

高輝度LHC-ATLAS実験に向けた 初段ミュオントリガー論理回路の検証システムと性能評価

東京大学理学系研究科 物理学専攻
修士課程2年 牧田藍瑠

高輝度LHC-ATLAS実験

LHC (Large Hadron Collider)

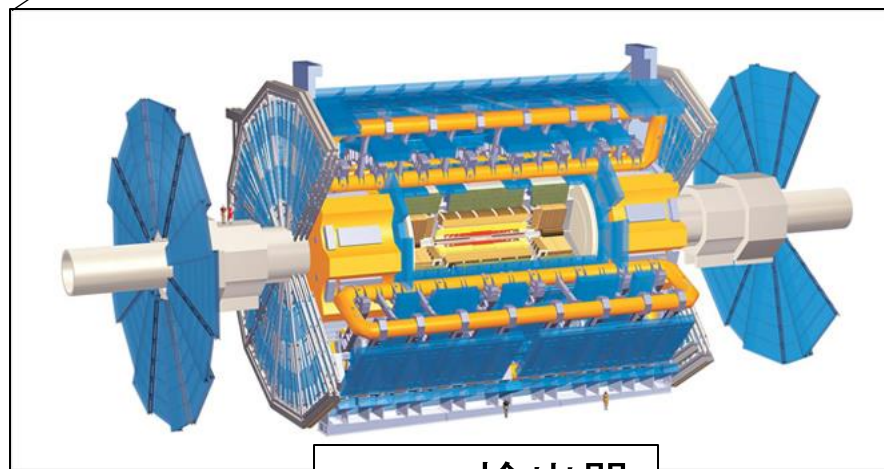
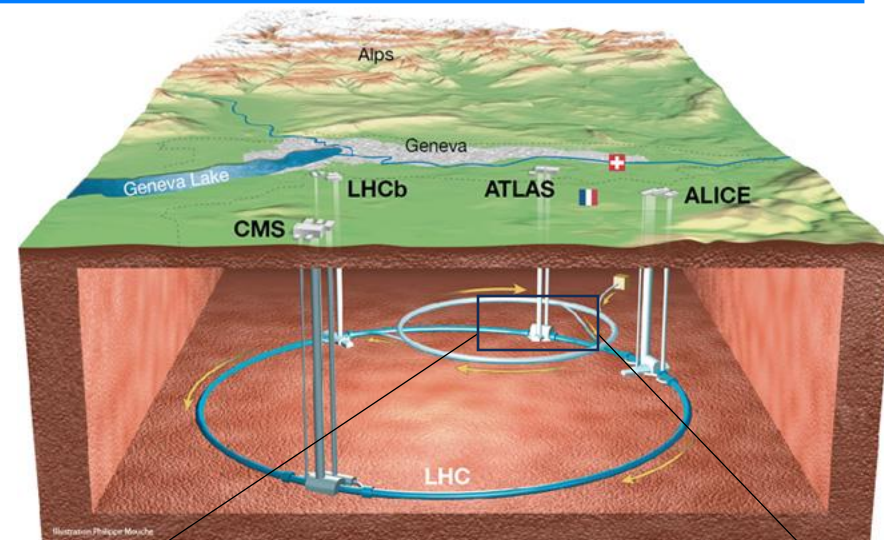
- 世界最高エネルギー 13.6 TeVの陽子-陽子衝突
- 40 MHzの陽子-陽子バンチ交差

ATLAS検出器

- 陽子衝突の生成粒子を検出する大型汎用検出器

高輝度LHC-ATLAS実験 2029年開始

- 高統計を活かした新物理探索+標準模型精密測定
- 瞬間最高ルミノシティ 約3倍に増強
 $2 \times 10^{34} [\text{cm}^{-2}\text{s}^{-1}] \rightarrow 5-7.5 \times 10^{34} [\text{cm}^{-2}\text{s}^{-1}]$
- イベントレートが増大
→ Trigger DAQシステムをアップグレード



ATLAS検出器

高輝度LHC-ATLAS実験のTDAQシステム

TDAQシステムアップグレード

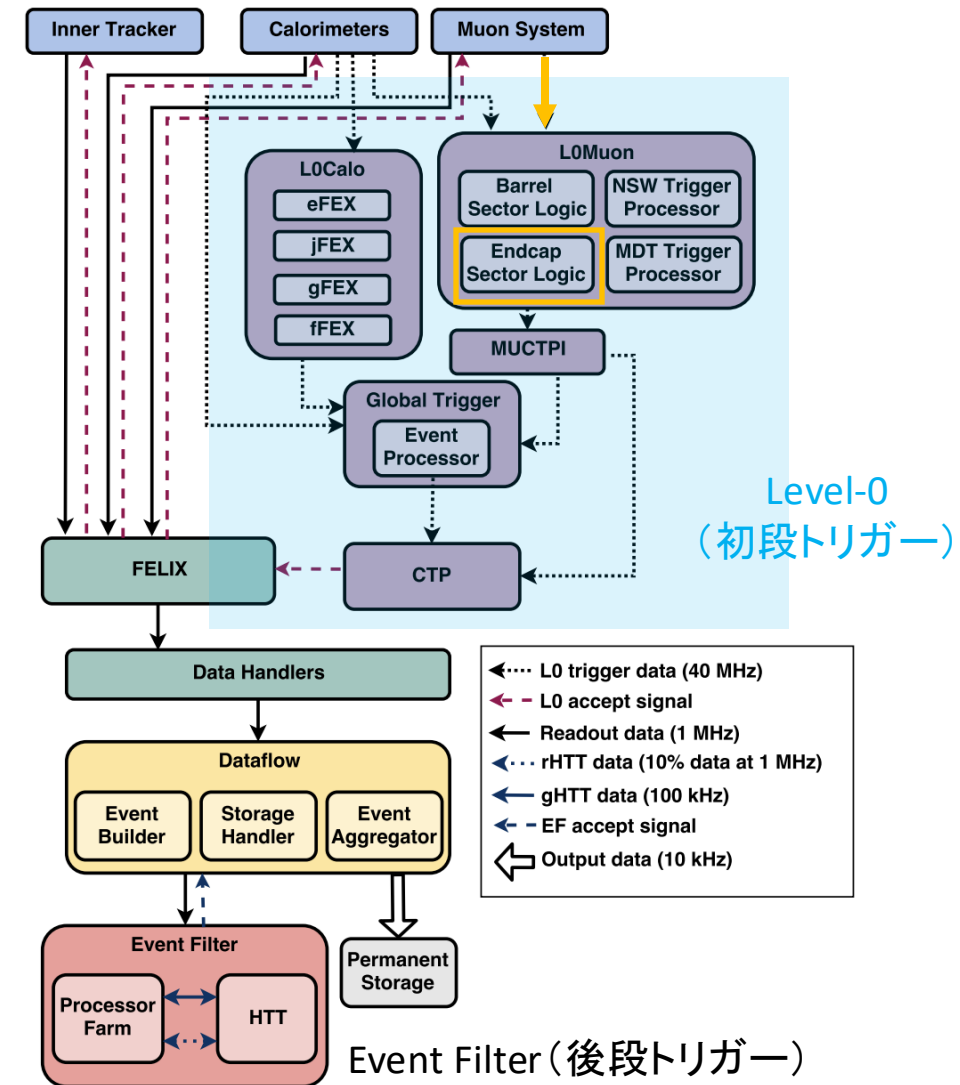
- アクセプタンスを落とさない読み出し
- 多様な物理事象を精度良くトリガー

初段トリガーの増強

- レート: 100 kHz \rightarrow 1 MHz (レート増大に対応)
- レイテンシー: 2.5 μ s \rightarrow 10 μ s (高精度トリガー)

高輝度LHC-ATLASのLevel-0(L0)トリガー

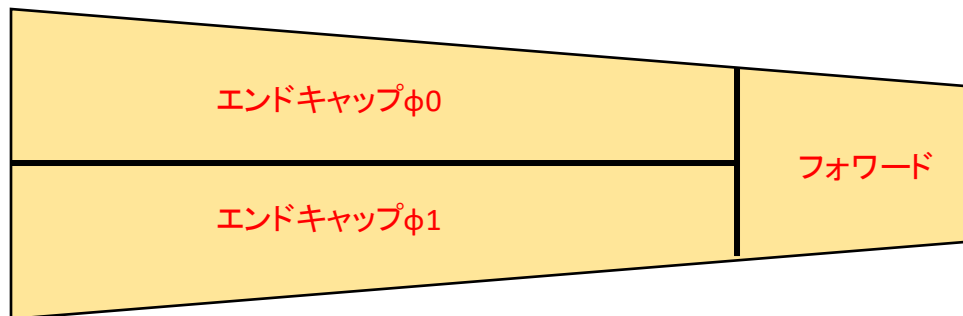
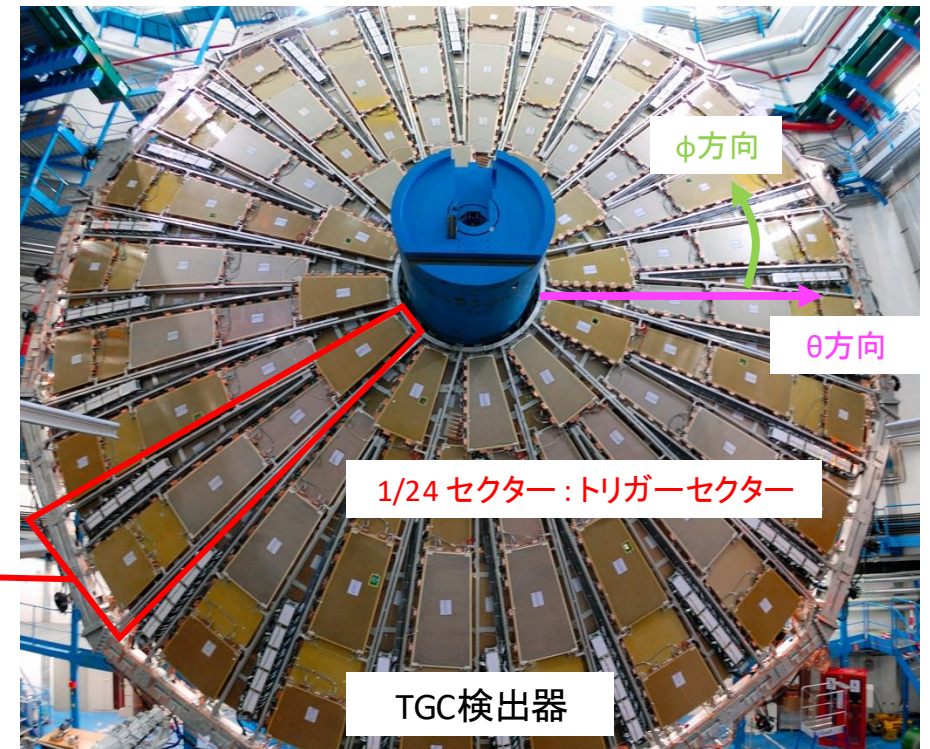
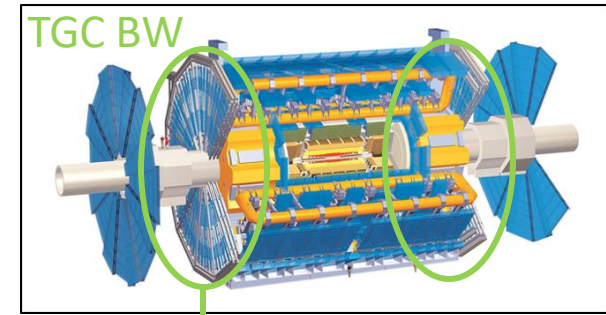
- ハードウェアベースのトリガー
- カロリメータとミュオン検出器からの情報でCTPがトリガー判定
- CTPはトリガー判定(L0 Accept)を各検出器に分配
- Endcap Sector Logic (SL) はTGC検出器のヒット情報からミュオン飛跡候補を出力
 \rightarrow TGC検出器のエレクトロニクスを刷新



TGC検出器

Thin Gap Chamber (TGC) 検出器

- θ 方向と ϕ 方向の2次元読み出しMWPC
- 高速ミュオン再構成
- エンドキャップ初段ミュオントリガー担当
- 3ステーション・各2または3層
- 24個 × 両側で48個のトリガーセクター



トリガーセクター: 独立したトリガー論理回路の区画

高輝度LHC-ATLAS実験のTGCエレクトロニクス

ASD

- TGCからのアナログ信号をデジタル化

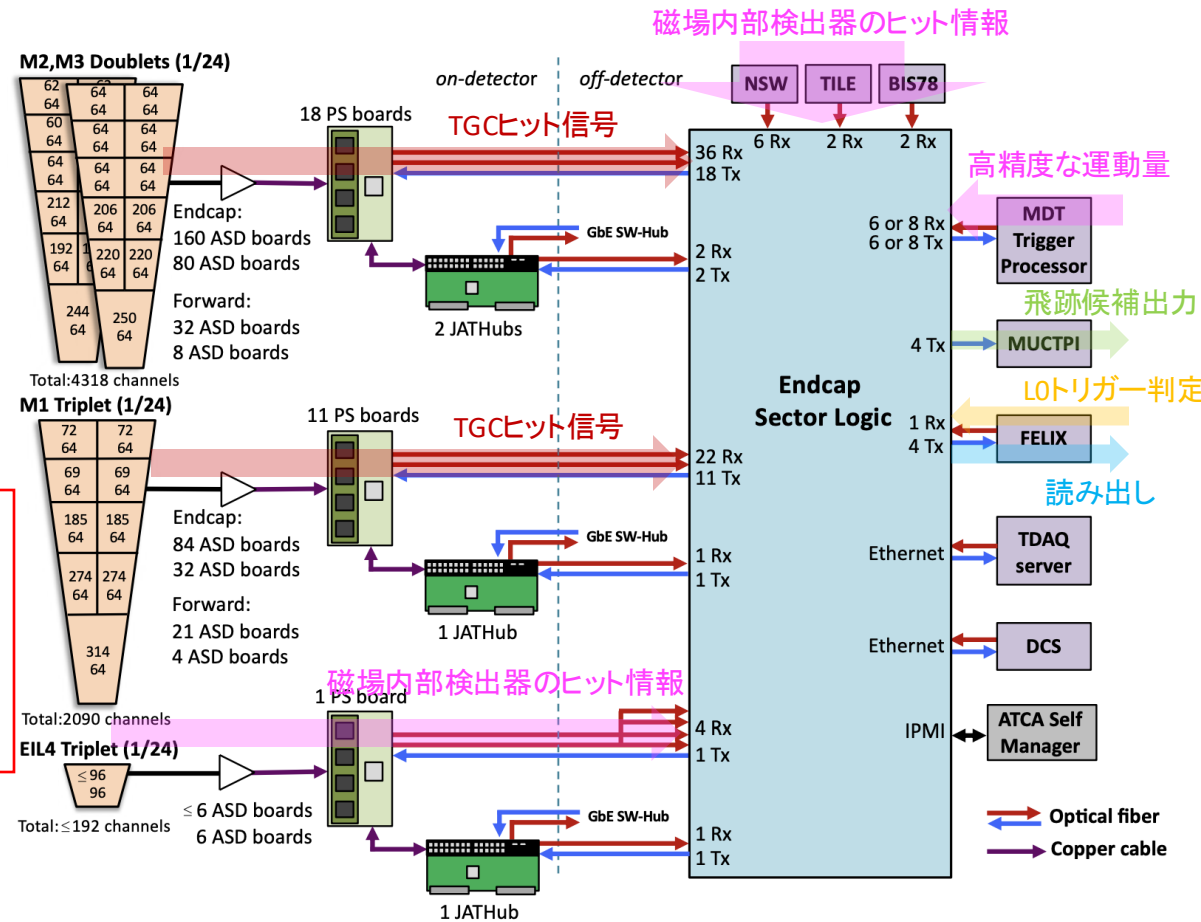
PS Board

- ASDからのヒット信号をLHCの陽子バンチ交差に同期
- バンチ交差毎のヒットを固定長のビットマップにして、シリアル通信で後段へ送信

Endcap Sector Logic (SL)

トリガー論理回路(本研究の対象)

- PS Boardからのヒット信号のビットマップをもとに
ミュオン飛跡を再構成し横方向運動量概算
- 磁場内部の検出器や高精度なミュオンスペクトロメータからの情報を使って高精度な飛跡情報を追加
- ミュオン飛跡候補を後段へ出力
- L0トリガー判定までヒット信号をバッファ
- L0 Acceptが出た場合にはヒット信号およびトリガー中間出力を読み出し



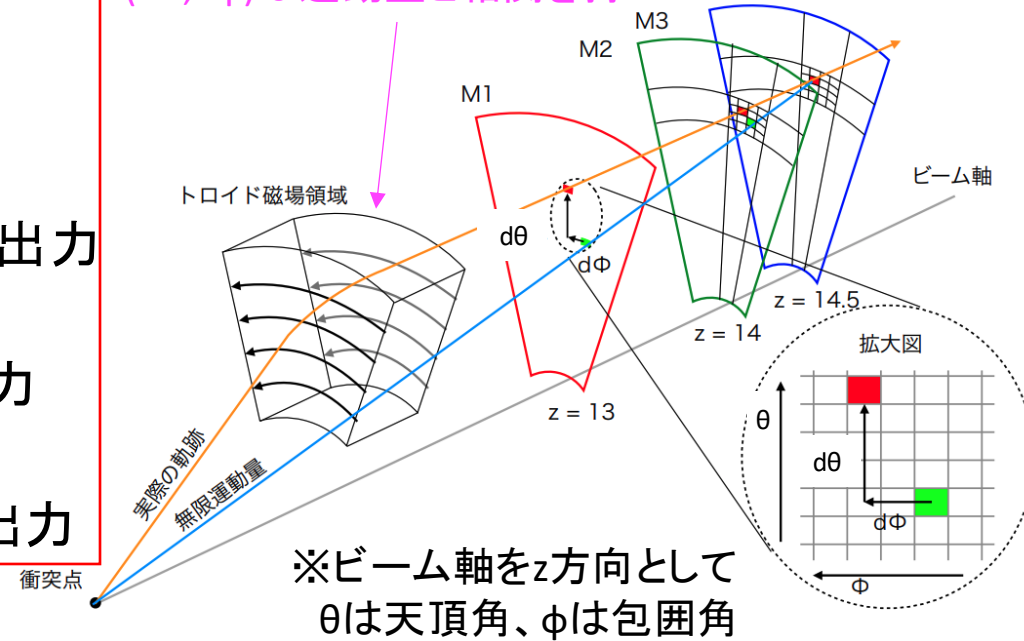
TGC SLトリガー論理回路の概観と開発状況

トリガー論理回路は以下の6段階

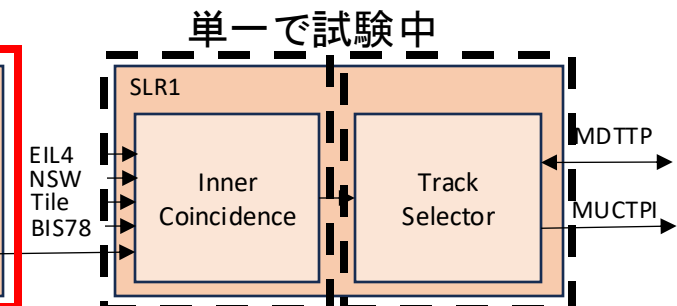
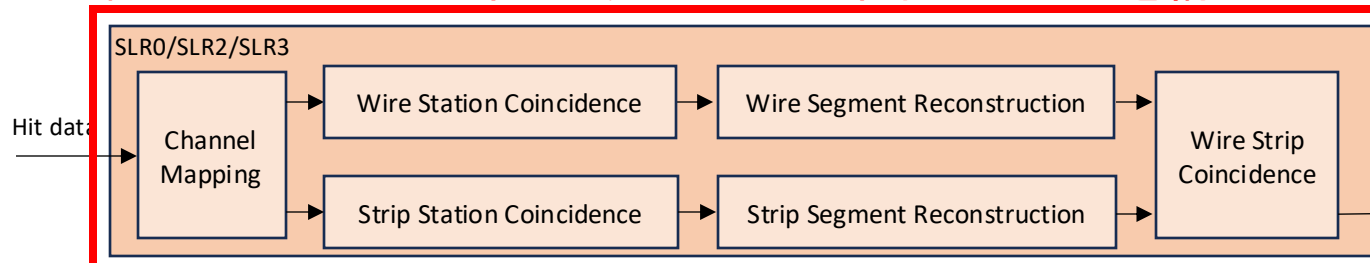
実装済み

1. Channel Mapping:
TGCのヒットデータをトリガー論理回路の入力に合う形で接続
2. Station Coincidence:
Station内でコインシデンスを取り、M1/2/3各々から代表点を出力
3. Segment Reconstruction:
代表点3つから直線飛跡を再構成して角度情報($d\theta, d\phi$)を出力
4. Wire Strip Coincidence
($d\theta, d\phi$)から横方向運動量閾値 p_T threshold(と座標情報)を出力
5. Inner Coincidence } 仮実装部分が残っている
6. Track Selector }

トロイド磁場で θ 方向に曲がる
($d\theta, d\phi$)は運動量と相関を持つ

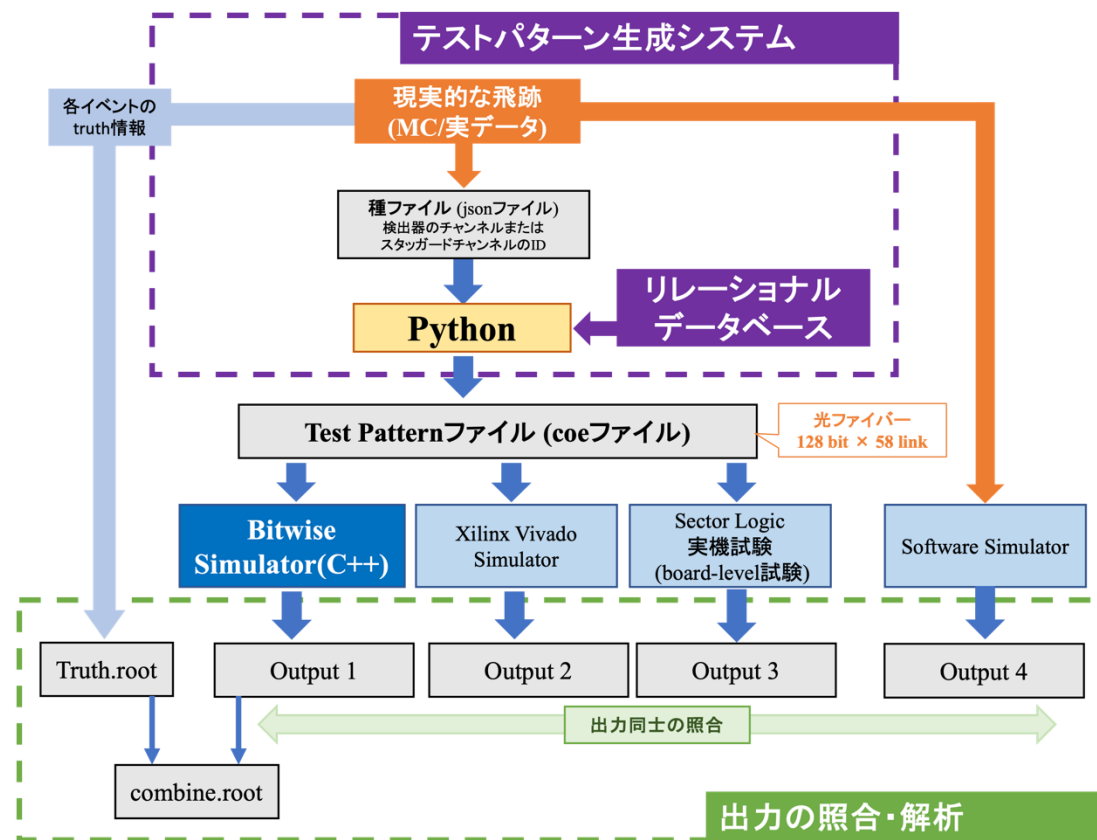


論理回路の性能に問題あり(後述) → 検証システムを構築



検証システムの全体像

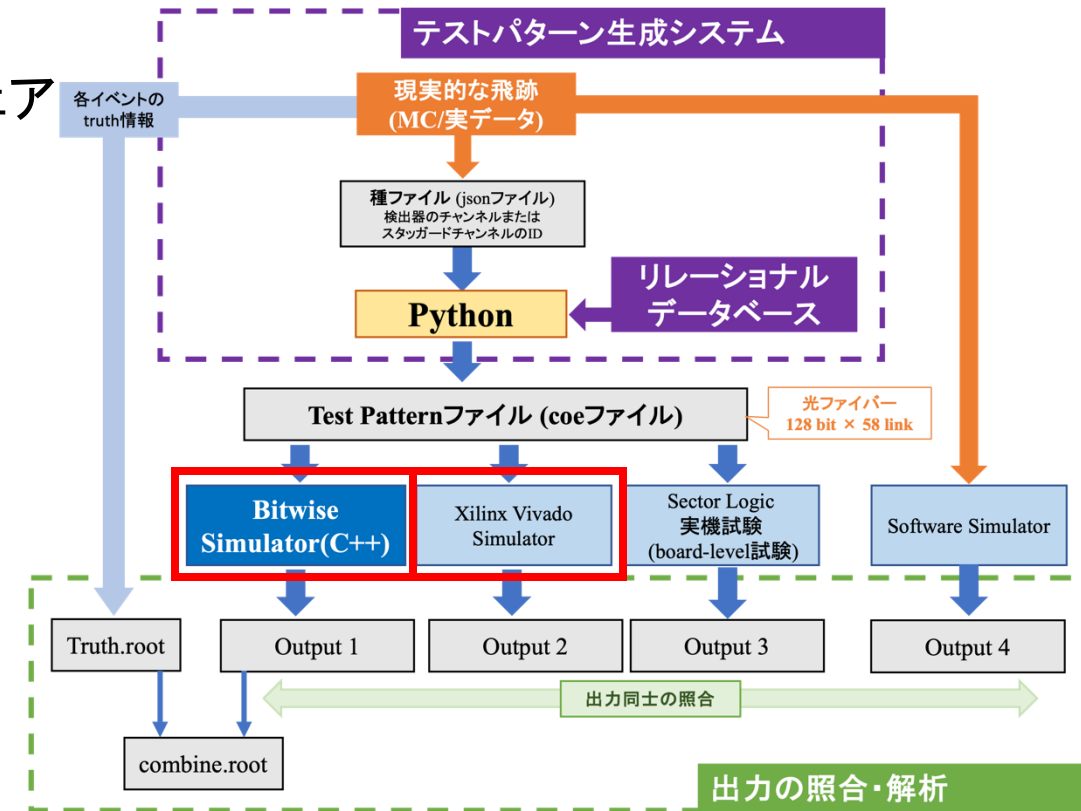
- テストパターン生成システム
 - ATLAS実験でのシミュレーションデータを入力データフォーマットに変換
 - 全システムに共通の入力テストパターン
- Bitwise Simulator
 - トリガー論理回路をトレースしたソフトウェア
- Vivado Simulator
 - トリガー論理回路ファームウェアをシミュレーションするテストベンチ
- SL実機試験
 - トリガー論理回路ファームウェアを実装したSL実機を使ったテストベンチ
 - 本番環境に近い性能評価が可能
- Software Simulator
 - トリガーアルゴリズムを実装したソフトウェア
 - 論理回路開発に先行して理想性能を示す



Bitwise SimulatorとVivado Simulatorの比較

- Bitwise Simulator
 - トリガー論理回路をbitレベルでトレースしたソフトウェア
 - 任意の信号線がプローブ可能
- Vivado Simulator
 - トリガー論理回路ファームウェアをシミュレートするソフトウェア (Xilinx, Vivado Simulationで実装)
 - 任意の信号線がプローブ可能

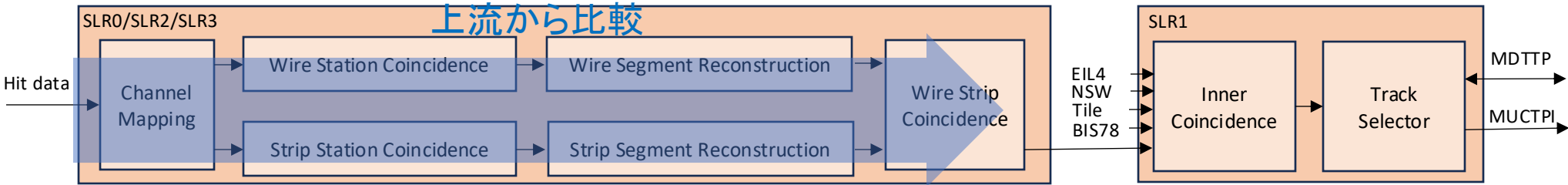
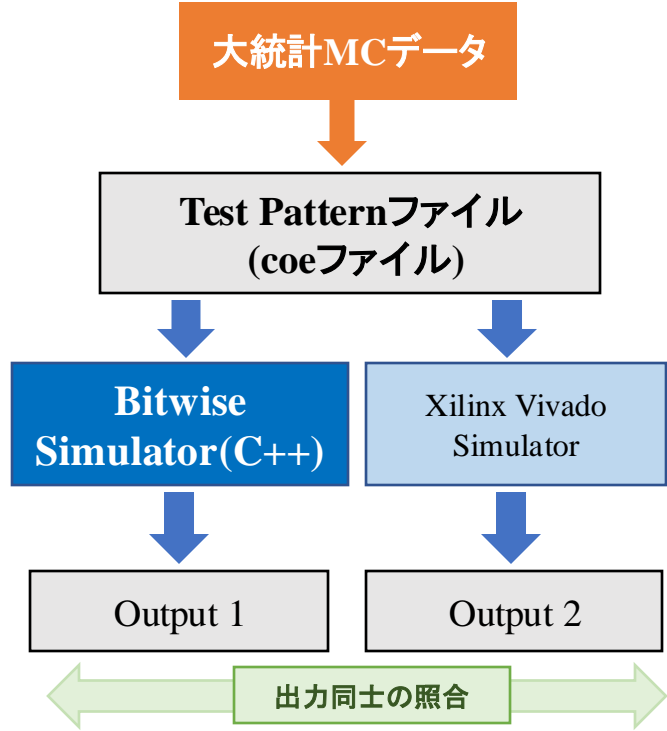
Bitwise simulatorとVivado simulatorでの、
任意の信号線を比較し、
不具合箇所を精密に特定可能



Bitwise SimulatorとVivado Simulatorの比較

具体的な検証手順

1. 共通の大統計テストパターンをBitwise SimulatorとVivado Simulatorに入力
 - シングルミュレーション 8603 イベント
2. 不一致が起きたイベントをリストアップ
3. 不一致イベントひとつに注目し、両者の出力をbit単位で上流から順に比較
4. 不一致が発生した箇所を特定→修正を繰り返す →両者で複数の不具合を特定・修正した



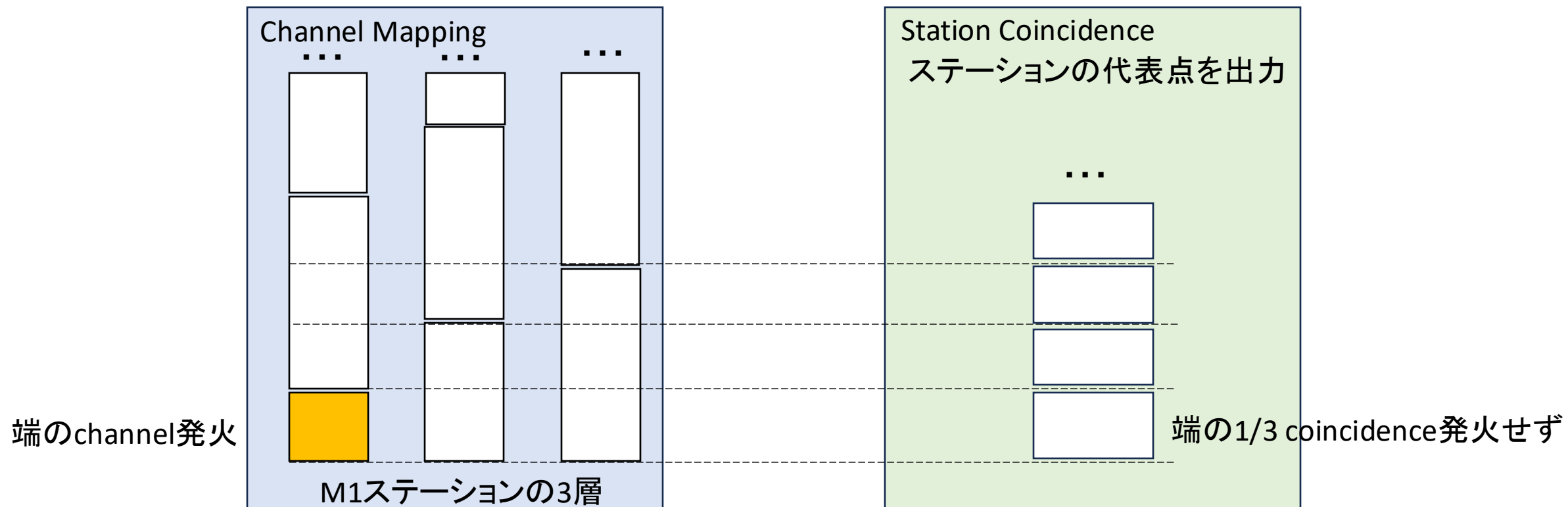
トリガー論理回路の不具合特定1/3

Station Coincidence

Channel Mapping - Station Coincidenceでの出力の比較

検出した不具合(ファームウェア)

- 端のchannelでのコインシデンスの論理式に誤り



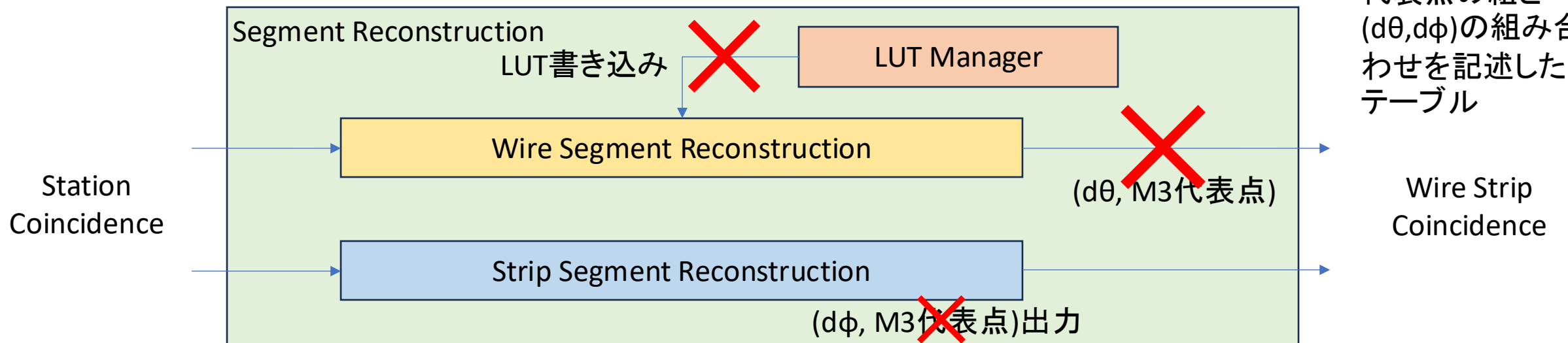
トリガー論理回路の不具合特定2/3

Segment Reconstructionの内部信号を比較

検出した不具合(ファームウェア)

- Wire Segment Reconstruction – Wire Strip Coincidence間の断線
- Strip Segment Reconstructionの出力するM3代表点情報に誤り
- Segment ReconstructionのLUT※書き込みの部分的なエラー

※LUT:
代表点の組と
($d\theta, d\phi$)の組み合
わせを記述した
テーブル



トリガー論理回路の不具合特定3/3

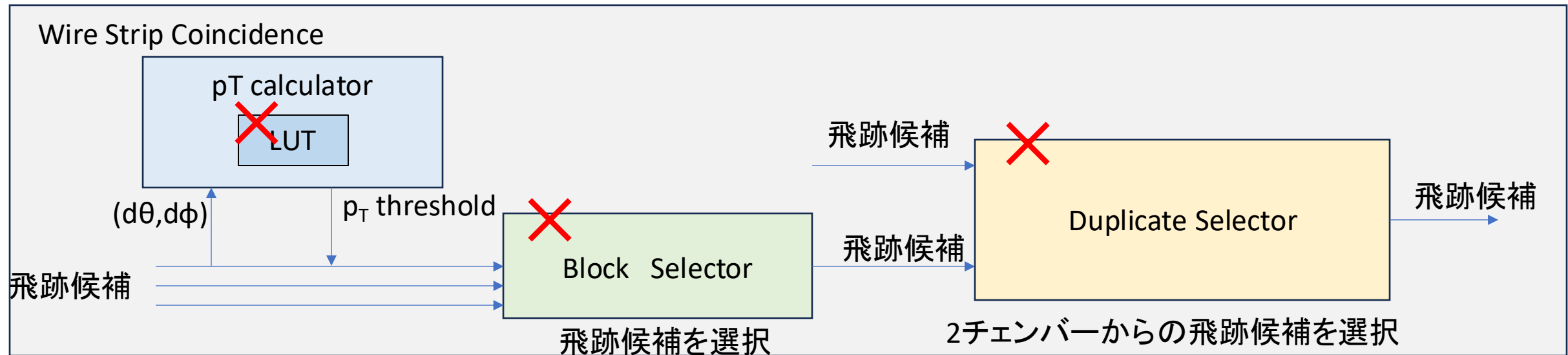
Wire Strip Coincidenceの内部信号を比較

検出した不具合(ファームウェア)

- Block Selectorの条件式に誤り
- Duplicate Selectorの条件式に誤り
- LUT※作成システムとファームウェア間の θ の符号の定義に不一致
→誤ったLUTデータが使用されていた

発火層数と同じときの $(d\theta, d\phi)$ が
小さい飛跡を選ぶ
 $(d\theta, d\phi)$ の大小評価の条件式に誤り

※LUT:
 $(d\theta, d\phi)$ と p_T の組
み合わせを記述
したテーブル



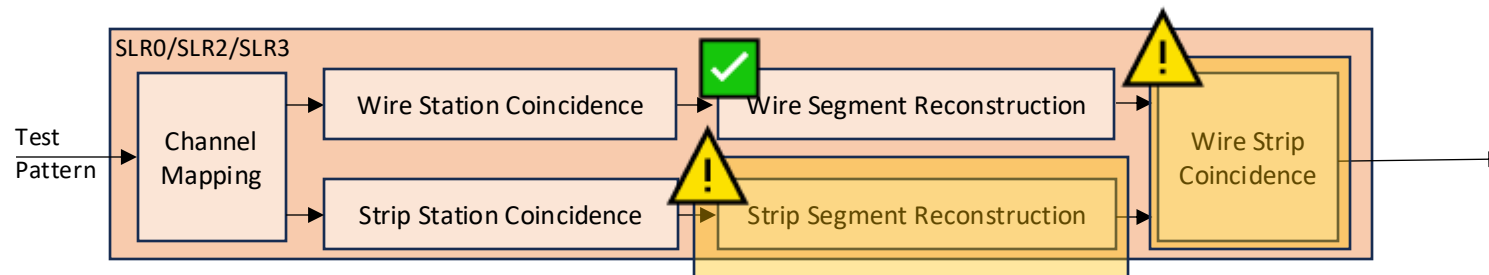
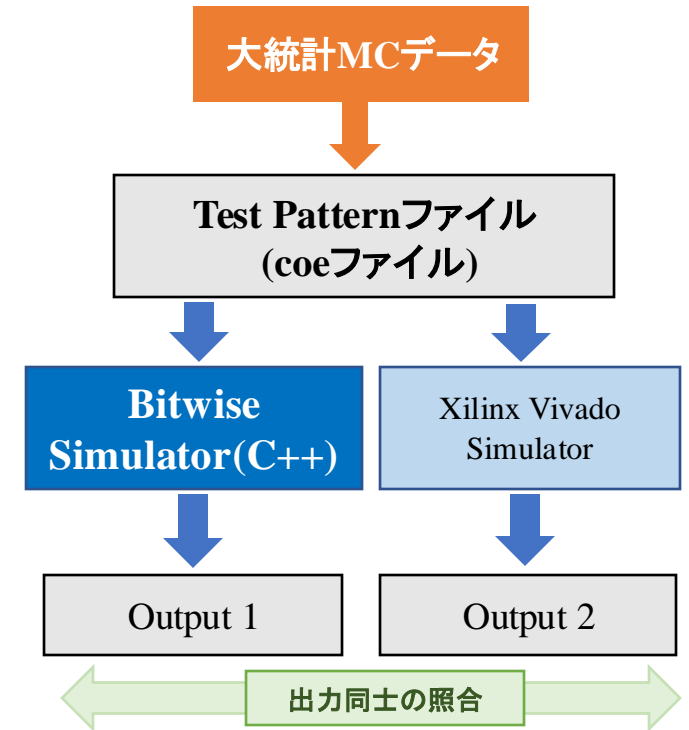
Bitwise sim vs Vivado simの比較での修正結果

現状での不一致イベント数

8603 events中

モジュール	不一致イベント数
Wire Segment Reconstruction	0
Strip Segment Reconstruction	7
Wire Strip Coincidence	4

大統計テストパターンを利用したことで、多くのレアな不具合を検知して修正した
引き続き、残りの不具合の検索を行っていく



トリガー性能評価

現実的なデータに対するトリガー応答を調べるため、大統計シミュレーションデータを用いてトリガー性能を評価した

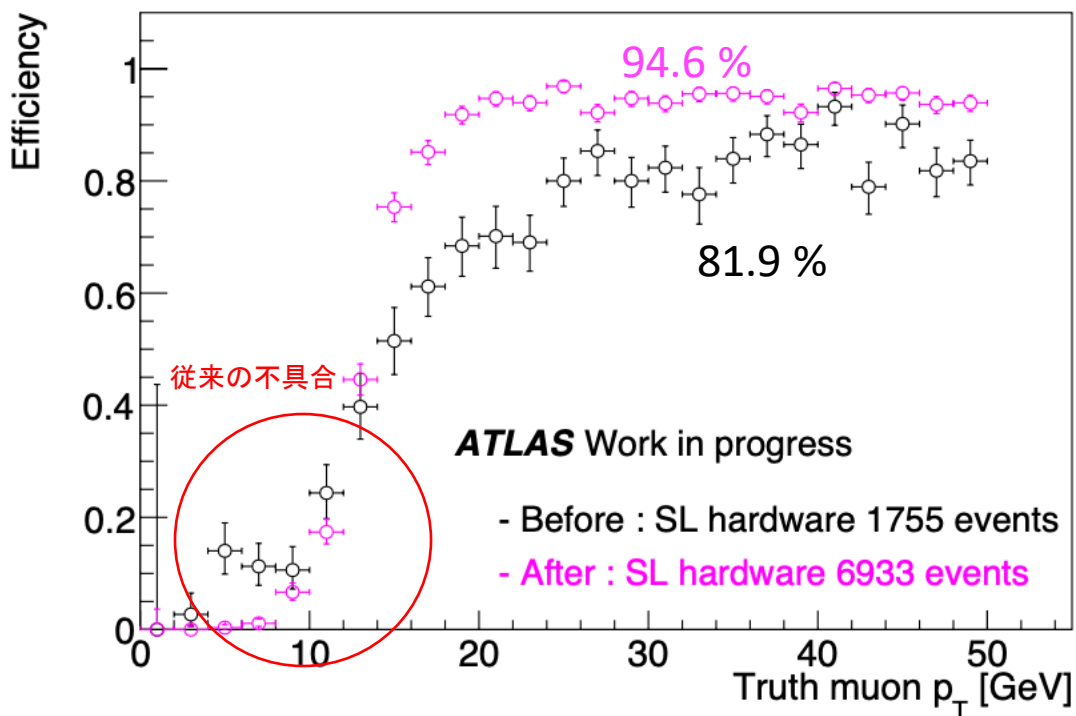
使用したミューオンのシミュレーションデータ

- 多重散乱やクロストークを含む
- パイルアップ無し
- $p_T \cdot \eta \cdot \phi$ について一様な分布を仮定

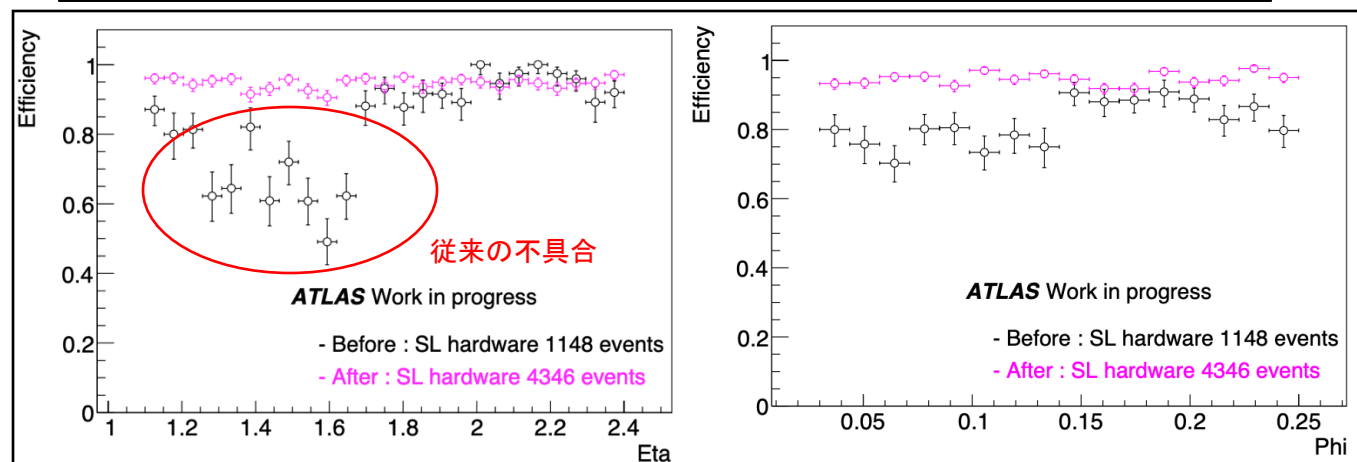
Parameter	カット条件
p_T	$0 < p_T < 50 \text{ GeV}$ (均等に分布)
η	$1.1 < \eta < 2.4$ (均等に分布)
ϕ	$0.03 < \phi < 0.25$ (均等に分布)

修正前後のトリガー性能評価

SL実機試験の修正前後の p_T threshold出力



$$\text{Efficiency} = \frac{p_T \text{ threshold} = 20 \text{ GeV と判定されたイベント数}}{\text{入力したイベント数}}$$



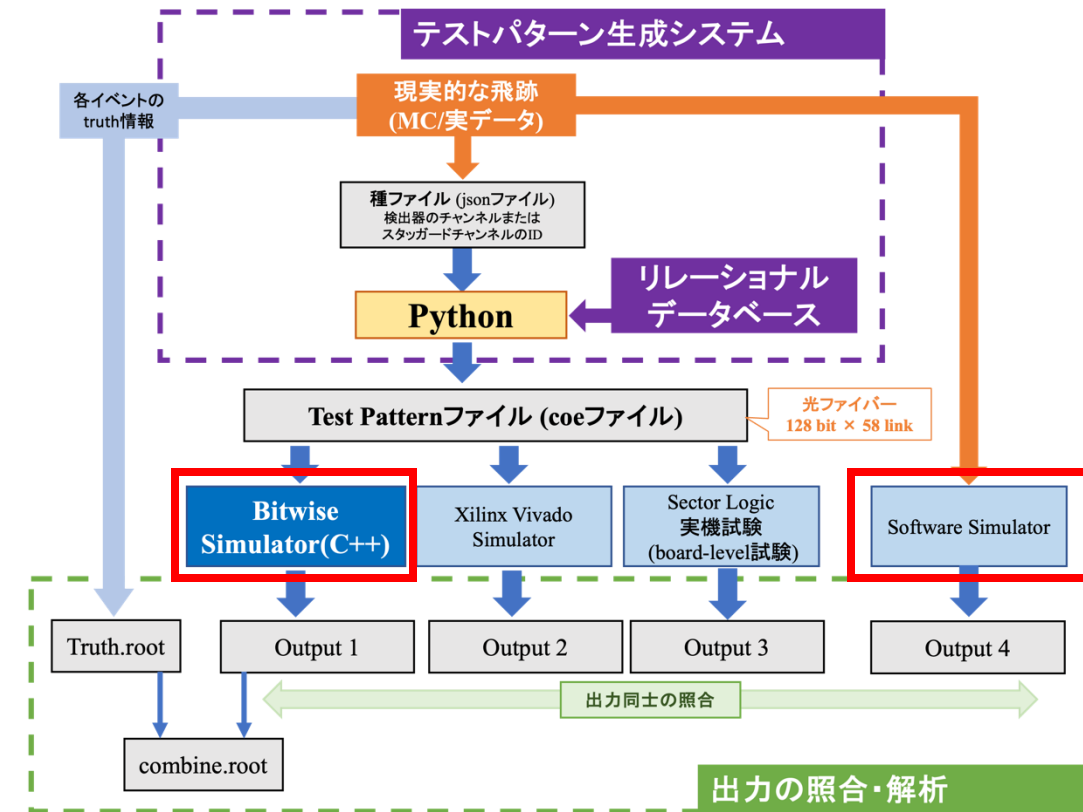
プレート領域 ($p_T > 20$ GeV) での η, ϕ 分布

論理回路の性能として、プレート領域のEfficiencyが81.9% → 94.6%に回復した
本研究による精密な検証システムによって、プレートEfficiency12.7%もの顕著な性能の改善に成功

Software Simulator (理想性能)との比較

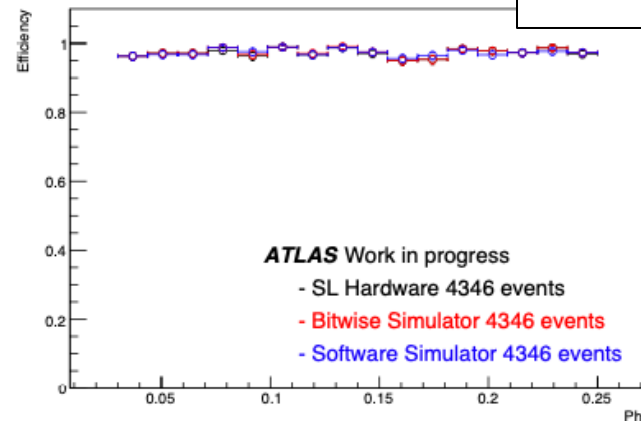
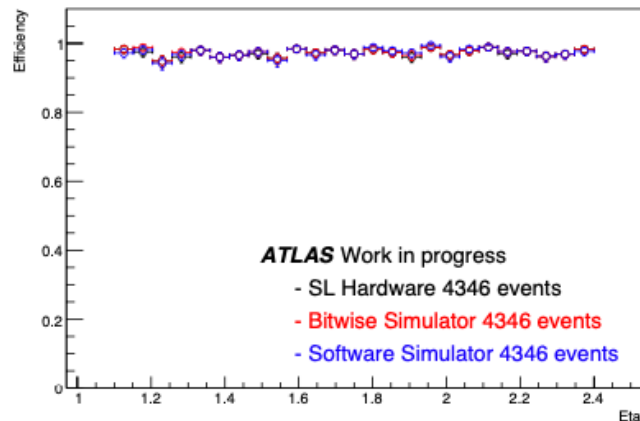
- Bitwise Simulator
 - トリガー論理回路をbitレベルでトレースしたソフトウェア
- SL実機試験
 - SL実機を使ったテストベンチ
 - 本番環境に近い性能評価が可能
- Software Simulator
 - トリガーアルゴリズムを実装したソフトウェア
 - トリガー論理回路に先行して開発された
 - **理想性能の指標**となる

修正したトリガー論理回路の性能が理想性能に到達しているか、検証する



理想性能との比較 (Segment Reconstruction)

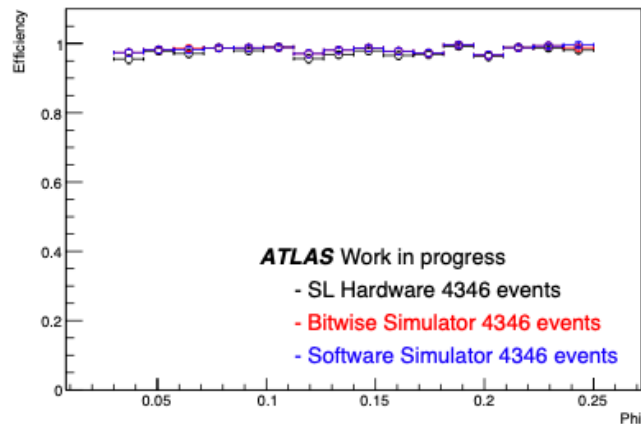
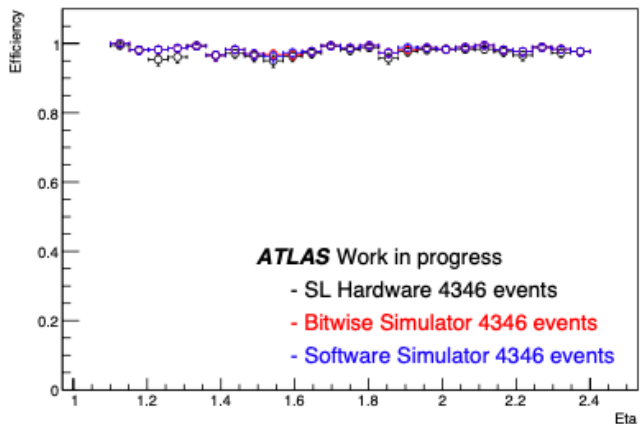
Wire Segment Reconstruction



$$\text{Efficiency} = \frac{\text{SegmentReconstruction において直線飛跡を再構成したイベント数}}{\text{入力したイベント数}}$$

※segment reconstruction:
ヒットパターンから直線飛跡を再構成し、
その角度情報($d\theta, d\phi$)を出力する回路

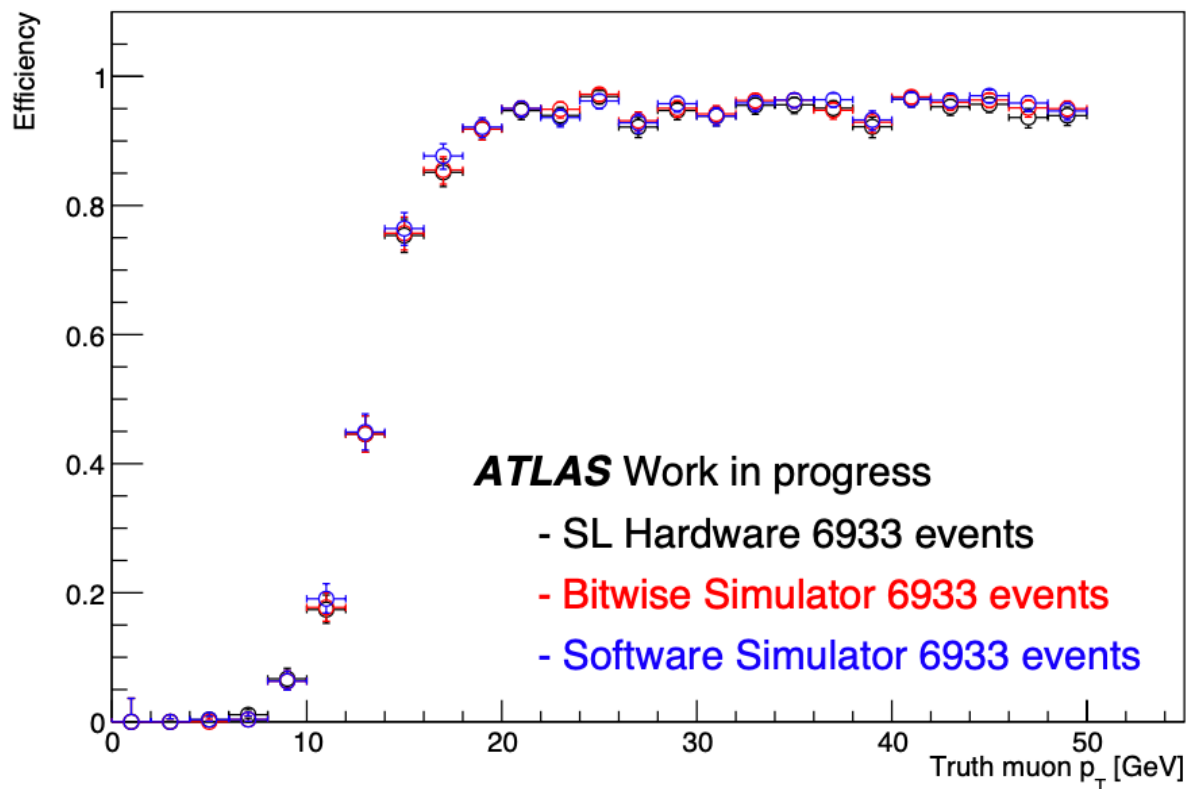
Strip Segment Reconstruction



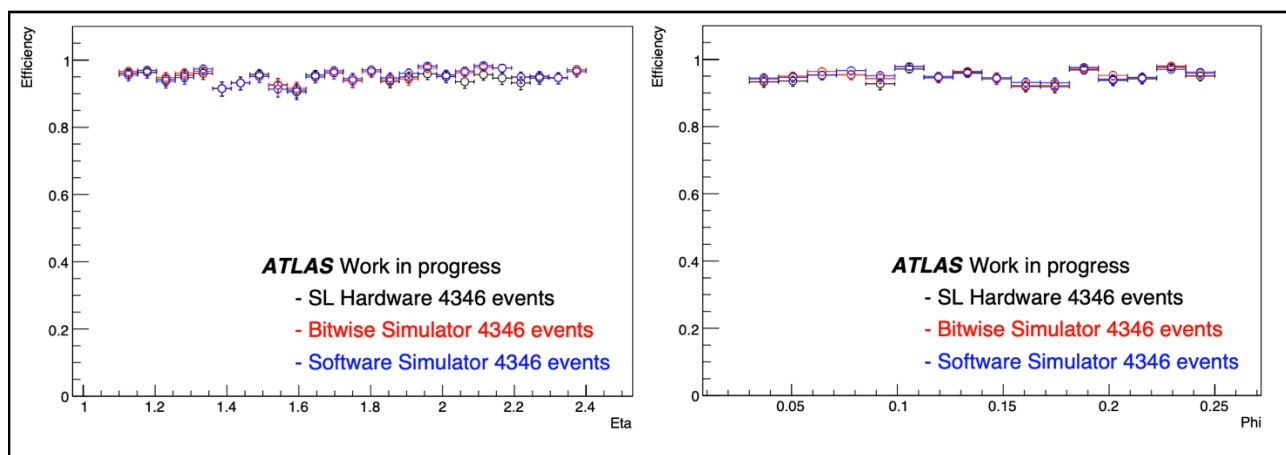
同程度のEfficiencyまで回復している
のが確認できた

プラトー領域 ($p_T > 20$ GeV) の η 分布、 ϕ 分布

理想性能との比較 (Wire Strip Coincidence)



$$\text{Efficiency} = \frac{p_T \text{ threshold} = 20 \text{ GeV と判定されたイベント数}}{\text{入力したイベント数}}$$



プラトー領域 ($p_T > 20 \text{ GeV}$) での η, ϕ 分布

※Wire Strip Coincidence: 角度情報 ($d\theta, d\phi$) から横方向運動量 p_T 等を出力する回路

Efficiency が Software Simulator (理想性能) と同程度となった

まとめと展望

本研究のまとめ

- トリガー論理回路をbit単位で検証する精密なシステムを構築
 - 本検証システムを使って大統計でなければ発見できなかった小さな不具合を精密に特定・修正
 - トリガー性能評価を行いトリガー論理回路が理想性能に比肩する性能まで回復
- 論理回路とソフトウェアが整合する状態を実現→今後の強固な開発基盤を確立した

今後の展望

- 残った不具合箇所の特定制と修正→シミュレータと論理回路の完全な一致
 - Inner Coincidence・Track Selectorを含む完全統合回路に対応した検証システムに拡張
- 2029年の高輝度LHC-ATLAS実験に向けてSLトリガー回路の開発・検証を進めていく

BACK UP

SLのハードウェア概要

Virtex Ultrascale+ (SL FPGA)

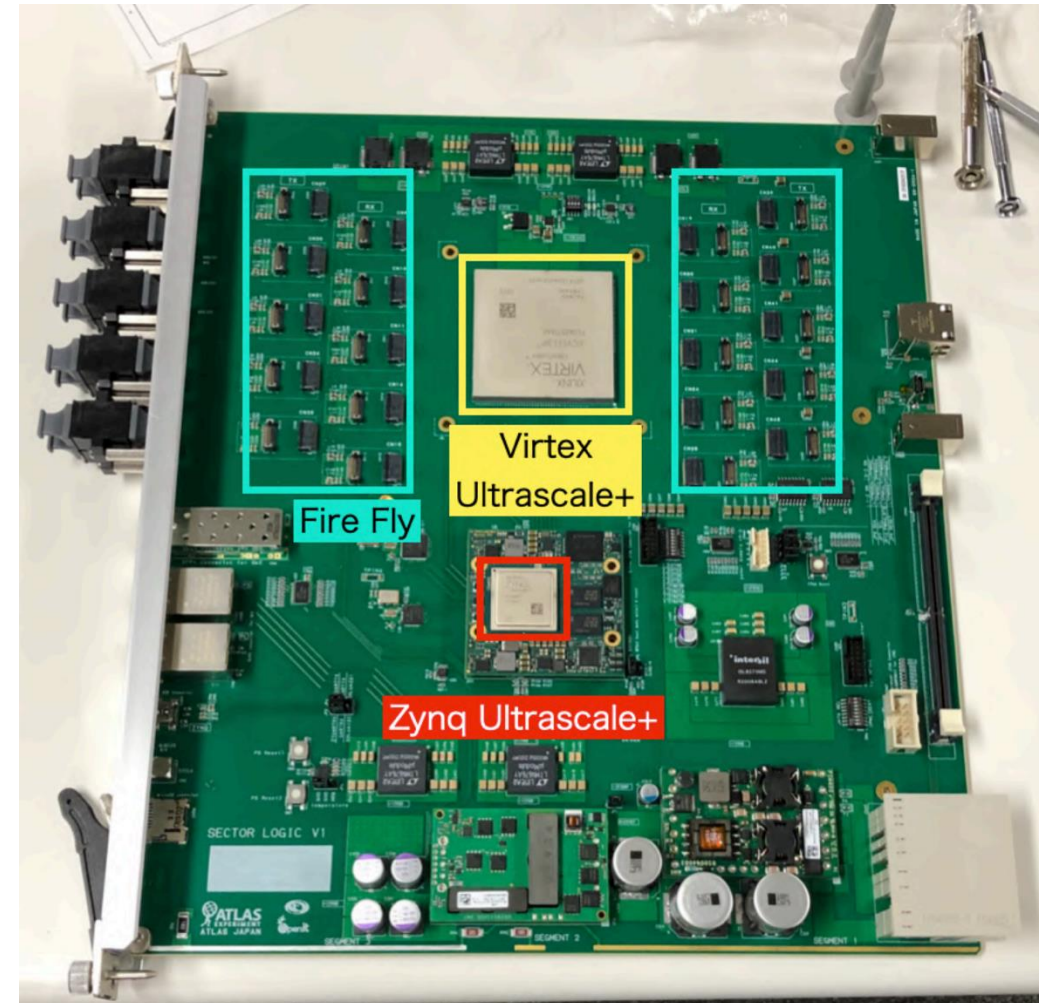
- 4つのシリコンダイ(SLR)から構成される大規模FPGA
- 読み出し回路・トリガー論理回路が実装

Zynq Ultrascale+ (MPSoC)

- プロセッサ領域とFPGA領域をもつSoC
- SL FPGAをコントロール・データ読み出し

FireFly

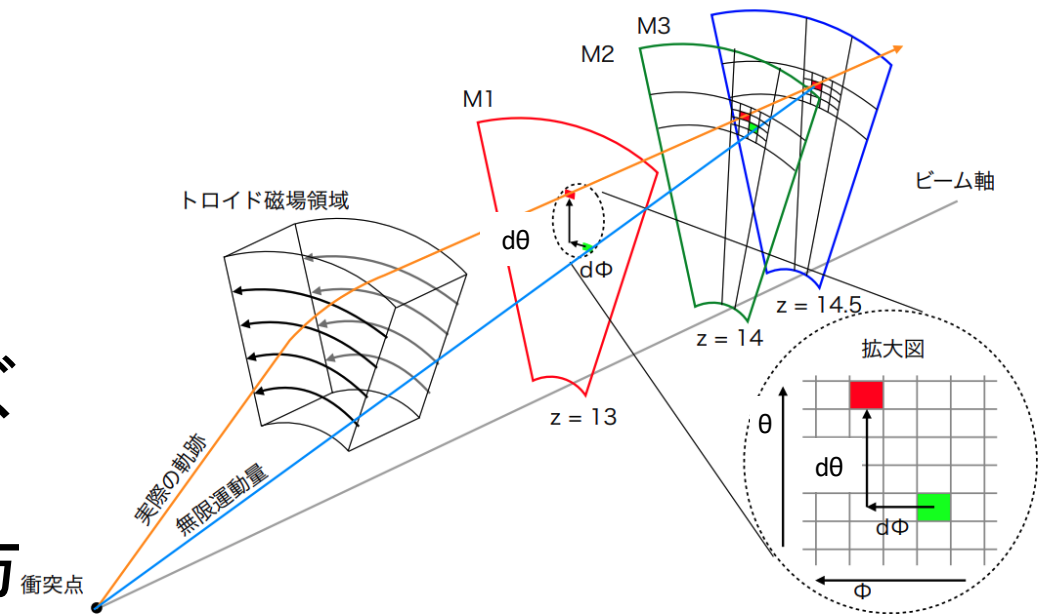
- 光リンクモジュール
- FPGAと各光リンクのインターフェース



TGC BWにおけるミュオン横方向運動量の概算

横運動量概算のアイデア

1. M1/2/3の各ステーションから2or3層コインシデンスで代表点を選ぶ
2. 代表点3つから飛跡を再構成する
3. ビーム衝突点とM3代表点を直線で結んだ無限運動量飛跡を考える
4. 再構成した飛跡と無限運動量飛跡のズレ($d\theta, d\phi$)を考える
5. トロイド磁場によって飛跡は θ 方向と ϕ 方向に曲がるので($d\theta, d\phi$)と横運動量 p_T は相関を持つ
6. ($d\theta, d\phi$)から p_T を求める。



※ビーム軸をz方向として
 θ は天頂角、 ϕ は包囲角

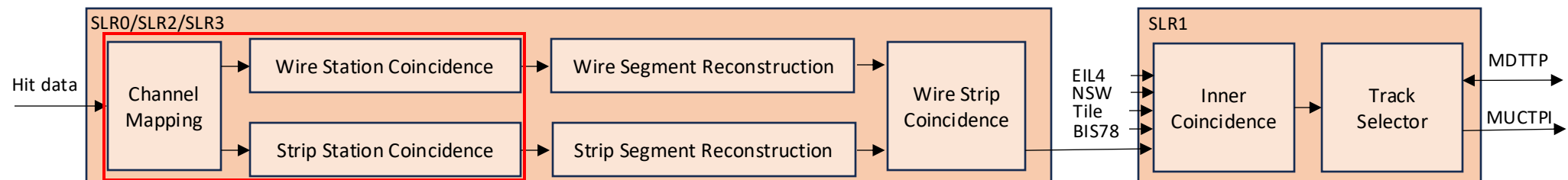
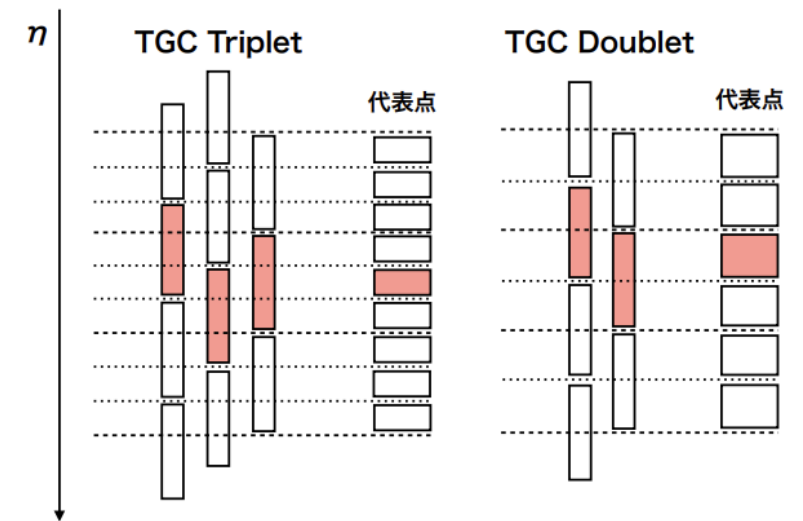
トリガー論理回路 - 1,2段階 / 6段階 -

- Channel Mapping

- hit dataを並べ替えトリガー論理回路に適した形に成形する
- Chamberのオーバーラップ部分などにORの処理を施す

- Station Coincidence

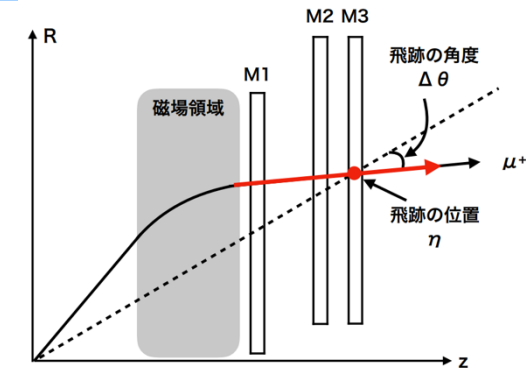
- Station内の2または3層のchannelでコインシデンスをとって代表点を決定する
- 代表点番号とコインシデンスパターン(3/3,2/3,1/3,2/2,1/2)を後段に送る



トリガー論理回路 - 3,4段階 / 6段階 -

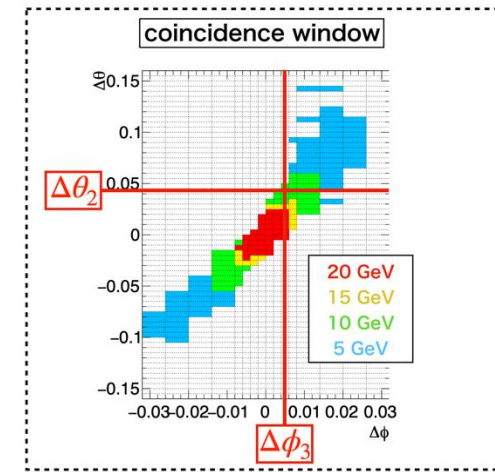
- Segment Reconstruction

- M1/2/3の代表点の組から($d\theta, d\phi$)を出力する*
- 発火層数が多い組み合わせを優先的に選ぶ
- ($d\theta, d\phi$)とM3代表点番号と発火層数を後段に送る

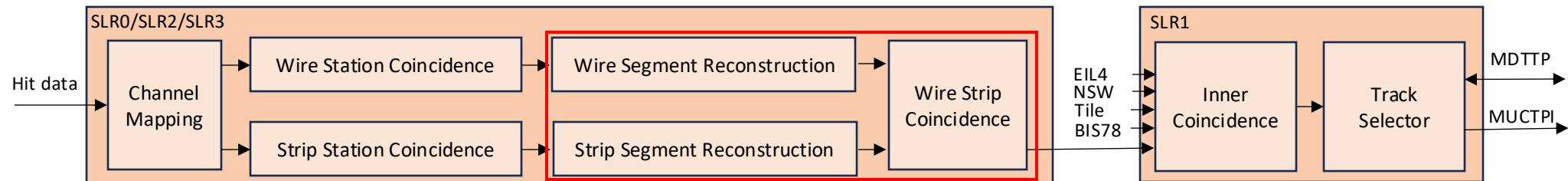


- Wire Strip Coincidence

- ($d\theta, d\phi$)から p_T thresholdを出力する*
- M3代表点番号から座標情報を出力する*
- 発火層数が多い飛跡を選ぶ

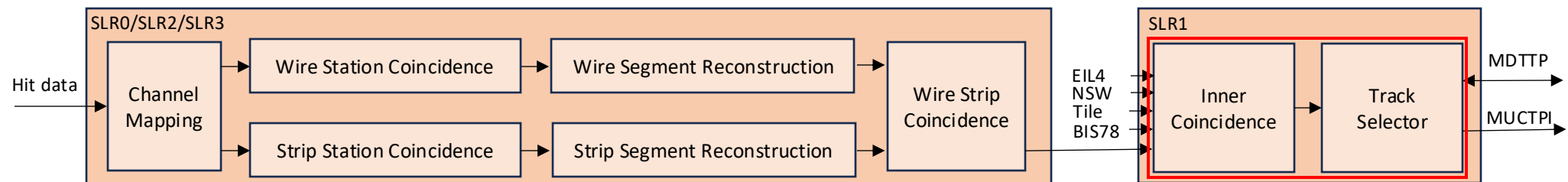
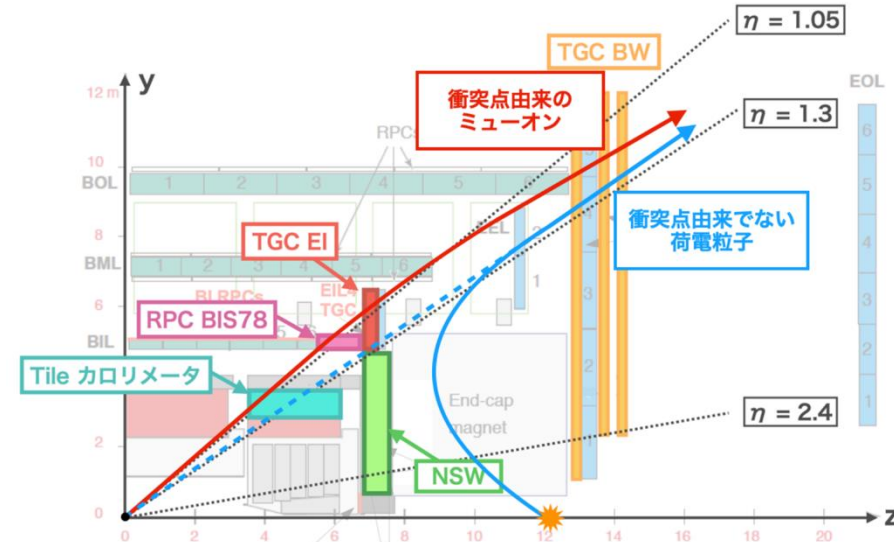


※LUTを利用して対応関係を求める。



トリガー論理回路 - 5,6段階 / 6段階 -

- Inner Coincidence
 - 磁場内部の検出器とコインシデンスをとって
 - フェイクミュオンを除外する
 - 運動量測定を精度を上げる
- Track Selector
 - 飛跡候補を p_T が大きい順に並び替える
 - 上位3つにMDTの高精度運動量情報を付加する
 - 上位6つをMUCTPIに送る



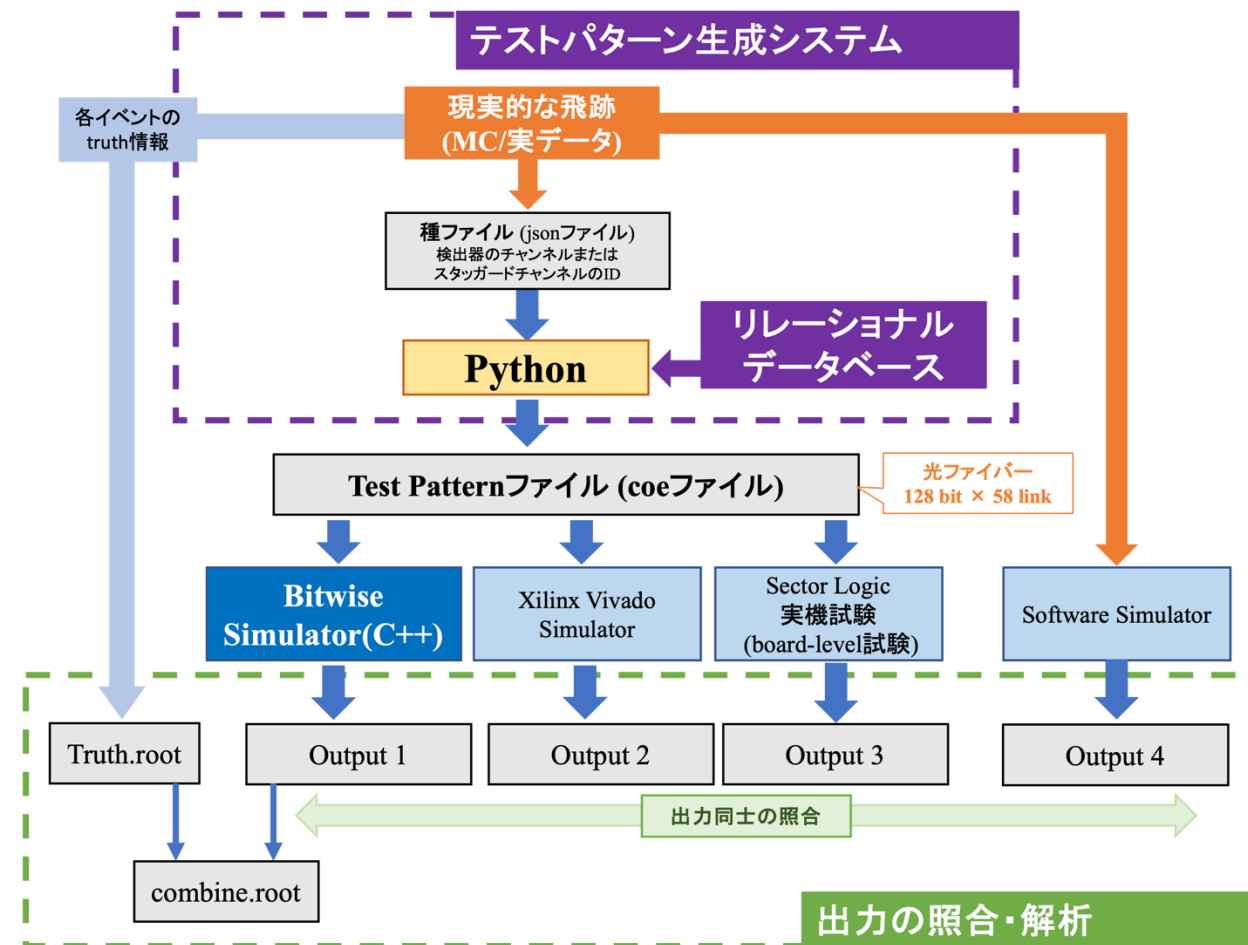
検証システムの全体像 1/2

• テストパターン生成システム

- ATLAS実験でのシミュレーションデータを入力
- トリガー論理回路に適した形式でのテストパターンを生成
- 全システムに共通の入力が可能でソフトウェア・ファームウェアの間でイベント毎の詳細な比較が可能

• Vivado Simulator

- XilinxのVivado Simulationで実装
- トリガー論理回路ファームウェアの動作をシミュレート
- **任意の**信号線のプローブが可能



検証システムの全体像 2/2

• Bitwise Simulator

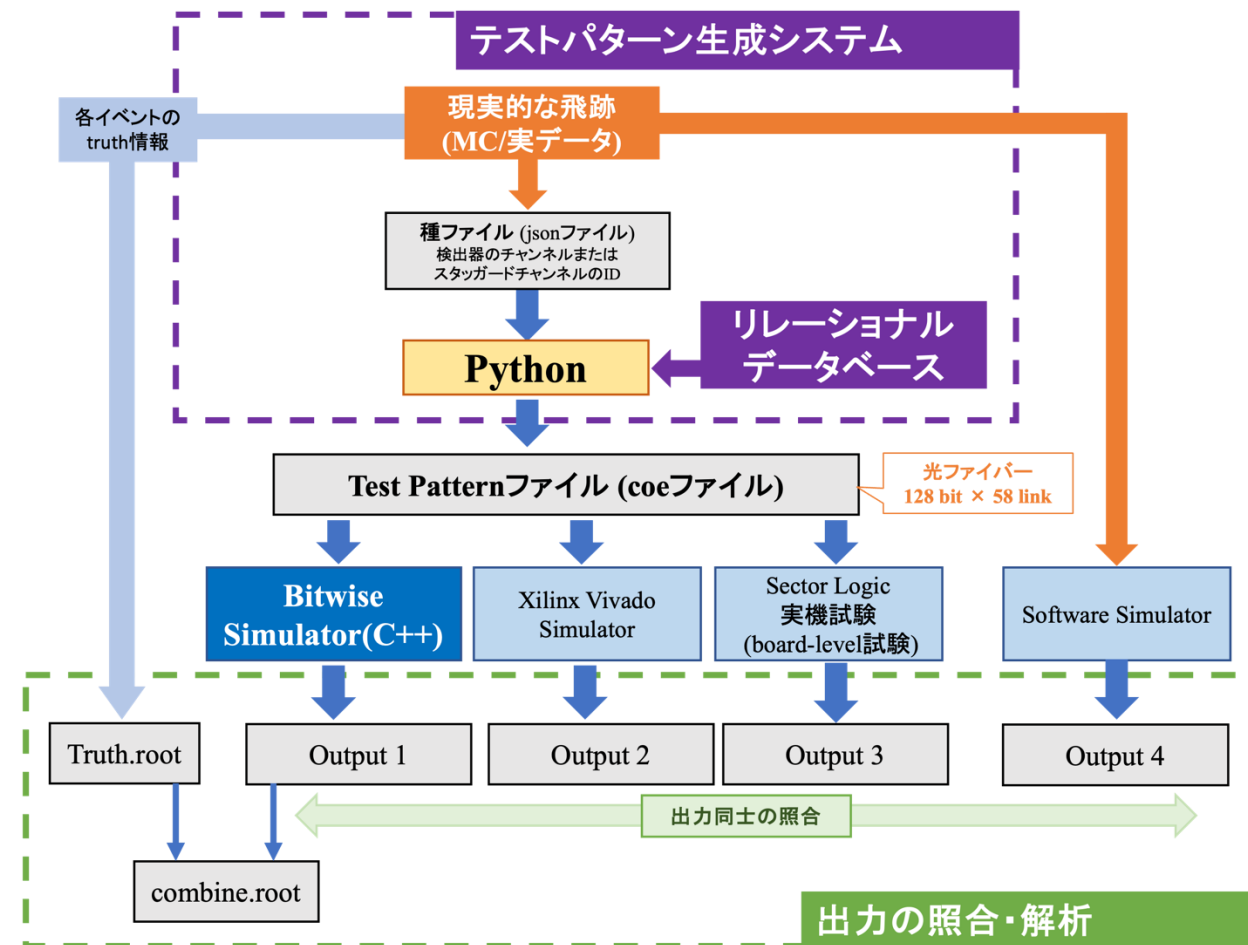
- トリガー論理回路をbitレベルでトレースしたC++プログラム
- **任意の**信号線のプローブが可能
- Vivado Simulatorとのbit単位での比較

• SL実機試験

- SL実機単体で試験可能
- テストパターンを入力
- トリガー論理回路の出力を読み出し

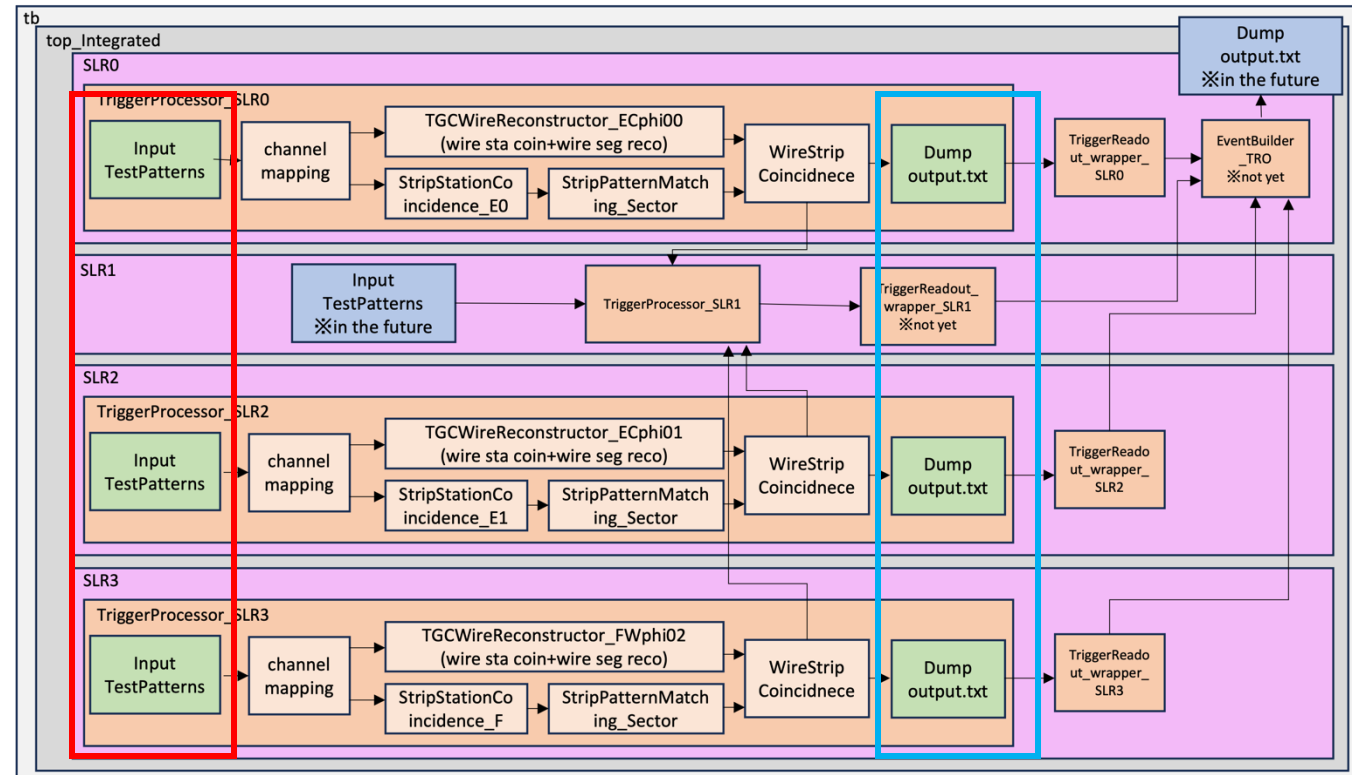
• Software Simulator

- トリガーアルゴリズムを実装したプログラム
- アルゴリズム自体の性能を示す
- トリガー論理回路に先行して開発された
- 理想性能として参照



Vivado Simulatorの開発と実装

- トリガー論理回路全体をテストベンチに含む
- Bitwise Simulatorと共通の大統計テストパターン
 - テストパターンのcoeファイルを1行ずつ読んで40 MHz毎に対応するテストパルスを入力
 - トリガー論理回路シミュレートに必要な機能のみを実装したことで高速化。~30 min/10,000 events
- Wire Strip Coincidenceの出力直後で読み出し
 - テストパターン入力の固定時間後に読み出し
 - 読み出し位置は任意に選択できて拡張性が高い実装
- 出力はBitwise Simulatorと同じ形式



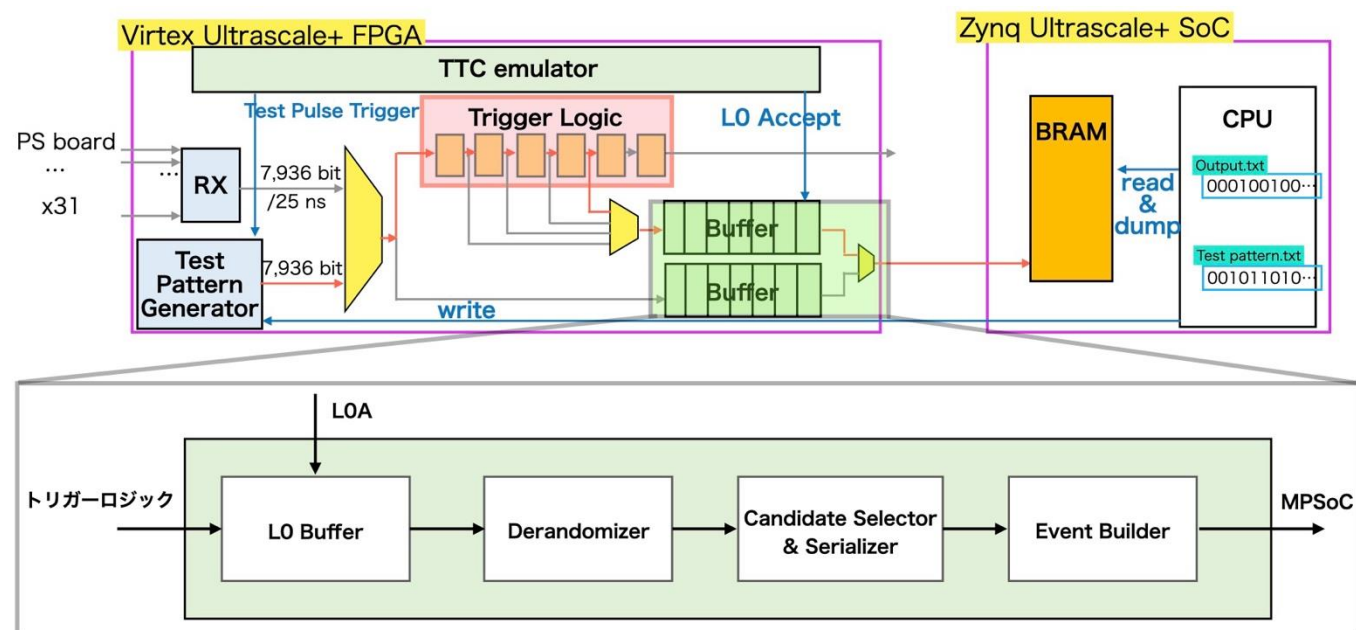
テストパターン入力

出力の読み出し

SL実機試験の実装

SL実機試験

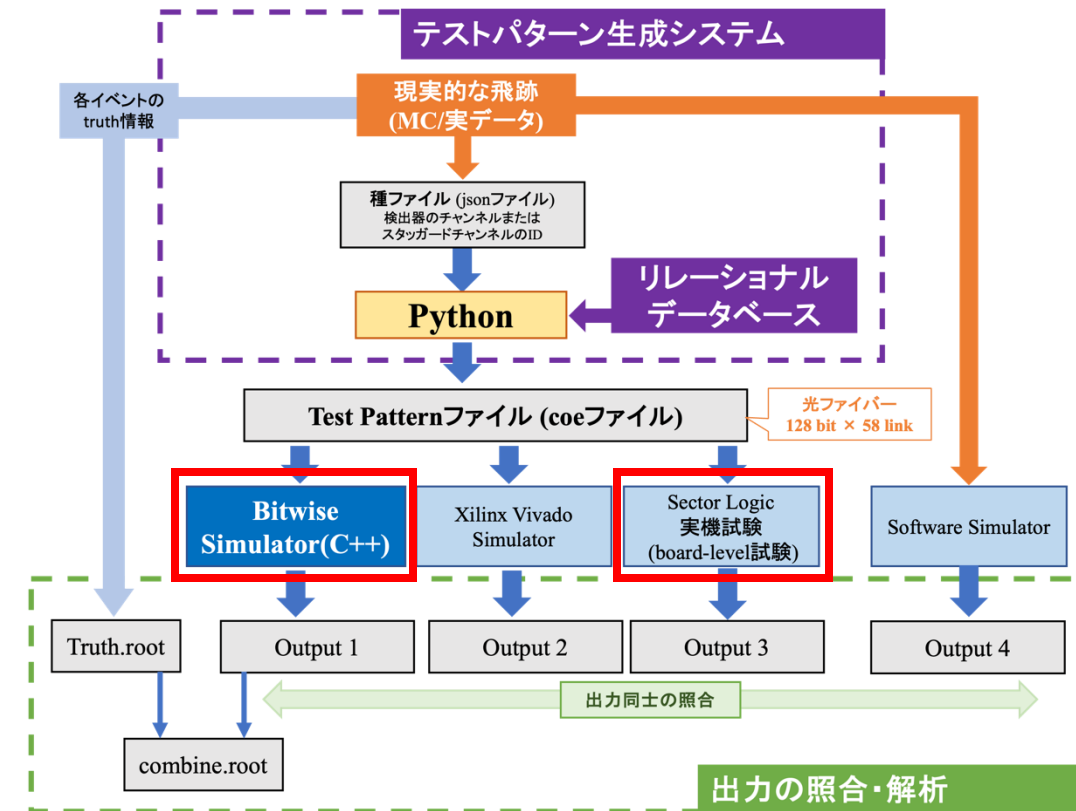
- Zynq Ultrascale+からVirtex Ultrascale+のTPGeneratorにテストパターンを入力
- TPGeneratorからテストパターンがトリガー論理回路に入力される
- トリガー論理回路の出力は読み出し回路を経てZynq Ultrascale+のBRAMに送る
- BRAMのデータを読んでテキストファイルとして出力する



検証① Bitwise Simulator vs SL 実機試験の比較

- Bitwise Simulator
 - トリガー論理回路をbitレベルでトレースしたソフトウェア
 - 任意の信号線がプローブ可能
- SL実機試験
 - トリガー論理回路ファームウェアをSL実機に実装した本番環境に近いテストベンチ
 - プローブ可能なのは**一部**の信号線のみ

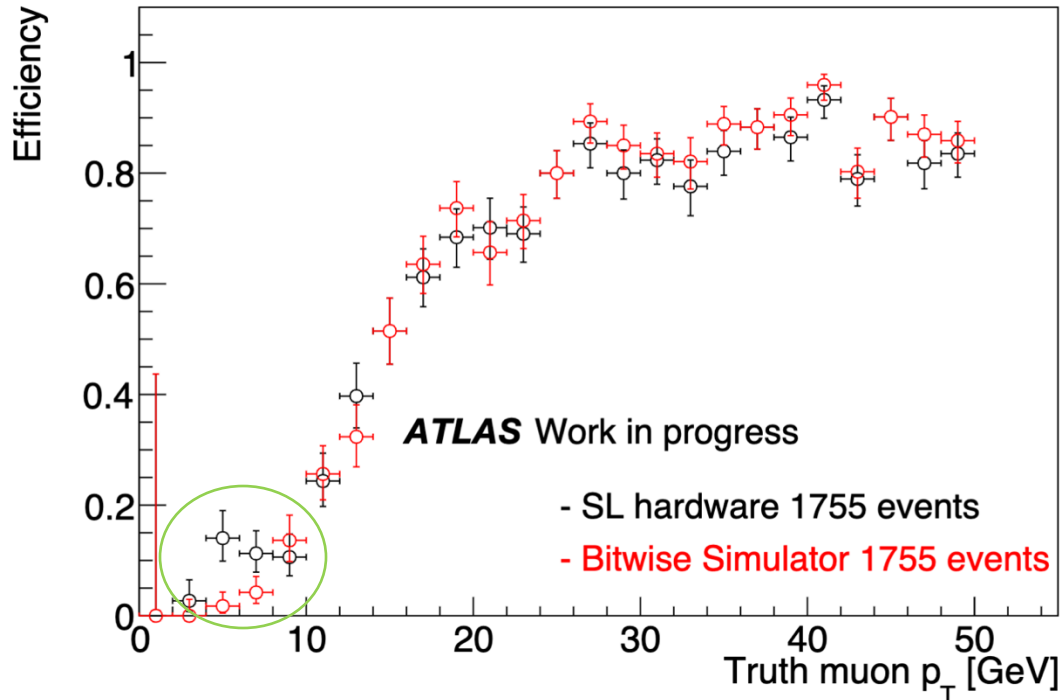
→ファームウェアとソフトウェアの比較により
トリガー効率のinefficiencyを検知



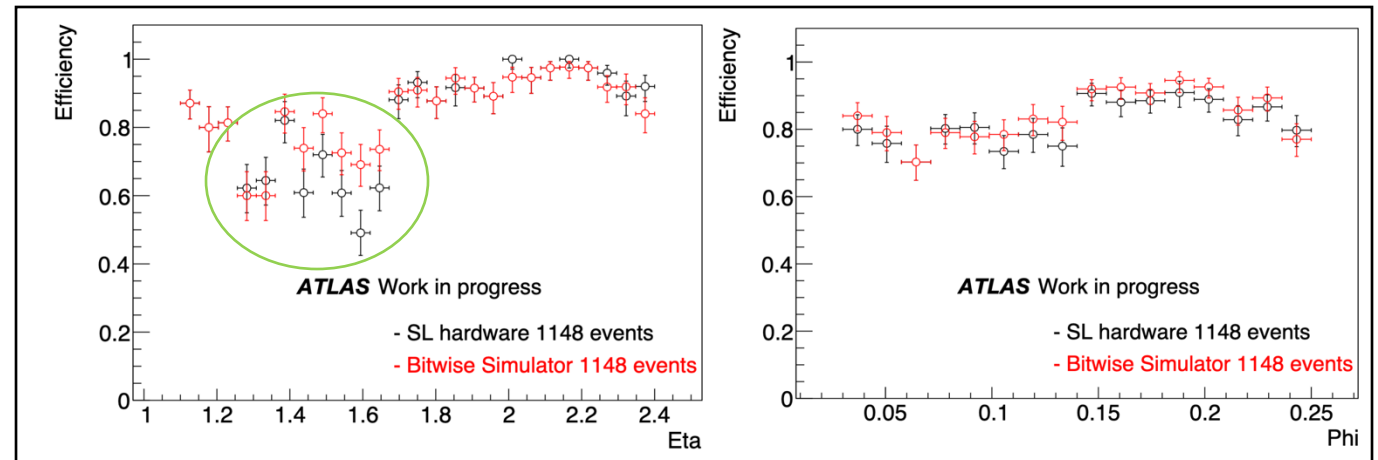
検証① Bitwise Simulator vs SL 実機試験の比較

Bitwise Simulator と SL 実機試験 のトリガー性能 (修正前)

※入カイベントの詳細は後述:



$$\text{Efficiency} = \frac{p_T \text{ threshold} = 20 \text{ GeV と判定されたイベント数}}{\text{入力したイベント数}}$$

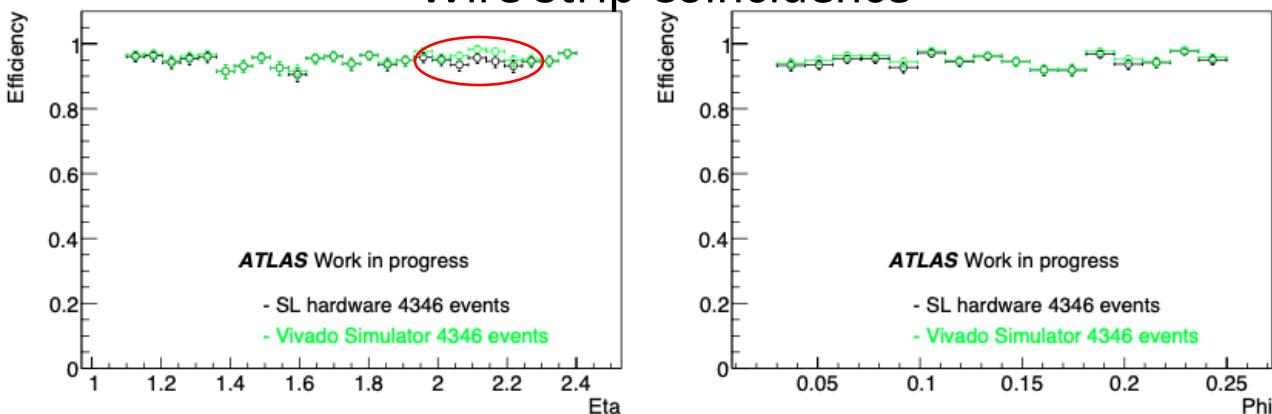


プラトー領域 ($p_T > 20 \text{ GeV}$) での η, ϕ 分布

全体的に efficiency が低い、局所的な inefficiency が複数箇所ある (緑丸)
 → 不具合箇所の特定のため、より精細な信号線のプローブが必要

SL実機試験とVivado Simulatorの性能比較

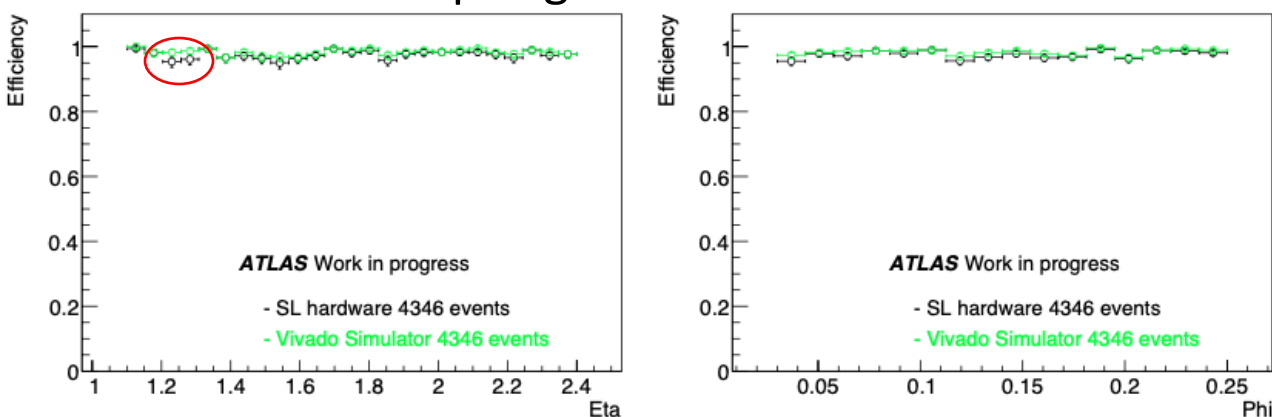
Wire Strip Coincidence



Vivado SimulatorとSL実機試験は
同じファームウェアを使用

前述の理想性能との比較で、
SL実機試験でのみ現れていたInefficiencyは
SL実機特有の不具合と考えられる

Strip Segment Reconstruction



Vivado SimulatorとSL実機試験の比較で
現れる差異と同じところである

プラト一領域での η 、 ϕ 分布

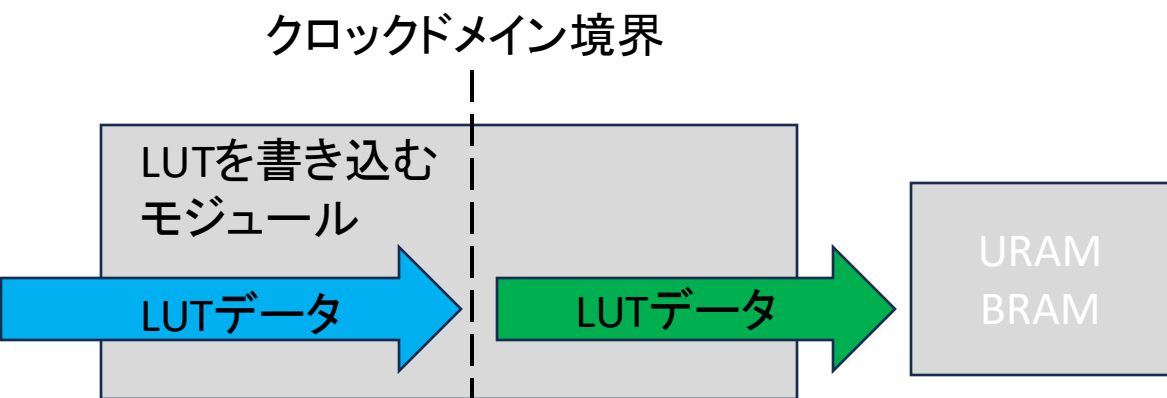
SL実機試験での不具合の検知

SL実機試験で検知した不具合（実機は信号線のプローブが非効率で未特定）

- SL実機試験とVivado Simulatorの出力に数%差異がある
- SL実機試験の試験アプリのrun by runで出力がイベント数で約0.1%だけ変化する

検証すべき項目1

LUT書き込みパスでクロックドメインの境界がある
クロックドメインクロッシングで不適切な処理の可能性



この箇所CDC修正済み→今後検証する

検証すべき項目2

Vivado SimulatorとSL実機試験は同一のファームウェア
ファームウェアの差異は一部のモジュールのみ

Vivado Simulator

トリガーロジック

SL実機試験

トリガーロジック

読み出し回路

→Vivado Simulatorに読み出し回路を含めて検証