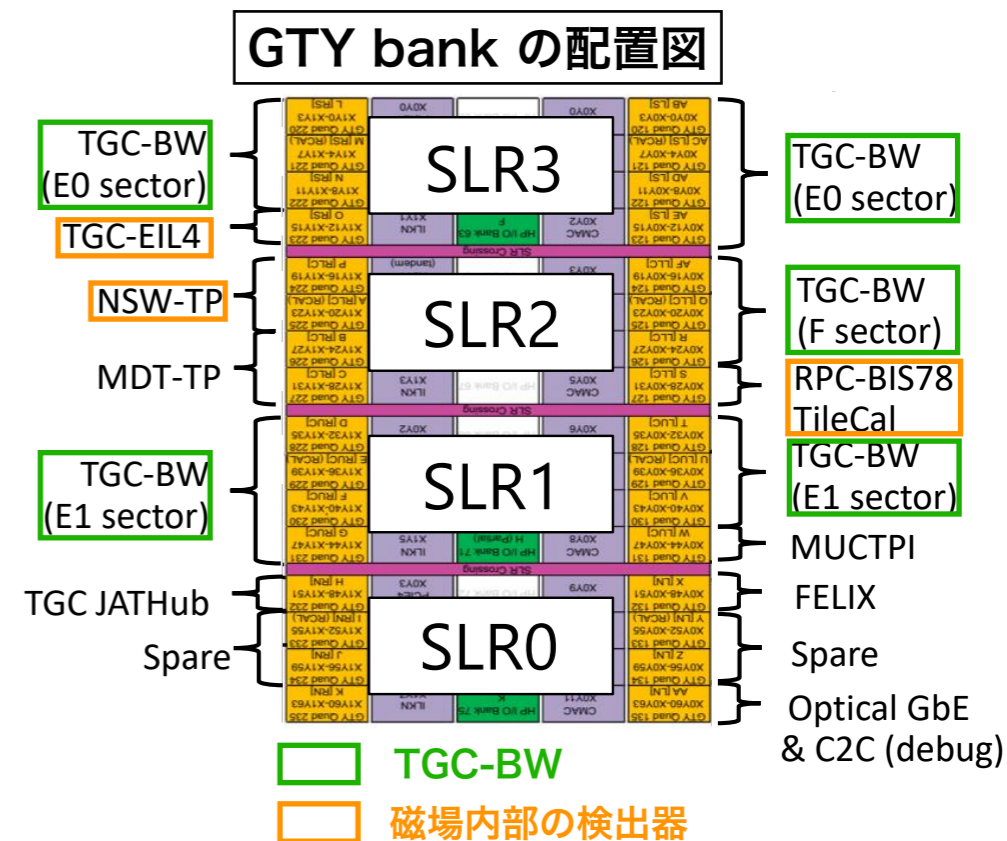




Firmware integration

フロアプランの決定

- FPGA 内のフロアプランは RAM のリソース量、処理時間や GTY bank の数などを考慮して決定した。
 - ▶ 飛跡再構成については TGC セクターごとに実行し、各実行ブロックを各 SLR に配置することで処理時間を低減。
 - ▶ 磁場内部の検出器を用いた p_T 判定はリソース使用量の比較的少ない Forward 領域の飛跡再構成と同じ SLR を使用する。



* TGC-EIL4 はヒット情報を SLR0 で受け取り、コインシデンスをとった結果を SLR2 に転送する。(ヒット情報は最初が届くため、十分な処理時間を確保できる。)

回路合成(ワイヤー飛跡再構成)

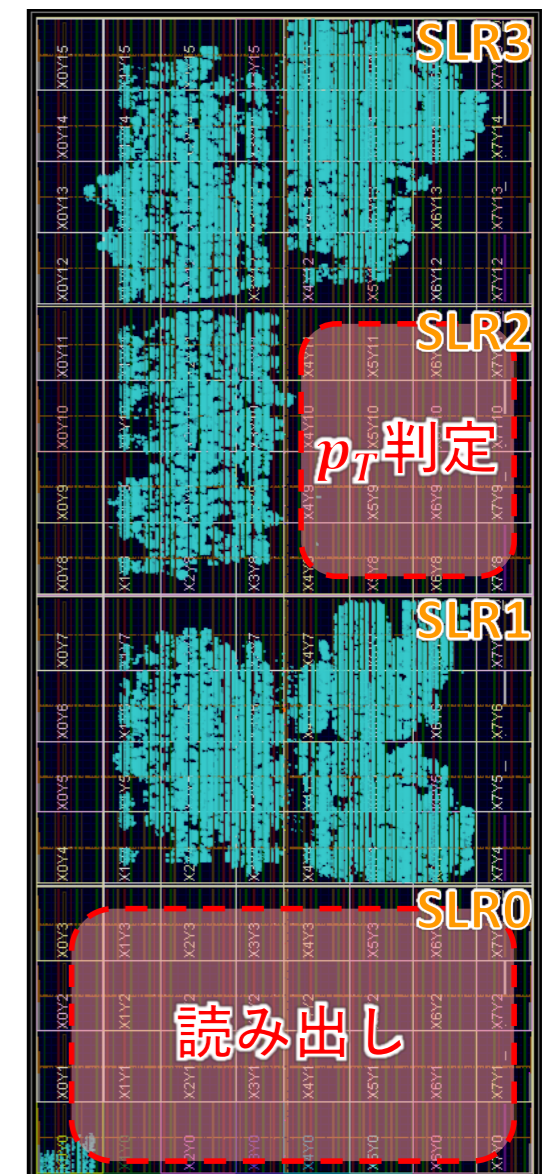
- 特にワイヤー飛跡再構成は 4,400 チャンネルの信号を扱う大規模なファームウェアとなるため、実際に回路合成を行い FPGA 全体での素子の使用率を確認した。

*タイミング違反を避けるため、FPGA全体での使用率は ~50% が目標。

	ロジックセル	ルックアップ テーブル	レジスタ	RAM
SLR3	41%	28%	5%	46%
SLR2	18%	12%	2%	20%
SLR1	40%	28%	5%	46%

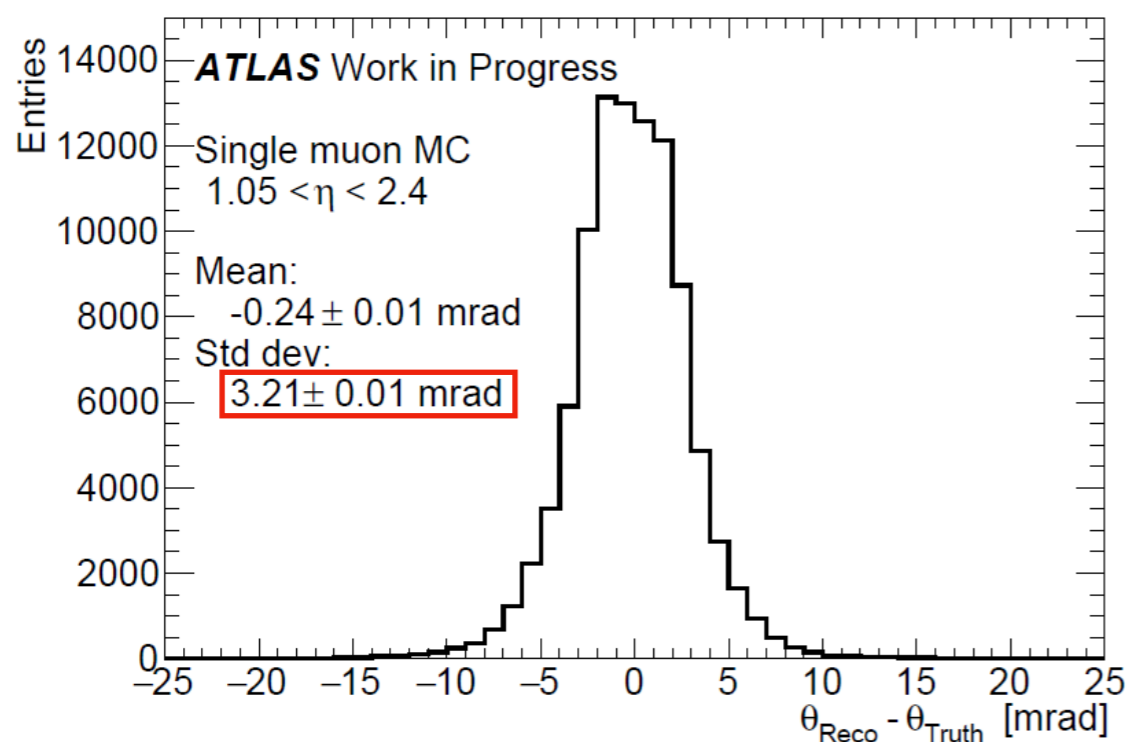
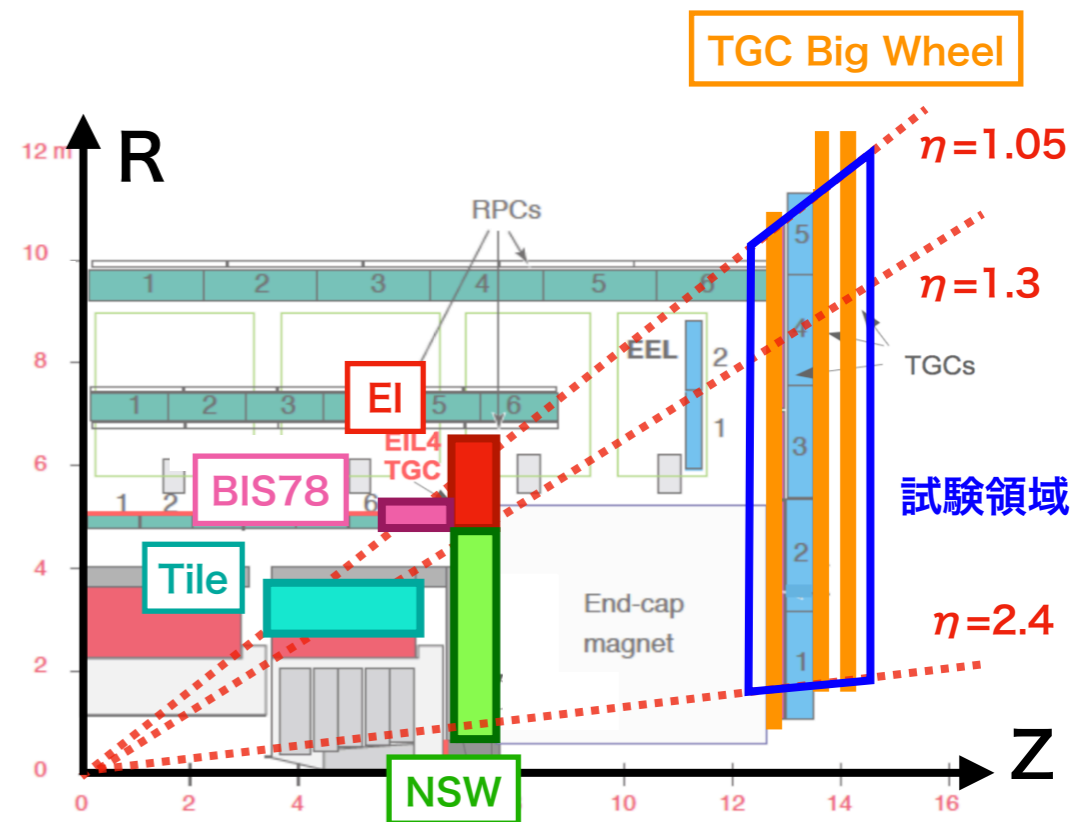
- 回路合成の結果
 - ▶ タイミング違反なし
 - ▶ 処理時間: 78 ns (< 要求 125 ns)
 - ▶ 各 SLR での素子使用率 ~40%
- ➔ 要求を満たす規模で回路を実現した。

FPGA上の回路配置図



ワイヤー飛跡再構成の性能評価(シミュレーション)

- Vivado の論理シミュレーションを用いてファームウェアの動作を検証した。
 - ▶ 単一ミュオン MC シミュレーション ($20 < p_T < 100$ GeV) から抽出した 10 万事象分のヒット情報を 40 MHz で連続入力。

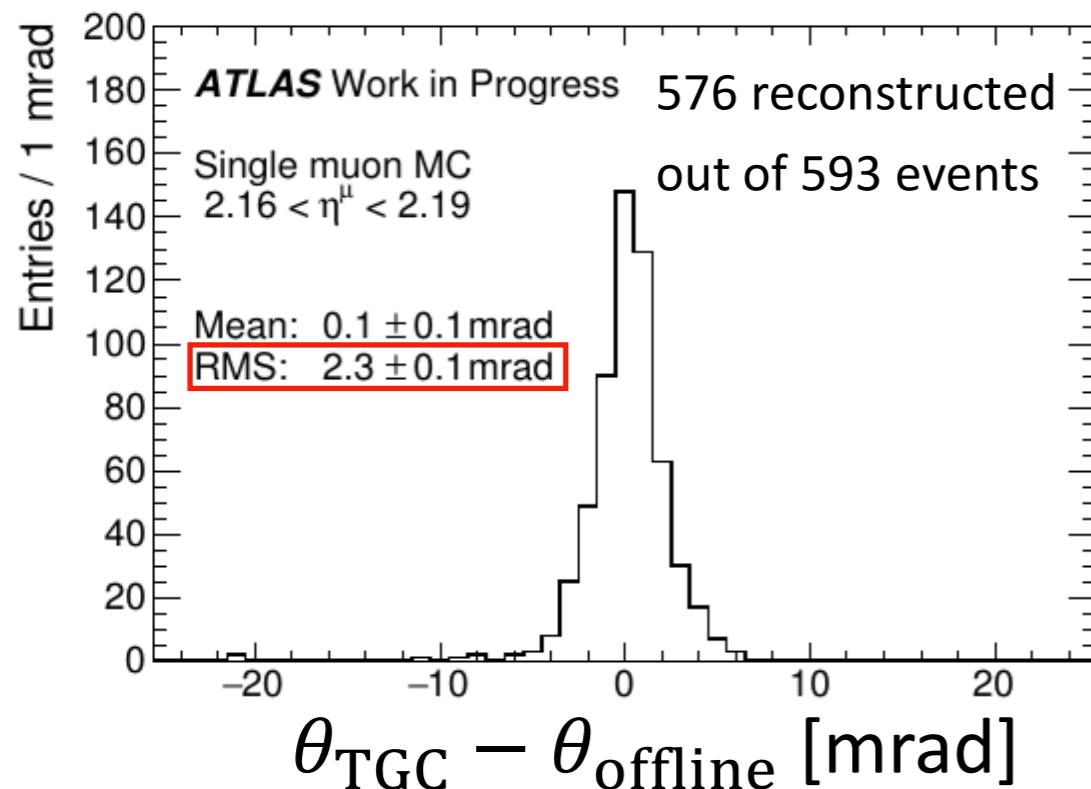
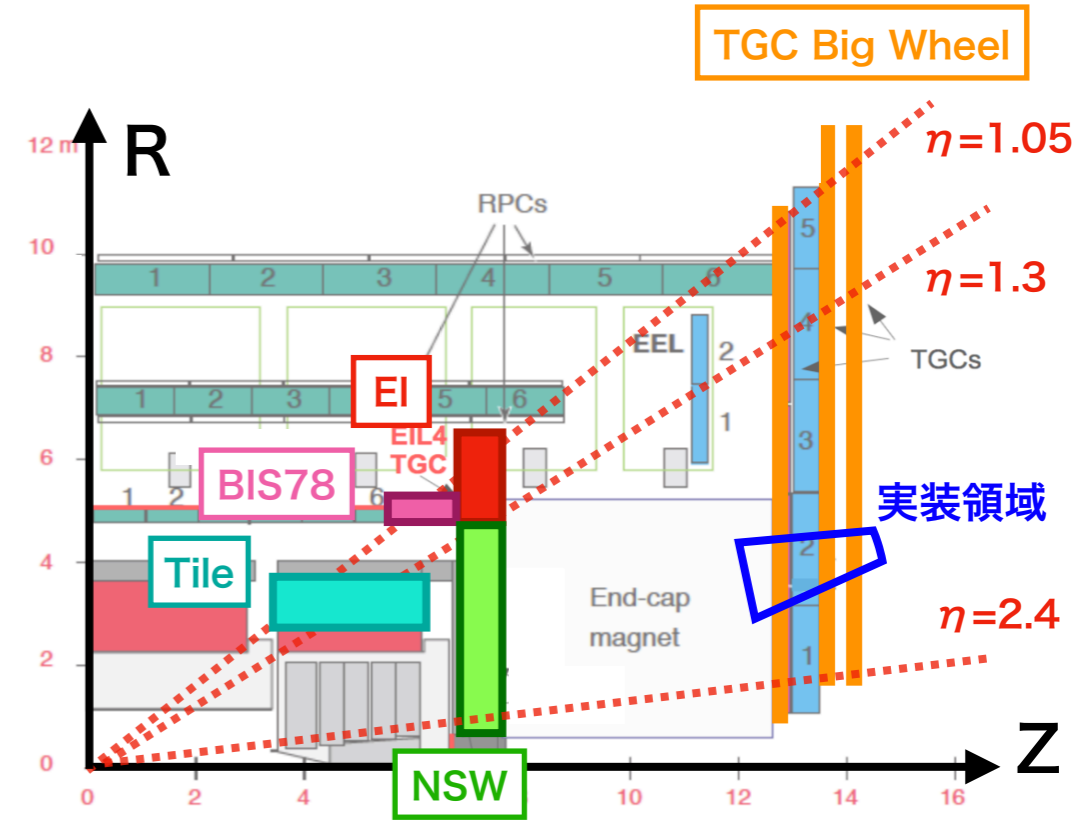


- 再構成された飛跡の角度(θ_{Reco})と真のミュオン飛跡の角度(θ_{Truth})を比較することで性能を評価した。
 - ▶ 角度分解能: 3.2 mrad (< 目標 4 mrad)
 - ▶ 再構成効率: 96.8 % (> 目標 93 %)

ワイヤー飛跡再構成のFPGA実機テスト

- Virtex UltraScale+ の評価ボード (VCU118*) を用いて、ワイヤー飛跡再構成ファームウェアの動作実証を行った。
- ▶ 実装したファームウェアは 1 ユニットに対応する一部領域のみ。 ($2.16 < \eta < 2.19$)

* 実装されている FPGA は XCVU9P



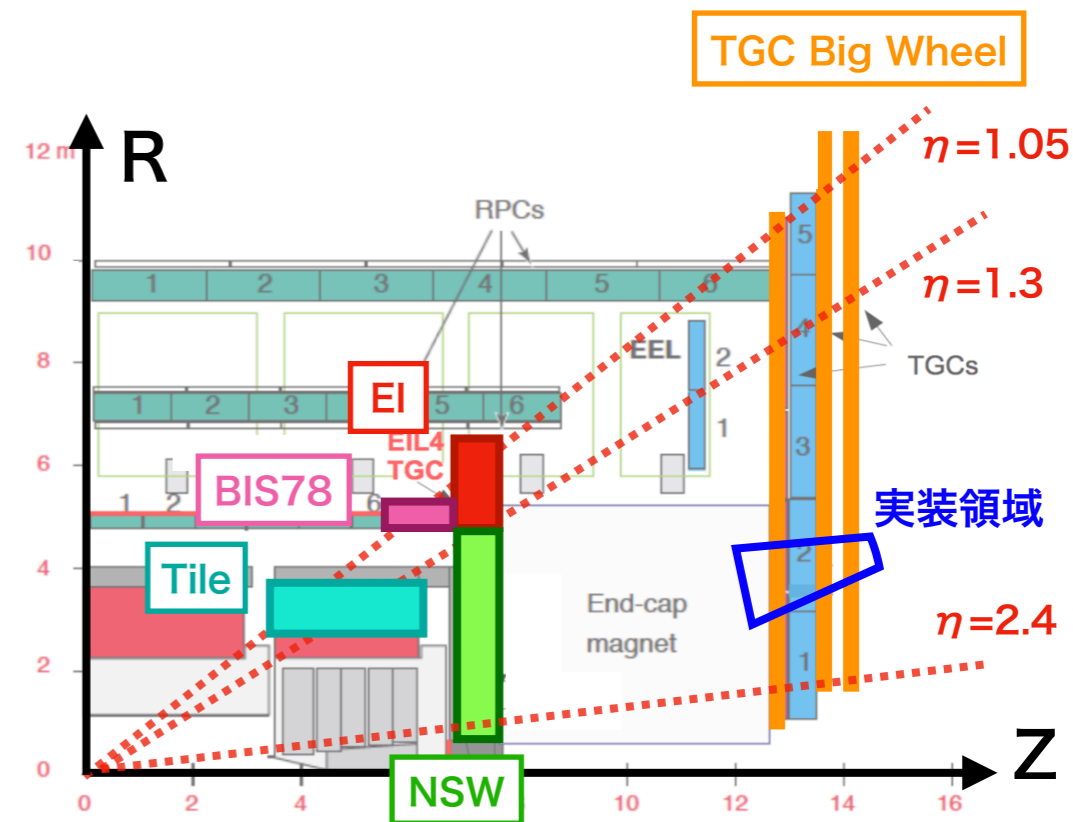
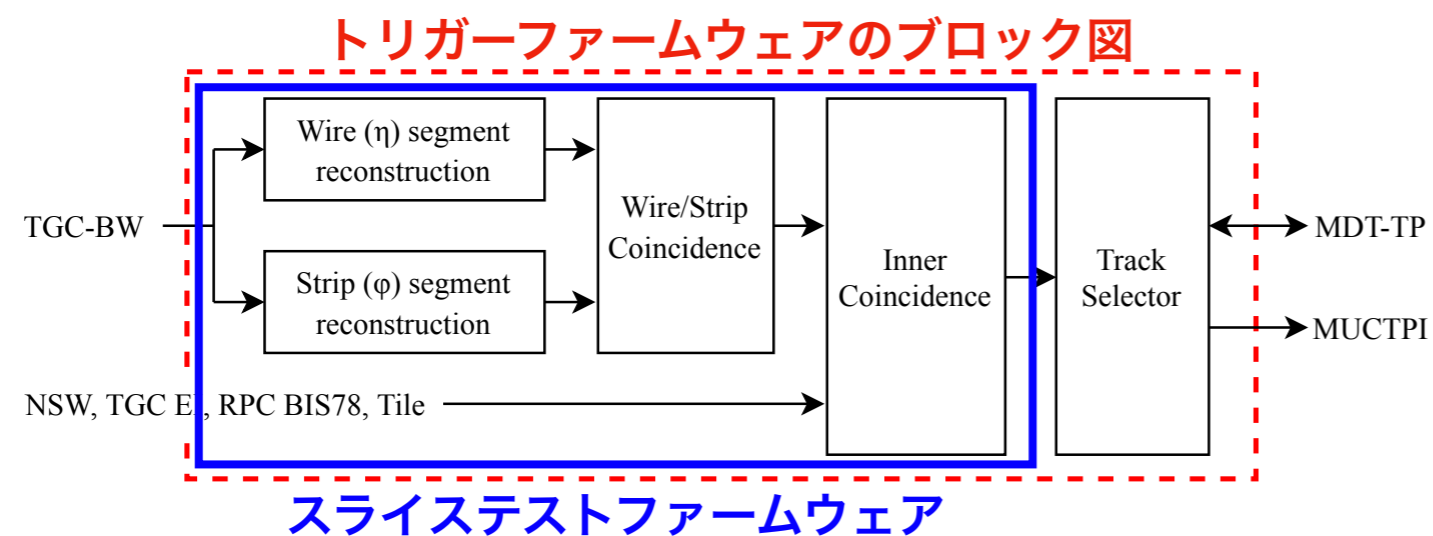
- 実機テストにおいても要求される性能を満たしていることを確認した。
 - ▶ 角度分解能: 2.3 mrad** (< 目標 4 mrad)
 - ▶ 再構成効率: 97.1 % (> 目標 93 %)
- ** 実装領域 ($2.16 < \eta < 2.19$) はチャンネル幅が狭く、良い分解能が得られやすい。

ファームウェアの統合試験

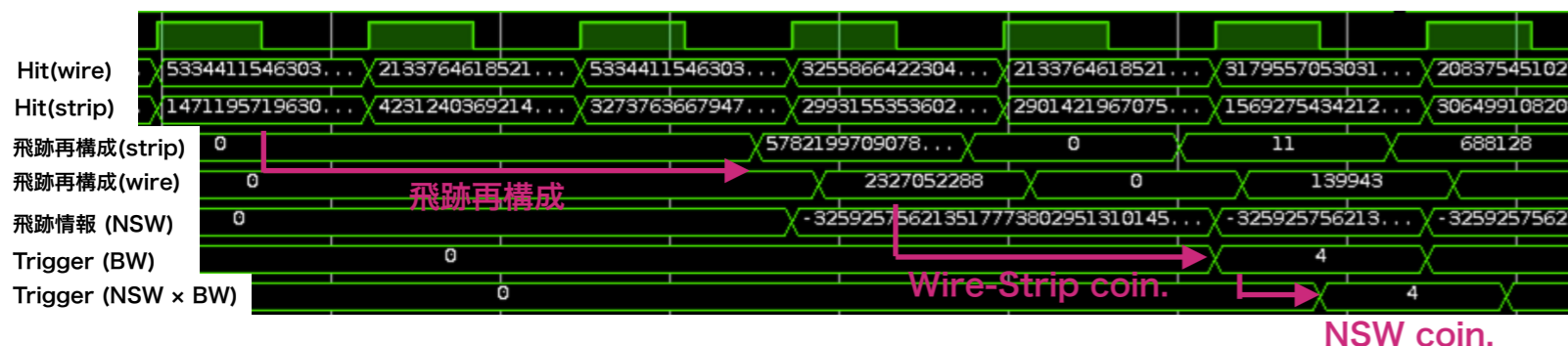
- ワイヤー飛跡再構成以外のトリガーロジックを加えたファームウェア*を一部領域について作成し、回路合成**・シミュレーションを行った。

* 飛跡の選別を行うモジュール (Track Selector) は加えていない。

** SLR を考慮した回路の配置は行っていない。



シミュレーションの波形解析



- 単一ミュオンシミュレーションの MC シミュレーションから抽出したヒット情報や NSW の飛跡情報を 40 MHz で連続入力
- 検出効率などの性能評価は現在進行中

まとめ

トリガーファームウェアの開発

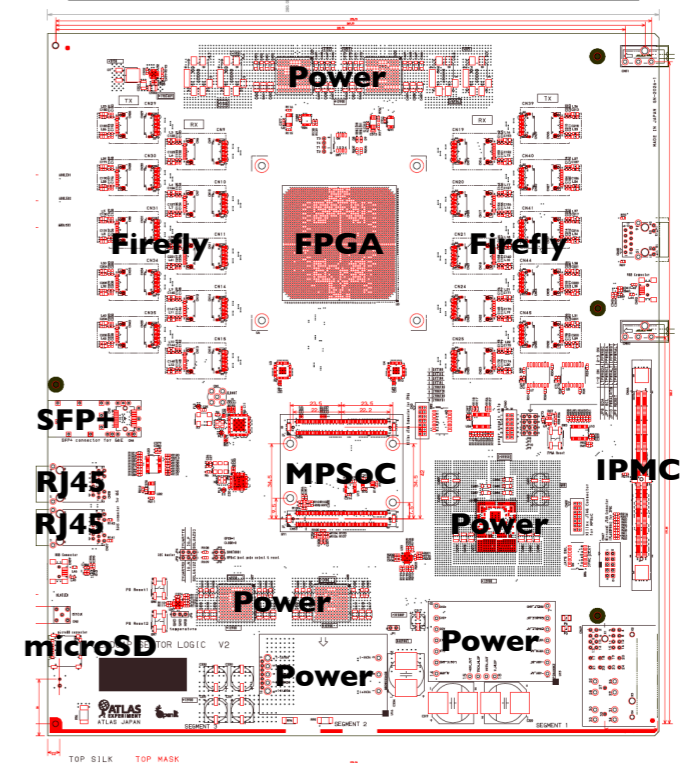
- XCVU13P FPGAの特徴 (SLR、UltraRAM) を考慮したファームウェアの作成が進行中
- 一部のファームウェアについては全領域に拡張し、回路規模および動作性能が要求を満たすことを確認した。

	ファームウェア開発		MC を用いた動作検証	
	一部領域	全領域	一部領域	全領域
ワイヤー飛跡再構成	○	○	○	○
ストリップ飛跡再構成	○	○	Ongoing	×
Wire/Strip coincidence	○	×	Ongoing	×
Inner coincidence	○	×	Ongoing	×

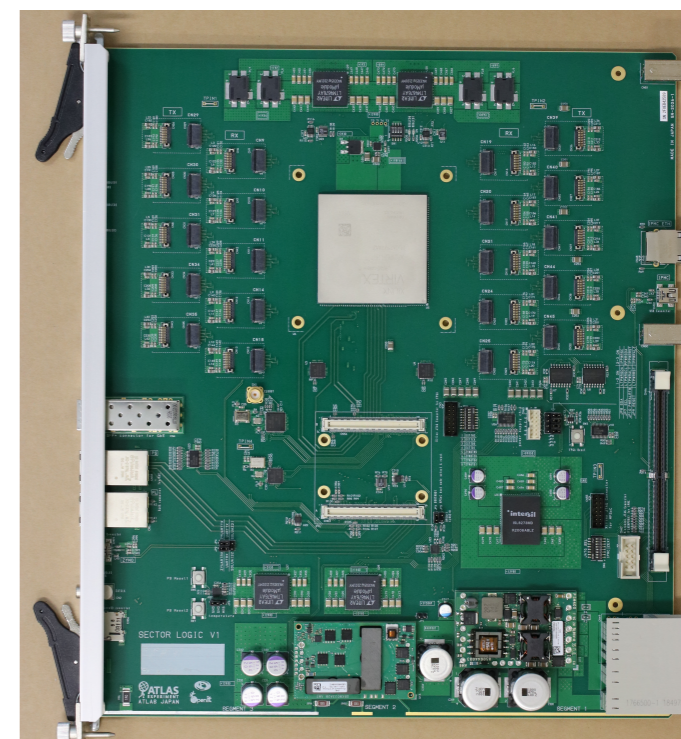
今後の計画

- SL 第一試作機が完成したため、ワイヤー飛跡再構成ファームウェア(トリガー領域全体)の実機テストを行う。
- スライステストの領域をトリガー領域全体に拡張し、FPGA 全体での素子の使用率を確認する。
- 後段に出力するミュオン候補の選別を行うアルゴリズム (Track selector) の開発を進める。

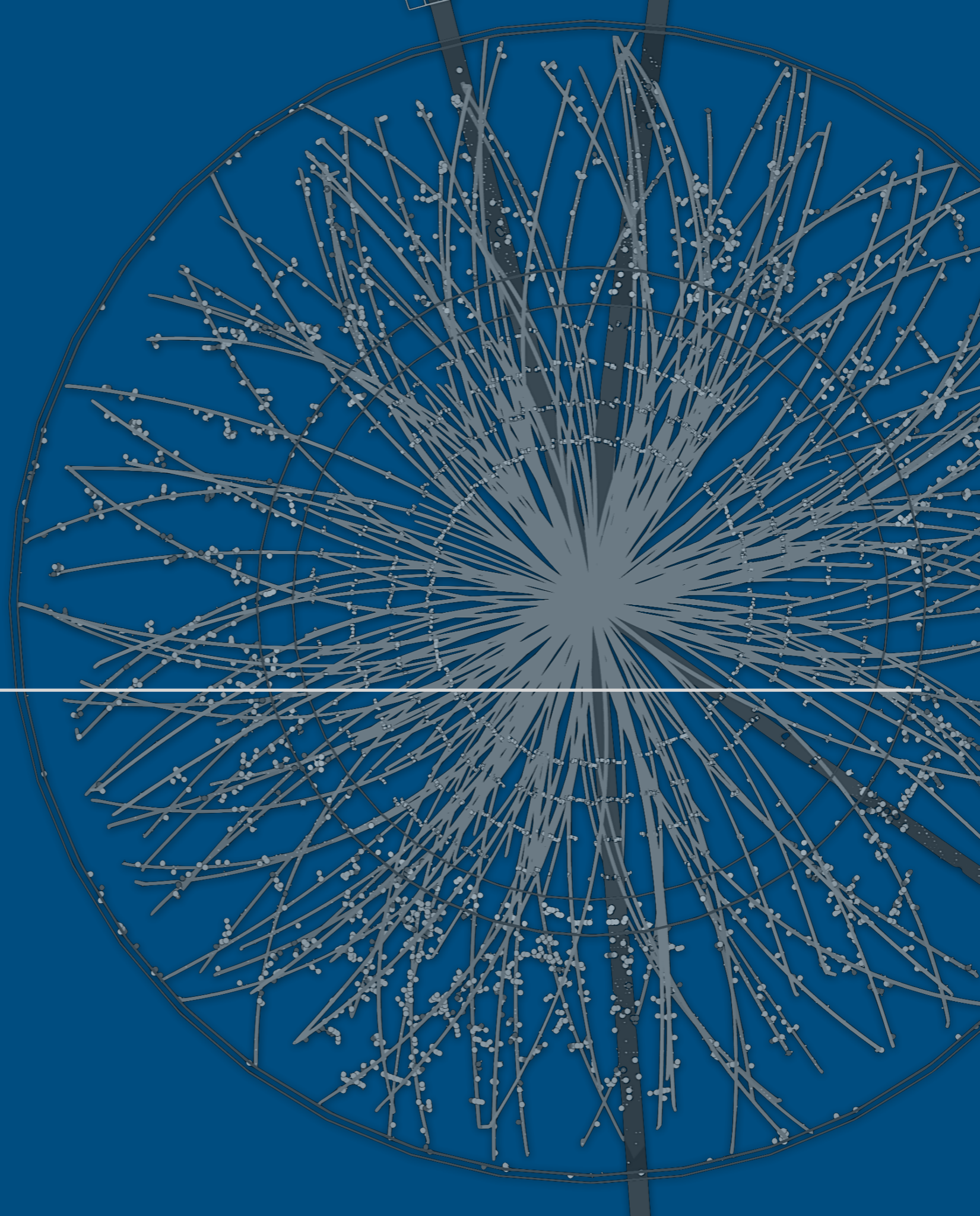
SL 第一試作機のレイアウト



SL 第一試作機

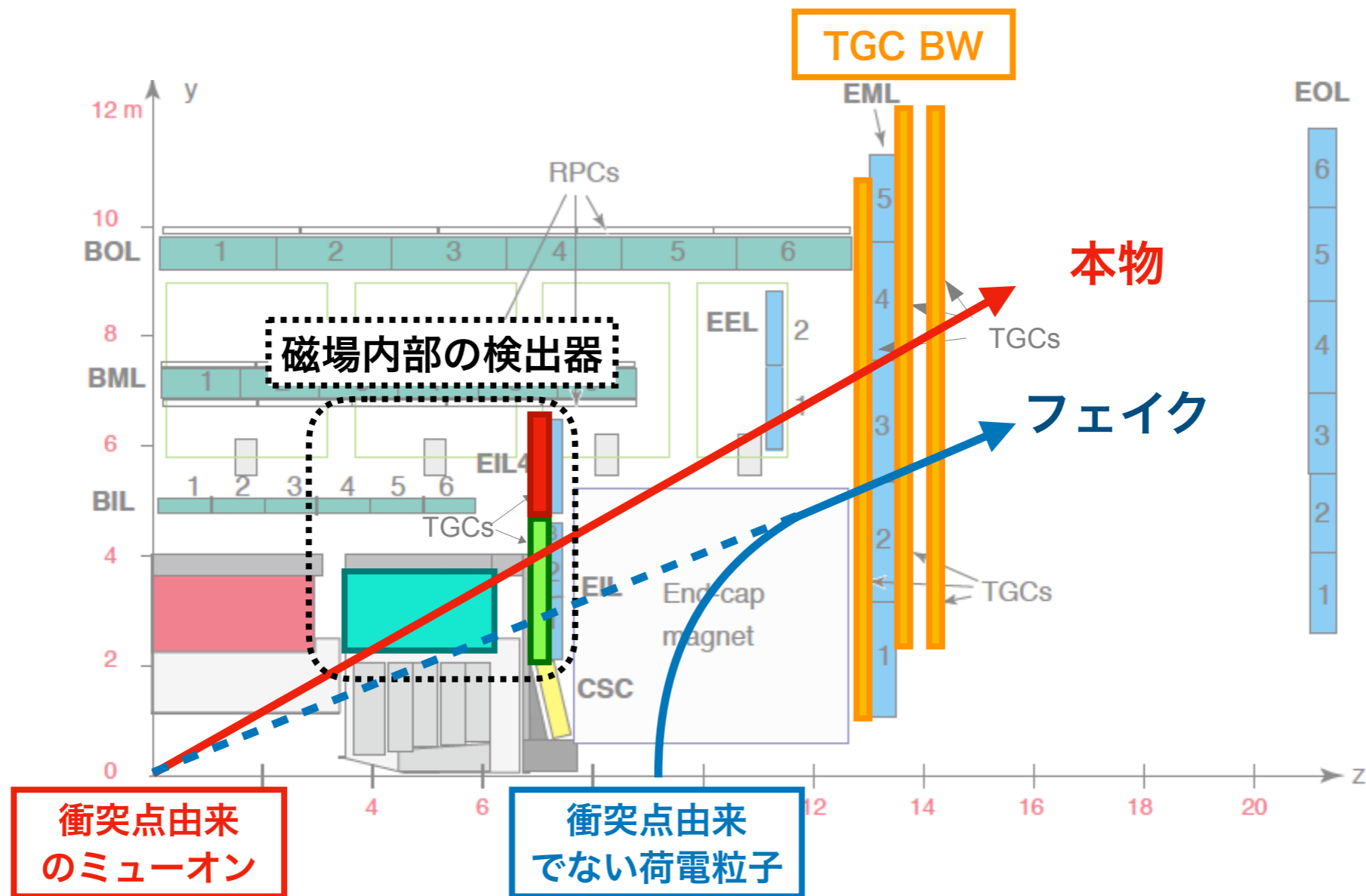


Backup



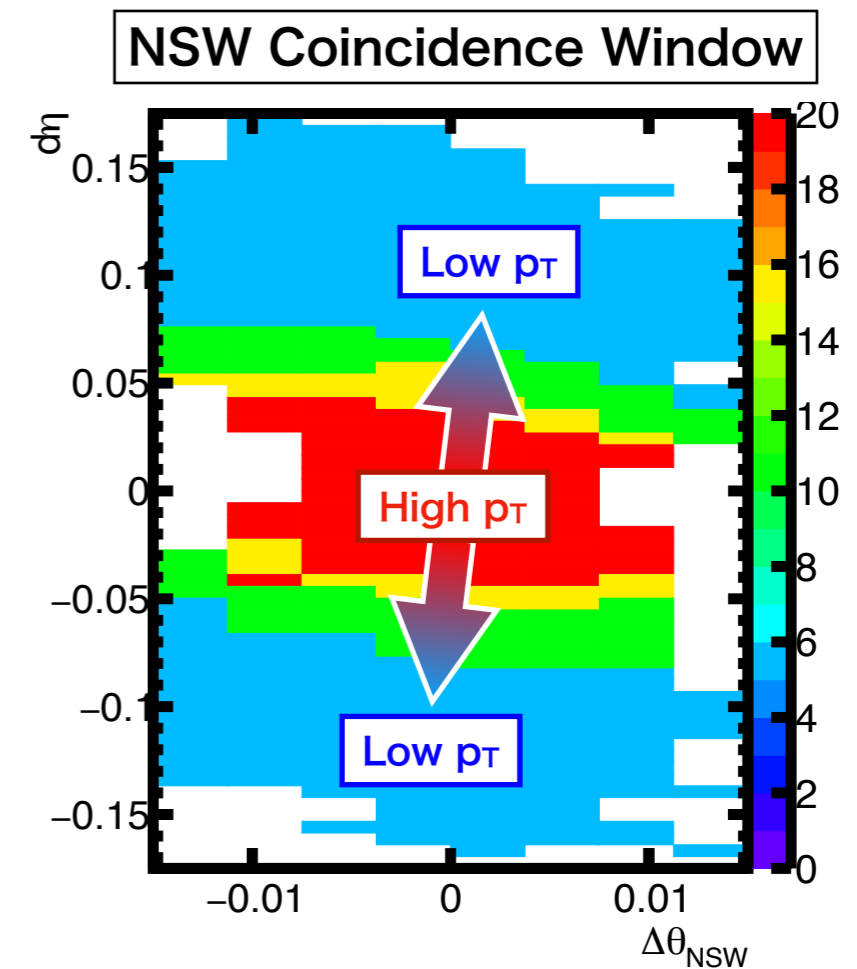
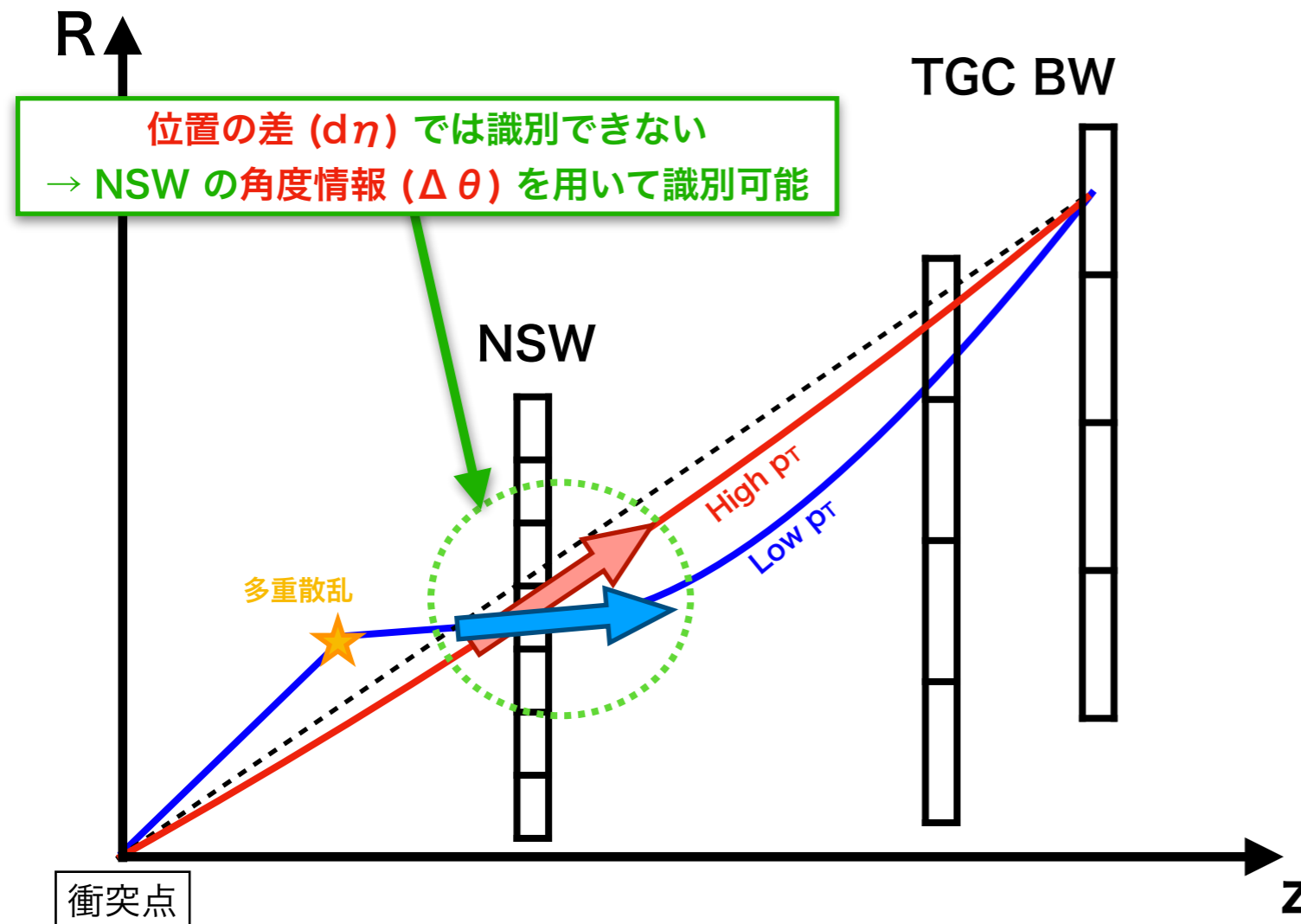
ミュオントリガーの背景事象

- エンドキャップ部におけるミュオントリガーの背景事象
 - ① 衝突点由来でない荷電粒子によるトリガー。(フェイクトリガー)
 - ➔ 磁場内部の検出器にヒットを要求することで削減する。
 - ② TGC BW の分解能では削減できない低い p_T のミュオンによるトリガー。



磁場内部の検出器を用いたトリガーロジック

- 低い p_T のミュオンは磁場領域で η 方向に大きく曲がる。
 - ▶ TGC BW で再構成された位置と磁場内部の検出器で測定した位置の差 ($d\eta$) に閾値を設けて低い p_T のミュオンによるトリガーを削減する。
- NSW では飛跡の再構成を行うため、角度情報も出力される。
 - ▶ 位置の差 ($d\eta$) に加えて飛跡の角度 ($\Delta\theta$) を用いることで多重散乱由来の低い p_T のミュオンによるトリガーを削減できる。



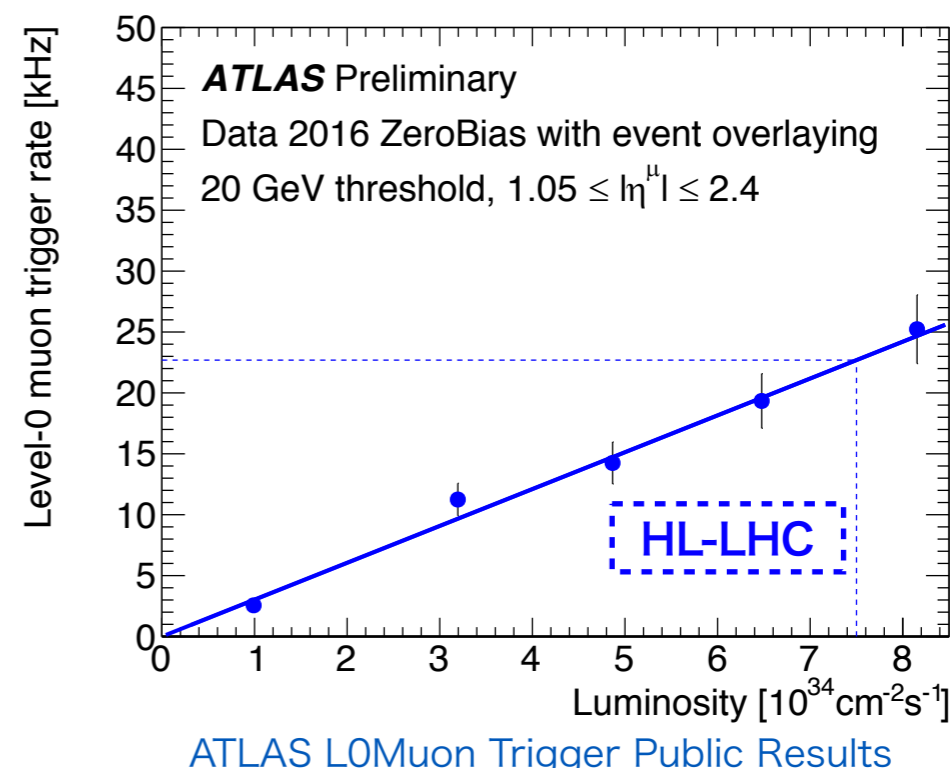
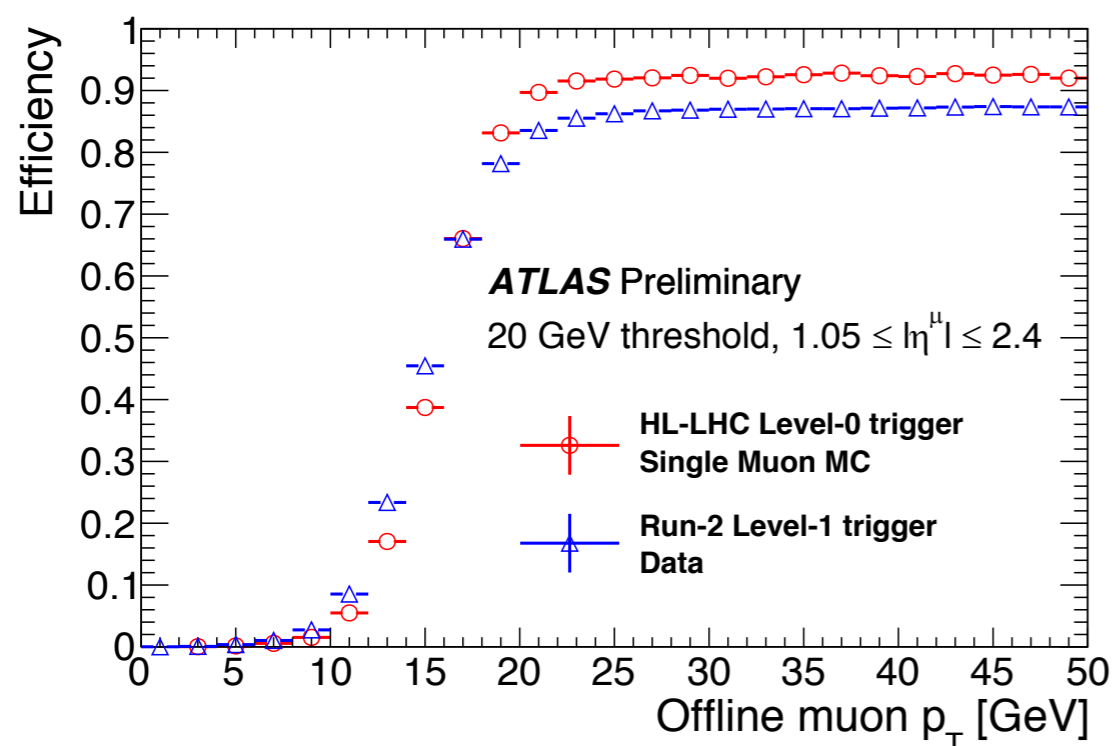
HL-LHC におけるミュオントリガーの性能

検出効率

- Run-2 トリガーとの検出効率を比較した。
 - ▶ 全ての TGC のヒット情報を使用することで p_T 閾値以上の検出効率が $\sim 4\%$ 向上。
 - ▶ 低い p_T のミュオンの削減能力が向上。

トリガーレート

- ランダムトリガーで取得した Run-2 データを用いて HL-LHC のトリガーレートを見積もった。
 - ▶ 20 GeV 閾値のトリガーレートは ~ 23 kHz。
(Level-0 全体のトリガーレート of 1 MHz の $\sim 2.3\%$ を占める。)
 - ▶ MDT を使用することで更なるトリガーレートの削減が期待される。



HL-LHC における処理時間の見積もり

HL-LHC において Sector Logic に要求される処理時間

Contents	Latency
TGC hit signal arrival	0.888 μs
Coincidence of TGC BW	1.013 μs
TGC BW and TGC EI coincidence	1.063 μs
TGC BW and RPC BIS78 coincidence	1.360 μs
NSW track candidate and TileCal signal arrival	1.425 μs
TGC BW and NSW (TileCal) coincidence	1.450 μs
Final selection of track candidate	1.475 μs
⋮	⋮

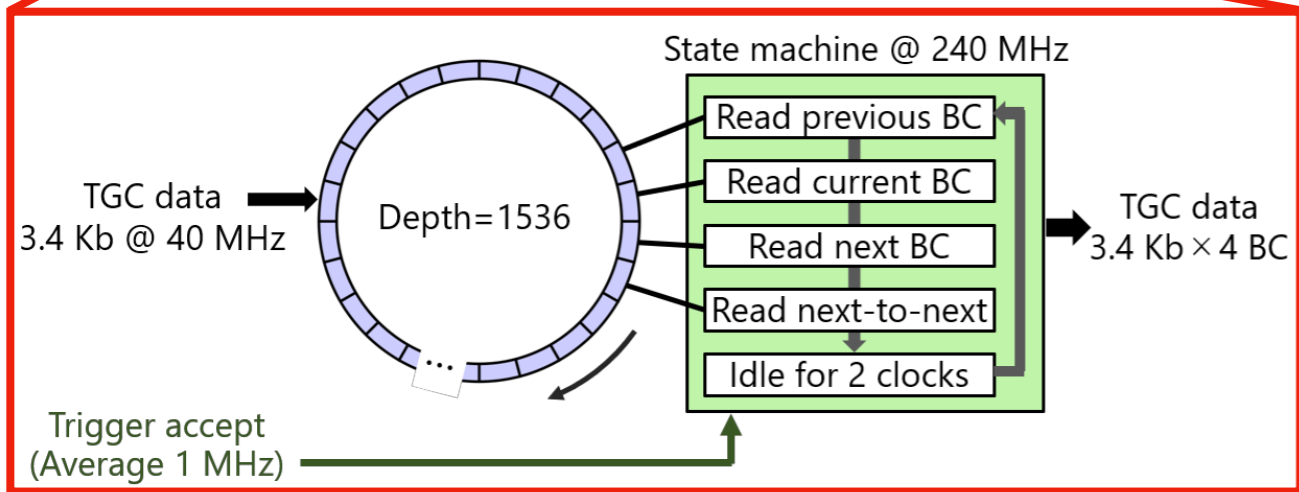
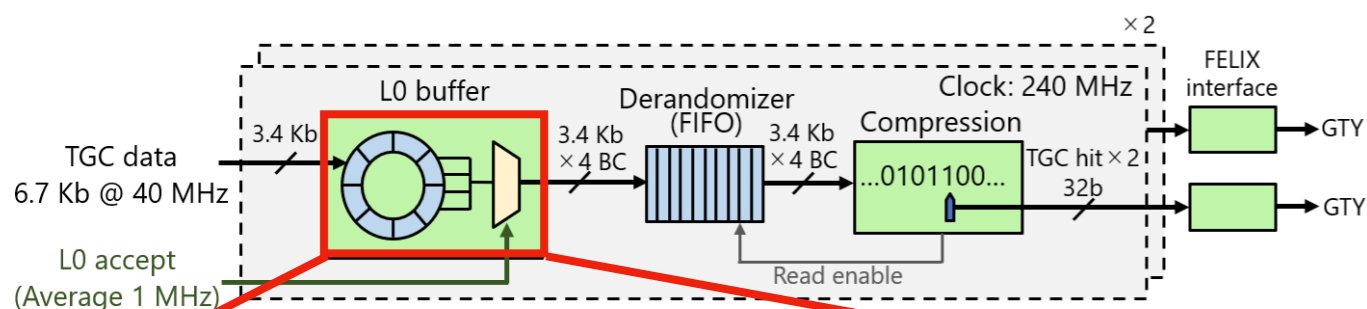
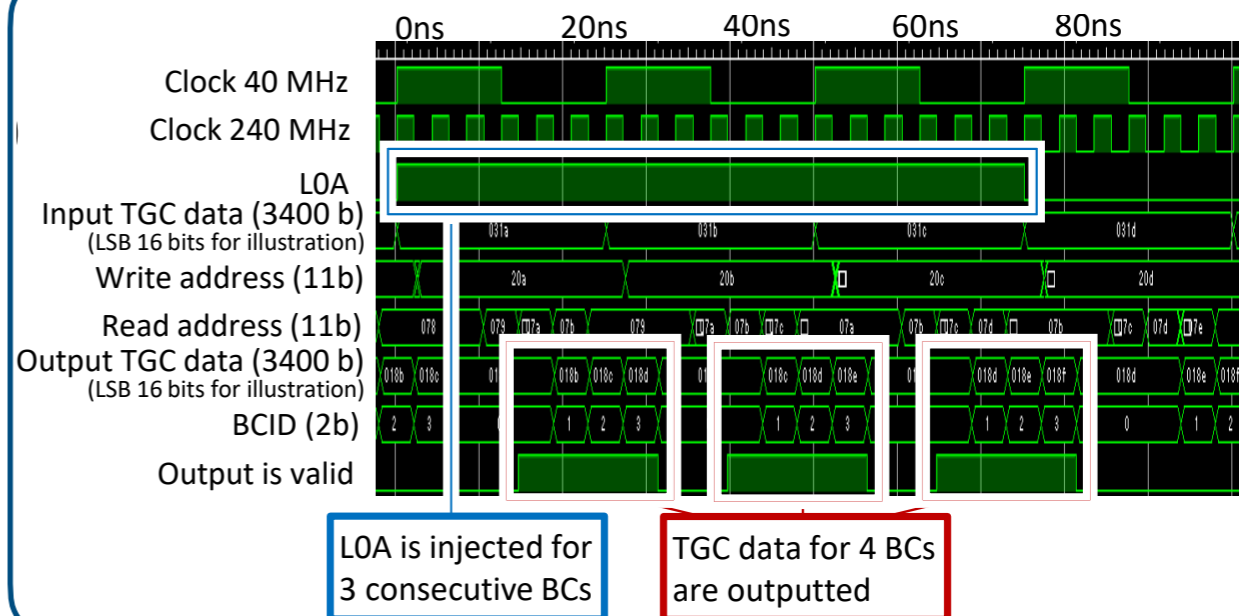
0.125 μs

- 飛跡再構成の処理時間は **0.125 μs** 以内であることが要求される。

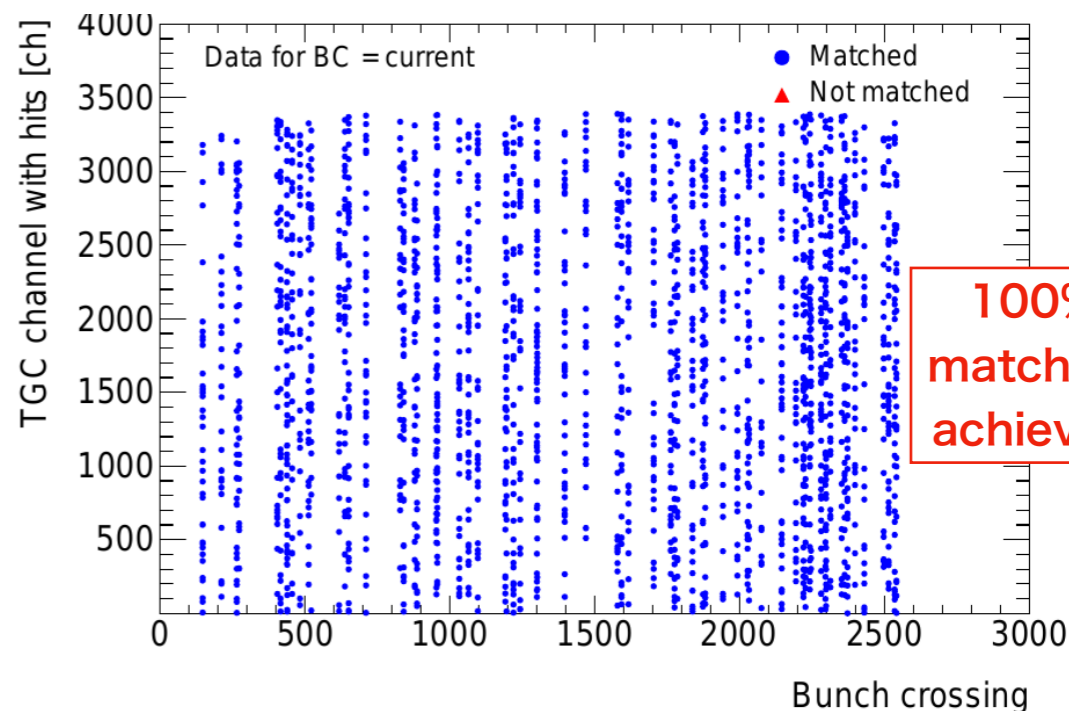
読み出しファームウェア

- HL-LHC では 4 バンチ交差分 (Bunch Crossing : BC) のデータを読み出す。
 - フルサイズのファームウェアを作成し、ランダムデータを入力として動作を検証した。

4 BC 読み出しファームウェアの動作検証

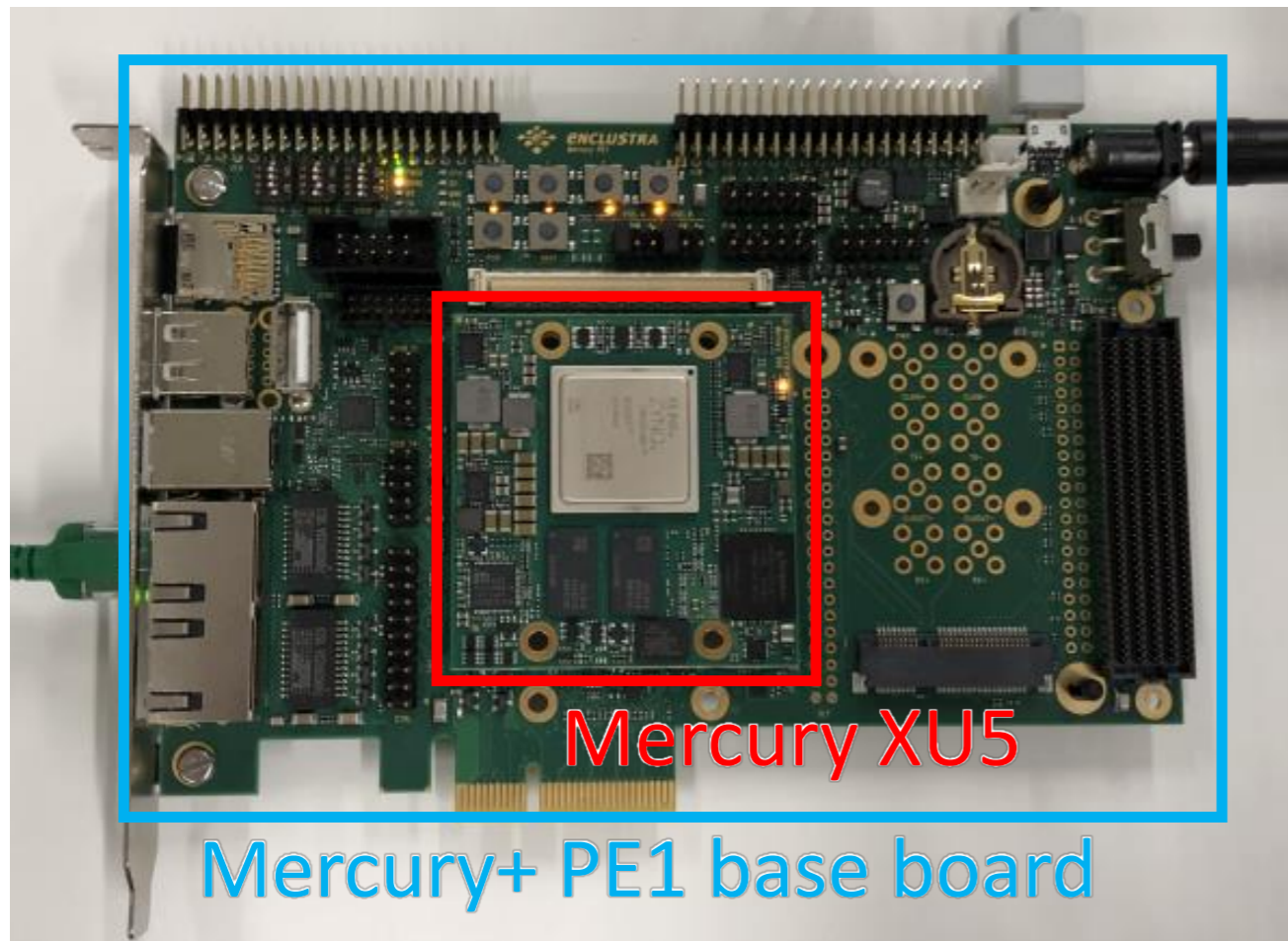


ランダムデータによる入力と出力の比較



コントロールファームウェア

- MPSoC は Xilinx Zynq UltraScale+ MPSoC を採用。
 - ▶ SL にマウントされる商用のメザニンカード(Mercury XU5) に実装されている。
- Xilinx Virtual CableとAXI C2Cにより、SL上のFPGAのコンフィギュレーションや制御を行う。



- Enclustra Mercury XU5 mezzanine
 - ▶ Xilinx Zynq UltraScale+ MPSoC
 - ▶ DDR4 SDRAM, eMMC flash, quad SPI flash,
 - ▶ Gigabit ethernet PHY, USB
- PetaLinux (Xilinx cross-compiler) を使用して MPSoC 上で Linux のブートに成功
- SL 上の FPGA をコントロールするためのファームウェアを開発中
- ファームウェアの動作検証のためにループバックテストと chip-to-chip テストを行う予定

TGC検出器回路系の模式図

