

2023 年度 修士論文

高輝度 LHC-ATLAS 実験に向けた L0 ミューオントリガー大規模
エレクトロニクスの統合試験とコミッショニングシステムの開発
-クロック・制御・読み出し系の開発と統合-

東京大学理学系研究科物理学専攻
素粒子物理国際研究センター
奥村研究室

博士課程（前期課程）2 年
学籍番号 35-226058

長坂 錬

2024 年 2 月 25 日

概要

高速通信によるデータ送受信の技術などのエレクトロニクスの進歩により、近年の加速器実験は大型化・高度化の流れにある。エレクトロニクスシステムが大型かつ高度になってくると、オペレーションモデルの並列化・自動化は必須である。また、回路が物理的に直接アクセスできない場所や放射線環境下にあったりする場合は、エレクトロニクスのモニター・制御の遠隔化や、放射線損傷に対する回復手続きなどが必要である。本研究では、このような大規模エレクトロニクスシステムである、高輝度 LHC-ATLAS 実験における TGC 検出器エレクトロニクスシステムにおいて、

1. 完全に遠隔でのモニター・制御・デバッグが可能なテストベンチのインフラストラクチャーの確立
2. LHC バンチ交差クロック (40.079 MHz) に対して固定位相でのタイミング信号分配・受信・位相合わせの実装と、再現性や系統誤差も含めた検証及び、最終的な不確かさの評価
3. 運用へ向けた読み出し回路及び、包括的なテストシステムの実装と検証
4. 不揮発性メモリを用いた自律型ステートマシンの評価と読み出し回路におけるデータロスの評価

といった、新たなシステムの開発、実装、検証及び運用レベルへの洗練を実現した。ここで実装された個々の技術や、システム、検証結果は、ATLAS 実験などの高エネルギー加速器素粒子実験にとどまらず、様々な大規模エレクトロニクスへの応用が可能なものである。

標準模型の精密測定や、新物理探索を目的として、瞬間最高ルミノシティを $7.5 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ まで増強した高輝度 LHC-ATLAS 実験が 2029 年から始まる。ATLAS 検出器のエンドキャップ領域に位置し、ミューオンの運動量・飛跡再構成を行う Thin Gap Chamber 検出器も、LHC の高輝度化に対応して、一部を除き全てのエレクトロニクスを刷新し、トリガーレートを従来の 10 倍の 1 MHz へ拡張する。具体的には、高輝度 LHC-ATLAS 実験における TGC 検出器エレクトロニクスでは、検出器からのヒットビットマップを、ヒットの有無に関わらず、前段回路から全て後段回路に転送する仕様(トリガーレスでの 40 MHz でのデータ転送)となり、1 MHz のトリガーレートが可能になるとともに、後段回路でより柔軟かつ総合的なトリガー判定を行うことが可能になる。前段回路には FPGA を導入し、前段回路は後段回路と高速シリアル光通信でデータ転送を行う。

前段回路では、各ヒット信号に対して、どのバンチ交差に対応するヒット信号かを同定する BCID(Bunch Crossing Identification) を行う。正しく BCID を行うためには、全 1434 台の前段回路におけるクロックの位相が十分な精度 (< 1 ns) で一致している必要がある。本研究では、全 1434 台の前段回路の位相を合わせる機構を確立した。まず、前段回路で、固定位相で送受信を行うための技術を、安定かつ再現性を持って使用できる段階へと洗練させた。また、位相調整の仕組みにおいて、その役割の一部を担う新たなモジュール (TAM) の実装を行なった。実際に KEK に立ち上げたテストベンチを用いて、十分な精度 (~ 50 ps) での位相合わせが達成されていることを確かめるとともに、前段回路が再構成するクロックの位相の再現性や、クロックスキューの詳細な検討も行うことで、最終的な不確かさの評価も行った。本研究の結果は、そのまま 1434 台の前段回路の位相合わせへと拡張できるものである。

TGC 検出器からのヒット信号は ASD ボード、前段回路、後段回路の順で処理され、各前段回路は最大 16 台の ASD ボードからの信号を受け取る。ASD ボードと前段回路の FPGA にはテストパルス発生機構が搭載されており、本研究では、これを用いることで読み出しパスの検証を行なった。ASD ボードに搭載されたテストパルスを用いることで、読み出しパスを包括的に検証することができ、読み出しパスが正しく働くこと及び、各エレクトロニクスの reboot 等に対しても再現性を持つことを確かめた。この際、後段回路に接続された全ての前段回路を、並列かつ自動で、制御・データ取得を行うファームウェア・ソフトウェアを実装することで、ワンクリックでいつでも読み出しパスを包括的に検証することが可能となった。ASD テストパルスを用いた包括的な読み出しパスのテストは、約 5 分で完了するため、コミッショニングやオペレーションの際に、ハードウェアの故障やケーブルの配線ミスを

短時間かつ一挙に検証することができ、価値のあるテストシステムを作成することができたと言える。また、前段回路の FPGA に搭載されたテストパルスを用いることで、1 MHz での読み出しに耐えうる読み出し回路が実装されているかのストレスチェックを行い、前段回路から後段回路を含む読み出し部分が十分な安定性と occupancy に対する耐性を持っていることを確かめた。

前段回路は放射線環境下に設置され、FPGA の SEU に対する回復手続きを行う。前段回路に回復不可能な SEU が発生した場合は、FPGA の再コンフィギュレーションを行うため、一部の読み出し回路にデータロスが生じる。本研究では、テストベンチにて回復不可能な SEU が発生した際のデモンストレーションを行い、実際の高輝度環境下で予想されるデータロスを評価し、それが十分に小さいことを確かめた。

さらに本研究では、遠隔化の観点から、完全に遠隔でのモニター・制御・デバッグが可能なインフラストラクチャーの整備をテストベンチに対して行なった。加えて、大規模エレクトロニクスシステムへの応用を踏まえて、制御の並列化・自動化を行い、1434 台の前段回路を有する大規模エレクトロニクスシステムにも耐えうるような安定性・拡張性を有したシステム、運用方法を確立するに至った。

目次

第 0 章	本研究におけるユニークな点のまとめ	1
第 1 章	序論	3
1.1	素粒子物理学における標準模型	3
1.2	LHC-ATLAS 実験	4
1.3	LHC-ATLAS 実験における TGC 検出器	5
1.4	高輝度 LHC-ATLAS 実験に向けたアップグレード計画	7
1.5	本論文の目的と構成	8
第 2 章	高輝度 LHC-ATLAS 実験での TGC 検出器エレクトロニクス	11
2.1	高輝度 LHC-ATLAS 実験における TDAQ システム	11
2.2	TGC 検出器エレクトロニクス	13
第 3 章	システムレベルの機能説明と実装	25
3.1	ブート・コンフィギュレーションパス	25
3.2	放射線損傷に対するモニターと回復パス	26
3.3	モニター・コントロールパス	29
3.4	固定位相でのタイミング信号分配・受信と位相合わせ	34
3.5	リードアウトパス	40
3.6	テストパルス発生機構	43
第 4 章	Timing Alignment Master (TAM) モジュールの機能実装	45
4.1	コントロールパスの実装	46
4.2	固定位相でのタイミング信号と位相合わせの実装	50
4.3	放射線損傷に対する回復と自律型制御機構の実装	55
4.4	TAM の量産と QAQC に向けて	55
第 5 章	高輝度 LHC-ATLAS 実験に向けた TGC 検出器エレクトロニクスシステムの統合試験	57
5.1	TGC 検出器エレクトロニクスシステムの統合試験の概要	57
5.2	遠隔でのモニター、制御、デバッグが可能なインフラストラクチャーの整備	59
5.3	タイミング信号分配・受信と位相合わせのデモンストレーション	60
5.4	読み出しシステムのデモンストレーション	66
5.5	自律型制御機構の完了時間及び、読み出し回路への影響の評価	69
第 6 章	結論と今後の展望	73

謝辞	75
付録	77
A 固定位相でのタイミング信号分配・受信に必要な実装	77
B 統合試験における全 11 台の ASD テストパルスを用いた delay curve の結果	81
参考文献	115

目次

1.1	LHC 加速器の概要	4
1.2	ATLAS 検出器の全体像	5
1.3	ATLAS 検出器の座標系	5
1.4	ATLAS 検出器のスライス図と衝突点付近で発生した粒子の応答	6
1.5	ミューオン検出器の $R-z$ 平面の断面図	6
1.6	TGC 検出器における 1/12 セクターの構造	6
1.7	TGC チェンバーとステーションの構造	7
1.8	高輝度 LHC 実験のタイムスケジュール	8
1.9	高輝度 LHC-ATLAS 実験の主なターゲットとなる事象における、レプトンの p_T 閾値とアクセプタンスの関係	9
2.1	高輝度 LHC-ATLAS 実験における TDAQ システムの概要図	11
2.2	TGC エレクトロニクスの 1/24 セクターの概要図	13
2.3	フロントエンドエレクトロニクスの設置場所	14
2.4	ASD ボードの概要	15
2.5	PS board の概要	16
2.6	PP ASIC のブロック図	17
2.7	ミューオンが TGC 検出器に入射してから、検出器から信号が出力されるまでの時間分布	18
2.8	PP ASIC 内の陽子バンチ識別回路のタイミングチャート	19
2.9	JATHub の写真	20
2.10	TAM の写真	22
2.11	SL の写真	23
2.12	IPMC の写真	24
3.1	ブート・コンフィギュレーションパスの概要	26
3.2	TGC フロントエンドエレクトロニクスの回復手続きの概要図	27
3.3	PS board における自律型制御機構の概要図	28
3.4	自律型制御機構におけるユーザーパラメータの流れ	28
3.5	モニター・コントロールパスの概要	30
3.6	CPU 側から見たアドレス空間と FPGA のレジスタ空間の対応関係	31
3.7	Up link(PS board が送信側、SL が受信側) における、モニター・コントロールに関するデータ	33
3.8	タイミング信号分配と位相分配の概要図	34
3.9	タイミング信号分配と位相分配の概要図	35
3.10	PS board における GTX 周りのクロックパス	36

3.11	Block RAM(BRAM) の IP design	37
3.12	clocking wizard(MMCM) の IP デザイン	38
3.13	位相測定 の概念図	39
3.14	JATHub における位相測定の結果	40
3.15	読み出しパスの概要図	41
3.16	SL のヒット受信部	42
3.17	SL から FELIX へ送るデータのフォーマット	43
3.18	TTC Emulator 内部のテストパルストリガーの発行の仕方	43
3.19	TPT が発行される時間間隔の分布	44
4.1	TAM とその他のモジュールの接続図	45
4.2	TAM モジュールに実装した固定位相でのタイミング信号と位相合わせのブロック図	50
4.3	TAM が再構成するクロックの位相	51
4.4	TAM の機能実装のデモンストレーション	53
4.5	TAM の位相調整機構のデモンストレーション (位相調節前)	54
4.6	TAM の位相調整機構のデモンストレーション (位相調節後)	54
4.7	自律型制御機構の全体的な手順	55
5.1	TGC 検出器エレクトロニクスシステムの統合試験の概要	58
5.2	KEK テストベンチにおける各モジュールへのアクセスの概要図	60
5.3	PS board の reboot に対する PS board が再構成するクロックの位相の再現性	61
5.4	TAM の reboot に対する PS board が再構成するクロックの位相の再現性	62
5.5	SL の reboot に対する PS board が再構成するクロックの位相の再現性	62
5.6	統合試験におけるクロック分配系及び位相測定系の接続図	63
5.7	11 台の PS board を用いた、位相合わせのデモンストレーション (位相合わせ前)	65
5.8	11 台の PS board を用いた、位相合わせのデモンストレーション (位相合わせ後)	65
5.9	11 台の PS board を用いた、位相合わせのデモンストレーション後の PS board のクロックの位相	66
5.10	delay curve 作成の際の手続き	67
5.11	ASD テストパルスを用いて作成した delay curve	68
5.12	PS board FPGA テストパルスを用いた読み出し回路のストレスチェック	69
5.13	PS board の自律型制御機構の reconfiguration に要する時間	70
5.14	PS board の自律型制御機構の reconfiguration に伴う、SL における光リンク切れ時間の分布	70
5.15	JATHub のリセットによる PS board FPGA reprogramming に伴う、SL における光リンク切れ時間の分布	71
5.16	PS board に搭載されている FPGA における、積分ルミノシティと SEU が起こる回数の関係	71
A.1	GT トランシーバーのデータ受信部分 (RX) のブロック図。	77
A.2	RX CDR におけるクロックの再構成の仕方	78
A.3	GT トランシーバのデータ送信部分 (TX) のブロック図	80

表目次

1.1	標準模型におけるフェルミオンの性質	3
1.2	標準模型におけるボソンの性質	4
2.1	現在行われている LHC-ATLAS 実験 Run 3 と高輝度 LHC-ATLAS 実験の比較	12
3.1	SL と PS board 間のデータフォーマット (Down link)	31
3.2	SL と PS board 間のデータフォーマット (Up link)	32
4.1	GTX のリンク 1 番のデータフォーマット	46
4.2	GTX のリンク 2 番のデータフォーマット	47
4.3	VME プロトコルのタイミングチャート (write cycle)	48
4.4	VME プロトコルのタイミングチャート (read cycle)	49
4.5	TAM の LEMO OUT のポートごとの系統誤差。	52
5.1	各コンポーネントのスキューの評価	63
A.1	HEADER からのクロック再構成手法 (RX packet deformer 内部)	79
A.2	LHC CLK の立ち上がりとデータパッキングの関係 (TX packet former 内部)	81

第 0 章

本研究におけるユニークな点のまとめ

本研究は高輝度 LHC-ATLAS 実験に向けた TGC 検出器エレクトロニクスのアップグレードに関するものである。本研究の題材のシステムレベル試験である、「TGC 検出器エレクトロニクスの統合試験」(第 5 章)では、一部すでに実装されているモジュールを使いながら試験を行ったため、自分がやったことと、これまですでにやられていたことが区別しにくい。よって、最初に自分のやったことのまとめとして、本章を設けた。

TGC 検出器エレクトロニクスには、FPGA もしくは SoC が搭載されたモジュールが、Sector Logic、PS board、JATHub、TAM の 4 種類ある。Sector Logic、JATHub、PS board に関しては、本研究の開始時点で、一部の機能の実装は完了している。自分がやったことは、統合試験を行なっていくにあたり、新たに必要になった機能や、並列化や遠隔化、自動化のために必要な機能の追加実装と、統合試験を通じて長時間かつ大規模なシステムレベルの試験をすることにより運用レベルの安定性や再現性の確保のための実装を行った。詳細は第 3 章で述べる。TAM モジュールに関しては全ての機能実装を自分が行い、その詳細は第 4 章で述べる。

これまでの先行研究と大きく異なる点は、今まではボード単体の試験であったり、短時間の試験で、実装の評価を行ってきたが、本研究で初めて、大規模なエレクトロニクスシステムの運用に向けたシステムの開発と、大規模かつ長時間の検証を行い、その結果、運用に向けた堅牢なオペレーションモデルを開発することができた。具体的には、KEK にシステムレベルのテストベンチである、「統合試験」を自ら立ち上げ、既存のモジュールと新たに実装したモジュールを組み合わせ、制御・モニターパス、コンフィギュレーションパスといったインフラストラクチャーを整備し、それらが安定して動作していることを確かめた。また、LHC バンチ交差クロック (40.079 MHz) に対して固定位相でのタイミング信号分配・受信と位相合わせに関しては、全体のシステム設計から、ファームウェアのデザイン・実装、デモンストレーション、再現性の検証、測定の不確かさの評価までを全て行った。その際に全く新しいモジュールである TAM モジュールを設計、実装し、デモンストレーションを行うことで、今回得られた結果をそのまま、1434 台の PS board に拡張できることを確かめた。また、運用へ向けた読み出し (DAQ) システムの実装と、大規模かつ長時間の読み出しパスの検証もユニークな点と言える。今までは、実際のオペレーションと比べると少ないイベントかつ短時間の試験しか行われていなかったため、読み出しパスの再現性や安定性まで議論が及ばなかった。しかし、本研究で初めて、運用を見据えた読み出しシステム及び、その検証システムを開発することで、大規模かつ長時間の制御、データ取得が遠隔かつ自動で行うことが可能になり、その結果、今までは発見することができなかった再現性や安定性に関する問題点が統合試験を通じて明らかになった。この検証システムは、約 5 分で読み出しパスの検証を包括的に行うことができるため、ファームウェアの検証のみならず、実際のコミコミシングやオペレーションの際にハードウェアの故障やケーブルの配線ミスをチェックする際にも利用できるため、価値のある研究だと言える。この検証システムを用いて、系統的に発見することができた問題点に関しては、修正を行い、その結果運用を見据えた読み出しシステムとして、質の高いものを完成させることができた。

第 1 章

序論

1.1 素粒子物理学における標準模型

素粒子物理学において、物質を構成するクォークとレプトン、及び電磁気力、強い力、弱い力を定式化した理論は素粒子標準模型 (単に標準模型) と呼ばれる。標準模型は、場の量子論で記述され、量子論に特殊相対論を組み込んだ理論体系であり、電磁気力と弱い力を包括的に記述した電弱統一理論と呼ばれる、 $SU(2)_L \times U(1)_Y$ ゲージ理論と、強い力を記述する量子色力学 (QCD: Quantum Chromo Dynamics) と呼ばれる、 $SU(3)_c$ ゲージ理論からなる。表 1.1 に標準模型におけるフェルミオンとその性質、表 1.2 にボソンとその性質を示す。

		名称	質量	スピン
クォーク	第 1 世代	アップ (u)	2.2 MeV	1/2
		ダウン (d)	4.7 MeV	1/2
	第 2 世代	チャーム (c)	1.3 GeV	1/2
		ストレンジ (s)	93 MeV	1/2
	第 3 世代	トップ (t)	173 GeV	1/2
		ボトム (b)	4.2 GeV	1/2
レプトン	第 1 世代	電子ニュートリノ (ν_e)	< 2 eV	1/2
		電子 (e)	511 keV	1/2
	第 2 世代	ミューニュートリノ (ν_μ)	< 0.19 MeV	1/2
		ミューオン (μ)	106 MeV	1/2
	第 3 世代	タウニュートリノ (ν_τ)	< 18.2 MeV	1/2
		タウオン (τ)	1.78 GeV	1/2

表 1.1: 標準模型におけるフェルミオンの性質 [14]。

2012 年に欧州原子核研究機構 (CERN) にある、大型ハドロン衝突型加速器 (LHC) 実験にて標準模型が予言する最後の素粒子、ヒッグス粒子 (表 1.2) が発見された。また標準模型は現在まで、非常に多くの実験と高い精度で一致している。しかし、ヒッグス機構における自発的対称性破れのメカニズムなど、標準模型の検証が不十分であると同時に、暗黒物質やヒッグスボソンの質量に対する階層性問題といった、標準模型では説明のできない現象も観測されている。これらの未解決問題に対する新物理として、超対称性理論やテクニカラーモデルなどが提案されているが、現時点では、それらの新物理の証拠となるような実験的事実は観測されていない。

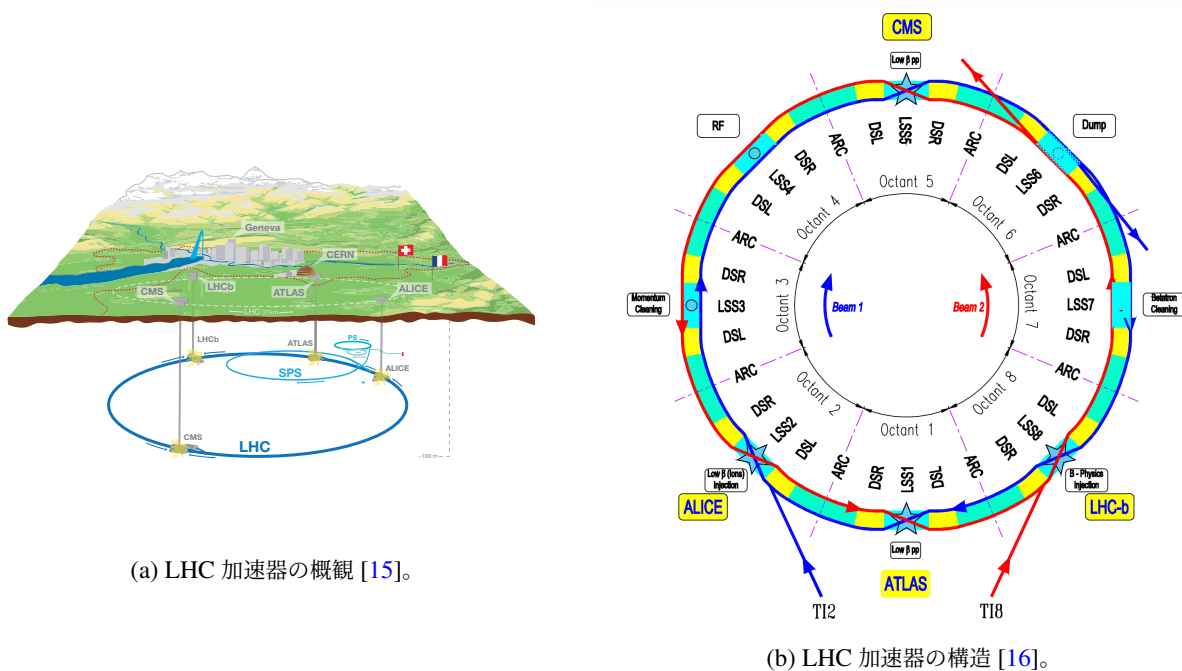
LHC 実験はヒッグスファクトリーと呼ばれるように、効率的にヒッグス粒子を生成することができるため、いまだに検証が進んでいない、標準模型における電弱対称性の部分をヒッグス粒子をプローブとして探索することがで

	名称	質量	スピン
ベクトルボソン	グルーオン (g)	0	1
	W ボソン (W^\pm)	80.4 GeV	1
	Z ボソン (Z^\pm)	91.2 GeV	1
	光子 γ	0	1
スカラーボソン	ヒッグス (h)	125 GeV	0

表 1.2: 標準模型におけるボソンの性質 [14]。

きる。また、LHC 実験は世界のエネルギーフロンティア実験であり、今まで到達することのできなかったエネルギー領域に到達し、一般的に質量が大きいとされている新粒子を直接生成することで新物理探索が可能である。

1.2 LHC-ATLAS 実験



(a) LHC 加速器の概観 [15]。

(b) LHC 加速器の構造 [16]。

図 1.1: LHC 加速器の概要。

LHC-ATLAS 実験は、スイスはジュネーブにある欧州原子核研究機構 (CERN) で行われている、エネルギーフロンティアの衝突型加速器実験である。図 1.1a に Large Hadron Collider (LHC) 加速器の概観を示す。LHC 加速器は、周長 26.7 km の円形加速器であり、図のようにスイスとフランスの国境をまたいで、地下約 100 m の深さに設置されている。

LHC 加速器の構造を図 1.1b に示す。LHC 加速器は、8 つの Arc-section(曲線部分)と 8 つの Straight-section(直線部分)が交互に配置されており、Straight-section の中心部は Point 1 から Point 8 と呼ばれている。そのうち、4 箇所陽子衝突が起こる設計になっており、Point 1 では ATLAS 実験、Point 2 では ALICE 実験、Point 5 では CMS 実験、Point 8 では LHCb 実験が行われている。陽子は約 10^{11} 個が塊になってビームバンチを形成し、個々の陽子が所定のエネルギー(設計値では 7 TeV、Run 3 時には 6.8 TeV)を持って、40.079 MHz の頻度で衝突する。

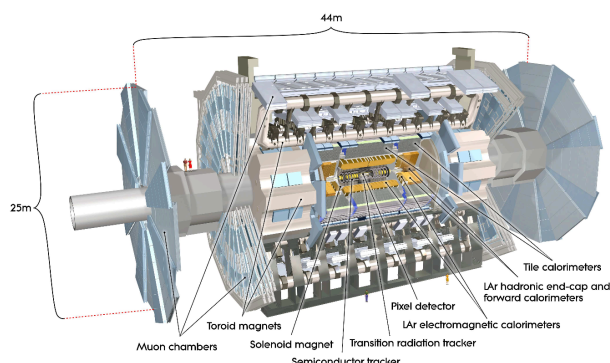


図 1.2: ATLAS 検出器の全体像 [22]。

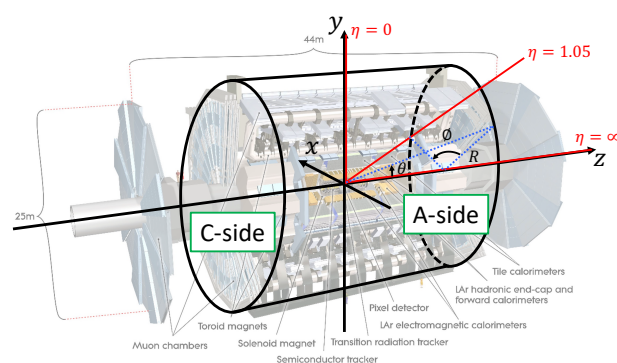


図 1.3: ATLAS 検出器の座標系。

図 1.2 に ATLAS 検出器の全体像を示す。ATLAS(A Toroidal LHC ApparatuS) 検出器は衝突点を囲むようにいくつかの検出器が配置された、大型汎用検出器である。最内層にある内部飛跡検出器は、荷電粒子の飛跡を再構成し、solenoid magnet 磁場中での飛跡の曲率から荷電粒子の電荷と運動量を再構成する。その外側にある電磁カロリメータと、ハドロンカロリメータを用いて、電子や光子、ハドロンを検出し、エネルギーを測定する。最外層にはミューオン検出器が設置されており、内部飛跡検出器とカロリメータを通過したミューオンを検出し、運動量を再構成する。

ATLAS 検出器の座標系を図 1.3 に示す。ATLAS 検出器の座標系は、衝突点を原点に取り、LHC の中心を向いて正とする x 軸、上向きを正として y 軸、ビームに沿って z 軸をとった直交座標系(右手系) (x, y, z) を用いる。 $z > 0$ は A-side、 $z < 0$ は C-side と呼ばれる。また ATLAS 実験では、その検出器の構造から、円筒座標系による記述もよく用いられる。円筒座標系では、ビーム軸からの動径を R 、方位角を ϕ 、天頂角を θ と定義する。また θ の代わりに、擬ラピディティ (Pseudorapidity) η

$$\eta = -\ln\left(\tan\frac{\theta}{2}\right) \quad (1.1)$$

を用いる。 η はその定義から、図 1.3 のように、 $\eta = 0$ は $\theta = \pi/2$ 軸、 $\eta = \infty$ は $\theta = 0$ 軸と一致する。本研究で扱うミューオン検出器は、 $|\eta| < 1.05$ の領域をバレル (Barrel) 部、 $|\eta| > 1.05$ の領域をエンドキャップ (Endcap) 部と呼ぶ。

図 1.4 に、ATLAS 検出器のスライス図と、標準模型の粒子が通過した際に検出器に残す典型的な飛跡を示す。図に示すように、衝突点で生成された標準模型の粒子は、それぞれの検出器に対して異なる応答を示す。ここから、粒子の種類を同定し、ニュートリノを除いて、粒子のエネルギーと運動量を再構成する。しかしニュートリノに関しては、ATLAS 検出器のどの検出器にも飛跡やエネルギーを残さないため、大きな損失横エネルギー (Missing Energy Transverse: MET) として観測される。

1.3 LHC-ATLAS 実験における TGC 検出器

ここでは、本研究で取り扱う TGC 検出器についてより詳しく述べる。ATLAS 検出器の最外層に位置するミューオン検出器は、カロリメータを通過したミューオンを捉え、運動量を再構成する。図 1.5 に高輝度 LHC-ATLAS 実験におけるミューオン検出器の $R-z$ 平面の断面図を示す。TGC 検出器は、 $1.05 < |\eta| < 2.4$ のエンドキャップ領域を担当する検出器であり、Endcap Toroid Magnet の外側にある Big Wheel (BW) と内側にある Endcap Inner (EI) の 2 種類に分かれる。

BW の外側 ($1.05 < |\eta| < 1.92$) はエンドキャップ領域、内側 ($1.92 < |\eta| < 2.4$) はフォワード領域と呼ばれる。

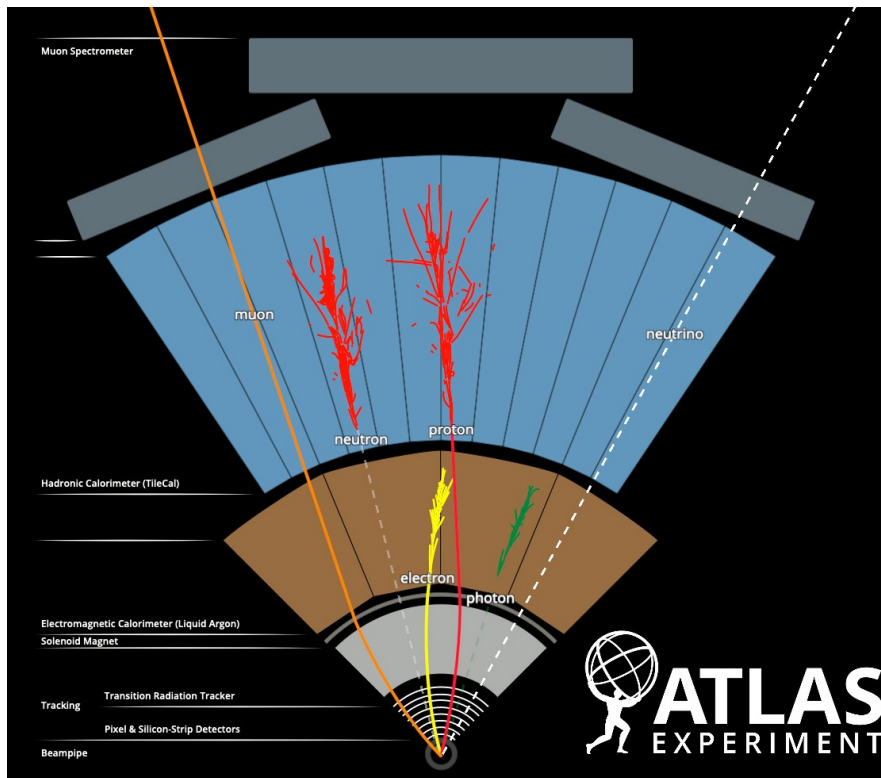


図 1.4: ATLAS 検出器のスライス図と衝突点付近で発生した粒子の応答 [23]。

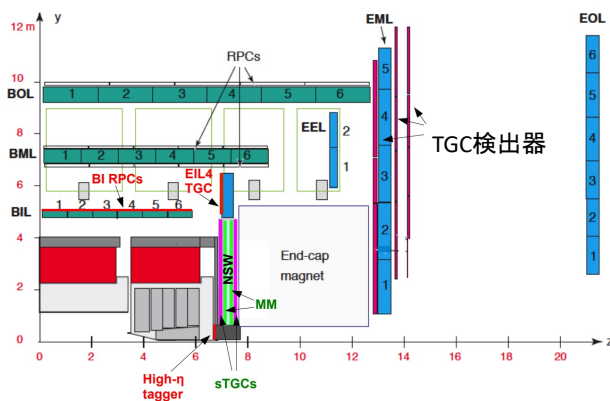


図 1.5: ミューオン検出器の $R-z$ 平面の断面図 [11]。

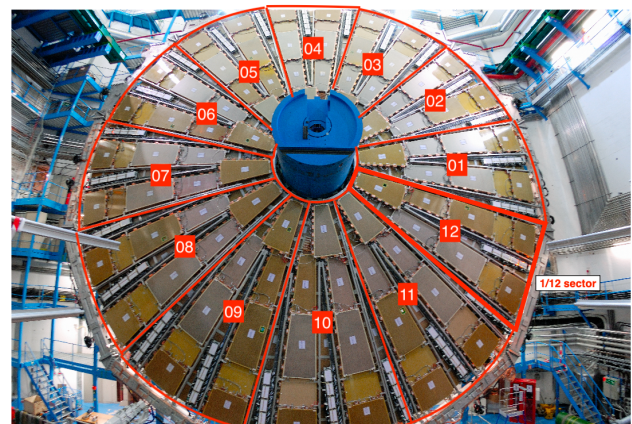


図 1.6: TGC 検出器における 1/12 セクターの構造。

エンドキャップ領域では、 ϕ 方向に 48 回対称になるように、フォワード領域では、24 回対称になるようにチェンバーが設置されている (図 1.6 参照)。エンドキャップ領域の 1/48 とフォワード領域の 1/24 は、独立したトリガー処理を行うため、それぞれがトリガーセクターと呼ばれる。後の 2.2.1 節で述べるように、エンドキャップ領域の 1/48 が 2 つとフォワード領域の 1/24 が 1 つ合わせて、1/24 セクターと呼び、トリガー演算を行う後段回路 1 台で、1/24 セクターを担当する。また、電源供給、ガス供給、電気回路制御、読み出しの観点から、1/24 セクターを 2 つ合わせたものを 1/12 セクターと呼ぶ (図 1.6)。各 1/12 セクターには、 x 軸の正の方向から y 軸の正の方向に向かって順に、A-side では A01 から A12、C-side では C01 から C12 と呼ぶ。さらに、 z 方向に関して、図 1.5 に示すように、TGC BW 検出器は M1、M2、M3 の 3 つのステーションから構成されている。

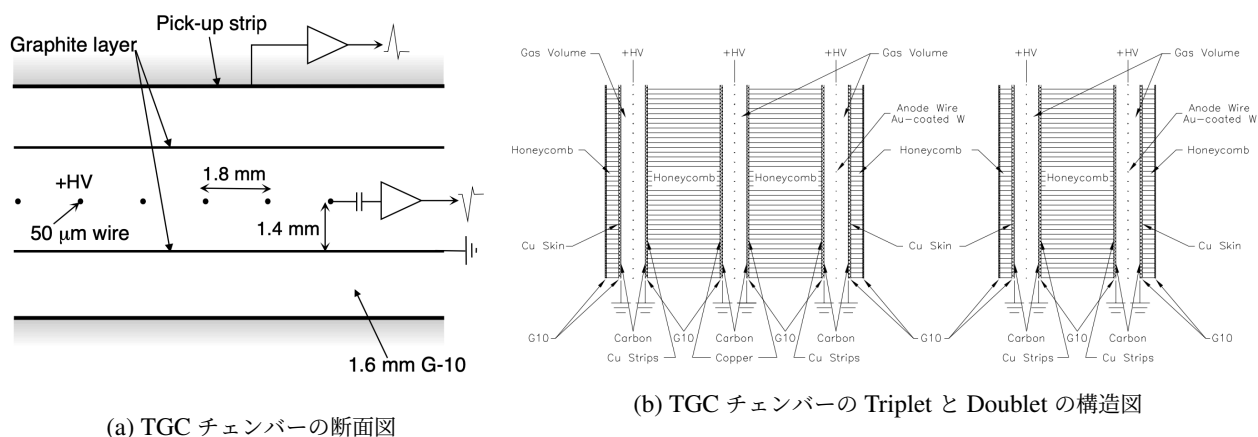


図 1.7: TGC チェンバーとステーションの構造 [22]。

図 1.7a に TGC 検出器のチェンバーの構造図を示す。TGC チェンバーはワイヤーとストリップの 2 次元の読み出しチャンネルをもつ、**Multi-Wire Proportional Chamber (MWPC)** である。ワイヤーとストリップは直行していて、ワイヤーは R 方向の位置測定、ストリップは ϕ 方向の位置測定に対応する。

図 1.7b に TGC チェンバーにおける Triplet と Doublet の構造図を示す。先ほども述べた通り TGC 検出器は M1、M2、M3 の 3 つのステーションから構成されているが、このうち、M1 は Triplet の構造を持ち、M2、M3 は Doublet の構造を持つ。図 1.7b に示すように、Triplet は 3 層のガス層を持ち、3 層のワイヤーと 2 層のストリップがある。一方、Doublet は 2 層のガス層を持ち、2 層のワイヤーと 2 層のストリップを持つ。

TGC チェンバーのガス層を荷電粒子が通過すると、ガス分子の電離をして、エネルギーの一部を失う。ガス分子から電離した電子は、アノード・カソード間に印加された電圧による電場にしがたって、アノードの方向に移動し、アノードワイヤー近傍の強い電場によって電子雪崩を引き起こす。アノードワイヤーでは、電子雪崩の際に発生した正イオンのドリフトによる電流が、カソードストリップではそれらの鏡像電荷による電流が検出される。

TGC 検出器はオンラインでのトリガー発行のための検出器であるため、高い時間分解能が求められる。よって、図 1.7a のようにワイヤーのピッチを短く張り、最近接一次電子のドリフト時間の分布を小さく抑えることで、バンチ交差間隔 25 ns に対して、十分な時間分解能を達成している。一方で、TGC 検出器は、位置分解能は比較的低くても問題はないため、チャンネル数の調整のために、ワイヤー電極を 4~20 本程度まとめてから、読み出しを行っている。その結果、ワイヤーとストリップを合わせて、TGC 検出器の読み出しチャンネルは 32 万チャンネルである。

1.4 高輝度 LHC-ATLAS 実験に向けたアップグレード計画

図 1.8 に高輝度 LHC-ATLAS 実験に向けた LHC のタイムスケジュールを示す。高輝度 LHC 実験は、標準模型の精密測定や新物理探索のために、現在の LHC 実験の高輝度化を目的として、2029 年から運転を開始する予定である。ビームの輝度はこれまでの約 3 倍に向上する予定である。重心衝突エネルギーを 14 TeV、瞬間最高ルミノシティを $5 \sim 7.5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ まで向上し、10 年間の運転で、 $3000 \sim 4000 \text{ fb}^{-1}$ の積分ルミノシティを目指す。

高輝度 LHC-ATLAS 実験が目的とする主な物理事象としては、ヒッグスボソン対生成事象を用いた、ヒッグスポテンシャルの形を決定するヒッグス粒子の自己結合定数の測定や、超対称性粒子の探索などがある。LHC 加速器の高輝度化により、統計量を増やし、これらの物理に対してより精密な測定が可能になる。高輝度 LHC-ATLAS 実験では、ルミノシティの向上に対し、エレクトロニクス改良や刷新により、読み出し・トリガーの高速、高帯域



図 1.8: 高輝度 LHC 実験のタイムスケジュール [24].

化を行うとともに、パイラアップの増加に伴う精度の低下を防ぐために内部飛跡検出器の高精細化を行う。

ルミノシティの向上に統計量が増えると、目的の信号事象も増えるが、同時に背景事象も増える。高輝度 LHC-ATLAS 実験では、検出器やエレクトロニクスのアップグレードを行うことで、背景事象は落として、信号事象は効率的に収集することを目指す。具体的には、内部飛跡検出器が全てシリコン検出器に置き換わったり、本研究で扱う TGC 検出器のように全てのエレクトロニクスの刷新を行う。

図 1.9 に高輝度 LHC-ATLAS 実験の主なターゲットとなる事象における、レプトンの p_T 閾値とアクセプタンスの関係を示す。現行のトリガーシステムのまま、LHC 加速器の高輝度化が行われると、レプトンの横運動量 p_T 閾値を 50 GeV まで上げる必要があり、目的とする物理事象に対するアクセプタンスは低い。よって高輝度 LHC-ATLAS 実験では、トリガーレートの拡張を行うとともに、トリガーレイテンシーを増やすことによって、より高度なトリガーアルゴリズムの導入が可能になる。例えば、2.2.1 節で示すように、初段トリガーレートを従来の 10 倍の 1 MHz へ拡張するとともに、初段トリガーレイテンシーを従来の 4 倍の 10 μ s へ増やす。その結果、 p_T 閾値を 20 GeV 付近まで下げることができる見込みである。このように、LHC 加速器の高輝度化に伴って、ATLAS 検出器及び、エレクトロニクスのアップグレードを行うことで、より興味のある物理事象に対してアクセスが可能になる。

1.5 本論文の目的と構成

本研究は、高輝度 LHC-ATLAS 実験に向けた L0 ミューオントリガーの刷新に関して、大規模なエレクトロニクスシステムのコミショニングやオペレーションシステムの研究を行なったものである。TGC 検出器エレクトロニクスシステムは、非常に多数の回路・エレクトロニクスが複雑に接続され、コミショニングやオペレーションシステムの実装や運用は自明ではないため、本研究を通して洗練されたシステムを作成することは、大変価値が高いと言える。本研究では、複数のエレクトロニクスに渡って横断的に実装された機能を検証しつつ、自動化や並列化

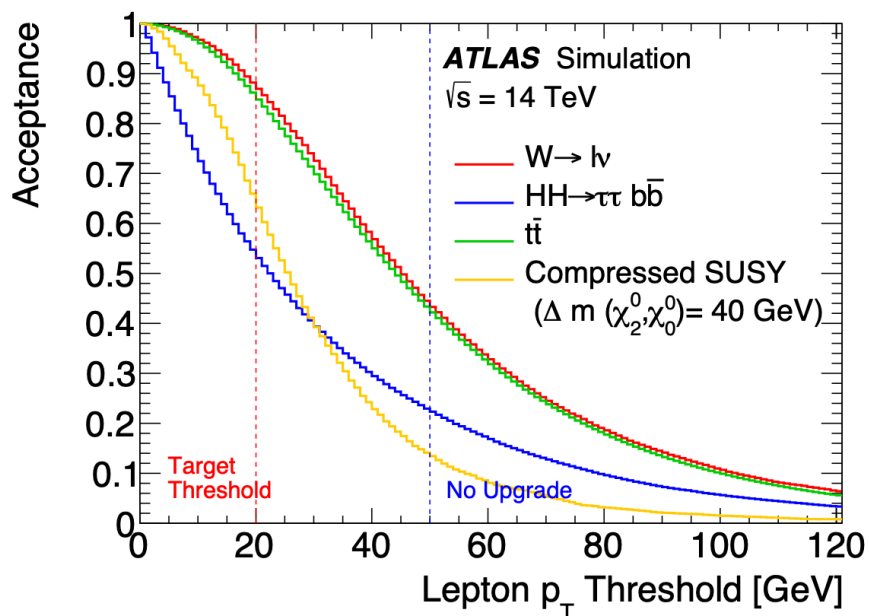


図 1.9: 高輝度 LHC-ATLAS 実験の主なターゲットとなる事象における、レプトンの p_T 閾値とアクセプタンスの関係 [12]。

を行い、洗練することで、再現性や安定性を持つシステムにまで昇華させた。

本研究の構成は以下の通りである。まずは第 2 章で、高輝度 LHC-ATLAS 実験における TGC 検出器エレクトロニクスについて述べる。ここでは各エレクトロニクスの概要を述べる。第 3 章では、TGC 検出器エレクトロニクスのシステムレベルの機能説明と、各モジュールへの実装について述べる。第 4 章では、TAM モジュールの機能とデモンストレーションについて述べる。第 5 章では、TGC 検出器エレクトロニクスシステムの統合試験について述べる。最後に、第 6 章では、本研究の結論と今後の展望について述べる。

第 2 章

高輝度 LHC-ATLAS 実験での TGC 検出器エレクトロニクス

2.1 高輝度 LHC-ATLAS 実験における TDAQ システム

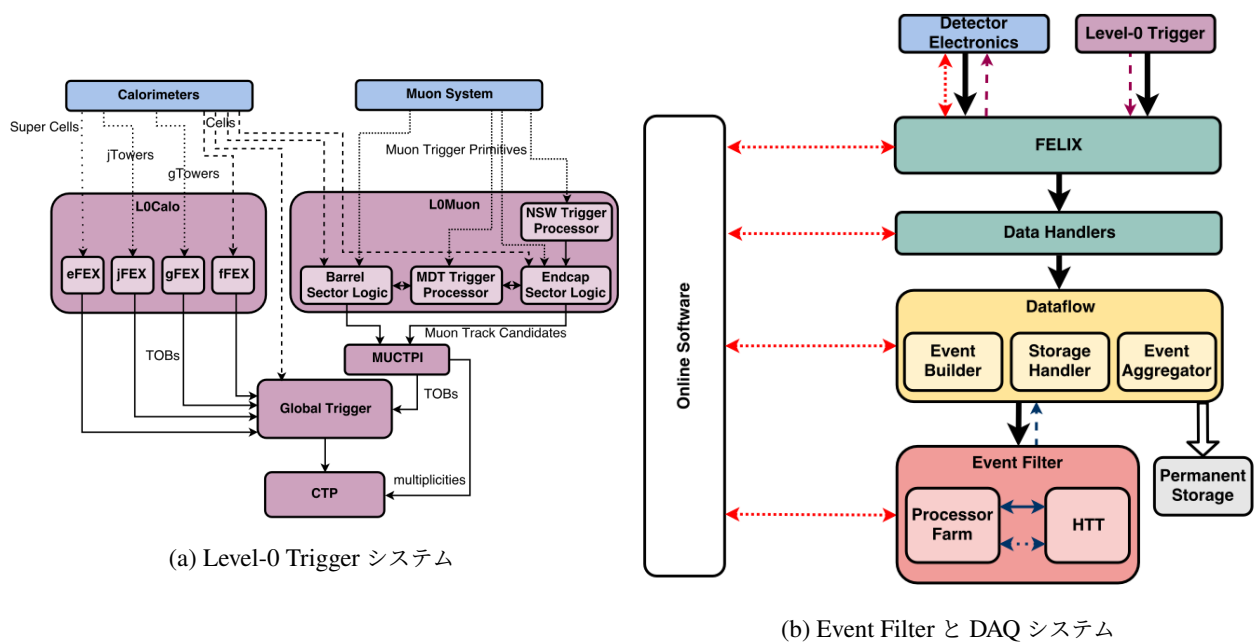


図 2.1: 高輝度 LHC-ATLAS 実験における TDAQ システムの概要図 [12]。図 2.1a に Level-0 Trigger システムの概要を示す。Level-0 Trigger はハードウェアベースの初段トリガーであり、大まかに Level-0 カロリメータシステム (L0Calo) と Level-0 ミューオンシステム (L0Muon) から構成される。最終的なトリガー判定は、Central Trigger Processor (CTP) で行われ、後段に渡すべきデータと判断された場合には Level-0 Accept (LOA) 信号が CTP から発行される。図 2.1b に Event Filter と DAQ システムの概要を示す。CTP から発行された LOA 信号は Local Trigger Interface (LTI)、Front-End Link eXchange (FELIX) を介して、各システムへ分配される。LOA 信号を受けた各システムはバッファ (L0 Buffer) に一時的に保存しておいた該当するヒット情報を読み出し、FELIX に送信する。FELIX は受け取ったデータをさらに後段に渡し、後段トリガーでは Event Filter がソフトウェアベースのトリガー処理を行い、通過したものが CERN の Permanent Storage に保存される。

LHC では、40.079 MHz の頻度で陽子バンチ交差が発生するが、衝突によるすべての信号を記録することはそもそも不可能である。また、陽子バンチ交差のうち、エネルギーフロンティア物理の観点から見ると、ほとんどは興味のない移行運動量が小さい (ハドロンスケール) 反応であるため、興味のある事象とそうではない事象を選別して、興味のある事象のみを記録するオンライン選別システム (トリガーシステム) がハドロンコライダーでは特に重要である。トリガーと読み出しをまとめて、**Trigger and Data Acquisition (TDAQ)** システムと呼ぶ。高輝度 LHC-ATLAS 実験では、トリガー性能の向上を目指して、TDAQ システムを刷新する。高輝度 LHC-ATLAS 実験における TDAQ システムの概要を図 2.1 に示す。高輝度 LHC-ATLAS 実験におけるオンライントリガーは 2 段階で構成される。一つはハードウェアベースのトリガーを行う **Level-0 Trigger** と呼ばれる初段トリガーである。Level-0 Trigger は、カロリメータの情報を入力としてトリガー判定を行う Level-0 カロリメータシステム (LOCalo) と、ミュオンスペクトロメータの情報を入力としてトリガー判定を行う Level-0 ミュオンシステム (LOMuon) から構成される。LOMuon で再構成されたミュオンの運動量等の情報は、**Muon-to-Central Trigger Processor Interface (MUCTPI)** を経由して、**Central Trigger Processor (CTP)** に送られる (図 2.1a)。CTP で最終的なトリガー判定がなされ、興味のあるデータだと判定された場合は、Level-0 Accept (LOA) 信号が CTP によって発行され、**Local Trigger Interface (LTI)**、**Front-End Link eXchange (FELIX)** を介して、各システムへ分配する (図 2.1b)。LOA 信号を受けた各システムはバッファ (LO Buffer) に一時的に保存しておいた、該当するヒットデータの情報を読み出す。読み出したデータを FELIX に送信し、それを受け取った FELIX はさらに後段に渡す。後段トリガーでは、Event Filter がソフトウェアベースのトリガー処理を行い、最終的に、後段トリガーもパスしたデータがストレージに保存される。

	瞬間最高ルミノシティ [$\text{cm}^{-2}\text{s}^{-1}$]	初段トリガーレート [kHz]	初段トリガーレイテンシー [μs]
Run 3	2×10^{34}	100	2.5
高輝度 LHC	$5 \sim 7.5 \times 10^{34}$	1000	10

表 2.1: 現在行われている LHC-ATLAS 実験 Run 3 と高輝度 LHC-ATLAS 実験の比較。瞬間最高ルミノシティは従来の約 3 倍に増強され、それに対応するため及び、今までの LHC-ATLAS 実験で取りきれなかったイベントを取得するために、初段トリガーレートは従来の 10 倍に増強される。また初段トリガーレイテンシーは従来の 4 倍に増強され、より複雑かつ洗練されたトリガーアルゴリズムの導入が可能になる。

表 2.1 に現在行われている LHC-ATLAS 実験 Run 3 及び高輝度 LHC-ATLAS 実験における、瞬間最高ルミノシティ、初段トリガーレート、初段トリガーレイテンシーを示す。瞬間最高ルミノシティは従来の約 3 倍に増強され、それに対応すべく初段トリガーレートは従来の 10 倍に増強される。瞬間最高ルミノシティの 3 倍の増強に対して、初段トリガーレートが従来の 10 倍に増強されているのは、単純にルミノシティが増えた分に加えて、今までの LHC-ATLAS 実験で落としていたイベントも取得しようという挑戦的なトリガーだと言える。また初段トリガーレイテンシーを $2.5 \mu\text{s}$ から $10 \mu\text{s}$ へ伸ばすことにより、複雑かつ洗練されたトリガーアルゴリズムの導入が可能になる。

本研究では、ハードウェアベースのトリガー判定を行う Level-0 Trigger である、Level-0 ミュオンシステムのうちの、エンドキャップ領域に設置された TGC 検出器システムの高輝度 LHC-ATLAS 実験に向けたアップグレードに関するものである。

2.2 TGC 検出器エレクトロニクス

2.2.1 TGC 検出器エレクトロニクスの全体像

高輝度 LHC-ATLAS 実験における、TGC 検出器エレクトロニクスシステムでは、TGC 検出器からの電流信号を ASD ボードが電圧信号へ変換、増幅し、閾値電圧と比較しデジタル信号に変換して、LVDS 規格で PS board に送信する。PS board は ASD ボードから受け取った信号が、どのバンチ交差に対応するヒット信号かを同定する **Bunch Crossing IDentification (BCID)** を行なった後に、SL へ送信し、SL がトリガー判定を行う。この他に、PS board の制御・タイミングキャリブレーションのためにデータパスとは独立した、JATHub と TAM が導入される。

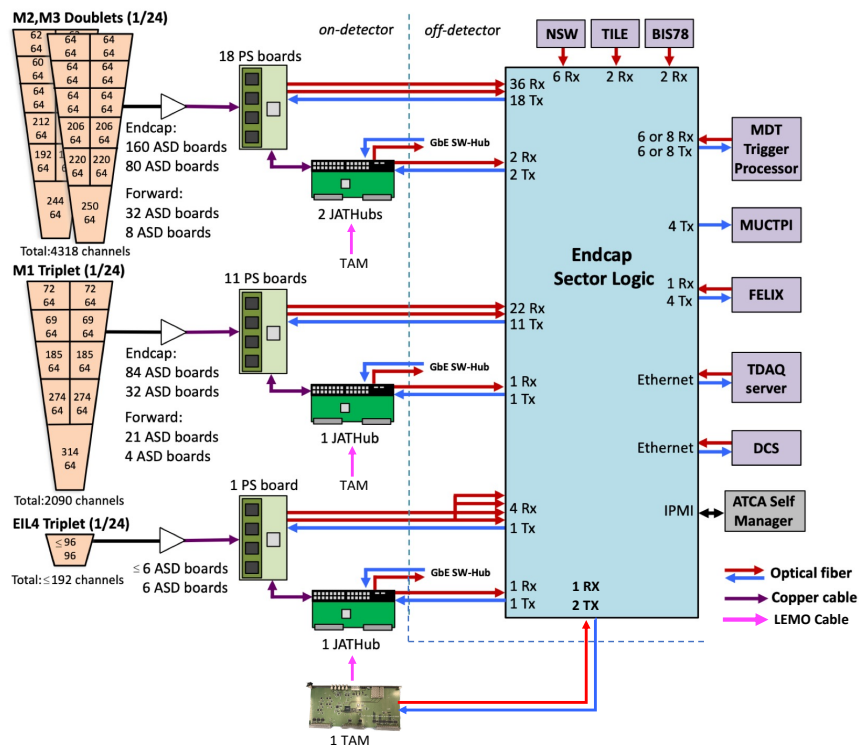


図 2.2: TGC エレクトロニクスの 1/24 セクターの概要図。[11] より引用の上、TAM に関する部分を加筆している。TGC 検出器からのヒット信号は、ASD ボードによってデジタル信号に変換され PS board へ入力される。それを受け取った PS board は、どのバンチ交差に対応するヒット信号かを同定する BCID を行い、ヒット信号の有無に関わらず全てのヒットビットマップを SL へ送信する。各 PS board や TAM には SL を通じてコントロール信号とともに TTC 信号が分配される。JATHub はデータパスとは完全に独立しており、PS board や TAM の FPGA のファームウェアコンフィギュレーション、回復不可能な SEU エラーへの対処、及び PS board 上の LHC バンチ交差クロックの位相のモニター等を行う。TAM は JATHub に対して VME master として働くとともに、SL から受けた TTC 信号を隣の 1/24 セクターに属する TAM と位相を比較して合わせた後、JATHub へ分配する働きをもつ。

図 2.2 に TGC エレクトロニクスの、1/24 セクターの概要図を示す。TGC 検出器全体では、この 1/24 セクターが、エンドキャップ領域の C-side, A-side に 24 セクターずつ、合計 48 セクター存在する (1.3 節参考)。読み出しやトリガー判定のためのモジュールである、ASD、PS board や Sector Logic は 1/24 セクターで閉じた設計になっ

ている。図 2.2 にあるように、1/24 セクターに 496 台の ASD、31 台の PS board と 1 台の SL が設置され、SL は主に 31 台の PS board からの入力を元にして、トリガー判定を行う。PS board のモニター、制御、タイミングキャリブレーションのためのモジュールである、JATHub や TAM は隣り合う 1/24 セクターを 2 つ合わせた 1/12 セクター (図 1.6) で閉じた設計になっている。つまり、1/12 セクターに 6 台の JATHub と 1 台の TAM が設置され、62 台の PS board のモニター、制御、タイミングキャリブレーションを行う。

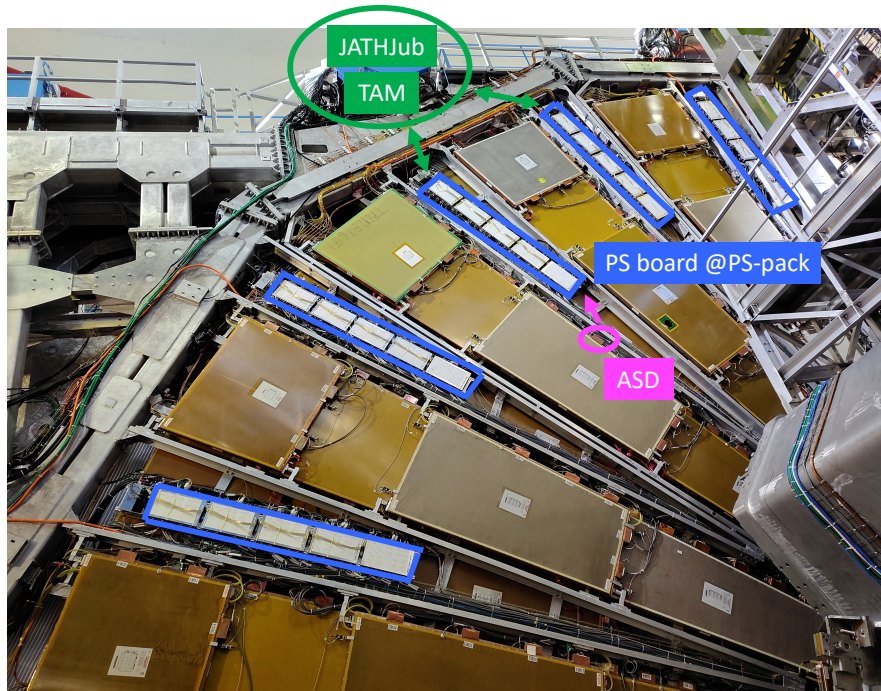
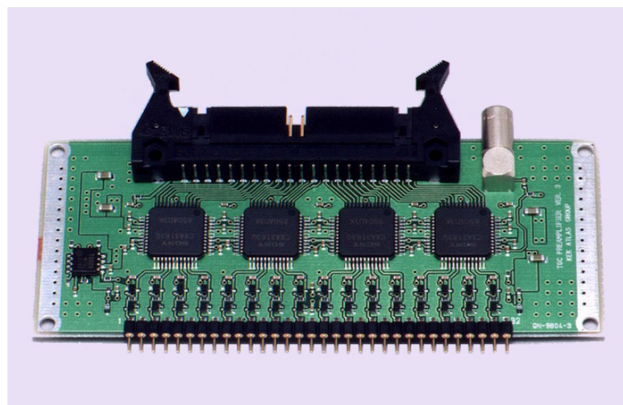


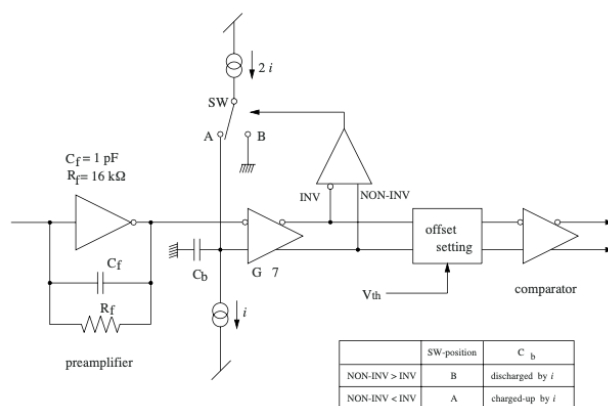
図 2.3: フロントエンドエレクトロニクスの設置場所。ASD ボードは TGC 検出器に直接取り付けられる。PS board は、写真のように TGC 検出器の外側に PS-pack と呼ばれる構造体に格納され、取り付けられる。JATHub 及び TAM は、TGC 検出器の近くに設置された Mini-Rack と呼ばれる 19 インチラック内の VME クレートに設置され、JATHub が VME slave、TAM が VME master として働く。

2.2.2 Amplifier-Shaper-Discriminator (ASD) ボード

図 2.4 に Amplifier-Shaper-Discriminator (ASD) ボードの概要を示す。TGC 検出器によるワイヤーとストリップのヒット信号は、最初に TGC 検出器のチェンバー部分に直接取り付けられている、ASD ボードに電流信号として入力される。ASD ボードでは、入力の電流信号を、電圧信号へと変換、増幅し、閾値電圧を超えている間のみ 1 を出力してデジタル信号に変換した後、LVDS 規格の差分信号に形成し、PS board へ出力する。1 枚の ASD ボードには計 4 枚の ASD チップが搭載されており、それぞれ TGC 検出器の 4 チャンネルを担当するため、1 台の ASD ボードで 16 チャンネルを担当することになる。TGC 検出器全体で 32 万チャンネルを取り扱うため、合計で約 2 万 3000 台の ASD ボードが、図 2.3 のように TGC 検出器に設置される。また ASD ボードには擬似信号であるテスト電荷を出力する機構 (ASD テストパルス) が搭載されている。その出力タイミングは衝突に由来するミューオンを擬似するために、2.2.3 節で述べる PP ASIC からタイミングを制御することで、バンチ交差クロック (LHC CLK) に正確に同期して出力が可能である。また、後で述べるように、ASD テストパルスを用いて読み出しパスを包括的に検証することができる。



(a) ASD ボードの写真 [2]



(b) ASD チップの回路のブロック図 [5]

図 2.4: ASD ボードの概要。図 2.4a が ASD ボードの写真。ASD ボード 1 枚あたり 4 枚の ASD チップが載っており、1 枚の ASD ボードでは 16 チャンネル分の TGC 検出器からの信号を処理する。図 2.4b は ASD チップの回路ブロック図。

2.2.3 Primary Processor board (PS board)

ASD ボードで、LVDS 信号として処理された、TGC 検出器からのヒット信号は、**Primary Processor board (PS board)** に入る。ASD ボードから受け取った信号を、PP ASIC で可変遅延をかけて、ミューオンの飛行時間 (**Time of Flight: ToF**) 差や ASD ボードから PS board までのケーブル長の違いを吸収した後、どのバンチ交差に対応する信号かを同定 (BCID) する。1 枚の PS board には 8 枚の PP ASIC が搭載されており、PS board FPGA は 256 チャンネル分のヒット信号を扱う。PS board FPGA はヒットの有無に関わらず、256 ビットの固定長のヒットビットマップをバンチ交差周に同期して、高速光通信を用いて後段 (SL) 転送し続ける。また、PS board は SL から高速光通信を通して配布される、**Timing Trigger and Control (TTC)** 信号を受信する。TTC 信号には、LHC のバンチ交差に同期した 40.079 MHz クロック (LHC CLK) や、BCID の際に付与する番号のリセットを行う **Bunch Counter Reset (BCR)** 等が含まれる。

PS board には、**Patch-Panel ASIC (PP ASIC)** と Xilinx 社製の Kintex-7 FPGA*1 の 2 種類の集積回路が搭載されている。また、SL との高速光通信のための SFP+ が 2 つ、JATHub との通信のための RJ45 ジャックが 2 つ搭載されている。その他にも、ASD ボードの閾値電圧を設定するための **Digital to Analog Converter (DAC)** や、DAC からの出力をモニターするための **Analog to Digital Converter (ADC)**、FPGA で再構成したクロックのジッターを小さくするためのクロックジッタークリーナー、ファームウェア及びボードに固有なパラメータを書き込んでおくための QSPI フラッシュメモリといった IC も搭載されている。

PS board は、PS-pack と呼ばれる箱に入れ、図 2.3 のように TGC BW 構造体に配置される。TGC 検出器の 1/24 セクターの M1 を読み出すために 11 台、1/24 セクターの M2、M3 を読み出すために 18 台がまとめて設置される。TGC 検出器全体で、PS board は合計 1434 台設置される。

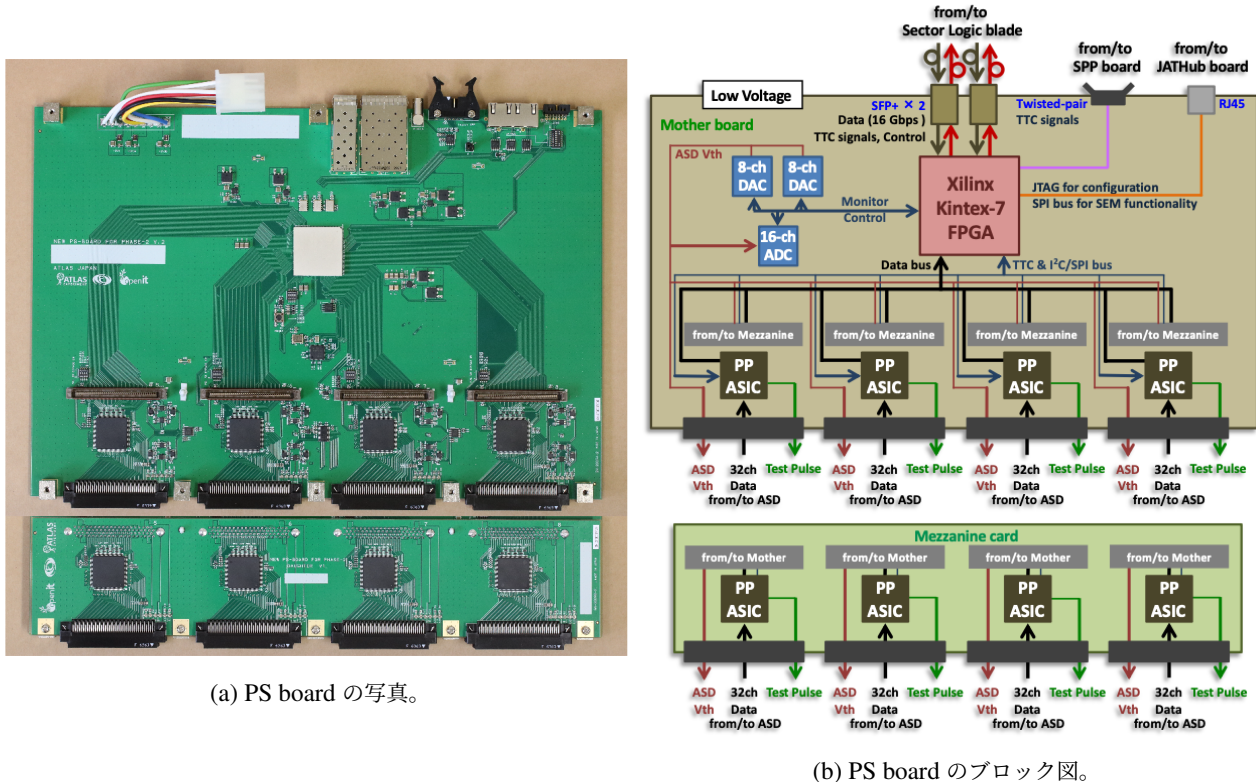


図 2.5: PS board の概要 [2]。図 2.5a が PS board の写真。図 2.5b が PS board のブロック図。1 枚の PS board には 8 枚の PP ASIC が搭載されており、PS board FPGA は 256 チャンネル分のヒット信号を扱う。PP ASIC において BCID が行われた後、ヒット信号は PS board FPGA に入力され、ヒットの有無に関わらず全てのヒットビットマップを SL へ高速光通信を用いて送信する。

Patch-Panel ASIC (PP ASIC)

図 2.6 に PP ASIC のブロック図を示す。ASD ボードからの信号はまず、PS board 上の PP ASIC に入る。1 つの PP ASIC には、2 つの ASD ボードが接続され、合計で 32 チャンネルのヒット信号を処理する。ヒット信号は PP ASIC で初めて BCID がなされる。そのために、PP ASIC には可変遅延回路と陽子バンチ識別回路が搭載されている。

まず、可変遅延回路について述べる。PP ASIC に入力される、ヒット信号は、衝突点からのミュオン ToF の違い (45 ns ~ 64 ns) や、ASD ボードから PP ASIC までのケーブルの配線長の違い (1.8 ~ 25 m) により最大で、各 PP ASIC に入ってくるタイミングが M1 内で最大 ~ 15 ns、M2、M3 内で最大 ~ 40 ns 異なる。可変遅延回路は、各 ASD ボードごとに信号遅延をかけることで、最もタイミングが遅れている ASD ボードからのヒットにその他のヒット信号を合わせる。可変遅延回路の刻み幅は、0.74 ns, 0.84 ns, 0.99 ns, 1.19 ns の中から選択できて、ステップ数は最大 48 ステップである。例えば、1.19 ns の刻み幅であれば、最大で $1.19 \text{ ns} = 55.93 \text{ ns}$ の遅延が可能である。

一方で、単一のチャンネルからの信号であっても図 2.7 のような時間分布を持ち、到着時間は 20 ns から 30 ns 程度の幅を持つ。これは、ミュオン検出器の入射位置の違いによる、イベント毎の検出器中の電子のドリフト時

*1 Field Programmable Gate Array の略。ユーザーがファームウェアを編集することで、内部のデジタル回路を何度でも書き換え可能な集積回路である。

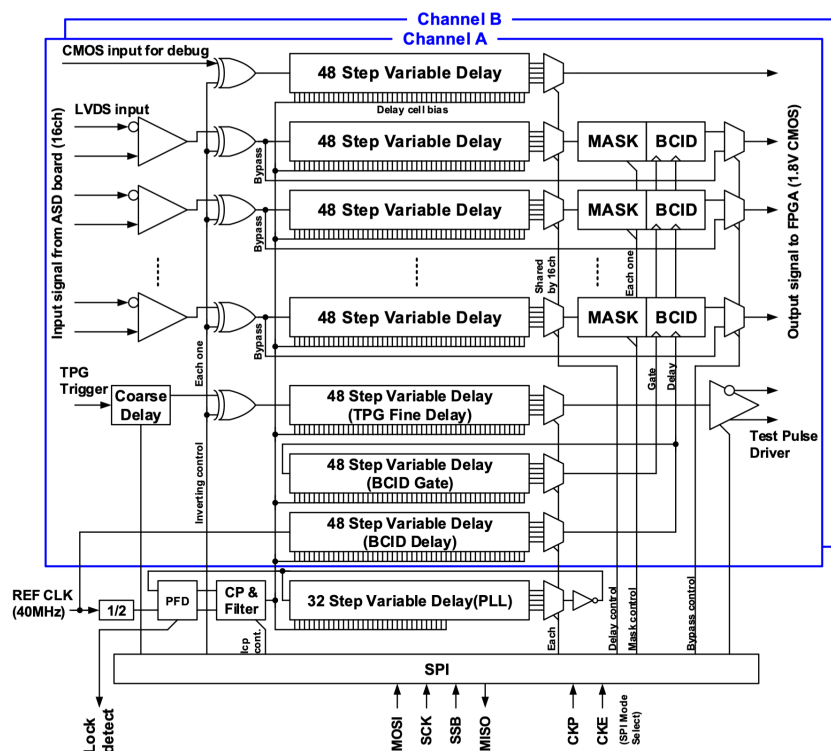


図 2.6: PP ASIC のブロック図 [17]。チャンネル A とチャンネル B はそれぞれ 1 つの ASD ボードに対応し、1 つの ASD ボードは TGC 検出器の 32 チャンネル分のヒット信号を取り扱う。PP ASIC には可変遅延回路と陽子バンチ識別回路が搭載されており、BCID が行われる。

間の違いや、ASD ボードまでの信号の伝搬速度の違いに起因するものである。この時間分布の幅の違いは、チャンネル毎のミュオン入射角度の違いに起因するものであり、チャンネルの η 座標に依存する。この時間分布幅に対しては、可変遅延回路を用いて、最も信号の到着時間が遅いチャンネルに立ち上がり具合を調整し、後述する陽子バンチ識別回路の有効ゲート幅 (BCID ゲート幅) を広く取ることに対応する。

次に、陽子バンチ識別回路について述べる。陽子バンチ識別回路では、ヒット信号がどのバンチ交差に対応するものかを同定する。図 2.8 に PP ASIC 内の陽子バンチ識別回路のタイミングチャートを示す。陽子バンチ識別回路はヒット信号の立ち上がりを検出する。可変遅延回路で、立ち上がりのタイミングが揃えられたヒット信号が、陽子バンチ識別回路で、40.079 MHz の LHC CLK と同期され、どの陽子バンチに由来するヒットであるかが同定される。同じ BCID を付与する時間幅を BCID ゲート幅と呼び、BCID ゲート幅は、0.74 ns, 0.84 ns, 0.99 ns, 1.19 ns の中から選択できて、ステップ数は最大 48 ステップから設定できる。その際に、図 2.7 で示した時間分布幅を持つヒット信号に対しても、同一の BCID を付与するために、BCID ゲート幅は、ヒット信号の時間分布を十分に覆うように設定する必要がある。ヒット信号の時間分布幅は、25 ns を超えるチャンネルに対しても、BCID ゲート幅はこれを覆うように設定する。その場合、図 2.8 の INPUT2 のような、前後の BCID のオーバーラップ領域にヒット信号が入射した際は、2 つの BCID に渡って陽子バンチ識別回路の出力がなされることになる。このような 2 つの BCID に渡ってヒット信号が同定されても、TGC 検出器は、7 層のヒットのコインシデンスを取るの、最終的にはどのバンチ交差に由来する信号かを一意に決定することができる。一方で、広げすぎると、コインシデンスとして異なるタイミングでミュオンを検出する確立が無視できなくなったり、陽子バンチ交差のタイミングとはズレたノイズ (低エネルギーの中性子や光子に起因する) の影響を受けるようになるため、BCID ゲート幅は必要

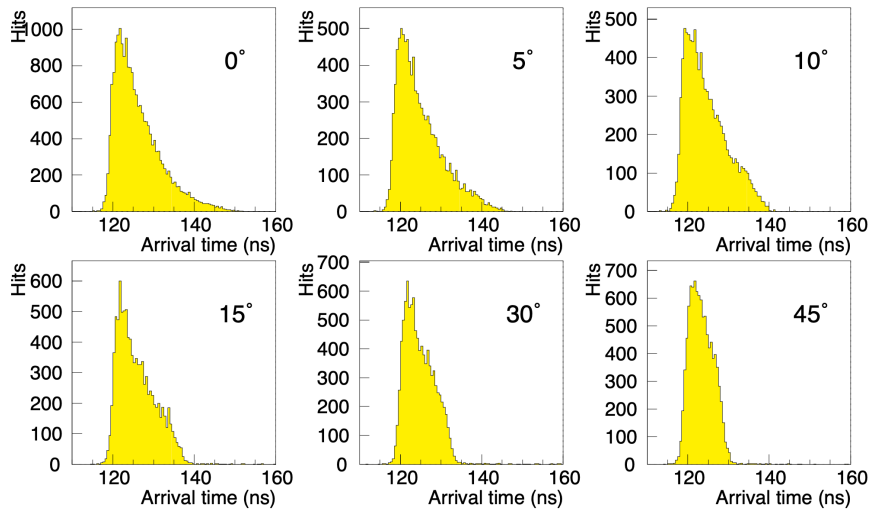


図 2.7: ミューオンが TGC 検出器に入射してから、検出器から信号が出力されるまでの時間分布 [20]。時間分布はミューオンが入射する角度に依存して、20–30 ns の幅を持つことが分かる。

最小限にすることが求められる。

そのためには、PS board が再構成する LHC CLK の位相をできる限り合わせる必要がある。上記で述べた PP ASIC の可変遅延回路や陽子バンチ識別回路は、PS board が SL から光リンクを通じて再構成した、LHC CLK で駆動する。よって、全 1434 台の PS board の再構成する LHC CLK の位相が揃っていないと、その分、陽子バンチ識別回路の BCID ゲート幅を広く取って、そのばらつきを吸収する必要が出てしまう。一方で、実際のオペレーションでは、PP ASIC の可変遅延回路の可変遅延や、陽子バンチ識別回路の BCID ゲートの刻み幅は、0.99 ns の使用を想定しており、これ以上の精度で、PS board の再構成する LHC CLK を合わせる必要はない。以上より、PS board の位相を十分な精度 (1 ns 以内) で合わせる必要があり、これに関するデモンストレーションは、5.3 節で述べる。

また、PP ASIC には、PS board からのコントロールを受けて、ASD ボードにテストパルストリガーを発行したり、ノイズが多いチャンネルをマスクする機能等も実装されている。特に ASD ボードにテストパルストリガーを発行する機能に関しては、2.2.2 節で述べたように、タイミングをサブナノ秒の精度で制御できる設計がなされている (図 2.6 の Coarse delay と TPG Fine delay に対応する)。この ASD テストパルスを用いて、5.4.1 節のデモンストレーションを行なった。

PS board FPGA

PS board には Xilinx 社製の kintex-7 FPGA が搭載されている。PP ASIC で BCID されたヒット信号は、FPGA に入る。PS board には 8 つの PP ASIC が搭載されていて、1 つの PP ASIC あたり 32 チャンネルを処理するので、PS board FPGA は毎バンチ交差ごと (25 ns 毎) に、256 チャンネルを入力として受けとる。PS board FPGA はヒットの有無に関わらず、256 チャンネルの固定長のヒットビットマップを毎バンチ交差ごと (25 ns 毎) に SL へ転送する。よってヒット信号の転送レートは、 $256 \text{ bit} \times 40.079 \text{ MHz} = 10.260 \text{ Gbps}$ となる。この転送レートを実現するために、PS board と SL 間の通信には高速光通信が実装される。また、PS board の FPGA や IC の状態を常に SL に転送したり、PS board のレジスタの読み書き、TTC 信号の分配等にもこの光リンクが使用される。詳しくは 3.3.2 節で述べるが、PS board と SL は 3 本の光ファイバーで接続され、うち 2 本が PS board が送信側とし SL を受信側とする通信 (Up link)、残る 1 本が SL を送信側とし PS board を受信側とする通信 (Down link) であ

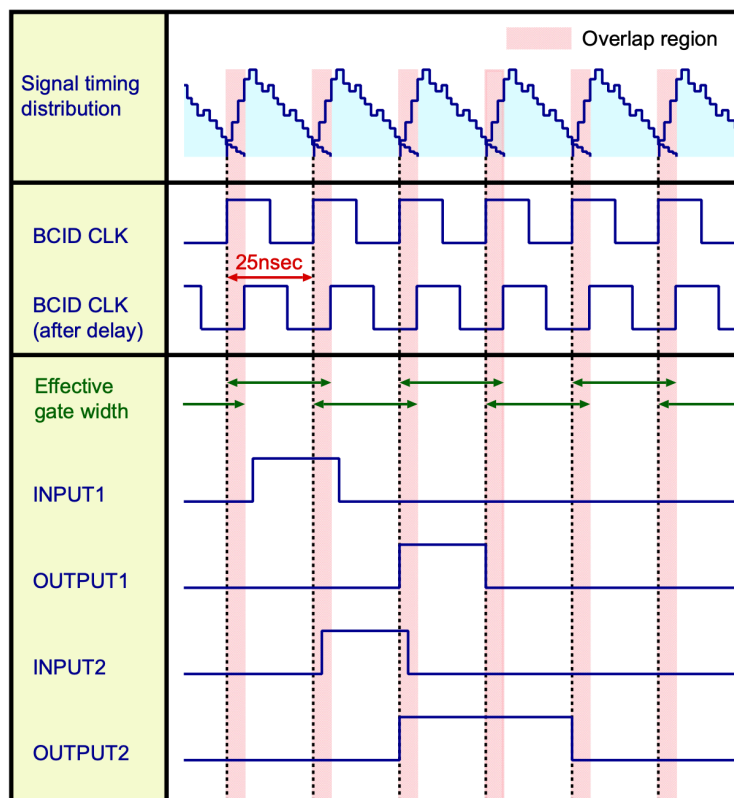


図 2.8: PP ASIC 内の陽子バンチ識別回路のタイミングチャート [4]。INPUT1 は有効ゲートが重ならないタイミングで陽子バンチ識別回路にヒット信号が入射した場合を示しており、OUTPUT1 がこれに対応する出力信号である。INPUT2 は有効ゲートが重なるタイミングで陽子バンチ識別回路にヒット信号が入射した場合を示しており、OUTPUT2 がこれに対応する出力信号であり、これを見ると 2 バンチに渡って出力されていることが分かる。

る。これらの光ファイバー線を用いて、PS board と SL の間には 1 本あたり、 $40 \text{ bit} \times 200.395 \text{ MHz} = 8.016 \text{ Gbps}$ の転送レート的高速光通信が実装される。つまり、各 PS board から SL へは、 $8.016 \text{ Gbps} \times 2 = 16.032 \text{ Gbps}$ 、SL から PS board へは、 8.016 Gbps の転送レートが確保されている。光リンクの実装には Xilinx 社の提供する高速シリアル通信対応のトランシーバの 1 種である 7 シリーズ GTX トランシーバ [7] を用いて行われる。これらの通信はバンチ交差と正確に同期する必要があるため、SL から配布される LHC CLK を基準に駆動される。

先ほど述べた PP ASIC や、ASD ボードに閾値電圧を供給する Digital to Analog Converter (DAC)、閾値電圧をモニターする Analog to Digital Converter (ADC) 等の PS board 上の IC の制御・コンフィギュレーションも PS board FPGA が行う。

FPGA 上にデジタル回路を構成するためには、FPGA 上の Static Random Access Memory (SRAM) をプログラムする。このことを FPGA のコンフィギュレーションと呼ぶ。SRAM は揮発性メモリのため、FPGA の電源を入れ直したりすると、ファームウェアの情報は失われる。そこで、PS board では、不揮発性の、QSPI フラッシュメモリにファームウェアを保管しておき、FPGA の電源が起動した際には、自動で QSPI フラッシュメモリからファームウェアを参照して、FPGA のコンフィギュレーションを行う設計をしている。一方で、PS board の FPGA 上のレジスタの値や SL との光リンクは、初期状態に戻り、FPGA のコンフィギュレーションのたびに、再設定やリンク確立の手続きが必要である。特に、2.2.4 節で述べるように高輝度 LHC-ATLAS 実験では、回復不可能な SEU が生じた際に、PS board は FPGA の再コンフィギュレーションを行うことでこれに対処する。この場

合、別々のタイミングで、個々の PS board の FPGA が再コンフィギュレーションを行うため、遅延パラメータといったボードに固有のパラメータのプログラムや SL との光リンクの再構成などをボードごとにそれぞれのタイミングで行う必要がある。これを実現しているのが、3.2 節で述べる自律型制御機構であり、PS board は初期状態から、外部からの特別な操作なしに自身をコンフィギュレーションする。

2.2.4 JTAG Assistance Hub (JATHub)

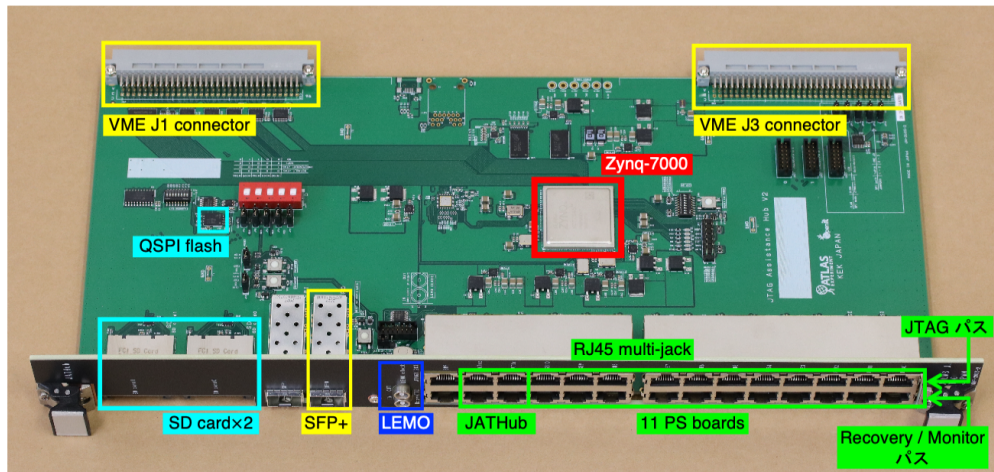


図 2.9: JATHub の写真 [2]。JATHub には FPGA と CPU が一体化した System-on-Chip(SoC) デバイスである、Zynq-7000 が搭載されている。JATHub はデータバスとは独立した PS board と TAM の制御用のモジュールであり、FPGA のファームウェアプログラミングや、回復不可能な SEU が生じた際に FPGA の再コンフィギュレーションを行う機能、PS board のクロックの位相をモニターする機能を RJ45 multi jack に接続した Cat 6 ケーブルを経由して行う。冗長性を持たせたブートのために、QSPI フラッシュメモリ及び 2 枚の SD カードも搭載されている。JATHub は TGC 検出器の近くに設置された Mini-Rack 内の VME クレートに設置されるため、TAM をマスターとする VME 通信が可能である。また光イーサネットのインターフェイスとして SFP+ も搭載されている。

フロントエンドエレクトロニクスのモニター、制御のためのモジュールとして、**JTAG AssisTance Hub (JATHub)** が導入される。JATHub はデータバスとは完全に独立しており、クロックドメインとしても、TAM から基準クロックを受け取る部分以外は、LHC CLK とは非同期でスローな制御を担当する。

図 2.9 に JATHub の写真を示す。JATHub には、Xilinx 社製の Zynq-7000 デバイスが搭載されている。これは、FPGA と CPU が一体化した **System-on-Chip (SoC)** であり、FPGA 部分を **Programmable Logic (PL)**、CPU 部分を **Processing System (PS)** と呼ぶ。また、光イーサネット通信のため SFP+ や TAM から基準クロックを受け取るための LEMO コネクタ、フロントエンドエレクトロニクスの回復手続き及びモニター、JTAG 通信のための RJ45 ジャック、TAM との VME 通信のための VME J1, J3 コネクタが搭載されている。加えて、JATHub は放射線環境下に設置されるので、冗長性を持たせたブートのために、QSPI フラッシュメモリ及び 2 枚の SD カードも搭載されている。

JATHub は、TGC 検出器の外枠に設置された Mini-Rack と呼ばれる 19 インチラック内の VME クレートに設置される (図 2.3)。1 台の JATHub に対して、最大 11 台の PS board もしくは TAM を接続し制御することができ、1/12 セクターのフロントエンドエレクトロニクスをカバーするために、6 台の JATHub が 1 つの Mini-Rack に設置される。

JATHub SoC

JATHub の主な機能は、フロントエンドエレクトロニクスの、

1. JTAG 線経由での FPGA のコンフィギュレーション及び QSPI へのファームウェア書き込み
2. 放射線損傷に対する回復手続き
3. 再構成したクロックのモニター及び delay parameter の決定

である。まず、JTAG 線経由での FPGA のコンフィギュレーション及び QSPI へのファームウェア書き込みについて述べる。PS board や TAM に搭載されている FPGA のコンフィギュレーションを行う際は、Xilinx Virtual Cable (XVC) もしくは、Serial Vector Format (SVF) player を用いる。どちらの場合も Zynq のソフトウェアが、PS board に接続されている JTAG 信号バスを制御することで、PS board FPGA への JTAG 通信を実現する。XVC の場合は、JATHub は中継地点として機能し、イーサネットで接続される上位のサーバーからのコマンドを中継する。一方 SVF player の場合は Zynq のソフトウェアが SVF ファイルを参照して JTAG バスを制御する。

次に放射線損傷に対する回復手続きについて述べる。PS board や JATHub、TAM は ATLAS 実験室に設置されるため、放射線損傷に対する回復手続きが必要である。TGC 検出器は ATLAS 検出器の最外層に位置していることに加えて、PS-pack や Mini-Rack は TGC 検出器の外枠付近に設置されているので、ビーム軸から離れた場所に位置する。よって、PS board、JATHub、TAM が受ける放射線損傷はそれほど大きくはないことが期待されるが、FPGA や素子が受ける放射線損傷の中には、データ収集に影響を及ぼすものがある。放射線損傷には、電離作用による Total Ionizing Dose Effect (TID) や、Non Ionizing Energy Loss (NIEL) による弾き出し損傷効果といった半永久的な損傷を起こしうるものに加えて、半永久的ではないものの、メモリのビットが反転してしまう Single Event Upset (SEU) の 3 種類が考えられる。このうち、TID や NIEL に対しては、高輝度 LHC-ATLAS 実験における放射線環境に耐えうる素子の選定を行うことによって対処している。SEU に関しては、FPGA の SRAM に SEU が生じると、FPGA が設計通りに動かなくなる可能性があるため、FPGA の SEU に対する修復が必要である。修復が可能であるもの (1 ビットエラー及び隣接する 2 ビットエラー) に対しては、FPGA に Soft Error Mitigation (SEM) Controller を実装することで、SEM Controller が自動的に修復する。修復不可能な SEU (隣接しない 2 ビットエラー及び 3 ビット以上のエラー) に対しては、FPGA の再コンフィギュレーションを行う必要があるが、JATHub がこの役割を担う。PS board や TAM の FPGA において修復不可能な SEU が発生した場合、SEM Controller がそれを検知して、Cat 6 ケーブルを通じて JATHub に救難信号を送る。それを受け取った JATHub は自動的に、対応する PS board や TAM の FPGA の再コンフィギュレーションを行う。また、隣合う JATHub 同士も Cat 6 ケーブルで接続することで、JATHub の SoC に修復不可能な SEU が生じた際には、隣の JATHub に救難信号を送り、SoC の再コンフィギュレーションを行うことで、JATHub 自身の放射線損傷に対応する。

最後に、再構成したクロックのモニター及び delay parameter の決定について述べる。PS board は SL との光通信を介して TTC 信号を固定位相で受信するが、SL から各 PS board までの光ファイバーは、40 m の長さのものから 90 m のものまであり、対応してクロック分配時間に分布が生じる。したがって、1434 台の PS board における TTC 信号の位相を揃える必要があり、具体的には、PS board の PP ASIC で正しい BCID を行うために 1 ns より十分良い精度で位相を揃える。この目的のために JATHub と TAM が用いられる。JATHub の役割は、TAM から LEMO コネクタを通じて受信した基準クロックを元にして、自身の RJ45 ジャックに繋がる PS board (最大 11 台) のクロックの位相を測定することである。

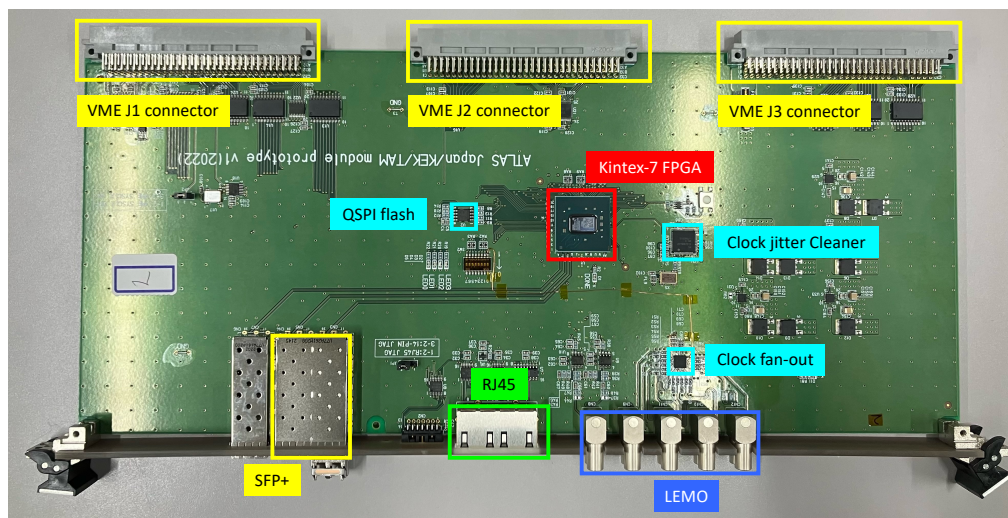


図 2.10: TAM の写真。TAM には Xilinx 社製の kintex-7 FPGA が搭載される。SL との光通信のための SFP+ が 2 つ、JATHub から制御を受けるための RJ45 ジャックが 2 つ搭載される。TAM は TGC 検出器の近くの Mini-Rack 内の VME クレートに設置され、VME マスターとして働く。また、再構成したクロックを 6 台の JATHub と隣の TAM、自身に分配するために、合計で 8 つの LEMO OUT と fan-out 素子、クロックジッタークリーナ、位相合わせのための LEMO IN が 2 つ搭載される。またファームウェアを収めておく空間として QSPI フラッシュメモリが搭載される。

2.2.5 Timing Alignment Master (TAM)

PS board のタイミングキャリブレーションと、Mini-Rack の VME マスターとして、Timing Alinment Master (TAM) が Mini-Rack に設置される。

図 2.10 に TAM の写真を示す。TAM には Xilinx 社製の kintex-7 FPGA が搭載される。また、SL との光通信のための SFP+ が 2 つ、JATHub との VME backplane との接続のための VME J1, J2, J3 コネクタ、JATHub との JTAG 通信及び回復パスのための RJ45 ジャックが 2 つ、JATHub への基準クロック分配及び隣合う TAM 同士の位相合わせのための LEMO コネクタ及びクロック fan-out、再構成したクロックのジッターを小さくするためのクロックジッタークリーナー、ファームウェアやボードに固有のパラメータを格納しておく QSPI フラッシュメモリなどが搭載されている。TAM の機能に関しては、4 章で詳しく述べる。

2.2.6 Sector Logic (SL)

図 2.11 に、SL の写真を示す。SL には、Zynq Ultrascale+ MPSoC、Virtex Ultrascale+ FPGA、IPMC の 3 つの集積回路が搭載されている。また、PS board 等との光リンクのための MPO24 コネクタ、Fire Fly やイーサネット接続のための RJ45 ジャック、ブートファイルを収める micro SD カード、クロックジッタークリーナ、ファンアウト素子等も搭載されている。

SL は、1/24 セクターに 1 台設置され、最大 31 台の PS board からのヒット信号と、磁場領域よりも内側にある検出器、NSW、RPC BIS78、Tile カロリメータからの情報も受信して、トリガー演算を行う。また、PS board のモニター・制御も SL が行う。

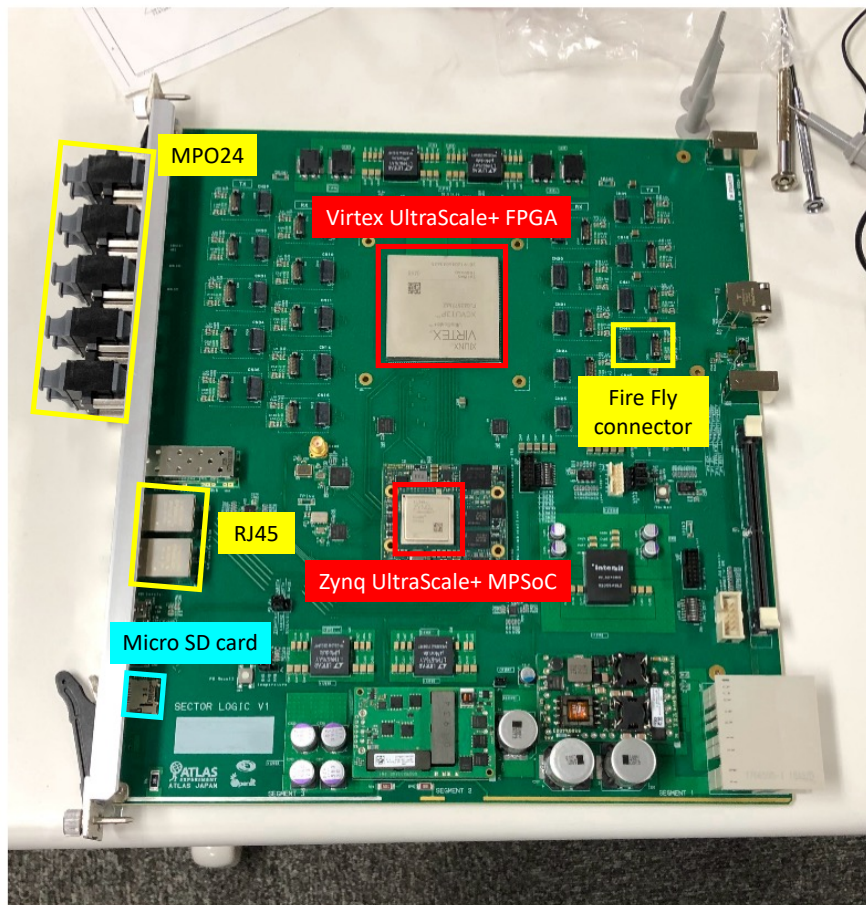


図 2.11: SL の写真。Zynq Ultrascale+ MPSoC、Virtex Ultrascale+ FPGA、IPMC の 3 つの集積回路が搭載されている。PS board や TAM との光リンクのための MPO24 コネクタ、Fire Fly やイーサネット接続のための RJ45 ジャック、ブートファイルを取る micro SD カード、クロックジッタークリーナ、ファンアウト素子等も搭載されている。

Virtex Ultrascale+ FPGA

Virtex Ultrascale+ FPGA には、

- PS board からヒット信号の固定位相での受信
- トリガー演算を行いミュオン飛跡と運動量の再構成
- 読み出し回路
- タイミング信号を CTP から LTI、FELIX を経由して固定位相で受信し、PS board へ固定位相で分配
- PS board や TAM のモニター・コントロール (LHC CLK に非同期)

といった機能が実装されている。1 台の SL には最大 31 台の PS board が接続され、2.2.3 節で述べたように、1 台の PS board ごとに 256 ビットの固定長のヒットビットマップを LHC CLK に同期して、固定位相で受信する。SL は PS board からのヒットビットマップに加えて、TGC より内側の磁場領域にある NSW、Tile、BIS78 といった検出器からのヒット信号を受け取り、トリガー演算として固定レイテンシーでのミュオンの再構成を行う。その

結果を MUCTPI と MDT Trigger Processor(MCTTP) に送信するとともに、イベント毎にデータをパケット化 (イベントビルディング) し、所定のデータフォーマットに整形して FELIX に送信する。読み出し回路では、PS board から固定位相で受信したヒットビットマップを一時的にバッファに保存しておき、FELIX からの L0 Accept 信号 (TTC 信号の一種) を受け取ると、それに対応するヒットビットマップのイベントビルディングを行い FELIX に送信する。SL MPSoC からの指示を受けて、PS board や TAM、SL FPGA 内のモジュールをコントロールしたり、モニターした値を SL MPSoC から読み出したりする。トリガーに関する機能以外は、本研究で扱うところであり、3 章で述べる。

Zynq Ultrascale+ MPSoC

JATHub の Zynq 7000 デバイスと同様に、SL の Zynq Ultrascale+ MPSoC も PS と PL と呼ばれる二つの領域から成る。詳しくは 3.3 節で述べるが、SL MPSoC の役割は、主に SL FPGA レジスタのモニター・コントロールである。また、5.4.1 節で述べるように、TGC リードアウトのデータ検証のために、読み出しパスの最終段から SL MPSoC 上の BRAM へのダンプパスも実装されている。

IPMC



図 2.12: IPMC の写真 [19]。IPMC はボード上の温度センサーや電圧センサーの値をモニターし、ATCA shelf manager に報告する役割を果たす。

図 2.12 に IPMC の写真を示す。SL は Advanced Telecommunications Computing Architecture (ATCA) 規格のボードであり、Intelligent Platform Management Bus (IPMB) を用いた、ATCA shelf manager とのインターフェイスとして、Intelligent Platform Management Controller (IPMC) を使用する。IPMC はボード上の温度センサーや電圧センサーの値をモニターし、ATCA shelf manager に報告する。また、ボードの電源管理を行い、shelf manager を通じた遠隔からの操作や、ホットスワップの操作に応じて、主電源である 12 V 電源を含むレギュレータを操作する役割を果たす。

第3章

システムレベルの機能説明と実装

この節では、TGC トリガーエレクトロニクスのシステムレベルの機能説明と実装に関する説明を行う。TGC トリガーエレクトロニクスのうち、PS board、JATHub、TAM は ATLAS 実験室、つまり TGC 検出器がある空間に配置され、放射線環境下にある (これらをフロントエンドエレクトロニクスと呼ぶ)。一方で、SL は ATLAS 回路室と呼ばれる、TGC 検出器のある空間から隔離された空間に配置され、放射線環境下ではない (これをバックエンドエレクトロニクスと呼ぶ)。

読み出しパスに関しては、PS board と SL のみで完結する。つまり TGC 検出器からのヒット信号を ASD ボードが、整形し、PS board で BCID 等がなされたヒットデータを SL 上で処理するといった流れである。また、PS board のコントロールや回復、キャリブレーション用のモジュールとして、JATHub や TAM が導入される。まず、PS board の放射線損傷に関しては、JATHub がその役割を担う。PS board の FPGA には、回復可能な SEU に対して、自己修復を行う機能を実装しているものの、自己修復不可能な SEU に対しては JATHub から reboot を行う。またタイミングキャリブレーションの観点では、JATHub と TAM がその役割を担う。まず、各 PS board の再構成したタイミング信号を測定するために JATHub が用いられる。また、その際に同一の 1/12 セクターに属する JATHub に共通の基準クロックを配るモジュールとして、TAM が用いられる。また、TAM は異なる 1/12 セクター間のタイミング信号の位相を合わせる役割も果たす。

3.1 ブート・コンフィギュレーションパス

図 3.1 に TGC エレクトロニクスにおけるブート・コンフィギュレーションパスの概要を示す。SL の起動時は、Zynq Ultrascale+ MPSoC は、micro SD 内のブートファイルを読み出してブートを行う。また Virtex ultrascale+ FPGA は、QSPI フラッシュメモリ内のファームウェアとユーザーパラメータを読み出し、自律型制御機構 [2] によってコンフィギュレーションを行う。QSPI フラッシュメモリへのファームウェアおよびユーザーパラメータの書き込みは Zynq Ultrascale+ MPSoC から行う。また、Zynq Ultrascale+ MPSoC から XVC(2.2.4 節参照) を起動して、Virtex ultrascale+ FPGA のファームウェアコンフィギュレーションを行うことも可能である。

PS board と TAM は、FPGA のコンフィギュレーションの際には、QSPI フラッシュメモリ内のファームウェアとユーザーパラメータを読み出し、自律型制御機構によってコンフィギュレーションを行う。QSPI フラッシュメモリへのユーザーパラメータの書き込みは SL から光リンクを経由して行う。QSPI フラッシュメモリへのファームウェアの書き込みは JATHub から Cat 6 ケーブルを経由して行う。また、同様のパスを用いて JATHub から XVC を起動して、PS board や TAM の FPGA に直接ファームウェアをコンフィギュレーションすることも可能である。JATHub は、QSPI フラッシュメモリおよび SD カード内のブートファイルを読み出してブートを行う。

以上のブート・コンフィギュレーションシステムの構築により、図 5.2 に述べるようなブート・コンフィギュレーションシステムが完成する。

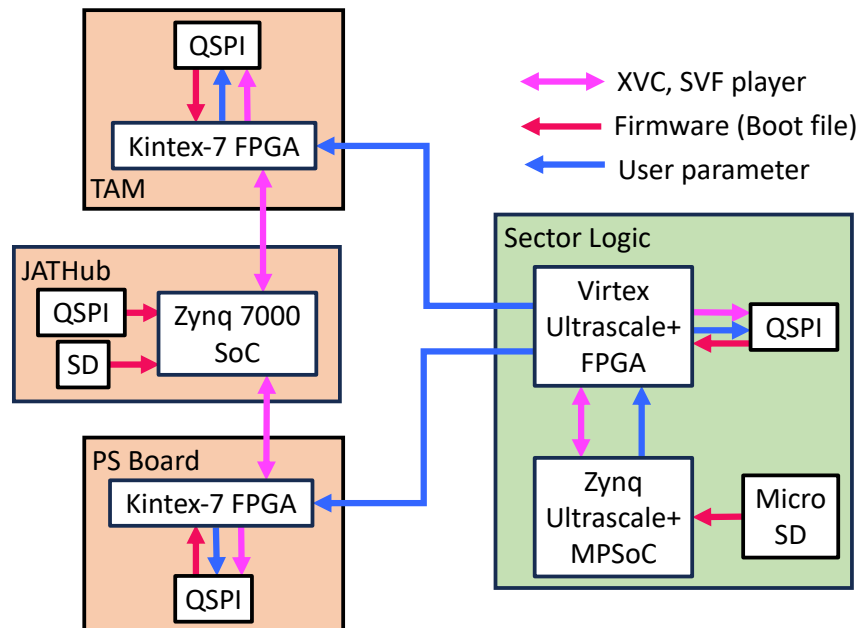


図 3.1: ブート・コンフィギュレーションパスの概要。赤線がファームウェア (ブートファイル)、青線がユーザーパラメータ、ピンク色の線が XVC/SVF player のパスを示す。SL の Zynq Ultrascale+ MPSoC は Micro SD 内のブートファイルを読み出してブートを行う。SL の Virtex Ultrascale+ FPGA 及び、PS board、TAM の FPGA は QSPI 内のファームウェアを読み出してコンフィギュレーションを行う。JATHub の Zynq 7000 SoC は SD 及び QSPI 内のブートファイルを読み出してブートを行う。SL の QSPI へのユーザーパラメータの書き込みは Zynq Ultrascale+ MPSoC から Virtex Ultrascale+ FPGA を経由して行う。PS board、TAM のユーザーパラメータの書き込みは、Zynq Ultrascale+ MPSoC から Virtex Ultrascale+ FPGA、光リンク、PS board(TAM) FPGA を経由して行う。SL の QSPI へのファームウェア書き込みや Virtex Ultrascale+ FPGA のコンフィギュレーションは、Zynq Ultrascale+ MPSoC から XVC、SVF player を用いて行う。PS board、TAM の QSPI へのファームウェア書き込みや FPGA のコンフィギュレーションは、JATHub から XVC、SVF player を用いて行う。

3.2 放射線損傷に対するモニターと回復パス

図 3.2 に TGC フロントエンドエレクトロニクスの回復手続きの概要図を示す。TGC フロントエンドエレクトロニクスは放射線環境下に設置されるため、放射線損傷に対する回復の手続きが必要である。2.2.4 節で述べた通り、修復可能な SEU に対しては、個々の FPGA に埋め込まれた SEM Controller で対処する。修復不可能な SEU に関しては、PS board、TAM は Cat 6 ケーブルを経由して JATHub に救難信号を送り、JATHub から FPGA の再コンフィギュレーションをしてもらう。その際、PS board や TAM には自律型制御機構 (Autonomous Control Mechanism) が実装されているため、自動で FPGA やボード上の IC のコンフィギュレーションを行う。JATHub に修復不可能な SEU が発生した場合は、隣の JATHub に Cat 6 経由で救難信号を送信し、隣の JATHub から Zynq の再コンフィギュレーションをしてもらう。

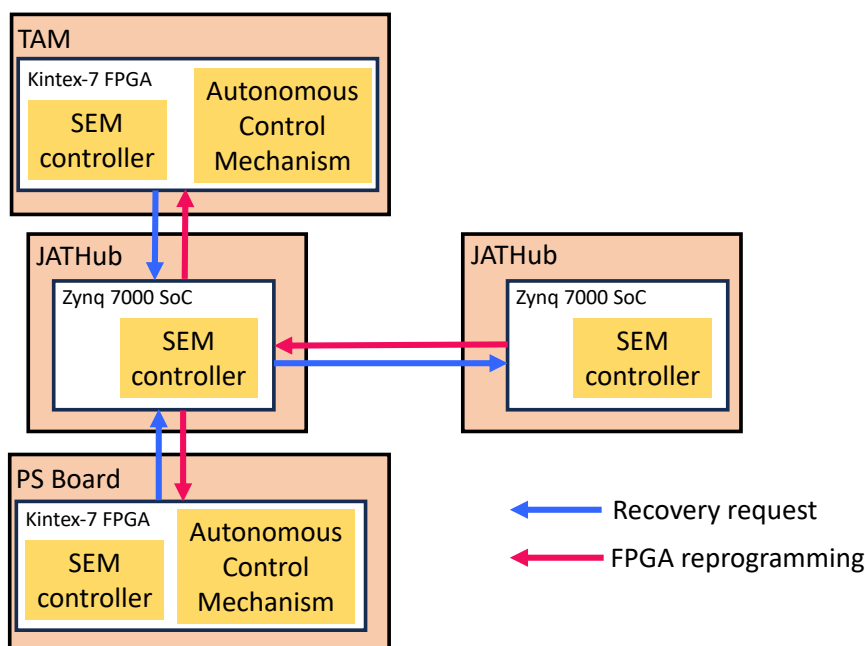


図 3.2: TGC フロントエンドエレクトロニクスの回復の概要図。赤線が FPGA(SoC) の再コンフィギュレーション、青線が回復不可能な SEU が起きた際の救難信号を示す。回復可能な SEU に対しては、FPGA(SoC) に実装された SEM Controller が自動で修復する。回復不可能な SEU に関しては、PS board、TAM は JATHub に救難信号を送り、FPGA の再コンフィギュレーションを行い、自律型制御機構によりパラメータの設定を行う。JATHub の SoC に回復不可能な SEU が生じた場合は、隣の JATHub に救難信号を送り、SoC の再コンフィギュレーションを行う。

3.2.1 SEM Controller

Xilinx 社の提供する、Soft Error Mitigation (SEM) Controller Core[1] をフロントエンドエレクトロニクスには実装してある。これを用いると、修復可能な SEU に対しては自動で SEM が自動で修復するとともに、修復不可能な SEU が生じた場合はフラグを立てる。このフラグを元にして、JATHub へ救難信号を送る。

3.2.2 自律型制御機構 (Autonomous Control Mechanism)

図 3.3 に PS board における自律型制御機構の概要図を示す。TAM に関しては、図 4.7 に示してあるが、手順としてはほとんど同じである。また SL は放射線環境下にはないが、コンフィギュレーションの簡便のため自律型制御機構が実装されており、これに関しても PS board とほとんど同じ実装がされている。

自律型制御機構は、自ら判断をしてスタートをトリガーするとともに、現在の操作が完了したことを自ら確認してから次のステップへの遷移するように設計されており、コンフィギュレーションに必要な手続きを適切な順序で行う。自律型制御機構は、電源の起動時、FPGA の再コンフィギュレーション、SL からのリセット等をトリガーとして、スタートする。自律型制御機構の最初のステップは、QSPI フラッシュメモリからのユーザーパラメータの読み取りと読み取った値を FPGA 上の各モジュールへ分配することである。並行して、SL との光リンクを駆動する、GTX トランシーバのリセットも行われる。3.4.1 節で述べるように、固定位相での光リンク確立は特別な操作を要するため、必要な手続きを適切な順序で行い、光リンクの確立を行う。

図 3.4 に自律型制御機構におけるユーザーパラメータの流れを示す。自律型制御機構におけるユーザーパラメー

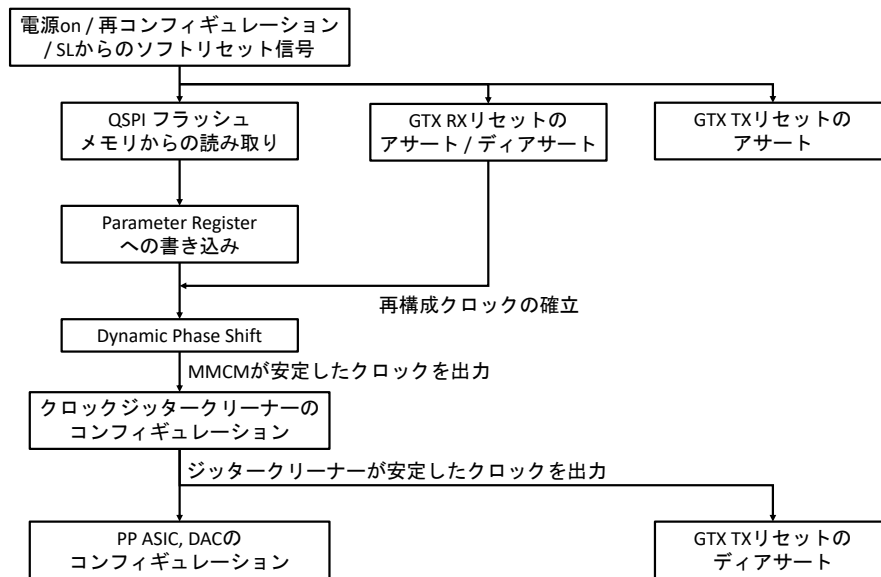


図 3.3: PS board における自律型制御機構の概要図。

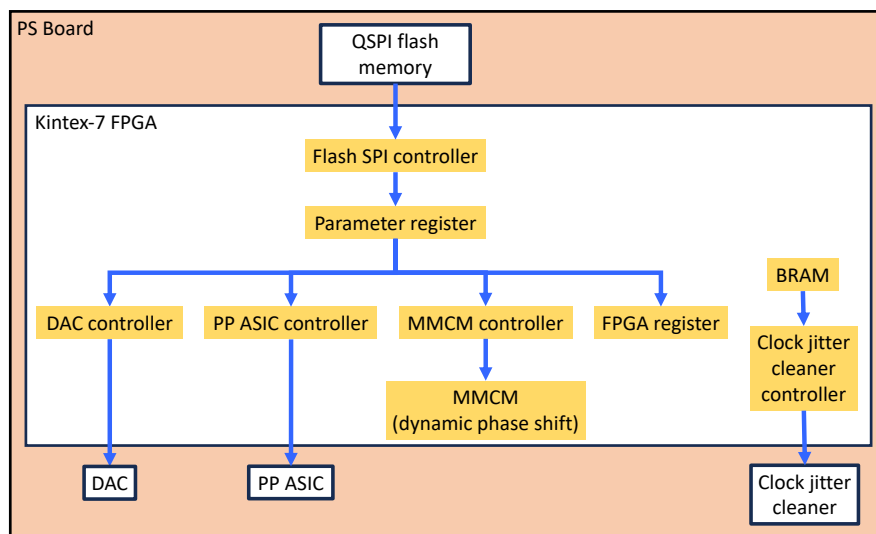


図 3.4: 自律型制御機構におけるユーザーパラメータの流れ。

タの扱いは、ボードに固有 (ボードごとに異なる) パラメータは QSPI フラッシュメモリに書き込んでおき、ファームウェアは共通のものを用いる。これにより、開発・実装するファームウェアは1つでよく、ボードごとの違いは QSPI フラッシュメモリに格納されるユーザーパラメータを用いて吸収することができる。TGC エレクトロニクスにおいて、クロックジッタークリーナーのコンフィギュレーションはボードごとに変わらず、PS board の場合は 1434 台共通である。よって、クロックジッタークリーナーのコンフィギュレーションパラメータは、QSPI フラッシュメモリに書き込む必要はなく、ファームウェア内にハードコードすることが可能で、具体的には FPGA 内の **Block RAM (BRAM)** と呼ばれるリソースを用いてこれを実現した^{*1}。一方でその他のパラメータはボードごと

*1 ファームウェアに初期値として、格納されたパラメータを変更する際には、ファームウェアの generate bitstream からやり直す必要がある。

に一般には異なる。例えば、MMCM(dynamic phase shift)の delay parameter などは PS board と SL 間のファイバー長に依存して変化する。このようなパラメータはファームウェア上ではなく QSPI フラッシュメモリに保管しておく。

QSPI フラッシュメモリからのパラメータの読み出しは、図 3.4 中の Flash SPI Controller というモジュールが行う。このモジュールは、SPI プロトコルを用いて、QSPI フラッシュメモリからパラメータを読み出して、FPGA 上のレジスタに保管する。次に FPGA 上のレジスタに保管されたパラメータは、Parameter register というモジュールが解釈して、個々のモジュールに分配される。

こうして各モジュールにパラメータが分配され、かつ SL から受け取る LHC CLK の再構成が完了すると、MMCM(dynamic phase shift)を用いた LHC CLK の delay が行われる。MMCM の IP デザインは後述 (図 3.12) するが、安定したクロックを出力できているかを示す出力ポートがあり、これが dynamic phase shift が完了した後、high であることを確認してから、クロックジッタークリーナーのコンフィギュレーションに進む。Clock jitter cleaner controller がクロックジッタークリーナーのコンフィギュレーションを行う。クロックジッタークリーナーのコンフィギュレーションパラメータは約 $24 \text{ bit} \times 600 = 1.8 \text{ KB}$ あり、これを SPI プロトコルにしたがってクロックジッタークリーナーに書き込んでいく。5.5 節で述べるようにクロックジッタークリーナーのコンフィギュレーションに関わる時間が、自律型制御機構の約 $1.3 \text{ s} / 1.416 \text{ s} \approx 92\%$ を占める。クロックジッタークリーナーのコンフィギュレーションが終わると、クロックジッタークリーナーはコンフィギュレーションパラメータに従って、内部の PLL のロックを始め、ロックが完了すると Loss Of Lock Bar (LOLB) というピン出力が high に遷移する。自律型制御機構では LOLB をモニターしておき、これが high に遷移したことを確認して、PP ASIC や FPGA に安定した LHC CLK を供給できていると判断し最後のステップに進む。最後のステップでは、PP ASIC や DAC のコンフィギュレーションと、GTX トランシーバの TX 部分のリセットを行う。一つの PP ASIC に書き込むパラメータは 224 ビットで、PS board ごとに 8 つの PS board が搭載されているので、計 $224 \text{ bit} \times 8 = 224 \text{ Bytes}$ である。PP ASIC controller というモジュール (図 3.4 参照) が SPI プロトコルを用いて、PP ASIC にパラメータを書き込んでいく。PP ASIC へのパラメータの書き込みが完了すると、今度は読み出しを行なって、書き込んだ値と一致しているかどうかをチェックする。加えて、PP ASIC 内の PLL がロックされているかを確認し、全て問題なくコンフィギュレーションが完了していれば、FPGA のレジスタとして用意された、reconfiguration done のフラグを high にする。SL はこの reconfiguration done のフラグを光リンクを経由して読むことによって各 PS board の状態をいつでもモニターすることができる。

3.3 モニター・コントロールパス

ここでは、TGC 検出器エレクトロニクスの各モジュールのモニター・コントロールパスについて述べる。図 3.5 に TGC エレクトロニクスのモニター・コントロールパスの概要図を示す。SL 上の Zynq MPSoC の PS (CPU) に実装された Linux にはイーサネットを通じてアクセスする。SL 上の FPGA レジスタ空間には、Zynq MPSoC の PS(CPU) から、Aurora 64B66B に実装された AXI Chip2Chip 通信等を経由して、アクセスする。

SL と PS board (TAM) は光ファイバーで接続されており、PS board (TAM) の FPGA レジスタ空間には、SL の GTY 及び PS board(TAM) の GTX を介してアクセスする。PS board (TAM) の QSPI フラッシュメモリも同様に光通信を介して、ビットバンキングを用いてアクセスする。また、PS board の PP ASIC には後で述べる自律型制御機構を用いてモニター・コントロールを行う。

JATHub の Zynq SoC の PS (CPU) には、光イーサネットを介してアクセスする。また PL に実装された FPGA レジスタ空間には、JATHub の PS を介してアクセスもしくは、TAM から VME backplane を介してアクセスする。JATHub の QSPI フラッシュメモリには、TAM から VME backplane を介してアクセスする。

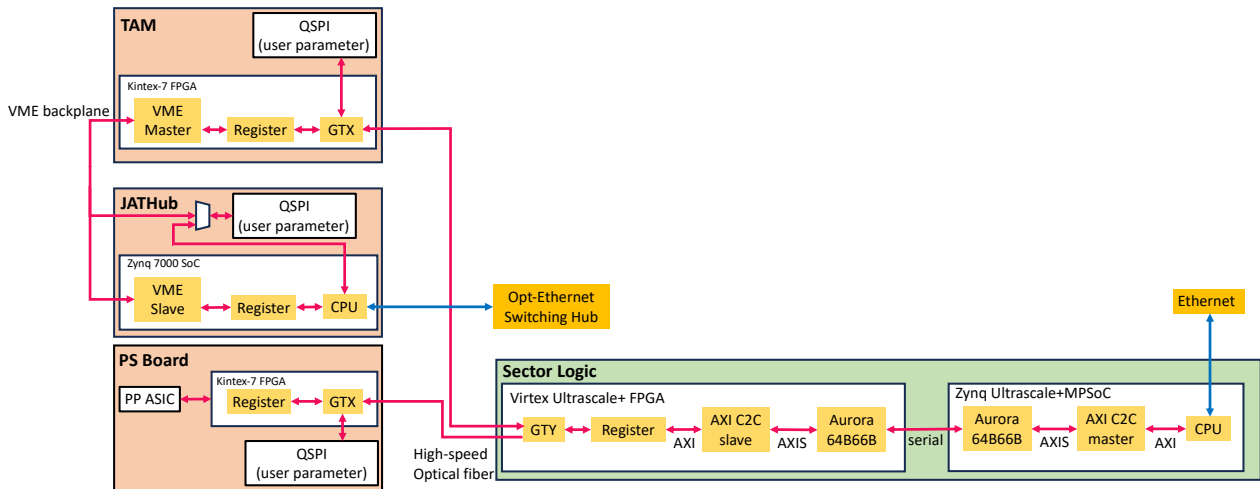


図 3.5: モニター・コントロールパスの概要。青線がイーサネットを介したアクセス、赤線がその他のパスを示している。SL の Zynq Ultrascale+ MPSoC 及び JATHub の Zynq 7000 SoC 上の CPU にはイーサネットを介してアクセスする。SL の Virtex Ultrascale+ FPGA には Zynq Ultrascale+ MPSoC からアクセスする。PS board や TAM の FPGA 及び QSPI フラッシュメモリには、高速光通信を経由して、SL からアクセスする。JATHub の PL 及び QSPI フラッシュメモリには、JATHub の PS からのパスと、TAM を経由した VME backplane からのアクセスが可能である。

3.3.1 SL のモニター・コントロール

図 3.5 に示すように、Zynq MPSoC PS の Linux から、PL 上に実装された AXI Chip2Chip に AXI プロトコルを用いてアクセスする。具体的には、Linux のルートファイルシステム内の/dev/mem が提供するキャラクターデバイスからアプリケーションから開き、PL モジュールが配置されている物理アドレスに直接読み書きを行う方法を取っている*2。図 3.6 に CPU 側から見たアドレス空間と FPGA のレジスタ空間の対応関係を示す。PS と AXI プロトコルを用いて接続される、PL 上のモジュールは、それぞれ個別のアドレスが CPU のアドレス空間に割り当てられるが、AXI Chip2Chip Master モジュールは、アドレス 0x80030000 から 0x80040000 までの 64 KB 分の範囲に割り当てられている。ここで指定されたアドレスはそのまま、SL の FPGA 側の AXI Chip2Chip Slave で指定されるアドレスとなり、よって CPU からアクセスできる SL の FPGA のレジスタも 64 KB 分である。

SL 上の Virtex Ultrascale+ FPGA 上のモジュールのモニター・コントロール線は MPSoC 上の PL を経由して、Virtex Ultrascale+ FPGA のレジスタ空間に接続され、CPU から Virtex Ultrascale+ FPGA レジスタをモニター・コントロールすることで全てのモジュールのモニター・コントロールを行う。例えば、とあるモジュールのリセットを行う場合は、その線に対応するレジスタを 0x0 から 0x1 に遷移させれば (active high の場合) リセットが完了するような設計になっている。このような設計のもとでは、CPU から Virtex Ultrascale+ FPGA レジスタまでのパスは変えることなく、新しくモニター・コントロール線を導入することが可能となる。PS board のモニター・コントロールの際にも、とある PS board のモニターした値を格納しておくレジスタや、PS board に送る値を格納しておくレジスタが Virtex Ultrascale+ FPGA に用意されており、それらを CPU から MPSoC を経由して、モニター・コントロールするような設計となっている。

*2 他にも PL モジュールを User space I/O (UIO) として登録し、アプリケーションから UIO ドライバを開いてアクセスする方法もある。

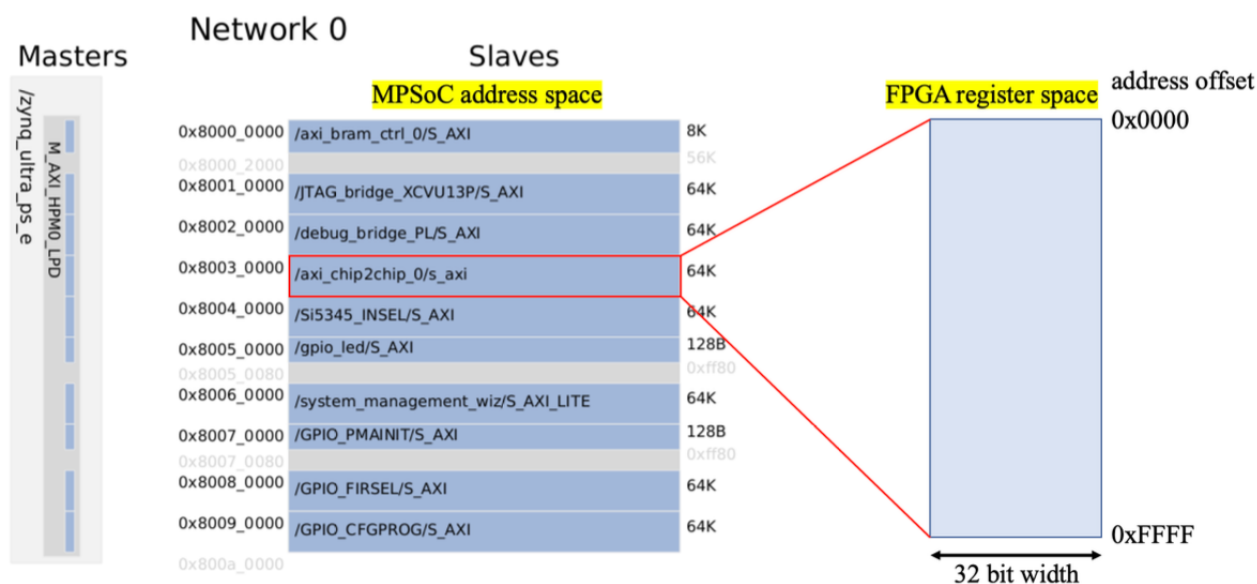


図 3.6: CPU 側から見たアドレス空間と FPGA のレジスタ空間の対応関係 [3]。

3.3.2 光通信を用いた、PS board のモニター・コントロール

Link	Word	31...24	23 ...16	15...8	7...0
Downlink PS RX SL TX	0	0[31...28], TTC Reset[27], Test pulse trigger[26], BCR[25], 0[24]	0[23...20], BCID[19:8]	K28.5[7:0]	
	1	0[31...20], DAC select[19...18], SCL[17], SDA[16]	0[15], CS[14...4], SPIRST[3], SIRSTB[2], SCLK[1], SDI[0]		
	2	address[31:16]	command[15:0]		
	3	FPGA Test pulse pattern number[31:16]	data[15:0]		
	4	Footer (0x0e0d)	full board ID		

表 3.1: SL と PS board 間のデータフォーマット (Down link)。

表 3.1 と表 3.2 に SL と PS board 間の光通信のデータフォーマットを示す。表にあるように、各行の 32 ビットの単位を word と呼び、5 つの word を合わせて packet と呼ぶ。SL と PS board 間の光通信は、1 本の Down link (表 3.1) と 2 本の Up link (表 3.2) の計 3 本である。SL 側には Xilinx 社が提供する GTY[8]、PS board 側にも同じく Xilinx 社の提供する GTX[7] という高速シリアル通信のモジュールが実装されている。SL と PS board 間の高速シリアル通信は 8b/10b を用いて実装される。表 3.1 と表 3.2 に示すように、SL と PS board 間のデータフォーマットでは、1 word = 32 bit である。32 ビットのデータを GTY に入力すると、8b/10b にしたがって 40 ビットにエンコードされた後、シリアライズされて光通信へと載せ替えられる。同様にそれを受けた側もシリアライズされたデータをパラレルデータ (40 ビット) に変換し、8b/10b にしたがって 32 ビットデータへとデコードする。SL と PS board 間の高速シリアル通信は、1 word を 200.395 MHz 毎に送受信する。以上より、SL と PS board 間の高

Link	Word	31...24	23 ...16	15...8	7...0
Downlink PS RX SL TX	0	Monitoring data and control signal read back[31:8]			K28.5[7:0]
	1	32-bit hit data			
	2	32-bit hit data			
	3	32-bit hit data			
	4	32-bit hit data			
Downlink PS RX SL TX	0	Monitoring data and control signal read back[31:8]			K28.5[7:0]
	1	32-bit hit data			
	2	32-bit hit data			
	3	32-bit hit data			
	4	32-bit hit data			

表 3.2: SL と PS board 間のデータフォーマット (Up link)。

速シリアル通信は、ラインレート 8.016 Gbps^{*3}である。また、3.4 節と 3.5 節で述べるように、SL と PS board 間の通信は、タイミング信号とヒットデータを固定位相で送受信するために、GTX と GTY 及びその周辺のモジュールは固定位相での実装がされている。

Down link には主に PS board のコントロールとタイミング信号分配のためのデータがパッキングされている。タイミング信号分配に関するデータは、Word 0 の 31 ビットから 8 ビットに格納されている、TTC Reset、Test Pulse Trigger、BCR、BCID と Word 3 の 31 ビットから 24 ビットに格納されている FPGA test pulse pattern number である。これらについては 3.4 節と 3.5 節で詳しく述べる。PS board のコントロールに関するデータは、主にレジスタとビットバンギング (DAC、QSPI フラッシュメモリ) のコントロールに分けられる。まずは FPGA レジスタに関して述べる。対応するデータは、Word 2 の、address と command 及び Word 3 の 15 ビットから 0 ビットの data である。PS board のレジスタ空間のアドレスとデータが address, data に対応しており、command が読み出し・書き込みの 2 種類に対応している。PS board のレジスタ空間はアドレス幅 16 ビット、データ幅 16 ビットの実装がされている。次にビットバンギング用のデータフィールドについて述べる。SL から DAC をモニター・コントロールする際には、I²C プロトコルにしたがって、Word 1 の 17 ビットから 16 ビットを SCL と SDA に使用し、19 ビットと 18 ビットでどの DAC を制御するかを選択する。また、QSPI フラッシュメモリをモニター・コントロールする際には、4-wire SPI プロトコルにしたがって、Word 1 の 15 ビットから 0 ビットまでの部分を制御する。残りの K28.5 は 8b/10b で定義されたコマワードと呼ばれるものである。固定値である Footer(0x0e0d) は、PS board 側で、SL と PS board とのリンクが正常に確立しているかのチェックのために現在は使用している。full board ID は PS board の ID を表しており、これは SL から各 PS board に配布され、PS board の FPGA 上のレジスタに格納される。

Up link はヒットデータとモニター・コントロールに関するデータが格納されている。ヒットデータに関しては、2 本の Up link それぞれに対して 4 word 分、つまり 128 ビット分が用意されている。PS board はヒット信号の有無に関わらず、全てのヒットビットマップを SL に転送する使用になっているので、1 台の PS board が受け持つ TGC チャンネルは 256 チャンネルであるから、40.079 MHz ごとに 256 チャンネルを SL へ転送できるようになっている。モニター・コントロールに関しては、Word 0 の 31 ビットから 8 ビットを有効に活用し、図 3.7 に示すように、状況に応じてモードを切り替えるような実装を行っている。実験中に用いるデータフォーマット (本番用フォーマット: 図 3.7a) では、xADC、ADC、DAC、QSPI、Si5395、PP ASIC、FPGA の SEM など各素子のモ

*3 = 40.079 MHz × 5 words × 32 bit/word × 10/8

モニター値や FPGA レジスタから読み出した値が、PS data、Data type の部分を用いて分けられて送信されるようになっている。特に各素子のモニター値は常に SL に送信する仕様になっており、SL が PS board の状態を特別なコマンドを送らなくてもモニターできるようになっている。SL から PS board のボード上の素子 (QSPI フラッシュメモリ等) をビットバンギングを用いて制御する際に用いるデータフォーマット (ビットバンギング用フォーマット: 図 3.7b) では、SDA、SDO の部分に I²C や 4-wire SPI における返り値が格納されている。SL と PS board 間の接続を確認及び識別を行うためのデータフォーマット (接続確認用フォーマット: 図 3.7c) では、PS board の ID と、リンク番号が SL へ送信されるようになっている。また全てのモード共通して BCID が SL へ転送されるようになっている。これは SL から受け取った BCID をそのまま返しているのではなく、SL から受け取った BCR を元にして、PS board 自身でカウンターを回して生成した BCID である。これを SL 側に送信し、SL 側が BCID が正しくカウントアップしているを見ることで、SL と PS board 間のリンクが安定的に動作していることを確かめることができる。また、それぞれの PS board における BCID の値を比較することで、3.5 で述べるように、SL の各 PS board に対する可変遅延を決定することができる。3つのデータフォーマット (本番用フォーマット、ビットバンギング用フォーマット、接続確認用フォーマット) の切り替えは、SL から PS board の FPGA 上に用意された所定のレジスタの値を変えることによって行う。

Normal

SFP+ 0

Word (32 bit)	Forth byte [31:24]							Third byte [23:16]		Second byte [15:8]	First byte [7:0]
Word-0	Status bit [1:0] = 2'b11	xADC read	ADC read	DAC read	DAC write	QSPI status	Si5345 Lock	PP ASIC status	SEM status [2:0]	BCID [11:0]	Comma = 0xbc

SFP+ 1

Word (32 bit)	Forth byte [31:24]							Third byte [23:16]		Second byte [15:8]	First byte [7:0]
Word-0	PS data [7:0]							Data type [3:0]		BCID [11:0]	Comma = 0xbc

(a) 本番用フォーマット。実験中に用いる。

Bit banging

SFP+ 0

Word (32 bit)	Forth byte [31:24]							Third byte [23:16]		Second byte [15:8]	First byte [7:0]
Word-0	Status bit [1:0] = 2'b10	xADC read	ADC read	DAC read	DAC write	2'b0		SDA [1:0]	2'b0	BCID [11:0]	Comma = 0xbc

SFP+ 1

Word (32 bit)	Forth byte [31:24]							Third byte [23:16]		Second byte [15:8]	First byte [7:0]
Word-0	SDO [10:0]							1'b0		BCID [11:0]	Comma = 0xbc

(b) ビットバンギング用フォーマット。PS board 上の素子 (QSPI フラッシュメモリ等) をビットバンギングを用いて制御する際に用いる。

Connection check

SFP+ 0

Word (32 bit)	Forth byte [31:24]							Third byte [23:16]		Second byte [15:8]	First byte [7:0]	
Word-0	Status bit [1:0] = 2'b01	PS board ID [10:0]							Link number = 1'b0	2'b0	BCID [7:0]	Comma = 0xbc

SFP+ 1

Word (32 bit)	Forth byte [31:24]							Third byte [23:16]		Second byte [15:8]	First byte [7:0]	
Word-0	Status bit [1:0] = 2'b01	PS board ID [10:0]							Link number = 1'b1	2'b0	BCID [7:0]	Comma = 0xbc

(c) 接続確認用フォーマット。SL と PS board 間の接続を確認及び識別を行う際に用いる。

図 3.7: Up link(PS board が送信側、SL が受信側) における、モニター・コントロールに関するデータ。表 3.2 の word 0 に相当する。3.7a、3.7b、3.7c の 3つのモードを切り替えて通信を行う。

3.3.3 TAM のモニター・コントロールについて

詳しくは 4.1 節で述べるが、基本的な設計としては PS board と同じである。

3.3.4 JATHub のモニター・コントロールについて

JATHub の PS (CPU) には光イーサネット経由でアクセスする。JATHub の PS から AXI プロトコルを用いて、JATHub の PL に実装されたモジュールのモニター・コントロールを行う。SL と同様に、JATHub の PL 上のモジュールは、それぞれ個別のアドレスが CPU のアドレス空間に割り当てられており、Linux から個別のアドレスにアクセスすることで、レジスタや QSPI フラッシュメモリを含めた JATHub の PL 上のモジュールにアクセスすることができる。特にレジスタは JATHub の PS から読み書きができるものと、VME backplane を経由して TAM から読み書きができるものが用意されている。

また QSPI フラッシュメモリに関しては、外部から制御するパスを設けてある。ボード上のジャンパーピンを変更することで、14 pin header か VME backplane (TAM を経由) かを変更することができる。

3.4 固定位相でのタイミング信号分配・受信と位相合わせ

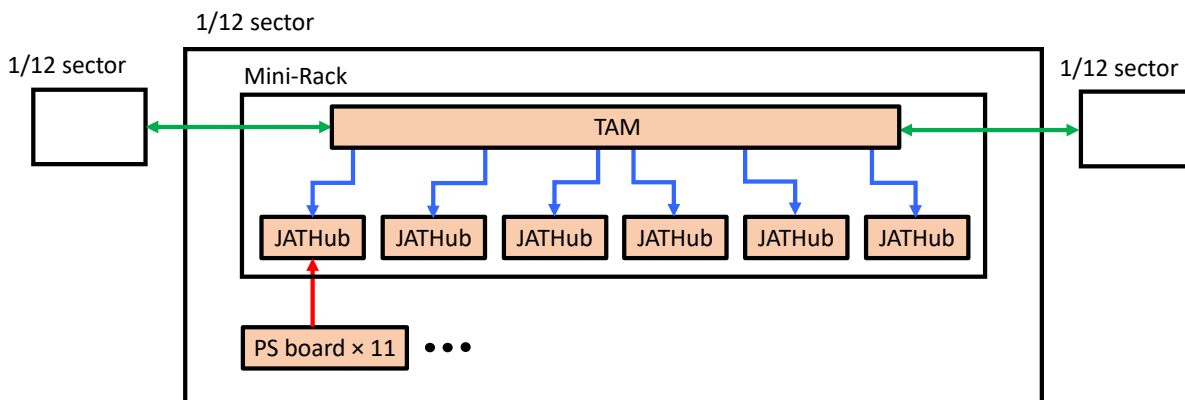


図 3.8: タイミング信号分配と位相分配の概要図。赤線が PS board から JATHub へのクロック送信、青線が TAM から JATHub への基準クロック送信、緑線が TAM 同士のクロック送受信を示している。

2.2.3 節で述べたように、正確な BCID を行うためには、1 ns 以内の精度で 1434 台の PS board の位相を一致させる必要がある。TGC 検出器エレクトロニクスでは、固定位相 (reboot やリンクの再確立等で位相が変化しない) でのタイミング信号分配・受信を行い、その上で、1434 台の PS board の位相合わせを行い、1434 台の PS board の位相を十分な精度 (< 1 ns) で一致させる。

図 3.8 にタイミング信号分配と位相測定の概要図を示す。図中の線はクロックパスを表す。簡単のため、SL から、PS board、TAM へのクロック分配パスは省略してある。TGC エレクトロニクスにおける位相合わせの概要は、大きく 2 つの手順に分けられる。一つは、TAM を用いて、異なる 1/12 セクター間の位相差を吸収し、全ての JATHub に同一の基準クロックを分配すること。もう一つは、その基準クロックを元にして、各々の JATHub が自身に接続された 11 台の PS board の相対位相を測定し、位相を合わせるために必要な遅延パラメータを決定することである。事前に TAM 間で位相を合わせておいて、全ての JATHub に同一の基準クロックを分配することで、JATHub は目の前の PS board の位相を合わせることを独立に行うことで全 1434 台の PS board の位相が揃うことになる。

タイミング信号分配及び位相合わせの手順は

1. SL から光ファイバーを通じて PS board と TAM に TTC 信号を分配する

2. PS board が固定位相で TTC 信号を再構成し、JATHub に Cat 6 ケーブルを通じて送信する (図 3.8 内赤線)
3. TAM が固定位相で TTC 信号を再構成し、隣り合う TAM 同士で位相合わせ (Coarse, Fine) を行い、全 24 台の TAM の位相が揃った状態にする。(図 3.8 内緑線)
4. TAM が基準クロックとして、JATHub に LEMO ケーブルと VME backplane を通じて送信する (図 3.8 内青線)
5. JATHub が TAM から受けた基準クロックを元にして、PS board の位相測定を行い、ボードごとの delay parameter (Coarse, Fine) を決定する
6. SL からビットバンギングを用いて、各 PS board の QSPI フラッシュメモリに先ほど求めた delay parameter を書き込む
7. PS board のソフトリセット (自律型制御機構を用いて PS board を再コンフィギュレーション) する。

である。最後にもう一度、JATHub による位相測定を行い、位相合わせが十分な精度で達成されているかを確認する。

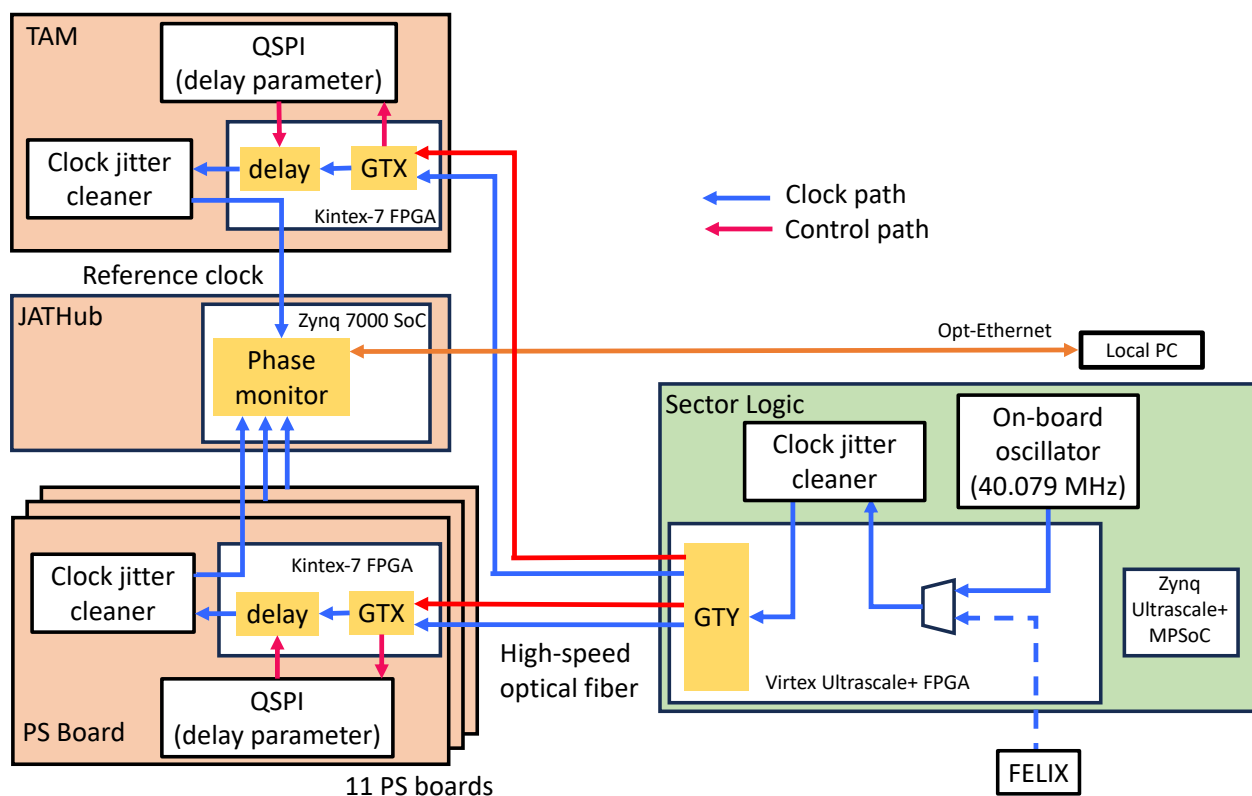


図 3.9: タイミング信号分配と位相分配の概要図。赤線が delay parameter のコントロールパス、青線がクロックパス、オレンジ色の線がイーサネットを経由した JATHub へのアクセスを示す。

図 3.9 に 1/12 セクター内の位相合わせのうち 1 台の JATHub に関する部分の概要図を示す。実際には 1/12 セクター内で 6 台の JATHub を用いて 62 台の SP board の位相を合わせるが、図 3.9 で示した構造が繰り返し存在し、それぞれ独立に位相合わせを行えば良いのでここでは全体は示さない。

SL の TTC 信号は、セレクターを SL FPGA 内部に設置し、FELIX から受けた TTC 信号を用いる場合と、SL のオンボード発振器のクロックを用いる場合を選択できるような仕様になっている。実際のオペレーションでは、SL は FELIX を通じて TTC 信号を受信し、それを PS board や TAM に分配する。一方で、5.1 で述べる 3 つ目の目

的の、ケーブルの配線ミスやハードウェアの故障を見つけるためのスタンドアロンテストベンチの作成という観点でみると、SL が仮想的なタイミング信号を内部で生成し、分配を行えるような機能が必要である。本デモンストレーションでも図 3.8 にあるように FELIX から TTC 信号を受信するパスは用いずにオンボード発振器からのクロックをソースとしている。このような仕様で TTC 信号の受信を行った場合、SL の power cycle(電源 on/off) のよって、PS board や TAM の相対位相は変化せず、JATHub の位相測定の結果も SL の power cycle によって変化しない。

3.4.1 固定位相でのタイミング信号分配・受信

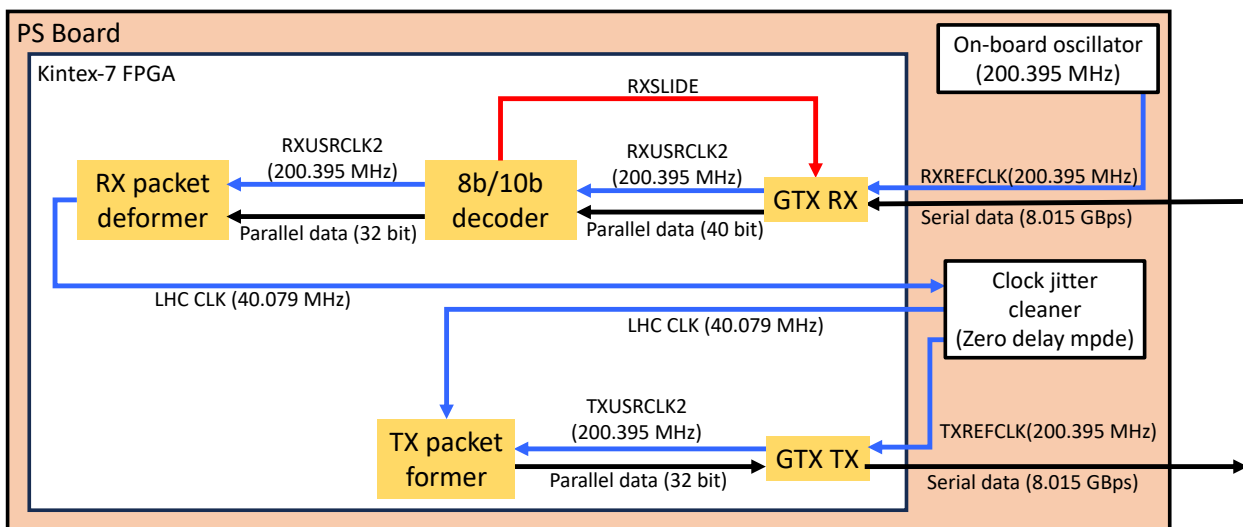


図 3.10: PS board における GTX 周りのクロックパス。

TGC 検出器エレクトロニクスでは、SL と各 PS board 及び TAM は光ファイバーで接続され、8b/10b を用いたシリアルデータ転送が行われる。8b/10b を用いたシリアルデータ転送においては、データにクロックの情報を乗せて同時に分配することとなる。図 3.10 に PS board に実装した GT トランシーバ周りの実装を示す。固定位相でのタイミング信号分配を行うために、特別な実装が必要な箇所は GT トランシーバ (GTX, GTY) と受け取ったデータを解釈するモジュール (図 3.10 中 RX packet deformer)、PS board 上のクロックジッタークリーナ、データの PACKET 化を行うモジュール (図 3.10 中 TX packet former) である。

ここでは実装の詳細には立ち入らないが、付録 A に示すような GT トランシーバ及び周辺のモジュールの特別な実装を、SL や TAM の固定位相が必要な部分にも同様にを行うことにより、TGC エレクトロニクス全体で固定位相でのタイミング信号送信・受信が達成されることになる。TAM に関する実装は今まで述べた PS board に関する実装と全く同じである。SL に関しては、3.5 節で、PS board から固定位相でのデータ受信に関する説明を行う。

また、3.4.2 で述べるように、全 1434 台の PS board の中には、最大で位相差が 25 ns を超える場合もあるので、40.049 MHz LHC clock をだけではその位相差を吸収することはできない。よって BCR や TPT、TPP といったタイミング信号の、25 ns を超える伝播時間差の測定のために、BCR の立ち上がりに同期した、202.42 kHz クロック (1 UI = 198 BC) も各 PS board 及び TAM は再構成することにした。この場合、1 UI が 5 μ s となるため、最大で、5 μ s の位相差を測定することができる。BCR は 3564 バンチに一回発行されるので、BCR と 202.42 kHz クロック (1 UI = 198 BC) を立ち上がりを合わせれば、その後は周期的に必ず 2 者の立ち上がりは一致するので、この 202.42 kHz クロックの立ち上がりを測定し、合わせることで 25 ns 以上の位相差にも対応できる。ここで再構

成した 202.42 kHz クロックは、BCR と位相関係が固定されているため、今まで述べた実装で固定位相での受信が達成されている。

3.4.2 タイミング信号の位相合わせ

次にタイミング信号の位相合わせの手続きについて説明する。各 PS board が再構成したタイミング信号 (40.049 MHz BC clock, BCR, TPT, TPP) にはスキューが存在する。このスキューは各 PS board と SL を結ぶ光ファイバーの配線用の違いや 5.3.2 に述べるように IC の個体差に起因するスキュー (part-to-part skew) に起因する。配線長の観点で述べると、例えば、全 1434 台の PS board にうち最も配線長が異なる個体は、最大で 50 m 異なるので、クロックスキューで述べると数十 ns の位相差が現れることになる。また、一方で、正しい BCID のためには、全 1434 台の PS board の位相を 1 ns より良い精度で合わせる必要がある。以上より、タイミング信号の位相合わせのためには、Coarse delay (25 ns 幅) と Fine delay (1/56 ns 幅) の 2 種類の可変遅延を PS board や JATHub、TAM に実装するとともに、それらの可変遅延を用いて、位相を測定する機能を JATHub や TAM に実装した。

可変遅延回路の実装

まず、可変遅延の実装について述べる。



図 3.11: Block RAM(BRAM) の IP デザイン [9]。IP デザインで simple dual port を選択し、port A を書き込み、port B を読み出しに設定している。今回は、BRAM を用いて、40.079 MHz LHC clock に同期した、25 ns 幅の可変遅延を実装しているため、port A 及び port B に用いるクロックは共通の 40.079 MHz LHC clock を設定している。例えば、202.42 kHz クロックを遅延させる場合は、遅延前の 202.42 kHz クロックを dina に入力し、addr_a と addr_b の差が遅延数に対応、遅延後の 202.42 kHz クロックが dout_b から出力される。

Coarse delay (25 ns 幅) の可変遅延に関しては、Xilinx 社の提供する Block RAM(BRAM) IP を用いた。つまり、variable length shift register として BRAM を用いたことになる。図 3.11 に BRAM の IP デザインを示す。simple dual port に設定し、PORT A を書き込み、PORT B を読み出しに用いる。その場合、clka と alkb に 40 MHz クロックを接続し、dina に 202.42 kHz クロックを入力する。その上で、addr_a と addr_b を 40 MHz クロックに同期し

た形で、40 MHz クロックの立ち上がり毎に +1 していく。そうすると、 $\text{addrb-addra tick} (= 25\text{ns} \times \text{addrb-addra})$ だけ位相が遅れた 202.42 kHz クロックが `doutb` から出力されることになる。しかし、BRAM のアウトプットは FPGA 上ではデータ線としての扱いとなり、クロック線として扱われない。データ線として扱うよりもクロック線として FPGA に認識させた方が、FPGA 全体に低スキューで分配するようになるので、BRAM を用いて coarse delay させた 202.42 kHz クロックは、一度 Global Clock Simple Buffer (BUFG[6]) と呼ばれるプリミティブに通してから FPGA 全体に分配する。BUFG に通すと、信号をグローバル配線リソースに乗せることになる。

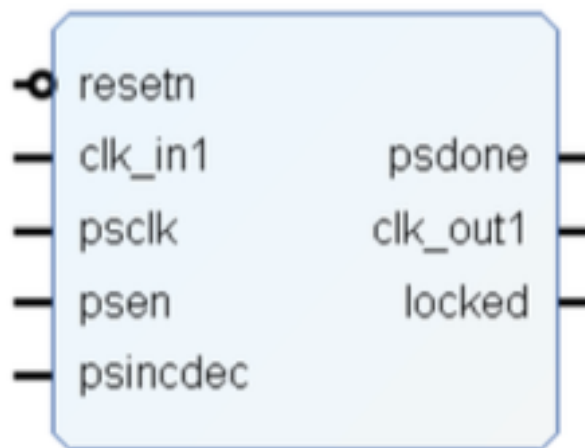


図 3.12: clocking wizard(MMCM) の IP デザイン [10]。clk in1 がクロックの入力ポート、clk out1 が遅延後のクロック出力ポートになり、locked が clocking wizard がクロックのロック状況を示し安定したクロックを出力できているかを示している。psclk は Dynamic Phase Shift に用いられるシステムクロックを入力するためのポート、psincdec はクロックの位相を遅らせるのか、早めるのかを選択するポートであり、“1”の場合には出力クロックの位相が進み、“0”の場合には出力クロックの位相が遅れる。psen に psclk の 1 tick 以上のパルスを入力すると Dynamic Phase Shift 機能は、psincdec の方向にしたがってクロックの位相をとある刻み幅で変化させる。Dynamic Phase Shift 機能が位相のシフトが完了すると、psdone を high (= “1”) にあげてシフト完了である。

Fine delay (1/56 ns 幅) の可変遅延には、Xilinx 社から提供されている Clocking Wizard IP コア [10] における、“Dynamic Phase Shift”と呼ばれる機能を用いる。Clocking Wizard の IP デザインを 3.12 に示す。Clocking Wizard には MMCM (Mixed-Mode Clock Manager) と PLL (Phase Locked Loop) のクロック生成機能が存在するが、Dynamic Phase Shift を使うことができるのは、MMCM なので、今回は MMCM を用いている。図 3.12 において、clk in1 がクロックの入力ポート、clk out1 が遅延後のクロック出力ポートになり、locked が clocking wizard がクロックのロック状況を示し安定したクロックを出力できているかを示している。また、psclk、psen、psincdec、psdone は Dynamic Phase Shift 機能をアクティブにした際に現れるポートである。psclk は Dynamic Phase Shift に用いられるシステムクロックを入力するためのポートで、これは clk in1 を同じクロックで構わない。また psincdec はクロックの位相を遅らせるのか、早めるのかを選択するポートであり、“1”の場合には出力クロックの位相が進み、“0”の場合には出力クロックの位相が遅れる。psen に psclk の 1 tick 以上のパルスを入力すると Dynamic Phase Shift 機能は、psincdec の方向にしたがってクロックの位相をとある刻み幅で変化させる。Dynamic Phase Shift 機能が位相のシフトが完了すると、psdone を high (= “1”) にあげてシフト完了である。ここで先ほど述べた位相シフトの 1 ステップ(刻み幅)は、MMCM において、安定した出力をもたらす、電圧制御発振器 (Voltage Controlled Oscillator; VCO) の周波数 (F_{VCO}) を用いて、 $1/(56F_{VCO})$ と表される。ここで F_{VCO} は入

クロックの周波数によって自動的に計算される値であり、例えば入力を 40 MHz、出力を 40 MHz と設定すると、 $F_{VCO} = 1 \text{ GHz}$ となり、したがって、遅延幅は、 $1/(56F_{VCO}) \approx 15 \text{ ps}$ となる。

位相測定回路の実装

次に位相測定の実装について述べる。この機能も先ほどと同様に、Coarse delay (25 ns 幅) と Fine delay (1/56 ns 幅) の 2 種類に関して実装する必要があるが、基本の仕組みは同様のため、Fine delay (1/56 ns 幅) についてのみ説明する。

図 3.13 に位相測定のコセプトを示す。まず基準クロックの立ち上がりで、測りたいクロックをサンプリングする (このサンプリングは D フリップフロップを用いて行われる)。次に先ほど述べたクロック位相制御機構を用いて、基準クロックの位相を 1/56 ns だけ進める。そしてこの遅延後の基準クロックを用いて再び、測りたいクロックをサンプリングする。この手順を fine delay (1/56 ns 幅) であれば 1400 回 (= $25 \text{ ns} / (1/56 \text{ ns})$) 繰り返すことで、40 MHz クロックの位相を 2π 全てスキャンする。仮に、測りたいクロックに対して、基準クロックの位相が $\pi/2$ だけ進んでいたとすると、その時は、基準クロックで測りたいクロックをサンプリングすると、必ず low と判定される。一方で、測りたいクロックに対して、基準クロックの位相が $\pi/2$ だけ遅れていたとすると、基準クロックで測りたいクロックをサンプリングすると、必ず high と判定される。また、測りたいクロックと基準クロックの位相が一致している、もしくは、ちょうど π ズレていたとすると、とある確率で high と判定され、とある確率で low と判定されるといった結果が得られる。したがって各位相関係ごとに十分な回数のサンプリングを行い、その結果をプロットすると、クロックの波形を得ることができる。今回の実装では、JATHub 及び TAM において、各位相関係ごとに 1000 回のサンプリングを 1000 回行い、その平均値から、結果をプロットし、エラーバーとしてその分散をつけることにした。また JATHub の場合はそのようにして得られた結果を PS に実装された Linux で解析して、各 PS board に必要な delay parameter を決定している。1 台の JATHub には最大 11 台の PS board が接続されるため、位相測定を 11 台並列行うことができるようなファームウェア、ソフトウェアを実装した。一方で TAM の場合は、FPGA にファームウェアレベルで実装された論理で、自身のクロックにフィードバックをかけて位相を合わせるところまで行っている。TAM に関しては詳しくは 4.2 節で詳しく述べる。

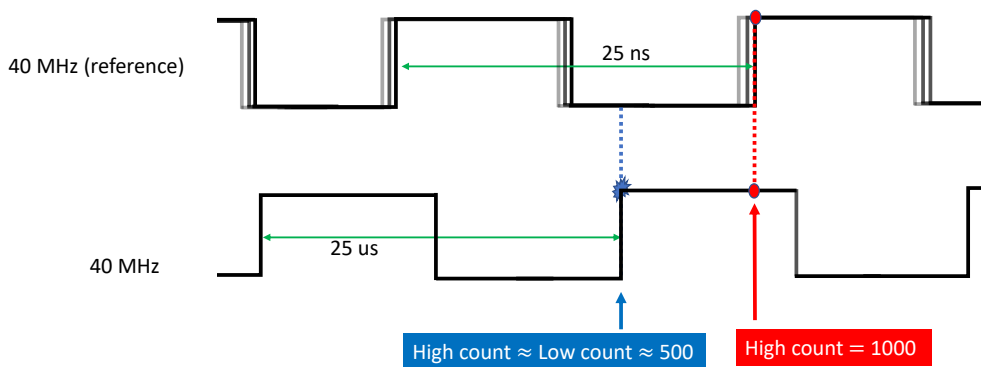


図 3.13: 位相測定のコセプト図。

JATHub における位相測定の結果を図 3.14 に示す。この図は、JATHub に基準クロックとして、TAM から 40.079 MHz を受け取り、とある PS board の 40.079 MHz クロックを測定している。図中の青線は、赤線を測定した後もう一度位相測定を行ったもので、緑線は JATHub を reboot した後に PS board や TAM のセットアップは変えずに再測定を行った結果である。これを見ると、確かに PS board のクロックの位相が測定できており、また再測定や JATHub の reboot に対する再現性が取れていることがわかる。また、この際、実際に PS board のクロック

クをオシロスコープでプローブし、この測定自体の検証も行い、PS board のクロックの相対位相が測定できていることも確かめた。

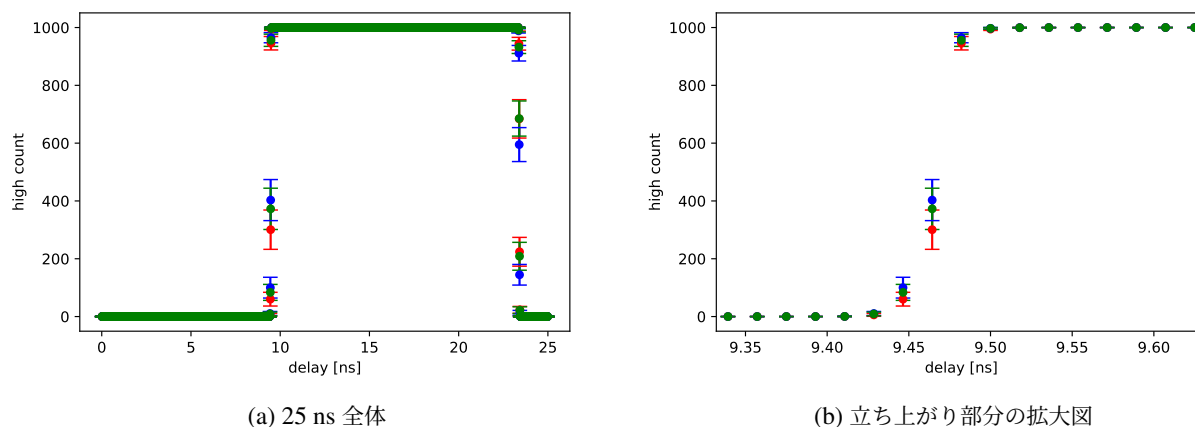


図 3.14: JATHub における位相測定の結果。青線は、赤線を測定した後にもう一度位相測定を行ったもの、緑線は JATHub を reboot した後に測定を行なったものである。

以上の議論は Fine delay (1/56 ns 幅) の delay parameter 決定に関する議論であったが、全く同様にして、Coarse delay (25 ns 幅) の delay parameter も決定することができる。その場合は、40.079 MHz クロックではなく、BCR の立ち上がりに同期した 202.42 kHz クロックを、測りたいクロックや基準クロックとして使用する。また、 2π スキャンする際は 25 ns 幅で、198 ステップ行うことで、202.42 kHz クロックの位相を測定することができる。よって FPGA の実装には、セレクターを用いて、クロックの周波数や遅延幅をスイッチすることで、大まかな機能はそのままに fine delay (1/56 ns 幅) での位相測定と、coarse delay (25 ns 幅) での位相測定をスイッチするような実装を行った。注意点としては、JATHub における Fine delay (1/56 ns 幅) での位相測定の際には、TAM から各 JATHub へ等長の LEMO ケーブルを通して、同位相で、40.079 MHz クロックを基準クロックとして分配している。一方で、BCR に同期した 202.42 kHz クロックの分配には VME バックプレーンを経由して、等長ではない分配の仕方をしているので、この位相差を吸収するために、各 JATHub は受け取った 202.42 kHz クロックを LEMO から受け取った 40.079 MHz クロックでラッチすることとした。なお、VME バックプレーンの端から端までで生じる最大の位相差は、数 ns 以内に収まることを確かめているので、この方法を用いて、同位相での 202.42 kHz クロックの分配も達成される。

このようにして、各 PS board の delay parameter (Coarse, Fine) を決定し、その delay parameter を各 PS board の QSPI フラッシュメモリに埋め込む。3.2 節で述べるように各 PS board はブートする際に自律型制御機構が働き、それらのパラメータを用いて自身をコンフィギュレーションする。その結果、全 1434 台の PS board の位相が揃うことになる。

3.5 リードアウトパス

この節では、読み出しシステムのデモンストレーションについて述べる。図 3.15 に読み出しパスの概要図を示す。黒線が読み出しパス、赤線が読み出しに関わるコントロールパス、青線がクロックパスである。

まずは、コントロールパスについて述べる。Test Pulse Pattern (TPP)、Test Pulse Trigger (TPT)、Bunch Counter Reset (BCR)、Level-0 Accept などがある。実際のオペレーションでは、BCR や L0 Accept は Central Trigger シ

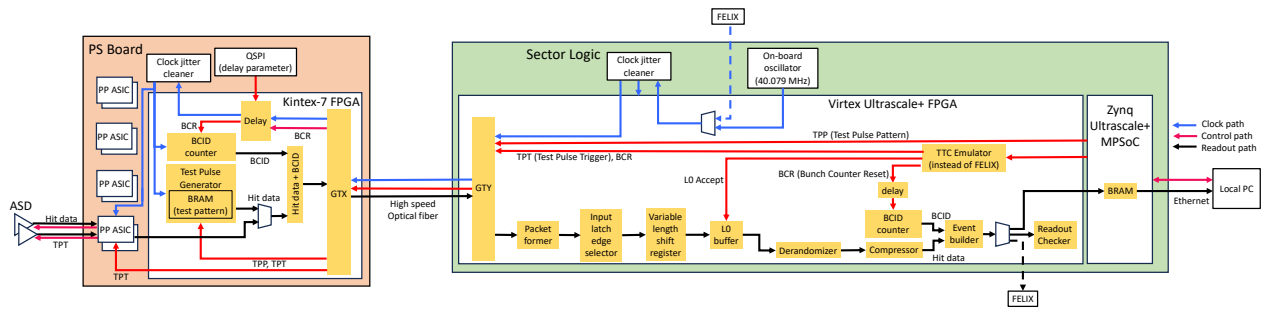


図 3.15: 読み出しパスの概要図。黒線が読み出しパス、赤線が読み出しに関わるコントロールパス、青線がクロックパスである。

システムから FELIX を通して供給される信号である。これに加えて、読み出しパスのテストのために、図 3.15 中の TTC Emulator というモジュールがタイミング信号のマスターを担う。5.4 節で述べる、読み出しシステムのデモンストレーションでは TTC Emulator を用いてこれらのコントロール信号を分配する方法を取っている。LHC クロックに同期した制御パス (TPT や BCR) は、発出するタイミングが LHC クロックのレベルで制御されてなければならない。また、これを受け取る PS board もこれらの制御パスの取り扱いには注意が必要である。具体的には、まず、固定位相でのタイミング信号分配や受信が読み出しパスで一貫して実装されていることが必要である。その場合、例えば SL が TPT を発出して、とあるレイテンシー (L0 latency と呼ぶ) 経過した後、ヒットデータが SL の L0 Buffer まで到着するが、その L0 latency は SL や PS board の reboot に対して変化しない。したがって、TPT の発出から、ちょうど L0 latency だけ経過した後に L0 Accept を発出することによって、テストパルスによるヒットデータを正しく取得することが可能になる。言い換えれば、SL の power cycle や PS board の reboot を行なった後に、固定の L0 latency で狙ったヒットデータが正しく取れているかをチェックすることによって、読み出しパスが固定位相での実装が達成されているかを検証することができる。なぜなら、SL の power cycle や PS board の reboot には、FPGA の再コンフィギュレーションやリンクの取り直し等の位相関係を変えうる全ての要素を含むからである。

次にクロックパスについて述べる。SL の FPGA 上のリードアウトに関わる論理のほとんどは LHC CLK に完全に同期して駆動するが、後で述べるように、PS board からのヒット受信部である GTY トランシーバ及び Packet former(図 3.15) は GTY トランシーバの RXUSERCLK2(3.4 及び図 3.10 参照) で駆動する。Input latch edge selector(図 3.15) で RXUSERCLK2 から LHC CLK への載せ替えが行われ、以降のモジュールは LHC CLK に同期して駆動する。PS board の FPGA 内のリードアウトに関わる論理及び、PP ASIC は、PS board が SL から受信した LHC CLK と完全に同期して駆動する。この際、個々の PS board に最適な delay parameter を用いて、全 11 台の PS board のクロックの位相が合わせられた状態を達成することにより、各 PS board がテストパルスに付与する BCID が一致する。またそれを受け取る SL は、固定位相での受信を実装するとともに、PS board から SL までのファイバー長の差に起因するズレを variable length shift register で吸収する。こうすることによって、独立なファイバー長で接続された SL と PS board においても同一のタイミング、BCID でデータ取得が可能である。

最後に読み出しパスについて述べる。TGC 検出器からのヒット信号は ASD ボードに入り、LVDS 規格の差分信号に形成された後、PS board 上の PP ASIC に入る。PP ASIC で、信号遅延と BCID が行われ、出力が PS board FPGA に入力される。PS board は、ヒットの有無に関わらず、全てのヒット信号を SL と PS board 間のデータフォーマットに沿った形でパッキングし、SL へ送信する。この際、SL から受け取った BCR と LHC CLK を元に自身の FPGA 上でカウンターを回して、ヒット信号に BCID を付与する。ここで付与された BCID はそのまま SL に送信され、SL と PS board 間のリンクの安定性の検証のために使われたり、SL が各 PS board に必要な遅延を決

定する際に用いられる。そして、光信号を通じて SL に送信されたヒットデータは SL の FPGA に入力される。

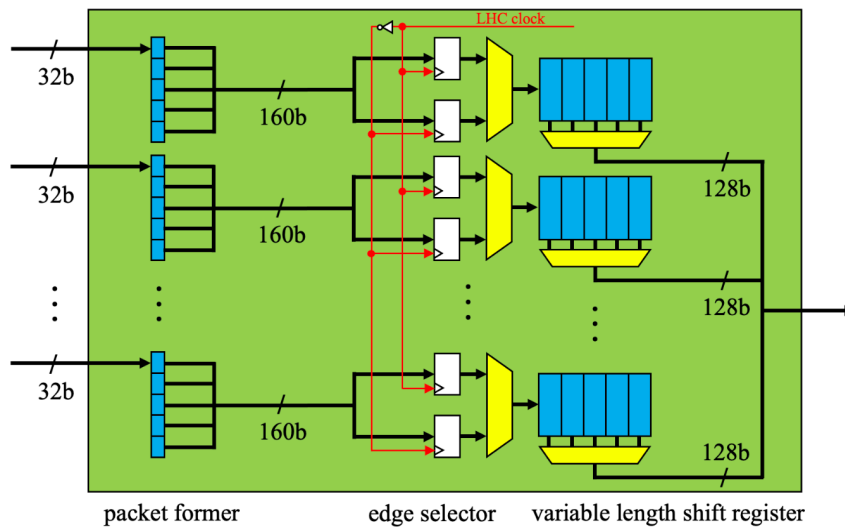


図 3.16: SL のヒット受信部 [3]。

SL のヒット受信部の概要を図 3.16 に示す。まずヒットデータが最初に入るモジュール (GTY は除く) は packet former である。packet former は SL の GTY RX の RXUSRCLK2 で駆動するモジュールである packet former では 32 ビット \times 5 ワードの平行データが 160 ビットの平行データに整形する。次に 160 ビットのデータは input latch selector に入る。ここでは、RXUSRCLK2 から LHC CLK への載せ替えが行われる。SL と PS board 間の通信は固定位相での転送・受信を実装しているため、GTY RX USRCLK2 と LHC CLK の位相関係は、SL から PS board 間のファイバー長にのみ依存している。ほとんどの場合は立ち上がり立ち下りのどちらかで、ラッチしてもデータの受け渡しは安定して行われるが、ファイバー長によっては、USRCLK2 と LHC CLK の立ち上がり同士が一致してしまい、単純に立ち上がりでラッチするという設計だと、データの受け渡しが不安定になり得る。したがって、図に示すように LHC CLK の立ち上がり立ち下りのどちらかでラッチするかを、選べるような設計を行なった。LHC CLK に載せ替えられた後ヒットデータが入るモジュールが variable length shift register である。これは、25 ns 幅の可変遅延回路である。SL と PS board 間のファイバー長は各々の PS board ごとに異なり、40 m から 90 m で分布するため、variable length shift register の長さを変えて全ての PS board からの同じタイミングでのヒット信号を、SL が同じタイミングで受信できるようにする。

次にヒットデータは L0 Buffer に入る。L0 Buffer は LHC CLK で駆動するモジュールである。L0 Buffer の depth は 400 であり、これは L0 latency が $10\mu\text{s}$ であることに対応する。L0 Buffer は、トリガーの処理待ちのためのバッファであり、L0 Accept が発行されると、対応するヒットデータをさらに後段に渡し、発行されないヒットデータは破棄する。トリガー判定され、L0 Buffer から出力されたヒットデータは、Compressor に入り、Zero suppress が行われる。Zero suppress は、ASD ボード (16 チャンネル) ごとに行われる。ヒットのあった ASD ボードのヒットビットマップと、その ASD ボードの SL から見た ID をセットにしてさらに後段に送り、ヒットのなかった ASD ボードに関しては何も送らない。こうして、圧縮されたデータは、Event Builder というモジュールに入力される。このモジュールが読み出しにおける最終段であり、ヒットデータを SL と FELIX 間の通信のデータフォーマット (図 3.17) にパッキングする。

パッキングされたヒットデータは実際のオペレーションでは、FELIX に送信されるが、これに加えて、SL Zynq MPSoC 上の BRAM にもダンプするパスと、SL FPGA 上に Readout checker というモジュールを実装した。

PS board FPGA、SL FPGA の全てのラインを一挙に検証することができる。もう一つは PS board の FPGA に埋め込まれた PS board FPGA テストパルスである。これは、SL から自由に TPP を設定することができる。2 種類のテストパルス発生機構は、PS board の FPGA に実装されたセレクターを用いて SL から選択する。

図 3.18 に TTC Emulator 内部のテストパルストリガーの発行の仕方を示す。テストパルストリガー (TPT) の発行の仕方は、Zynq MPSoC から変更することができる。一つは、Zynq MPSoC からパルスを打ち込み、そのタイミングで TPT を発行する方法である (図 3.18 中の shot)。もう一つは、特定の BC パターンごとに自動で TPT を打ち続ける方法である (図 3.18 中の auto)。auto モードの場合、何 BC 間隔で TPT を発行するかのパターンを BRAM に格納してあり、それを参照して、TTC Emulator が自動で TPT を発行する。

5.4.1 節においては、BRAM に格納するパターンを全アドレスに対して 3564 BC として試験を行なった。つまり、3564 BC ごとに 1 回の頻度 (≈ 11 kHz) で TPT を発行し続けることになる。一方で、5.4.2 節では読み出し回路のストレスチェックの観点から、実際のオペレーションに近い、平均 1 MHz の以下で述べる分布にしたがって、乱数を生成し、BRAM に格納することが妥当だと考え、そのように実装した。

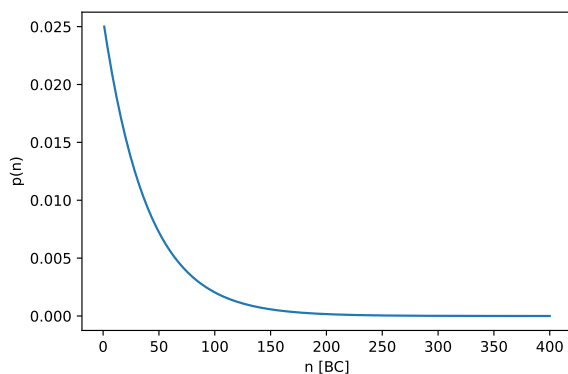


図 3.19: TPT が発行される時間間隔の分布。単位は 1 BC = 25 ns。

実際のオペレーションでは、1 MHz の頻度で、TPT が発行されるように、閾値が設定される。よって、とあるバッチ交差において、TPT が発行されるかは、1/40 の確率である。よって、ちょうど n BC が経過した時に TPT が発行される確率 $p(n)$ は、

$$p(n) = \left(\frac{39}{40}\right)^{n-1} \times \left(\frac{1}{40}\right) \quad (3.1)$$

$$= \frac{1}{39} \left(\frac{39}{40}\right)^n \quad (3.2)$$

で表される。この分布は図 3.19 のような分布になる。よってこの分布に従う n を乱数で生成して、それを BRAM に格納することで実際のオペレーションに近いテストパルスの発行を行うことができる。

第 4 章

Timing Alignment Master (TAM) モジュールの機能実装

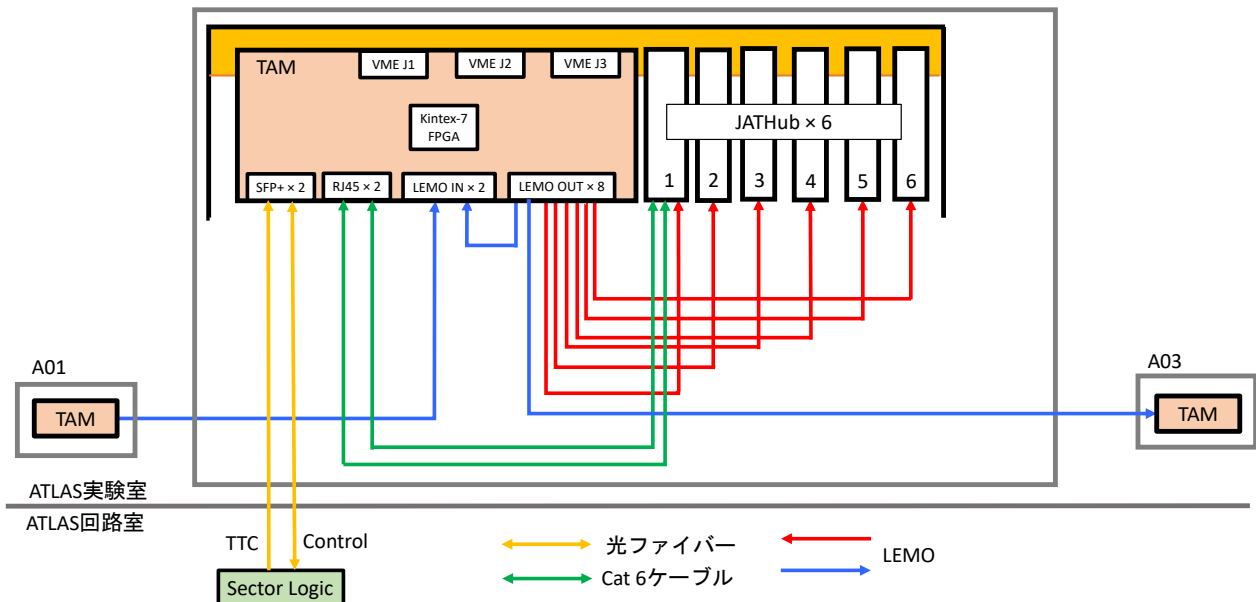


図 4.1: TAM とその他のモジュールの接続図。TAM モジュールと SL は 2 本の光ファイバーで接続され、SL から TTC 信号 (40.079 MHz BC clock, BCR) を受信すると共に、JATHub との VME 通信を行うために SL から制御される。また再構成した TTC 信号を JATHub へ分配したり、隣り合う TAM 同士の位相合わせを行うために、TAM にはクロック入力用に LEMO IN が 2 つ、クロック出力用に LEMO OUT が 8 つ搭載される。TAM モジュールは ATLAS 実験室に設置されるため、放射線損傷に対する回復手続が必要である。よって JTAG パスと加えて、PS board と同様に、RJ45 ジャックが 2 つ搭載され、JATHub と Cat 6 ケーブルで接続される。TAM モジュールは JATHub と同じ VME クレートに VME マスターとして挿入され、SL から制御信号を受け、JATHub の FPGA レジスタにアクセスしたり JATHub の QSPI フラッシュメモリにアクセスする。

タイミングキャリブレーションや、VME マスターの役割に担うフロントエンドエレクトロニクスとして、Timing Alignment Master (TAM) モジュールが実装される。本研究では、TAM モジュールの全機能を FPGA に実装したとともに、ハードウェアデザインの検証や試作機の運用を行ったため、この章で詳しく述べる。

図 4.1 に TAM モジュールとその他のモジュールとの接続の概要を示す。ATLAS 実験室に設置される VME ク

レートは 1/12 セクターごとに 1 つずつ設置され、それぞれの VME クレートに 1 台の TAM、6 台の JATHub が挿入される。TAM モジュールと SL は 2 本の光ファイバーで接続され、SL から TTC 信号 (40.079 MHz BC clock, BCR) を受信すると共に、JATHub との VME 通信を行うために SL から制御される。また再構成した TTC 信号を JATHub へ分配したり、隣り合う TAM 同士の位相合わせを行うために、TAM にはクロック入力用に LEMO IN が 2 つ、クロック出力用に LEMO OUT が 8 つ搭載される。1 台の TAM が 6 台の JATHub に基準クロック分配用に 6 つの LEMO OUT を使用し、隣り合う 1/12 セクター間の位相合わせのために 2 つの LEMO OUT と 2 つの LEMO IN を使用する。TAM モジュールは ATLAS 実験室に設置されるため、放射線損傷に対する回復手順が必要である。よって JTAG パスと加えて、PS board と同様に、RJ45 ジャックが 2 つ搭載され、JATHub と Cat 6 ケーブルで接続される。また、TAM モジュールは JATHub と同じ VME クレートに VME マスターとして挿入され、SL から制御信号を受け、JATHub の FPGA レジスタにアクセスしたり JATHub の QSPI フラッシュメモリにアクセスする。まとめると、以下の機能を FPGA に実装した。

- SL との高速光通信 (4.1.1 節)
- VME マスター (4.1.2 節)
- タイミング信号の受信・位相合わせ・JATHub への基準クロック分配 (4.2 節)
- 放射線損傷に対する回復 (4.3 節)

4.1 コントロールパスの実装

4.1.1 SL との通信 (データフォーマットの策定)

Link	Word	31...24	23 ...16	15...8	7...0
Link 1	0	0[31...27], Reset[26], BCR[25], 0[24]		0[23...20], BCID[19:8]	
	1	0[31...0]			
Downlink	2	0[31...0]			
	3	0[31...0]			
TAM RX	4	Footer (0x0e0d)		0[15...0]	
SL TX					

表 4.1: GTX のリンク 1 番のデータフォーマット (TTC 信号の分配用)。この光リンクを通して SL から TTC 信号を受信する。TAM の場合は、TTC 信号のうち、BCR と 40 MHz BC clock のみ受け取れば十分であるので、PS board とは異なり、TPT や test pattern といった情報は受け取らない。またリンクが安定かつ連続的に動作しているかをチェックするために、SL から TAM へも BCID を分配している。このリンクは TTC 信号に関わる場所なので、固定位相での受信部の実装を行なった。

SL と TAM は 2 本の光リンクで接続され、それぞれ、SL から TAM への TTC 信号分配と、TAM のコントロール・モニター及び VME 通信に使われる。

まず、リンク 1 番のデータフォーマットを表 4.1 に示す。リンク 1 番の GTX は、TAM が SL から TTC 信号を受信のために使用するので TAM が送信側の GT トランシーバは実装していない。また、PS board とは異なり、TAM はテストパルスに関連する機能を持たないので、TTC 信号のうち、JATHub に分配する、40.079 MHz BC clock と 202.42 kHz clock を生成する際に必要な BCR、SL と安定的なリンク確立が達成されているかのチェックのための BCID が実装されている。

次に、リンク 2 番のデータフォーマットを表 4.2 に示す。この光リンクでは、SL と TAM の送受信が実装され

Link	Word	31...24	23 ...16	15...8	7...0	
Link 2 Downlink TAM RX SL TX	0	0[31...27], Reset[26],0[25...8]			K28.5[7:0]	
	1	command[15:0]		0[15], CS[14...4], 0[3...2], SCLK[1], SDI[0]		
	2	address[31:0]				
	3	data[31:0]				
	4	Footer (0x0e0d)		0[15...0]		
Link 2 Uplink TAM TX SL RX	0	0[31...28], RXERR[27], RxSLP[26], QSPI status[25], Si5395 status[24]		SEM status[23...21], 0[20...8]		K28.5[7:0]
	1	0[31...21], read_fl[20], 0[19...17], write_fl[16]		0[15...11], SDO[10:0]		
	2	readback address[31:0]				
	3	readback data[31:0]				
	4	Footer (0x0e0d)		0[15...0]		

表 4.2: GTX のリンク 2 番のデータフォーマット (TAM コントロール・モニター用)。この光リンクを通して、SL から TAM のレジスタや QSPI フラッシュメモリの読み書きを行う。また、後で述べるように、この光リンクを経由して、VME 経由の JATHub のレジスタや QSPI フラッシュメモリの読み書きを行う。この光リンクは、TTC 信号とは全く無関係のパスで、比較的に遅いコントロール・モニターパスである。よって固定位相での受信部・送信部の実装は行っていない。

ており、TAM のレジスタの読み書きや QSPI フラッシュメモリの読み書きを行う。TAM の FPGA のレジスタはアドレス 32 bit、データ 32 bit の実装を行った。Down link(SL から TAM) では、レジスタの読み書きのための Command, Address, Data と、4-wire SPI ビットバンギング用の CSB, SCLK, SDI が用意されている。また、SL の光リンクを通じたりセットが行えるように、Reset も用意されている。Up link(TAM から SL) では、レジスタの読みに関して、読み出したデータとアドレスを返す、Address, Data と、アドレスの読み書きのモードを示す read fl, write fl が用意されている。TAM がレジスタの読みを行い、対応するレジスタのデータとアドレスを SL に返す際は、read fl を high にすることで、それを認識した SL がレジスタの読みの操作を完了する。また、4-wire SPI ビットバンギングの戻り値である SDO が用意されている。加えて、TAM の状態を常にモニターできるように、TAM は SL に対して、特別なコマンドを受けなくても、RxERR, RxSLP(SL と TAM 間の光リンクの確立状態を示す)、QSPI status, Si5395 Lock といった素子の状態に加えて、SEM の状態を常に SL に送り続ける実装を行なった。

4.2.4 節や 4 章で述べるデモンストレーションを通して、これらのデータフォーマットの定義及び実装が妥当であることを確かめた。

4.1.2 VME master の実装

TAM は VME クレート (HSC クレート) の向かって一番左側に挿入され、VME master としての役割を持つ。JATHub を VME slave として挿入し、J1 コネクタを通じて、JATHub の FPGA 上のレジスタを読み書きをすると共に、J3 コネクタを通じて JATHub のボード上の QSPI フラッシュメモリの読み書きを行う。

まずは、J1 を通じて JATHub の FPGA 上のレジスタをコントロールする機能について述べる。この際のプロトコルはいわゆる VME protocol の single cycle を用いて実装した。VME protocol は

- A01-A31: Address (Master)

- AM0-AM5: Address Modifier (Master)
- WRITE#: write (Master)
- DSA#: data strobe A (Master)
- DSB#: data strobe B (Master)
- D00-D31: Data (Master, Slave)
- DTACK#, BERR#: Data acknowledge, Bus error (Slave)

の線を用いて通信をする。ここで末尾に付いている"#"は負論理 (active low) を意味する。() 内に明記されている Master や Slave はその線に値を出力できるモジュールを明記している。例えば、アドレス線やその他の一部の線は Master のみが値を出力することができて、Slave はその値を読むことしかできない。同様に DTACK 線は Slave のみが触ることができて Master は読み込むことしかできない。一方でデータ線は、データを Master から Slave へデータを書き込む時は Master がデータを出し Slave はそのデータを読むことになるが、Master が Slave のデータを読み出す時は Slave がデータを出し、Master がそのデータを読み込むことになるので、Master も Slave も触ることができる線として定義されている。また、TAM と JATHub 間の VME 通信では VME 割り込みの機能は使わないので、VME 割り込みに関する説明は割愛する。TAM には J1, J2, J3 コネクタが、JATHub は J1, J3 コネクタが実装されており、TAM と JATHub 間の VME バックプレーンを経由した通信では、J1, J3 のみの実装となる。J3 は、TAM を経由した、JATHub の QSPI フラッシュメモリへのアクセスに用いられる*1。J1 コネクタのみを用いて VME 通信を行う場合は、アドレス幅 23 bit、データ幅 16 bit の通信となる。この際、VME プロトコルとしては、アドレス幅及びデータ幅が変更されるだけで、その他の部分は変わらない。

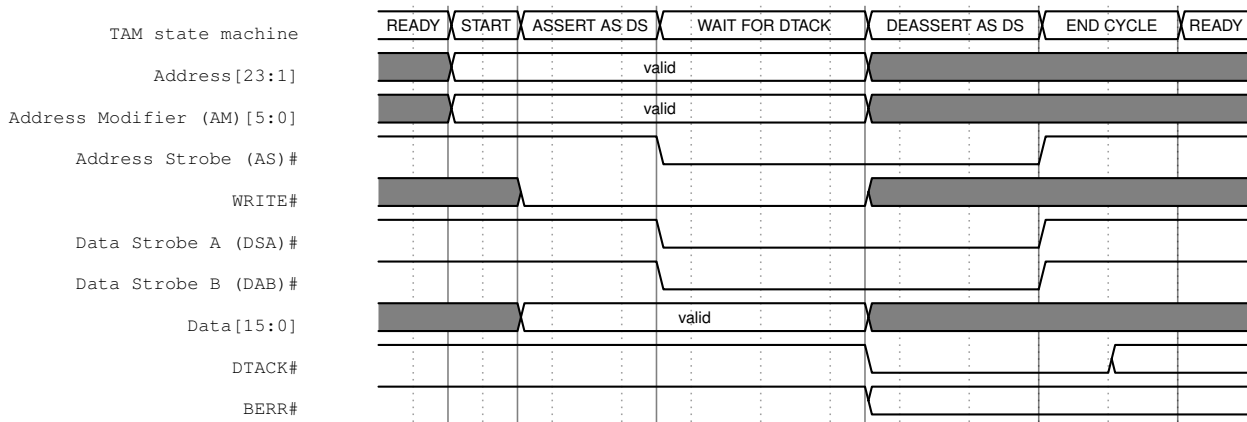


表 4.3: VME プロトコルのタイミングチャート (write cycle)。

VME プロトコルは、バス通信であり、基本的には Master 対複数の Slave の通信である。まず、write cycle について詳しく述べる。表 4.3 に TAM に実装した VME プロトコルの write cycle についてのタイミングチャートを示す。一番上の TAM state machine が TAM に実装したステートマシンの遷移を表している。ステートマシンは、TAM のオンボードの発振器のクロック (10 MHz) で駆動する。VME slave となる JATHub 側は、TAM のオンボード発振器とは独立した発振器のクロック (16 MHz) で駆動するため、VME master と slave でクロックドメインが異なるが、これから述べるように VME 通信はハンドシェイク的な通信なので問題はない。

アドレス幅 23bit とあるが、そのうち上位 8 ビットはどの JATHub (VME slave) かを指定して、残りの 15 ビッ

*1 J3 backplane が現行の TGC 検出器エレクトロニクスシステムのために特化した設計がされている。その中の Master から Slave の間で point-to-point の接続を持っている信号線を再利用するデザインとした。

トがそのうちのどのレジスタにアクセスするかを指定する。したがって、JATHub 側は上位 8 ビットを個々の JATHub に対して正しく (同一のクレートに差す JATHub 同士で被らないように) 設定する必要があり、ボード上の Dip switch を用いてハードウェア的に設定する。下位 15 ビットの方はファームウェアで自由に使えるアドレス空間であり、用途に合わせて定義しており、例えば 0 番は Master から読み書きできるテストレジスタである。以上より、Dip switch で上位 8 ビットが 0x84 と定義された JATHub のテストレジスタにアクセスする際は、A01-A23 を 0x840000 と設定する。さて、アドレスは上のように設定して、次は AM0-AM5 は以下のように設定する。AM0-AM5 は Address Modifier であり、これは、JATHub のファームウェア上に事前に設定された値に設定する必要がある。Master 側から受け取った AM0-AM5 と JATHub のファームウェアに設定された AM0-AM5 を比較し、一致した場合のみ JATHub のレジスタへのアクセスが許可される。WRITE# は cycle が write cycle か read cycle かどうかを示す。もし write cycle であれば WRITE# を low にして、read cycle であれば high に設定する。最後に書き込みたいデータを D00-D31 に設定する。以上の設定をして、Address strobe#, Data strobe#線を low に下げる。これにより、JATHub は Address strobe#, Data strobe#線の立ち下がりを読み取って、その時点で、アドレス線や write 線、データ線にある値を命令として認識して、自身のレジスタへの書き込みを行う。一通り、自身のレジスタへの書き込みが終わると、JATHub は DTACK#線を low にする。これを Master 側が認識して、Address strobe#, Data strobe#線を high に戻して、write cycle を終了する。一方 JATHub が Address strobe#, Data strobe#が高に戻ったことを認識して、DTACK#線を high に戻す。

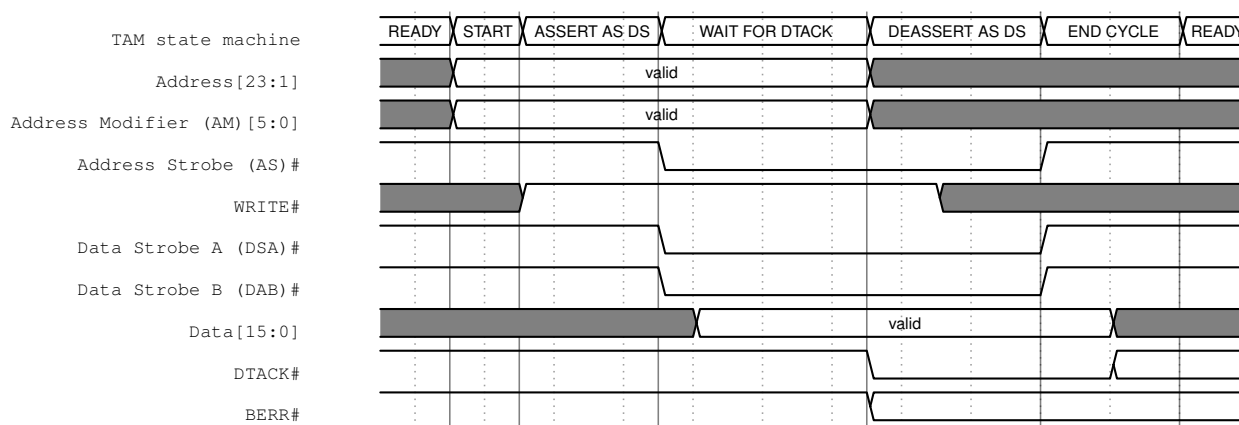


表 4.4: VME プロトコルのタイミングチャート (read cycle)。

次に read cycle について詳しく述べる。表 4.4 に TAM に実装した VME プロトコルの read cycle についてのタイミングチャートを示す。一番上の TAM state machine が TAM に実装したステートマシンの遷移を表している。アドレスや Address Modifier の設定は write cycle と同じであるが、WRITE# は high に設定し、Data は特に設定する必要がない。その上で、Address strobe#, Data strobe# は write cycle と同じタイミングで、操作する。これを受け取った JATHub は、指定されたアドレスのデータを、VME のデータ線に出力する。出力が完了すると、DTACK#線を low にする。Master 側がこれを認識して、その時にデータ線に出力されているデータを読み込み、Address strobe#, Data strobe#を high に戻して read cycle を終了する。一方 JATHub が Address strobe#, Data strobe#が高に戻ったことを認識して、DTACK#線を high に戻す。

以上の VME 通信の Master の機能をステートマシンとして、TAM の FPGA 上に実装した。また、VME を経由したレジスタの読み書きが安定して動作するかを、複数個のレジスタに対して、全データ (32 bit 幅のデータに対しては 0x0 から 0xffffffff まで、16 bit のデータ幅に対しては 0x0 から 0xffff まで) の読み書きを行い、書き込んだ値と読み出した値が一致するかをチェックした。その結果、エラーなく読み書き達成され、VME マスターの機能

が正しく実装されていることを確かめた。

4.2 固定位相でのタイミング信号と位相合わせの実装

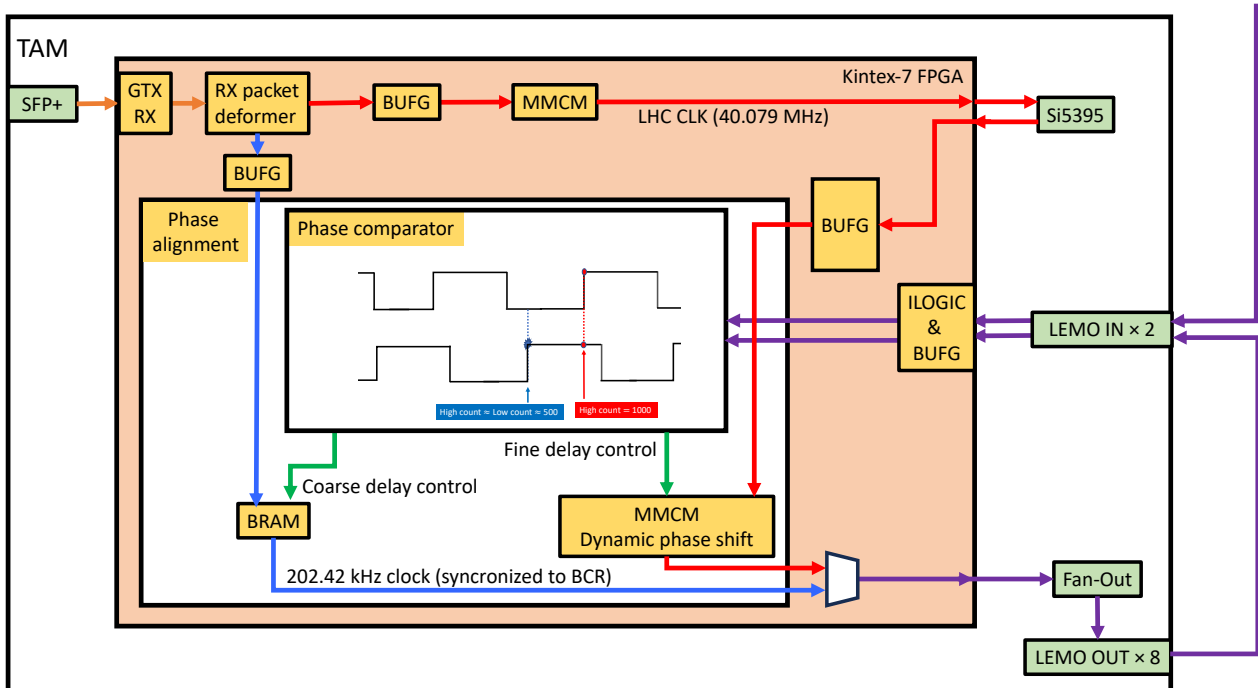


図 4.2: TAM モジュールに実装した固定位相でのタイミング信号と位相合わせのブロック図。オレンジ色の線が、SL から光リンクでクロックを受信する部分、赤線が主に LHC CLK に関わる部分、青線が 202.42 kHz クロック、紫線が LHC CLK もしくは 202.42 kHz クロック、緑線が可変遅延回路のコントロールを表している。

図 4.2 に TAM モジュールに実装した固定位相でのタイミング信号と位相合わせのブロック図を示す。TAM は SL と光リンクで接続され、TTC 信号を受信し、クロックを再構成して隣り合う TAM 同士で位相を合わせた後、JATHub に基準クロックを分配する (図 4.1 参照)。そのために、以下の節で、固定位相でのタイミング信号の受信、位相合わせ、JATHub へのクロック分配について述べる。

4.2.1 固定位相でのタイミング信号の受信

PS board と同様に、TAM も固定位相でタイミング信号を受信する必要がある。図 4.2 に示すように、LHC CLK(40.079 MHz) と BCR に同期した 202.42 kHz クロックの 2 種類を SL との光リンクから再構成する。実装は、3.4.1 節で述べたものと全く同様である。TAM 及び SL のリセットや reboot に対して、TAM が再構成するクロックの位相は変化せず、固定位相での送信及び受信が達成されていることを確認した (図 4.3)。

4.2.2 位相合わせ

次に位相合わせについて述べる。再構成したクロックは、SL から TAM までのケーブル長等によって、TAM 毎に個体差がある。TAM は 1/12 セクターに 1 台ずつ設置されるが、隣合う TAM 同士を等長の LEMO ケーブルで接続し、隣の TAM に位相を合わせるという手続きを 24 台で行った後 JATHub に基準クロックとして分配するこ

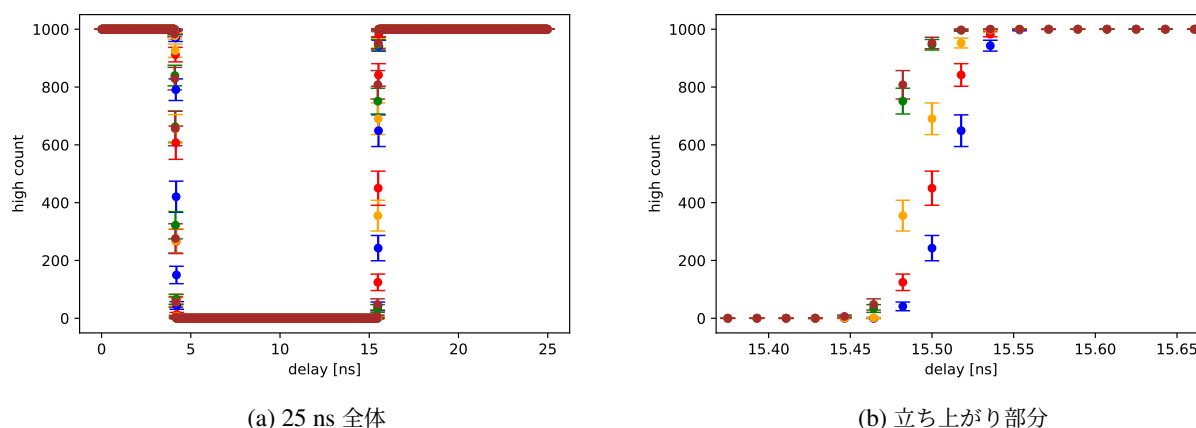


図 4.3: TAM が再構成するクロックの位相 (拡大図)。光リンクの再確立 (青) や TAM の FPGA reprogramming (緑)、TAM の power cycle (オレンジ、茶色) に対する、TAM が再構成したクロックの位相。ばらつきは 30 ps 程度に収まっており、十分な精度での固定位相での TTC 信号の受信が達成されていると言える。

とで、TGC 検出器の全ての JATHub が同位相のクロックを基準として PS board の位相を測定することができる。その結果、1 台の JATHub は自身にぶら下がる最大 11 台の PS board の位相を揃えることを、全ての JATHub が完全に独立に行うことで、全 1434 台の PS board の位相が揃うことになる (図 4.1 参照)。

本節では、再構成した 202.42 kHz クロックと 40.079 MHz BC clock を隣の TAM と比較して、自身のクロックに 2 種類の variable delay (Coarse delay = 25ns 幅, Fine delay = 18 ps 幅) を適切な分だけかけるような機能について説明する。まず可変遅延回路の実装について述べる。2 種類の variable delay の実装については 3.4.2 で述べたのと全く同じ方法で実装を行った。すなわち、図 4.2 に示すように、Coarse delay は BRAM、Fine delay は MMCN の dynamic phase shift 機能を用いて実装を行なった。

次に位相測定の実装について述べる。位相測定に関しても 3.4.2 と同じ手法を用いた。図 4.2 の、Phase comparator というモジュールに位相測定の機能を実装し、そこに LEMO IN で受けたクロックを入力する。この LEMO IN で受けたクロックは自身の LEMO OUT から出力したクロックと、隣の LEMO OUT から出力されたクロックを自身の LEMO IN に入力したものである。したがって、この 2 者を比較することにより、隣の TAM との位相差を測定することができる。その際、等長で 2 者のクロックを自身に引き入れる必要がある。そのためにも、LEMO ケーブルは等長のものを用いる。さらに、LEMO IN から、FPGA までの基板上的配線長も等長にする。加えて、FPGA 内での配線長も可能な限り等長にするために、図 4.2 に示すように、位相測定に用いるラッチを FPGA のピンに最も近い D フリップフロップを用いて行うようにする (ILOGIC)。最後に、LEMO IN から入ったクロックを受けるレシーバの系統誤差を加味する必要がある。LEMO IN から入力されたクロックは、DS90LV018A はデータシート [18] によると最大で 1.5 ns の個体差がある。実際に TAM の LEMO IN の入力のみスワップして、スキューを測定してみると、200 ps 程度のスキューが観測された。このスキューは、事前に測定をしておき、TAM の位相合わせのプロセスにおいてはこのスキュー分の校正を行う必要がある。後で述べる、TAM 同士の位相合わせのデモンストレーションにおいても、この校正を加味した位相合わせを行なっている。なお LEMO OUT のポートごとのスキューの測定も行なったところ、表 4.5 のような結果が見られた。これをみるとスキューは、典型的に 30 ps 程度と比較的小さいため、事前測定を行なってファームウェアで吸収するといったことはせず、系統誤差として計上する予定である (もちろん事前測定を行なって吸収することも可能である)。

こうして、自身のクロックと隣の TAM を比較しながら、自身のクロックに delay をかけて、 2π スキャンを行

LEMO OUT port	0	1	2	3	4	5	6	7
Skew [ps]	20	-60	20	-40	20	-60	0 (reference)	-60

表 4.5: TAM の LEMO OUT のポートごとのスキュー。これは最終的には校正せず系統誤差に計上する予定である。

う。適切な delay parameter を決定した後自身のクロックにその分の delay をかけることにより、隣の TAM に自身のクロックの位相を合わせることが可能になる。SL からのトリガーにより、この位相合わせのプロセスが全て自動に走るようなステートマシンを実装した。

4.2.3 JATHub へのクロック分配

TAM 同士で位相を合わせた後、そのクロックを JATHub に分配する。ここで、LHC クロックである 40.079 MHz と BCR に同期した 202.42 kHz クロックの両方を JATHub に分配する必要がある。

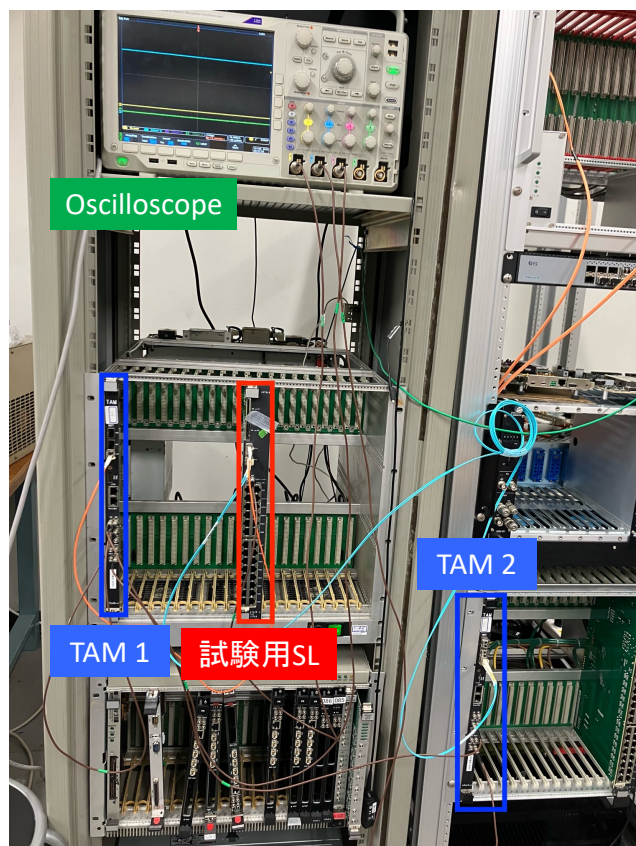
40.079 MHz クロックの分配は、TAM 上の fan-out 素子で 8 つに複製したあと、LEMO OUT 素子に出力する (スキューに関しては表 4.5 参照)。LEMO OUT 素子で出力された 40.079 MHz クロックを等長の LEMO ケーブルで 6 台の JATHub に分配し、JATHub は LEMO IN でそれを受け取る (8 - 6 = 2 つの残りの LEMO OUT から出力された 40.079 MHz クロックは先ほど述べた TAM 同士の位相合わせのために用いられる)。一方で 202.42 kHz クロックは、25 ns 以上のズレを同定するために用いられる。TAM 同士で、位相が合わせられた 202.42 kHz クロックの JATHub への分配には、VME backplane を用いる設計を考えた。VME backplane の配線はデータの送受信を目的に設定されているので、配線長は揃えられていない。そのため、VME backplane の配線を用いて分配された 202.42 kHz クロックは優位なスキューを持つ。よって、それぞれ JATHub は受け取った 202.42 kHz クロックを、LEMO IN から受け取った 40.079 MHz クロックでラッチすることで、スキューを吸収し、同位相での 202.42 kHz クロック分配を可能にした。この際、VME backplane において、TAM に一番近いところに差した場合と TAM から一番遠いところに差した場合で最もスキューが出るが、この場合でもスキューは 25 ns 以内に収まっていることを確認した。よって同位相の 40.079 MHz クロックでラッチすることでスキューは吸収できる。

4.2.4 TAM 間のクロックの位相合わせのデモンストレーション

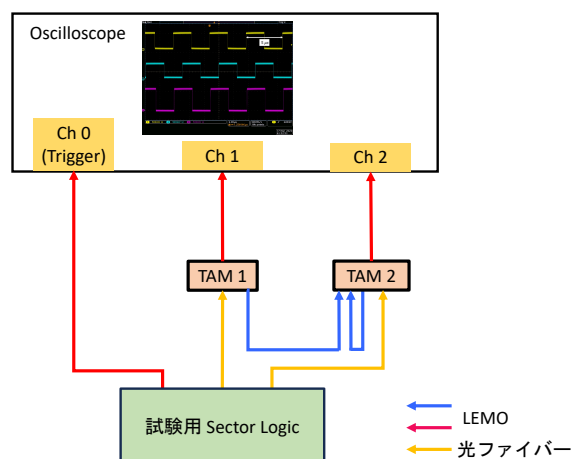
まずはセットアップについて述べる。2 台の TAM と試験用の SL を用いて図 4.4 のように接続する。ここで使用している TAM は第一試作機である。例えば、今回のセットアップでは TAM 2 の位相を TAM 1 に合わせるような配線にした。その場合、等長の LEMO ケーブルを用いて、TAM 1 のクロックを TAM 2 の LEMO IN に入力すること、及び TAM 2 のクロックを TAM 2 自身の LEMO IN に回し入れるような配線になる。さらに試験用 SL における 202.42 kHz クロックをトリガー (Ch 0) として、オシロスコープに入力し、Ch 1 に TAM 1、Ch 2 に TAM 2 の LEMO OUT を接続する。

デモンストレーションの手順は以下の通りである

1. それぞれの TAM は LHC CLK(40.079 MHz) と 202.42 kHz クロック (BCR に同期) を再構成する
2. LEMO OUT に 202.42 kHz クロックを出力し、位相調整機能を走らせる (Coarse delay)
3. LEMO OUT に LHC CLK(40.079 MHz) を出力し、位相調整機能を走らせる (Fine delay)
4. Coarse delay 調整後の 202.42 kHz クロックを Fine delay 調整後の LHC CLK(40.079 MHz) でラッチし、LEMO OUT から出力する。
5. オシロスコープを用いて、TAM 1 と TAM 2 の位相が一致しているか検証する。



(a) セットアップの写真



(b) セットアップの概要図

図 4.4: 異なる 1/24 セクター間の位相合わせの機能実装のデモンストレーション TTC 信号の分配用に、試験用の SL から 2 本の独立な長さの光ファイバーをそれぞれの TAM に接続する。TAM は再構成したクロックを LEMO OUT から出力する。今回のセットアップでは TAM 2 の位相を TAM 1 に合わせるような配線にした。その場合、等長の LEMO ケーブルを用いて、TAM 1 のクロックを TAM 2 の LEMO IN に入力し、TAM 2 のクロックを TAM 2 自身の LEMO IN に回し入れるような配線になる。その上で、試験用 SL、TAM 1、TAM 2 のクロックをオシロスコープに入力し、試験用 SL のクロックでトリガーを取ることによって、TAM 1 と TAM 2 の位相関係を確かめた。

6. また、ここで決定された Coarse, Fine delay parameter が接続した光ファイバーの配線長等と比較して妥当なものかチェックする。

ここでは、Coarse delay と Fine delay の二段階の位相調整機構をまとめて検証するために、位相調整後の 202.42 kHz クロックを位相調整後の LHC CLK(40.079 MHz) でラッチしたものをモニターする。

オシロスコープで測定した結果を図 4.5 に示す。調整後の TAM 1 と TAM 2 のクロックが十分な精度で揃っていることが確認できる。先ほど述べた通り、LEMO IN に関するスキューは、ファームウェアで校正し、吸収している。一方で LEMO OUT に関するスキューは吸収していないため、系統誤差として残る。これを見ると、100 ps よりも十分に良い精度で、TAM 1 と TAM 2 の位相が一致していることが分かる。また、ここで決定された、Coarse delay, Fine delay parameter が光ファイバーの配線長等の、仕掛けた遅延時間差から期待される値がえられていることも確認した。加えて、セットアップを保存し、FPGA の再コンフィギュレーション等を行なった上で、位相調整機構を再び走らせても、全く同じ Coarse delay, Fine delay parameter が選ばれることも確認し、再現性を確か

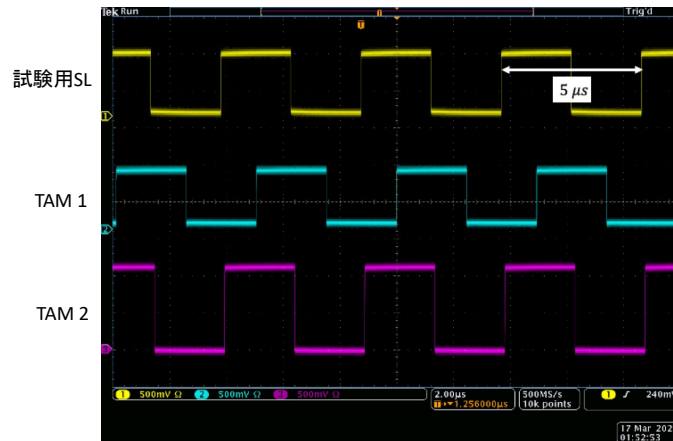
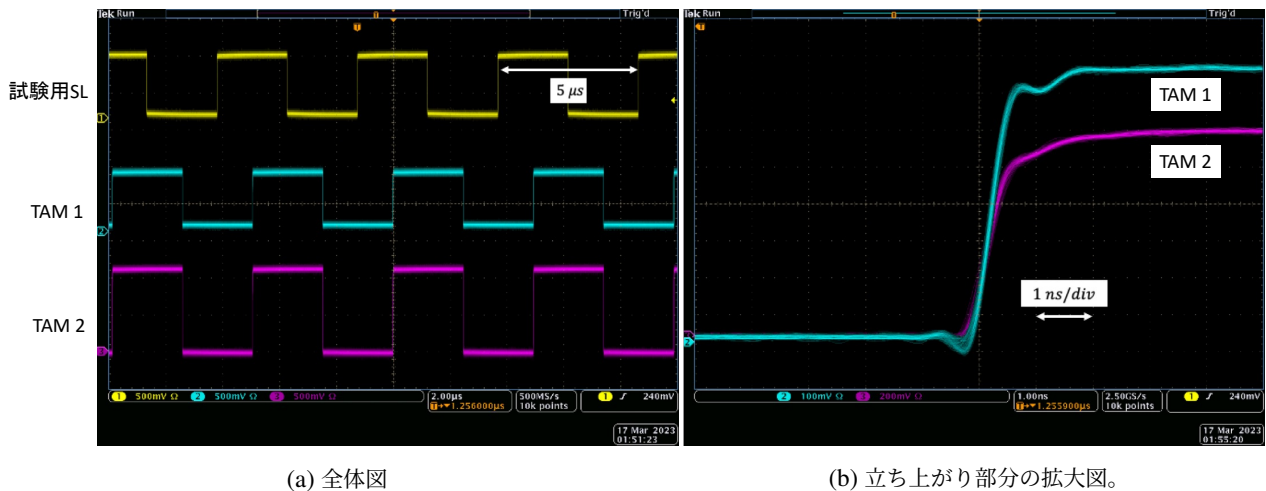


図 4.5: TAM の位相調整機構のデモンストレーション (位相調節前)。黄色の波形が試験用 SL、青色の波形が TAM 1、紫色の波形が TAM 2 である。



(a) 全体図

(b) 立ち上がり部分の拡大図。

図 4.6: TAM の位相調整機構のデモンストレーション (位相調節後)。黄色の波形が試験用 SL、青色の波形が TAM 1、紫色の波形が TAM 2 である。これを見ると 100 ps よりも十分に良い精度で位相が一致していることが分かる。

めた。

以上のデモンストレーションより、2 者の TAM 間での位相調整機構が十分に良い精度 (100 ps) で実装されたとと言える。実際のオペレーションでは、TGC システム全体で 24 台の TAM を用いて、TAM 同士の位相合わせを行うが、個々のステップはあくまでも上のデモンストレーションと全く同じである。よって、全く同一の TAM (ファームウェア) を 24 台用意し、順々に位相調整機構を走らせることによって、24 台の TAM の位相が十分に良い精度で一致した状態が達成でき、TGC 検出器エレクトロニクス全体で JATHub は 148 台あるが、その全ての同位相の基準クロックを分配することが可能となる。したがって、個々の JATHub が完全に独立に自身に接続された最大 11 台の PS board の位相を測定し、位相合わせを行うことで、TGC 検出器全体で合計 1434 台の PS board の位相を合わせることが可能になる。JATHub と 11 台の PS board における位相合わせのデモンストレーションに関しては、5.3 節で詳しく述べる。

4.3 放射線損傷に対する回復と自律型制御機構の実装

TAM も PS board と同様に放射線環境下に設置される。従って、TAM にも SEM Controller を実装し、回復可能な SEU に対しては自動で修復を行う。また、自律型制御機構を実装し、回復不可能な SEU に対しては JATHub に救難信号を出して、JATHub から FPGA の再コンフィギュレーションを行い、パラメータの設定等を行う。

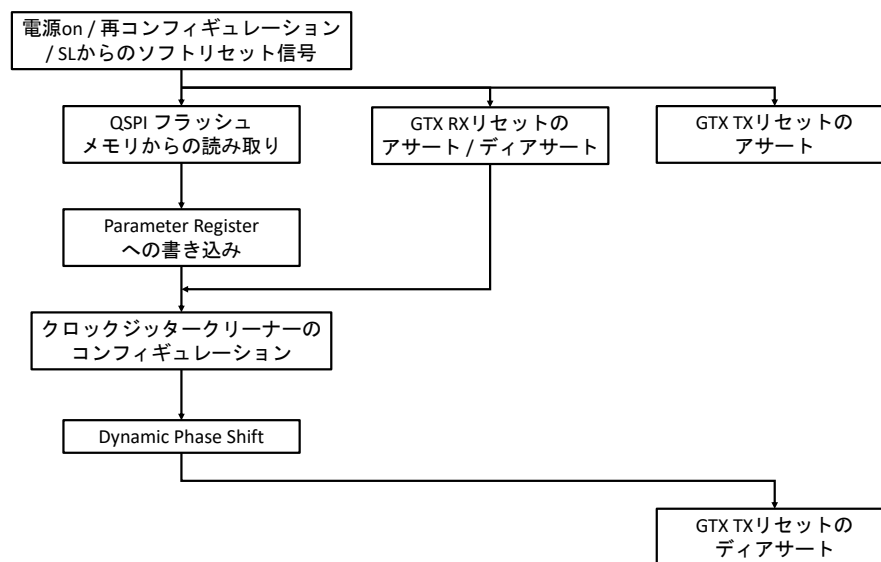


図 4.7: 自律型制御機構の全体的な手順

図 4.7 に TAM における自律型制御機構の手順の全体図を示す。自律型制御機構の実装は大まかには PS board と同じである。ただし、TAM は PS board とは異なり、PP ASIC や DAC、ADC といったデータ読み出しに関わる素子がないので、自律型制御機構のシーケンスにそれらのコンフィギュレーションは含まれない。TAM の場合、自身に必要な delay parameter は、4.2 節で述べたように、自身のスキャンによって決定する。一方で、一度決められた delay parameter は、光ファイバーの配線長等のセットアップを変更しない限り使い続ける値である。よって、一度スキャンが完了すると、reboot の度にスキャンし直すことはせず、PS board と同様に、delay parameter は QSPI に書き込んでおいて、自律型制御機構でクロックのシフトを行うという実装を行なった。つまり意図的に SL から位相調整機構を走らせない限り、TAM も PS board と同様に QSPI から読み出した delay parameter を用いて自身のクロックをシフトさせることになる。

4.4 TAM の量産と QAQC に向けて

現在 (2023 年 12 月) までで TAM は第一試作機と第二試作機が完成しており、ハードウェア的な不具合等を洗い出した上で、量産機の製作を行なっている最中である。予定では、2024 年 4 月ごろに予備も含めて 30 台の量産が完了し、Quality Assurance and Quality Control (QAQC) を始める予定である。TAM の QAQC では、実際にオペレーションで用いる予定のファームウェアを使い、TAM のハードウェアの不具合をチェックするとともに、LEMO OUT, LEMO IN のスキューの測定も行う。

第 5 章

高輝度 LHC-ATLAS 実験に向けた TGC 検出器エレクトロニクスシステムの統合試験

5.1 TGC 検出器エレクトロニクスシステムの統合試験の概要

高エネルギー加速器研究機構 (KEK) にて、図 5.1 に示すようなシステムレベル試験のためのテストベンチを立ち上げ、その機能実装を行なった。これまで実装した個々の技術を組み合わせて、安定して動かせるシステムを構築した。本章では、統合試験の構築及び機能実装、機能検証について述べる。

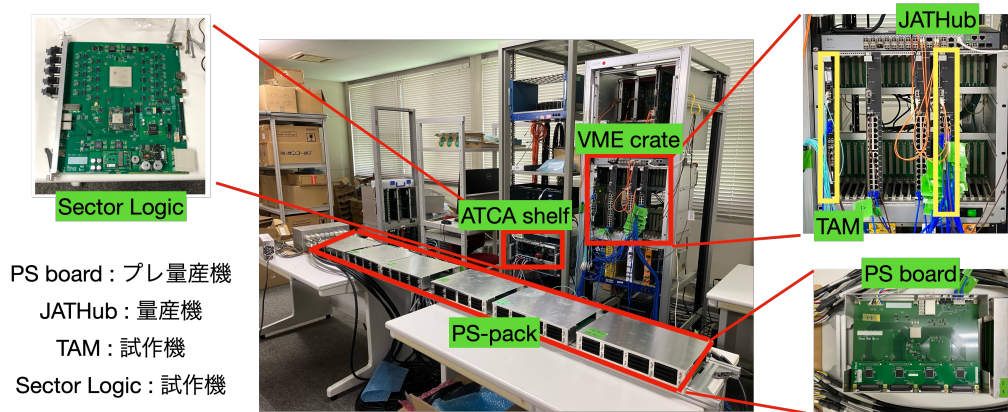
今回作成したテストベンチの目的は大きく 3 つあり、

1. 大規模エレクトロニクスシステムを見据えた、ファームウェアやソフトウェアのプロトタイプを作成
2. 実際のオペレーションで用いられるファームウェアやソフトウェアのプロトタイプの検証
3. ケーブルの配線ミスやハードウェアの故障を見つけるためのスタンドアロンテストベンチの作成

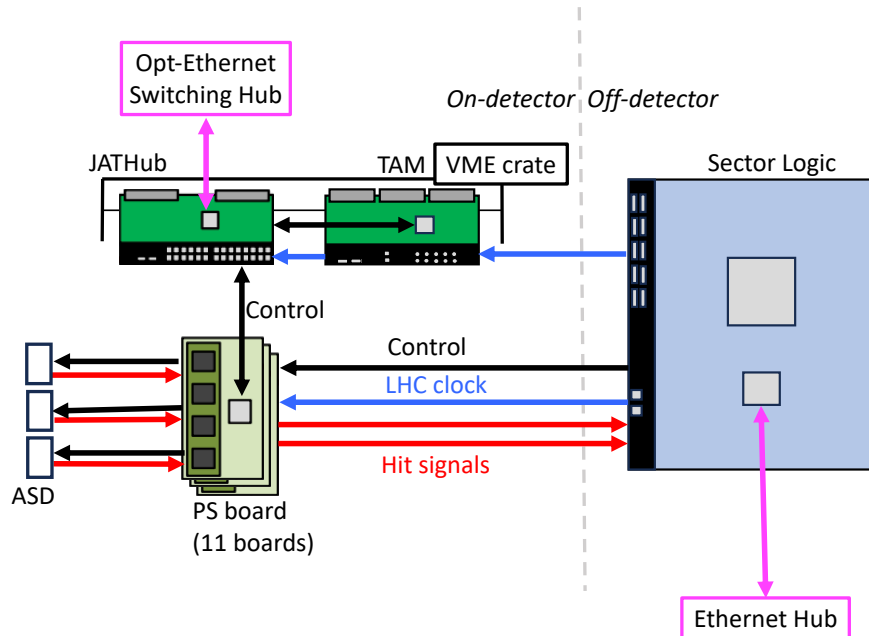
である。

一つ目は、システム全体の統合試験を通じて、実際のオペレーションで用いられる、大規模エレクトロニクスシステムを見据えた、ファームウェアやソフトウェアのプロトタイプを作成することである。ファームウェアやソフトウェアが開発段階であったり、試験に用いられるモジュールの数が少ない場合は、"手で"個々のモジュールをモニター、コントロールすることが妥当であったり可能であるが、実際のオペレーションでは膨大な数のモジュールを扱うことを見据えると、遠隔化や並列化は必須である。今回の統合試験では全 11 台の PS board を用いているが、実際のオペレーションを見据えて、遠隔で統合試験をモニター、制御、デバッグできるインフラを整備したこと及び、ワンクリックかつ、全 11 台の PS board のモニター・コントロール・読み出し等を並列に行うようなシステム、運用方法を構築した。また、5.4.1 節で述べるような、全 11 台の PS board の delay curve の作成等を通して、そのような実装が安定した形でなされていることを証明した。

二つ目は、システム全体の試験を行うことによって、実際のオペレーションで用いられるファームウェアやソフトウェアのプロトタイプを検証することである。種々のモジュールに渡って横断的に実装された、モニター、コントロール、タイミング信号分配・受信や位相合わせ、放射線損傷に対する回復、及び読み出し系といった機能の検証には、ボード単体の試験では不十分であり、ASD, PS board, JATHub, TAM, SL を接続した統合試験を行う必要がある。言い換えれば、個々のモジュールに正しく機能実装が達成されているかどうかを、統合試験を行うことによって、横断的かつ包括的に検証することが可能となる。例えば、5.4.1 節で述べるように、ASD ボードに搭載されたテストパルス発生機構を用いると、ASD から PS board、及び SL の読み出し回路の最終段までを一挙に検証することができる。また、この時、各ヒット信号 (ASD テストパルス信号) に付与される BCID やバンチタグが、各ボードの reboot やリセットに対して変化しないかを検証することにより、読み出しに関わる部分の固定位相



(a) KEK テストベンチの写真。



(b) KEK テストベンチの概要図

図 5.1: TGC 検出器エレクトロニクスシステムの統合試験の概要。図 5.1a に統合試験のための KEK におけるテストベンチの写真を、図 5.1b にテストベンチの概要図を示す。本テストベンチは Sector Logic, JATHub, TAM, 11 台の PS board を含む。Sector Logic は初号機、JATHub は量産機、TAM は第二試作機、PS board はプレ量産機を用いて試験を行った。それぞれの PS board には 16 台の ASD が装着され、PS board と SL は 2 本の光ファイバーで接続される。また JATHub と PS board (TAM) は 2 本の Cat 6 ケーブルで接続されており、JATHub と TAM は 1 本の LEMO ケーブルで接続されている。SL と JATHub の CPU 部分に PC からイーサネットを通じてアクセスし、コントロールやモニター、読み出しを行う。

での実装を検証することができる。

三つ目は、フロントエンド回路 (ASD, PS board, JATHub, TAM) を ATLAS 実験室に設置する際に、ケーブルの配線ミスやハードウェアの故障を見つけるためのスタンドアロンテストベンチの作成である。LHC が現在の運転期間である Run 3 を終えた後、高輝度化を目的とした Phase-2 アップグレードが行われる予定である。具体的なスケジュールとしては、Run 3 は 2025 年 11 月まで行われ、LS 3 (Long Shutdown 3) が 2025 年 12 月から 2029 年 1 月までの期間となる。したがって LS 3 の間に LHC や ATLAS 検出器のアップグレードを全て完了させなければならない。そのうち我々が担当する、TGC 検出器のエンドキャップ領域のハードウェア的なインストールとして、地下の ATLAS 検出器のある領域 (USA 15) で作業できる時間は約半年である。半年という期間の間に、フロントエンド回路のインストールを済ませなければならない。またこの期間をすぎるとフロントエンド回路へのハードウェア的なアクセスは制限される^{*1}。特に TGC 検出器全体で、PS board は 1434 台、JATHub は 148 台、TAM は 24 台をインストールすることになり、その数は膨大であるため、効率的に不具合をチェックできるシステムが必要である。各 1/12 セクターごとに設置された、mini-rack と呼ばれる VME クレートに 6 台の JATHub、1 台の TAM が設置され、ASD ボードが TGC 検出器に直接取り付けられる。PS board をインストールする際には、PS-pack と呼ばれるアルミニウム製の箱に PS board 入れて、それを 11 台 (もしくは 18 台) の単位で TGC 検出器のリムに装着する手順となる。したがって事前に、フロントエンド回路 (11 台の PS board、JATHub、TAM、176 台の ASD ボード) とそれらの不具合をチェックするためのバックエンド回路 (SL) を 1 単位として、delay curve の作成など一通りの試験を行うことができるスタンドアロンテストシステムを作成しておき、インストールと並行して、試験をその都度行う。試験が完了したフロントエンド回路 (11 台の PS board、JATHub、TAM、176 台の ASD ボード) に関しては、SL との接続以外は保存して、次のフロントエンド回路の検証に移るといった流れである。

図 5.1a に実際の統合試験におけるテストベンチの写真を示す。ただし、ASD ボードに関しては、簡単のためこの写真では省略してある。本テストベンチを用いた統合試験が行われた 2023 年 12 月現在では、PS board はプレ量産機、JATHub は量産機、TAM は第一試作機、Sector Logic は第一試作機の制作が完了しており、本テストベンチでもそれらを用いた。先ほど述べた、スタンドアロンテストシステムという意味合いから、本テストベンチでは 176 台の ASD ボード、11 台の PS board、1 台の JATHub、1 台の TAM、1 台の Sector Logic が使用されている。これは、TGC 検出器のうち、第 1 層目の M1 Triplet と呼ばれる部分に対応する。本テストベンチでは、配線等も含めて実際のオペレーションと同様のセットアップを組むことによって、コミッショニングやオペレーションへの移行が率直である。

5.2 遠隔でのモニター、制御、デバッグが可能なインフラストラクチャーの整備

実際のオペレーションを見据えた遠隔化という観点から、本テストベンチでは、すべてのモジュールにイーサネットを介して直接的、もしくは間接的にアクセスすることで、完全に遠隔でのモニター、制御、デバッグを行えるようなインフラを整備した。図 5.2 に本テストベンチにおいて、各モジュールのモニター、制御、デバッグに関するパスについての概要図を示す。まず、Sector Logic の Zynq ultrascale+ MPSoC 上にはイーサネットを通じて、Linux に直接アクセスする。Sector Logic 上のメイン FPGA (Virtex ultrascale+ FPGA) のレジスタには、Zynq ultrascale+ MPSoC の Linux からアクセスする。PS board や TAM の FPGA レジスタには、SL の Zynq ultrascale+ MPSoC から、SL のメイン FPGA、光ファイバーを介してアクセスする。同様に PS board や TAM の QSPI フラッシュメモリには、SL の Zynq MPSoC から、SL のメイン FPGA、光ファイバー、PS board や TAM の FPGA を介して、

^{*1} ATLAS 実験の他の検出器との干渉により、全てのエレクトロニクスにアクセスできる期間は限られている。例えば 2025 年 5 月以降は M3 側に設置される PS board にはアクセスできない。

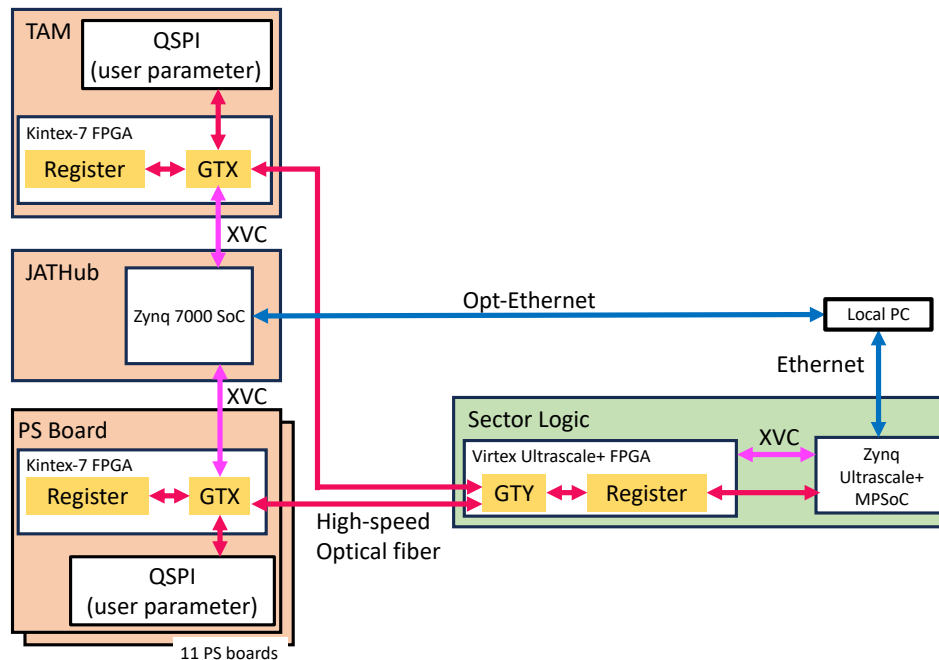


図 5.2: KEK テストベンチにおける各モジュールへのアクセスの概要図。青線がイーサネットを経由したアクセス、赤線が主にレジスタや QSPI フラッシュメモリへのアクセス、ピンク色の線が XVC(SVF player) を用いたコンフィギュレーションパスを示す。

ビットバンギングでアクセスする。JATHub の Zynq 7000 SoC には、PS にある Linux に光イーサネットを通じて直接アクセスする。また PL に実装されたレジスタには PS を経由してアクセスする。

また、SL 上の Zynq ultrascale+ MPSoC 及び JATHub のブートファイルを更新する際は、イーサネットを経由して、SD カード上に保存されたブートファイルを更新することができる。また、JATHub 及び Zynq ultrascale+ MPSoC には XVC (Xilinx Virtual Cable) を利用して、PS board、TAM、SL 上のメイン FPGA のファームウェアの更新が可能である。

以上より、完全に遠隔で、モニター、制御、デバッグが可能なテストベンチとなった。

5.3 タイミング信号分配・受信と位相合わせのデモンストレーション

4.2.4 節で述べたように、TGC 検出器全体で 148 台の JATHub に対して、同位相の基準クロックを TAM から分配するデモンストレーションの結果、個々の JATHub が自身につながる最大 11 台の PS board のクロックの位相測定を完全に独立に行うことで、TGC 検出器全体で 1434 台の PS board の位相を揃えることが可能になる。本節では、統合試験における、11 台の PS board への固定位相でのタイミング信号分配・受信と位相合わせのデモンストレーションについて述べる。なお、3.4.2 節で述べたように、JATHub を用いた位相測定は、再現性を持って相対位相を測定できていることは確かめており、測定自体のバリデーションは完了している。

まずは、5.3.1 節で、固定位相 (reboot やリンクの再確立等で位相が変化しない) でのタイミング信号の分配・受信が達成されているかを、各モジュールの reboot に対する再現性を検証することで確認する。次に、5.3.2 節で、今回のデモンストレーションに関わるクロックスキューの詳細な検討を行う。最後に、5.3.3 節で、位相合わせのデモンストレーションの結果について述べる。

5.3.1 再現性について

まずは、PS board が再構成するクロックの位相の再現性、つまり各種モジュールの再コンフィギュレーション等によって位相が変化しないことを確かめた。

ここでは JATHub の位相測定を用いて、各 PS board が再構成したクロックの位相を測定している。まずは、PS board の reboot に対して、測定結果を図 5.3 に示す。この結果を見ると、最大で 18 ps 程度の位相の不定性収まっており十分良い精度で、再現性が取れていると言える。この 18 ps の位相の不定性は、PS board の GTX の RX のロックの取り直しの際にも同等の位相の不定性が見えていることから、GTX の RX 部分の位相の不定性に対応している。

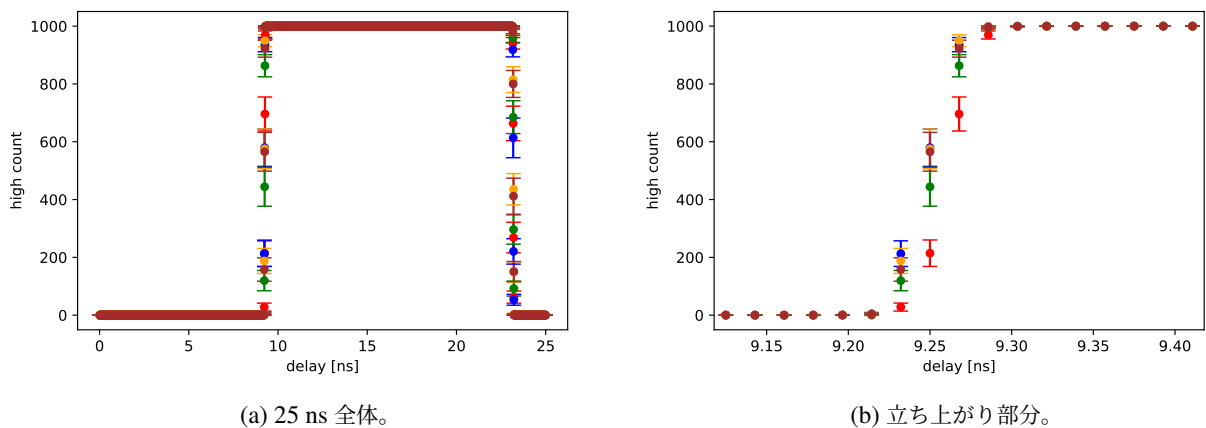


図 5.3: PS board の reboot に対する PS board が再構成するクロックの位相の再現性。PS board の reboot と位相測定を交互に行い、その時の結果を異なる色で示している。これを見ると、PS board の reboot のたびに 18 ps 程度の位相の不定性があるが、十分良い精度で再現性が取れている。

次に TAM の reboot に対する再現性について、測定結果を図 5.4 に示す。これを見ると最大でも 36 ps 程度の位相の不定性に収まっており、十分良い精度で再現性が取れていると言える。この 36 ps 程度の位相の不定性は、図 4.3 で示すように、TAM の GTX の RX のロックの取り直しの際にも同等の位相の不定性が見えていることから、TAM の GTX の RX 部分の位相の不定性に対応している。

最後の SL の reboot に対する再現性について述べる。SL を reboot すると、一時的に PS board も TAM も GTX の RX のリンクを失い、SL が正しいデータを送信し始めるとロックを取り直すことになるので、SL の reboot には先ほど述べた 2 つの不定性が必ず含まれることに注意する。測定結果を図 5.5 に示す。この結果を見ると、最大で 50 ps 程度の位相の不定性が見えているが、これは PS board と TAM の GTX の RX のロック取り直しに起因する位相の不定性であり、十分良い精度に収まっている。

5.3.2 システム内におけるクロックスキューの詳細な検討

本節ではタイミング信号分配・受信や位相合わせの際に問題となるクロックスキューについて詳しく議論する。この節では、JATHub の位相測定ではなくオシロスコープで PS board が再構成したクロックをプローブしてスキューを調べた。具体的には、図 5.6 にどの部分で主なクロックスキューが生じ得るかを示してある。ここで重要なのは、クロックスキューの中には、PS board が再構成するクロックに系統的なスキューとして現れるものと、単

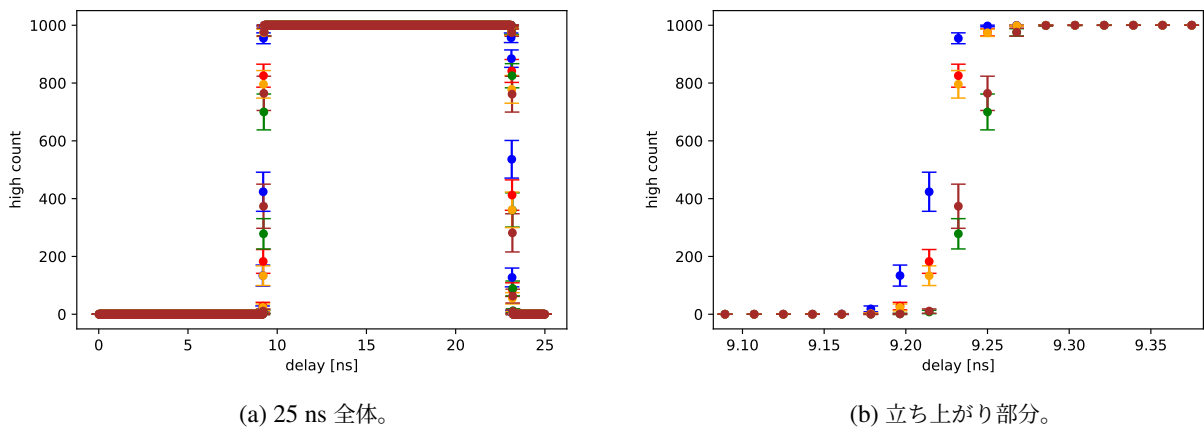


図 5.4: TAM の reboot に対する PS board が再構成するクロックの位相の再現性。TAM の reboot と PS board のクロックの位相測定を交互に行い、その時の結果を異なる色で示している。これを見ると、TAM の reboot のたびに 36 ps 程度の位相の不定性があるが、十分良い精度で再現性が取れている。この 36 ps のリブート毎の位相の不定性は図 4.3 で見えている位相の不定性である。

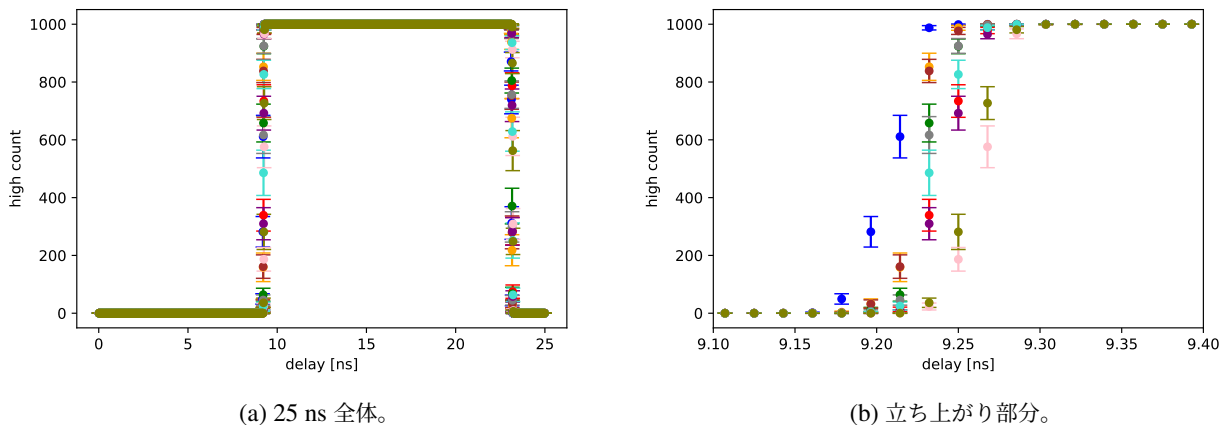


図 5.5: SL の reboot に対する PS board が再構成するクロックの位相の再現性。SL の reboot と PS board のクロックの位相測定を交互に行い、その時の結果を異なる色で示している。これを見ると、SL の reboot のたびに 50 ps 程度の位相の不定性があるが、十分良い精度で再現性が取れている。この 50 ps の位相の不定性は図 5.3 と図 5.4 で示した位相の不定性の重ね合わせで説明できる。

に位相測定に関して系統的なスキューとして現れるものがあるということである。前者であれば、位相測定の手続きを行うことによって、いわゆるケーブル長によるスキューをキャリブレーションで吸収するように、吸収することができる。しかし後者に関していうと、実際に PS board のクロックにそのスキューが乗っているのではなくあくまでも測定に関してその誤差が乗っているだけなので、位相測定を行なってその誤差を吸収するのは不可能であり、これを吸収しようとする、事前に独立した手続きでそのスキューを測っておいて、そのようなスキューを考慮した位相測定を行うことが必要となる。以上より、2つの観点のスキューが存在するが、図 5.6 でいうと、その2者は PS board までのクロック分配に関する部分のスキューと、その後の、再構成したクロックを JATHub に送信

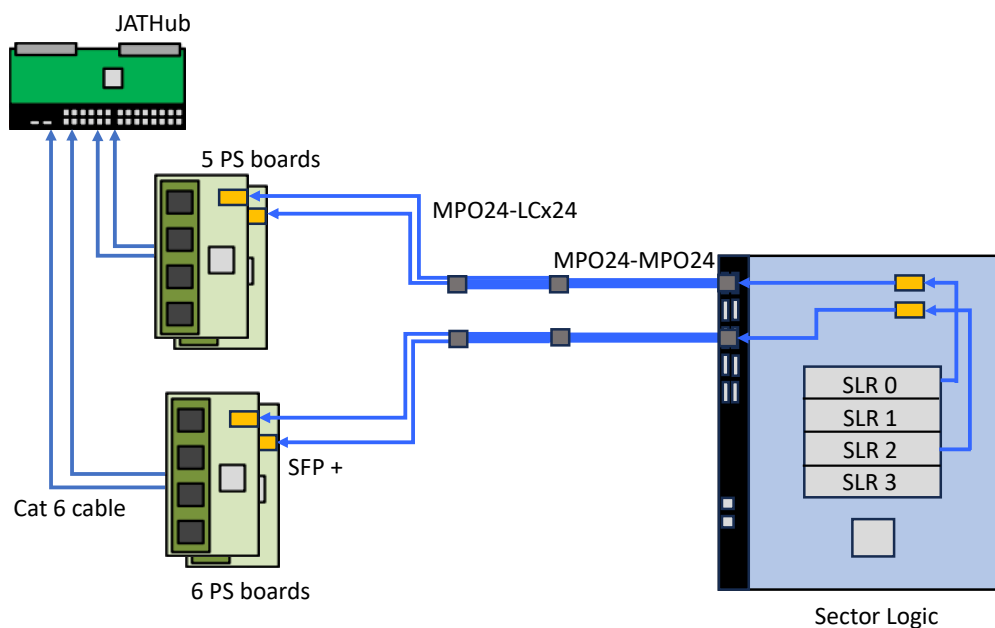


図 5.6: 統合試験におけるクロック分配系及び位相測定系の接続図。主なクロックスキューが生じ得る箇所に関して示してある。まずは、Sector Logic の光リンクのアウトプットのうちのどの port から出力されているかである。これは、FPGA (virtex ultrascale+) のどの SLR からの出ているかと、SL 上の配線長、及び Fire Fly コネクタの個体差が合わさったものである。次に、MPO24-MPO24 ケーブル、MPO24-LCx24 ケーブル、PS board に装着する SFP+ の個体差がある。また PS board の基盤上の配線や IC の個体差がある。そして最後に、位相測定に関するところとして、Cat 6 ケーブルの個体差と JATHub のどのポートに挿すかの個体差がある。

component	max [ns]	typical [ns]
SL port (same SLR)	1.5	0.7
SL port (different SLR)	3	1.5
MPO24-MPO24 (1 m)	0.2	0.1
MPO24-LCx24 (1 m-1 m)	<0.01	-
SFP+ (same type)	<0.01	-
SFP+ (different type)	0.1	0.1
PS board	0.1	0.1
Cat 6 cable (10 m)	0.6	0.4
JATHub (LVDS receiver)	0.4	0.2

表 5.1: 各コンポーネントのスキューの評価。スワップ試験を行なって評価を行なった。

する部分に対応する。

表 5.1 に各コンポーネントのスキューの評価結果について示す。繰り返しになるが、これは、各コンポーネントのスキューの測定方法は、その他のコンポーネントはそのまま特定のコンポーネントのみスワップして個体差を評価した。測定にはオシロスコープを用いた。まずはどのポートに差すかの誤差について述べる。これは、図 5.1 中の SLR port に対応するが、SLR に強く依存する結果となった。したがって Fire Fly コネクタの個体差に比べ

て、SLR もしくは、SL の FPGA から Fire Fly コネクタまでの配線長のスキューが大きいと言える。ただ、SL の FPGA から Fire Fly コネクタまでの配線長の差は大きく見積もっても、数 cm なので、スキューに換算すると、数 100 ps と見積もることができる。したがって支配的なスキューは、FPGA 中の SLR ごとの構造に由来すると考えられる。

次に、MPO24-MPO24 ケーブルの個体差について述べる。これは全て 1 m のものを使用しているが、最大で 200 ps のスキューが観測された。ただ全て 1 m と記載されているものであるが、実際に長さを測って見ると最大で 3 cm ほど長さが異なっており、上で述べたスキューはこの長さの違いからくるものだと考えられる。

次に、MPO24-LC24 ケーブルの個体差について述べる。これは MPO24 でバンドルされた、光信号を LC ケーブルへと分岐し、各 PS board へ分配する役割を持つ。MPO24-LC24 ケーブルに関しては、最大でも 10 ps 以内のスキューしか見られなかった。

次に SFP+ について述べる。SFP+ に関しては、同じ型番同士であれば、スキューは 10 ps 以内に収まっているが、異なる型番だと 100 ps ほどのスキューが観測された。

次に PS board について述べる。このスキューは、PS board 自体をスワップして測定を行った。実際には PS board のクロック再構成及び JATHub へのクロックの送信に関するパスには様々な IC が関与しているものの、それら IC のみをスワップすることは困難なのでこのような測定を行って、粗くスキューを見積もっている。具体的には PS board 上の、SFP+ との接続 IC (UE76-A20-3000T)、PS board の FPGA (xc7k325tffg900c-2)、ジッタークリーナー (Si5395) らのパーツごとのスキューまでが、クロックの再構成に関するところで、JATHub と繋がる RJ45 と接続されている IC (SN65LVDT348PW) のパーツごとのスキューまでが、位相測定にのみ関するところである。図 5.1 に示した結果は最大でも、100 ps ほどのスキューが見られたが、この結果は、これらのスキューが全て計上されている。

次に Cat 6 ケーブルのスキューに関して述べる。Cat 6 ケーブルに関しては 10 m のものを使用しているが、最大で 600 ps ほどのスキューが観測された。ここで、Cat 6 ケーブルを曲げたりして、曲率を変えてみてもスキューはほとんど変化しなかったことを明記しておく。このスキューを吸収しようとするすると全ての Cat 6 ケーブルの誤差を事前に測定して、かつそれをどの JATHub に差すかを意識しなければならないので、実際のオペレーションでは吸収することが難しいと考えている。よって典型的に 400 ps、最大で 600 ps ほどの系統誤差が位相合わせに計上される見込みである。

最後に JATHub のどの口に Cat 6 ケーブルを差すかに関するスキューについて述べる。これに関しては、オシロではなく、実際に JATHub の位相測定を用いて、評価を行った。結果は最大で 400 ps ほどのスキューが見られたが、これは JATHub 側の RJ45 からの信号を受ける LVDS レシーバ (SN65LVDT348PWR) のドキュメントに記載されているパーツごとのスキューに一致していることから、この IC 由来のものだと考えられる。この JATHub の LVDS レシーバに関するスキューは、事前に全ての JATHub に関して、スキューの測定を行うことによって、吸収する予定である。

以上をまとめると、Cat 6 ケーブル由来のスキュー (典型的に 400 ps、最大 600 ps) は吸収せず、その他のクロックスキューはキャリブレーションもしくは事前測定を行い、吸収することになる。

5.3.3 結果

位相合わせのデモンストレーションの結果を図 5.7 と図 5.8 に示す。図 5.7 が位相合わせ前の、11 台の PS board の位相測定の結果である。測定結果を見ると、最も位相の遅れているもの (図中灰色) に位相を一致するように、各 PS board の delay parameter を決定することができる。そうして決定した delay parameter を各 PS board の QSPI フラッシュメモリに SL からビットバンギングによって書き込み、自律型制御機構を最初から再び走らせることで、delay parameter を MMCM のシフトパラメータに反映させる。そうして、クロックの位相を合わせた後、もう一

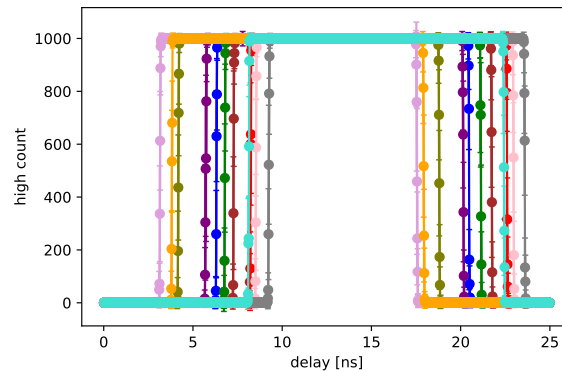
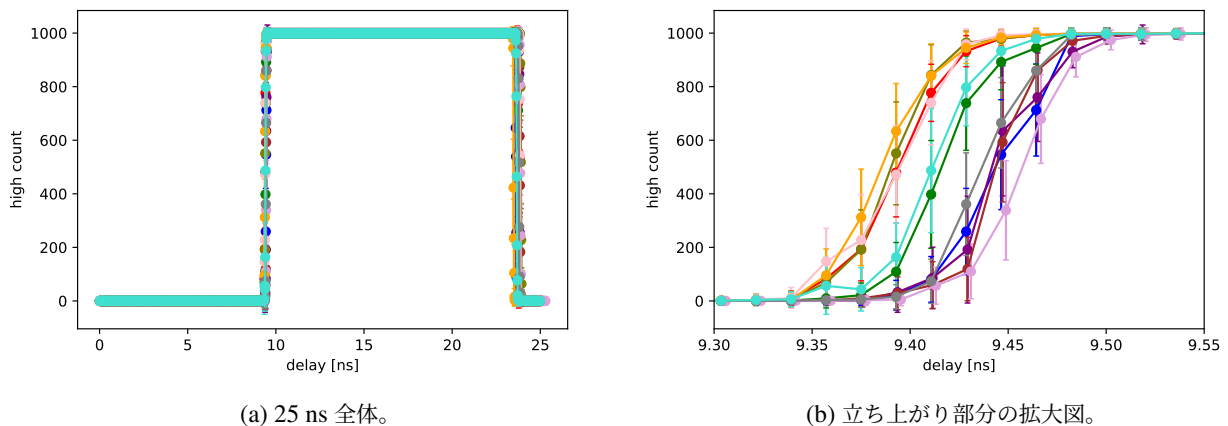


図 5.7: 11 台の PS board を用いた、位相合わせのデモンストレーション (位相合わせ前)。位相を合わせる前の、11 台の PS board の位相測定の測定結果を示している。この結果から 11 台の PS board の位相を最も位相が遅れているもの (図中灰色) に合わせるために必要な delay parameter を決定することができる。



(a) 25 ns 全体。

(b) 立ち上がり部分の拡大図。

図 5.8: 11 台の PS board を用いた、位相合わせのデモンストレーション (位相合わせ後)。位相を合わせた後の、11 台の PS board の位相測定の測定結果を示している。先ほど述べたように、最も位相が遅れていた PS board(図中灰色) に位相が一致するように残り 10 台の delay parameter を決定し、その delay parameter を QSPI フラッシュメモリに書き込んで、自律型制御機構を走らせ、MMCM の delay を変更している。これを見ると、11 台の PS board の位相が、約 50 ps の精度で一致していることが分かる。

度 JATHub による位相測定を行った結果を図 5.8 に示す。これを見ると、11 台の PS board の位相が、約 50 ps の精度で一致していることが分かる。この 50 ps は先ほど述べた 5.3.1 節で述べた、PS board の RX のリンクの取り直しに由来する。

ここで述べた結果は、5.3.2 節で述べたスキューを全て含んでいるが、このうち、Cat 6 ケーブルに由来するスキューと JATHub の LVDS レシーバに由来するスキューが位相測定にのみ影響を与える。JATHub の LVDS レシーバに関するスキューは本番のオペレーションでも事前測定を行い、位相測定に考慮に入れるが、一方で Cat 6 ケーブルに関しては運用上事前測定を行い、考慮に入れた位相測定を行うことが困難であるため、系統誤差として計上することになる。本テストベンチでもスキューに関してそのような扱いを行った。つまり、図 5.8 で示した結

果はあくまでも JATHub の位相測定を通した結果であるため、実際に PS board が再構成したクロックをオシロスコープでプローブして、最終的な位相のばらつきを見る。

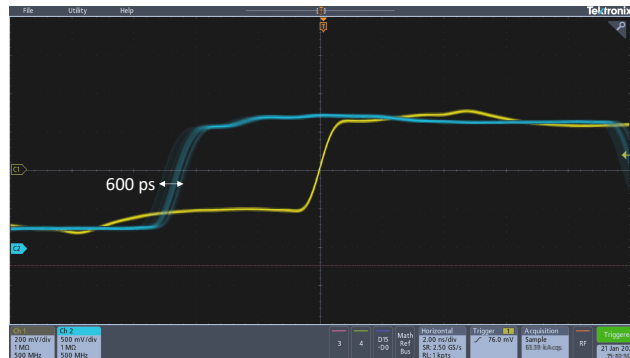


図 5.9: 11 台の PS board を用いた、位相合わせのデモンストレーション後の PS board のクロックの位相。

結果を図 5.9 に示す。この結果は、11 台の PS board の再構成したクロックをオシロスコープでプローブし、そのスナップショットを透明度を下げて重ねている。この結果を見ると、図 5.8 で示したクロックの位相からずれているが、これは Cat 6 ケーブルのスキューと一致することを確かめた。よって 4.2.4 節の結果と合わせて、実際のオペレーションでは、全 1434 台の PS board の位相が Cat 6 のスキューのみを系統誤差として受け入れた上で、図 5.8 で示したクロックの位相の不定性の約 50 ps を合わせた、650 ps が最終的な位相のばらつきとなり、これは要求値の 1 ns と比べて十分な精度である。また、この結果は、Cat6 ケーブルのスキューを事前測定し、位相測定を行う際に考慮に入れた場合は 50 ps 精度でのクロックの位相合わせが可能であることも示し、非常に高い精度で位相合わせが達成されていることが分かる。

5.4 読み出しシステムのデモンストレーション

この節では読み出しシステムのデモンストレーションについて述べる。まず、3.6 節で述べた、テストパルス発生機構のうち、shot モード (SL 上の MPSoC から 1 回テストパルストリガーを発生するように指示を出す) を用いて、単純なテストパルス発生機構に対して、ASD ボードから PS board、SL までの読み出し回路が上手く動いていることを確かめた。また、その際 SL や PS board 等の reboot に対しても再現性を持って、データ取得を行うことができた。これは、リードアウトパスが一貫して固定位相での実装ができていることを示す結果である。

次に 5.4.1 節で示すように、ASD ボードから PS board、SL までの読み出しシステムの検証と再現性および安定性の確認のために ASD delay curve の作成を行なった。これには 3.6 節で述べた、テストパルス発生機構のうち、auto モード (特定の BC パターンごとに自動で TPT を打ち続ける) を用いて、BCR に同期して、3564 バンチに 1 回 TPT を出す (≈ 11 kHz) 方法を取った。結果としては、全 11 台の PS board に対して、妥当な結果が得られ、読み出しシステムの再現性や安定性を示すものであると同時に、多数の ASD ボードや PS board (特に PS board 内の PP ASIC) の運用システムが洗練されていることを確認することができた。

最後に、5.4.2 節で述べるように、本番を想定した分布での平均 1 MHz での読み出し回路のストレスチェックを行なった。これには 3.6 節で述べた、テストパルス発生機構のうち、auto モード (特定の BC パターンごとに自動で TPT を打ち続ける) を用いて、図 3.19 に示すような分布毎に TPT を発行する方法を取った。結果としては、約 12% までヒット occupancy を上げて、読み出し回路は busy を出すことなく駆動することがわかり、これは本番の運用に対して十分耐えうる読み出しシステムが実装できていると言える。

5.4.1 ASD テストパルスを用いた delay curve の作成による、読み出しシステムの再現性・安定性の検証

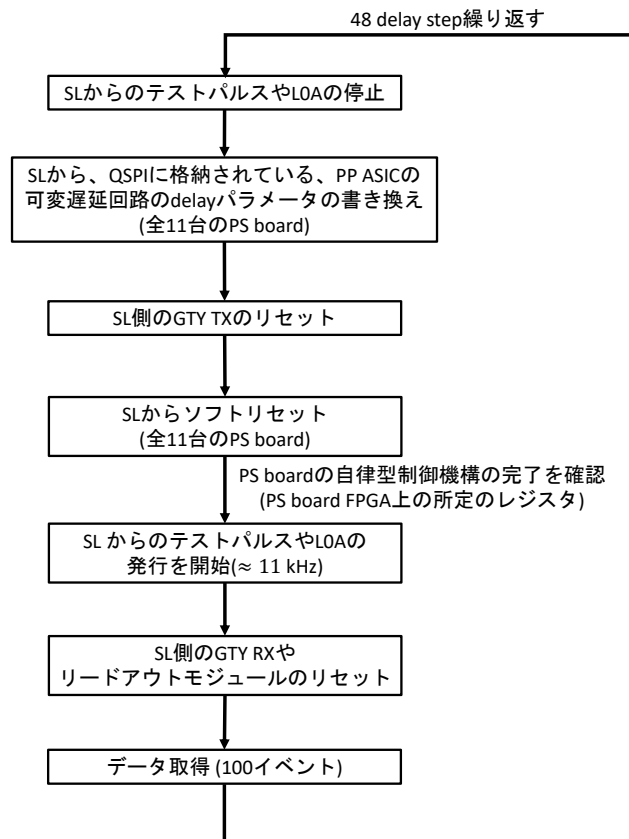


図 5.10: delay curve 作成の際の手続き。このステップは 11 台に対して並列に行われている。

まず、ASD テストパルスを用いた delay curve の作成のデモンストレーションについて述べる。delay curve は PP ASIC の可変遅延回路の delay を変化させながら ASD テストパルスの発出、データ読み出しを行い、その際、どのように BCID がされていくかを示すものである。これを行うことによって ASD から SL までの読み出しパスが正しく動いているかを検証することはもちろん、自律型制御機構を含めた制御系や固定位相での光リンクの実装など、タイミングパスが適切に実装されているかを検証することができる。また、delay curve の作成は実際のオペレーション及び、コミショニングの際に、適切な PP ASIC の遅延を決める際に行う手続きそのものである。図 5.10 に具体的な delay curve 作成の手続きを示す。これらの手続きは 11 台の PS board に対して並列に行われている。まず、SL から TPT や LOA の発行を停止する。次に、QSPI フラッシュメモリ内の、PP ASIC の可変遅延回路の delay パラメータを SL から、ビットバンギングを用いて書き換える。次に、SL から PS board のソフトリセットを行うことで、自律型制御機構が走り、QSPI フラッシュメモリ内のパラメータが PP ASIC に反映される。自律型制御機構が完了すると、PS board は FPGA 上の所定のレジスタに high を出力するので、SL は全 11 台の PS board が自律型制御機構が完了したことをこのレジスタを読むことで確認してから次のステップに進む。次に SL が TPT や LOA の発行を開始する。今回の delay Curve の作成では、BCR に同期した周波数 (≈ 11 kHz) で TPT や LOA の発行を行なった。最後にデータ取得を行う。今回は各 delay step 毎に 100 イベントを取得している。以上の手順を PP ASIC の可変遅延回路の delay パラメータを変えながら、合計 48 ステップ行う。

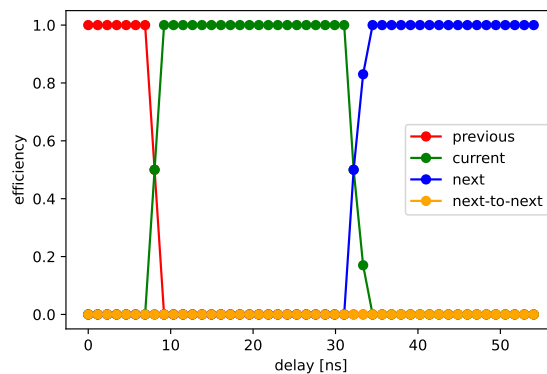


図 5.11: ASD テストパルスを用いて作成した delay curve。赤線が previous、緑線が current、青線が next、オレンジ色の線が next-to-next を示している。これを見ると PP ASIC の可変遅延回路の遅延を増やしていくと、バンチタグが予想通り変化していることが分かる。

図 5.11 に作成した delay curve を示す。残りの PS board 及び PP ASIC でも同様の delay curve が得られ、PP ASIC 間、PS board 間で立ち上がりと比較し、妥当な結果が得られた。ここでは割愛し、付録 B で示す。

本試験では、PP ASIC の可変遅延回路の 1 ステップを 1.19 ns に設定し、全体で 50 ns の遅延を行った。また、陽子バンチ識別回路の有効ゲート幅は 25 ns に設定したので、1 つのヒット信号には 1 つの BCID のみが付与される。図 5.11 における previous, current, next, next-to-next は SL 上でヒット信号に付与されるバンチタグであり、今は、SL の後段である、FELIX に送信するデータにおける情報をプロットしている。つまり、この delay curve を見ることで、ASD、PS board、Sector Logic の読み出しパスをくまなく検証することができる。図 5.11 を見ると PP ASIC の可変遅延回路の遅延を増やしていくと、ヒット信号に付与されるバンチタグは previous から始まり、current へと変化し、さらに 25 ns ほど遅延させると next へと変化しており、予想通りの結果を得ることができた。また SL や PS board の power cycle を行っても、全く同様の結果が得られたことから、Fixed latency でのタイミングの分配やヒットデータの送受信ができていないと結論できる。したがって以上より、ASD、PS board、SL の読み出しパスの検証、及びタイミング信号分配・受信や制御パスの検証ができたと言える。

この delay curve 作成は、11 台の PS board のデータ取得を並列に行なっており、また、本番で 1 台の SL には最大 31 台の PS board が接続されるが、全く同様のアプリを用いて 31 台の PS board を並列に扱い 31 台分の delay curve をワンクリックで作成することが可能である。delay curve の作成は、48 step 全体で約 5 分程度で完了するため、実際にここで実装されたファームウェア、ソフトウェアを用いて、ハードウェアやケーブルの配線ミス、コミショニングや実験の最中に手軽に検証することが可能であるため、価値のあるテストシステムを作成することができたと言える。

5.4.2 読み出し回路のストレスチェック

ASD テストパルスを用いた Delay Curve の作成は高速なデータ読み出しではなく、BCR に同期したデータ読み出し (≈ 11 kHz) で実装を行った。一方で、実際のオペレーションでは、トリガーレートは 1 MHz を想定しており、そのトリガーレートの元で読み出しシステムが busy を出すことなく動き続けられるかのストレスチェックも重要である。そこで、本テストベンチを用いて、1 MHz のテストパルストリガーレート及び LOA レートのもとの読み出しシステムの検証を行った。テストパルス発生機構は PS board の FPGA に実装したテストパルス発生機構を用いた。また、より実際のオペレーションに近づけるために、図 3.19 のような分布に従って、テストパルスを

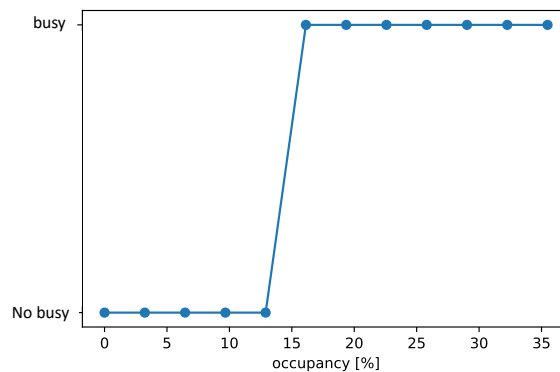


図 5.12: PS board FPGA テストパルスを用いた読み出し回路のストレスチェック。測定の間 busy が出なければ no busy と判定し、少しでも busy が出れば busy と判定しているような binary 判定である。

発行した。

結果を図 5.12 に示す。ここではそれぞれのステップに対して 2 時間読み出しを継続して行った。これは、約 7.2 ギガイベント分のテストパルスを発行したことになる。ここでは、イベントごとに occupancy の分布は変わらず、毎イベントで同じテストパターンを用いている。またこの結果は測定の間 busy が出なければ no busy と判定し、少しでも busy が出れば busy と判定している。つまり、どのくらいの間 busy が出ているかという busy rate を示したのではなく、no busy か busy が少しでも出たかの binary 判定であることに注意する。この結果を見ると occupancy は 12 % までは読み出し回路は busy を出さず、データロスなく、データ取得を行うことができているが、16 % 付近からは busy を出して、データロスが発生していることがわかる。

occupancy は大まかにパイラアップ μ に比例する。現在行われている LHC-ATLAS 実験 ($\mu = 20 \sim 60$) における、パイラアップと occupancy の関係を用いて、高輝度 LHC-ATLAS 実験 ($\mu = 200$) における occupancy を推定すると、TGC 全体で 0.127 % 程度になる見込みである。よって、これと比較すると、読み出し回路は十分、高輝度化に耐えうる実装がなされていると言える。

5.5 自律型制御機構の完了時間及び、読み出し回路への影響の評価

この節では、自律型制御機構の完了時間及び、読み出し回路への影響の評価について述べる。本テストベンチを用いて、自律型制御機構が完了するまでの時間を計測した。その結果と、実際のオペレーションで予想される SEU の発生確率から、SEU が発生した場合 (特に回復不可能な SEU) の読み出しシステムへの影響を評価した。

まずは、自律型制御機構の評価について述べる。図 5.13 に、自律型制御機構の完了に要する分布を示す。ここでは SL からソフトリセットを PS board にかけて、PS board がそれを受けてから、SL に GTX TX を返すまでの時間を計測している。ここでは、PS board における自律型制御機構の評価という意味で、SL における、リンクの取り直しに関しては含めていない。これを見ると、自律型制御機構が完了するまでにかかる時間 $\Delta t_{\text{reconfiguration done}}$ は、

$$\Delta t_{\text{reconfiguration done}} = 1.416 \pm 0.042 \text{ s} \quad (5.1)$$

と評価できる。

また図 5.13b を見ると、時間分布が見られているが、これは、図 5.13a に示すステップのうち、PS board の GTX RX 側が、RXSLIDE の回数が奇数になるまで、リセットを繰り返すことによる時間分布と、Si にパラメータの書き込みが完了してから、Si が安定したクロックを出力するようになるまでの時間分布の重ね合わせで説明できる。

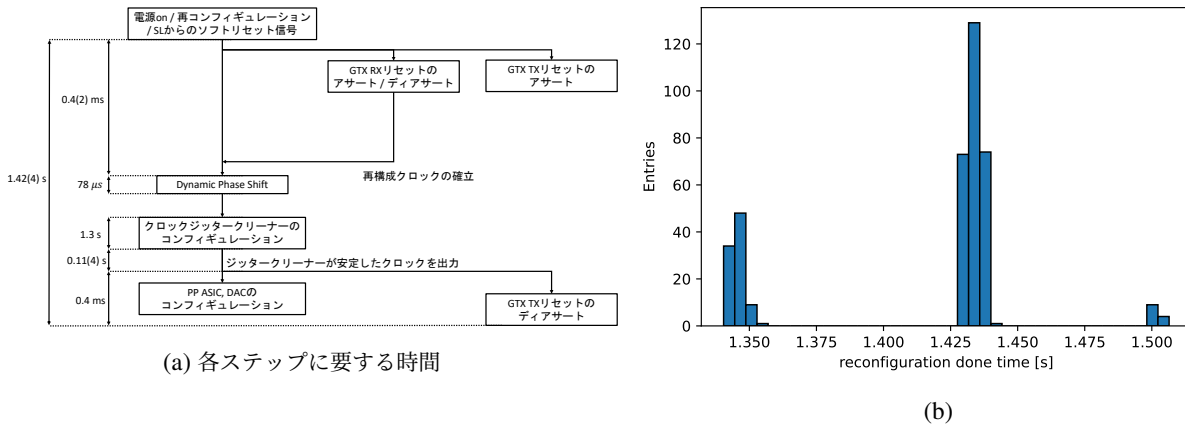


図 5.13: PS board の自律型制御機構の reconfiguration に要する時間。

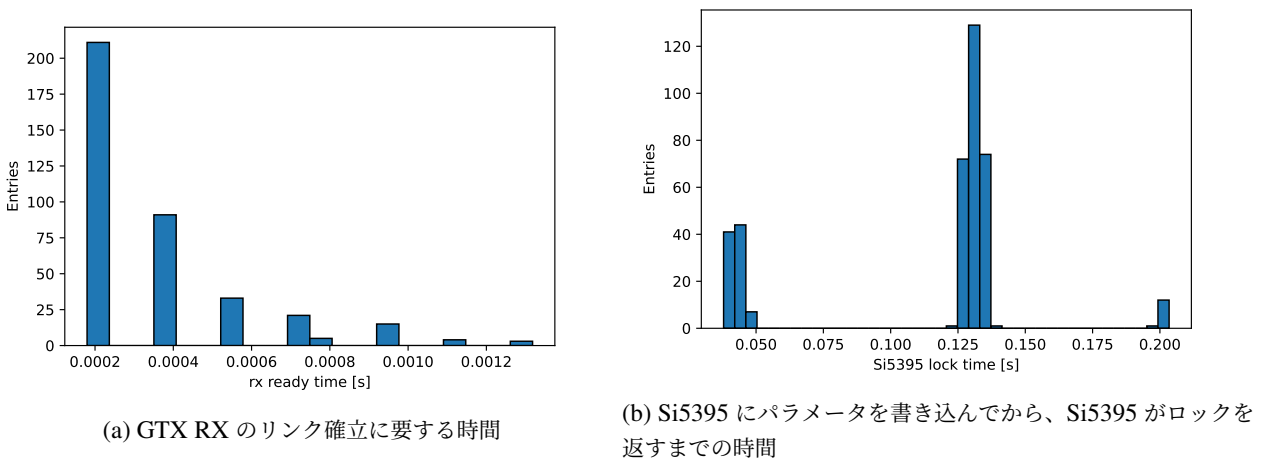


図 5.14: PS board の自律型制御機構の reconfiguration に伴う、SL における光リンク切れ時間の分布。

それらの時間分布を図 5.14 に示す。図 5.14a を見ると、おおよそ、0.2 ms ごとに、GTX RX のリンク確率に要する時間が離散的に分布していることが分かる。これは、RXSLIDE が偶数回だった場合、もう一度 GTX RX をリセットしてデータ・クロック再構成の手続きを最初から行うことになるが、これに要する時間はおおよそ 0.2 ms であり、また RXSLIDE の偶奇は完全にランダム、つまり 1/2 の確率で偶数、1/2 の確率で奇数になるため、図 5.14a のようなベルヌーイ分布が見られる。次に図 5.14b について述べる。これは Si のコンフィギュレーションに関するパラメータ空間に、FPGA からパラメータの書き込みが完了してから、Si が安定したクロックを出力するようになる (Si5395 の内部の PLL のロックを示す、LOLB ピンが High になる) までの時間分布を示している。これを見ると、全体の 25% ほどが 0.050 s 付近、72% ほどが 0.125 s 付近、3% ほどが 0.200 s 付近に分布をなしている。これに関しては特徴的な分布ではあるものの、現在使用している、クロックジッタークリーナーの Si5395 の Reference Manual [13] にそのような記述はなかった。しかし、PS board 間、つまりクロックジッタークリーナーの個体を変えても同様の分布が見られたことや、FPGA の reprogramming 等に対する再現性がとれていることから、この分布は現象として受け入れることとしている。

図 5.15 に、JATHub からのリセット信号によって PS board の FPGA reprogramming を行った際の、SL におけ

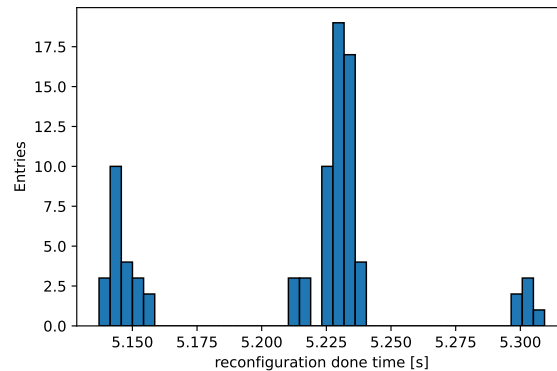


図 5.15: JATHub のリセットによる PS board FPGA reprogramming に伴う、SL における光リンク切れ時間の分布。

る光リンク切れ時間の分布を示す。ここでも先ほどと同様に、SL における、リンクの取り直しに関しては含めていない。リンク確立までに要する時間 $\Delta t_{\text{FPGA reprogramming}}$ は、

$$\Delta t_{\text{FPGA reprogramming}} = 5.213 \pm 0.044 \text{ s} \quad (5.2)$$

と評価できる。これを見ると、図 5.13 の時間分布に、加えて、おおよそ 3.8 s が FPGA が QSPI のファームウェア空間からファームウェアを読み込む際に要する時間であると結論できる。また、図 5.13 と図 5.15 を比較すると、時間分布の幅も大きくなっている。これは先行研究 [2] から、FPGA が QSPI のファームウェア空間からファームウェアを読み込む際に要する時間にはおおよそ 15 ms の時間ばらつきがあることがわかっており、本図でもその効果が重ね合わされていることが分かる。

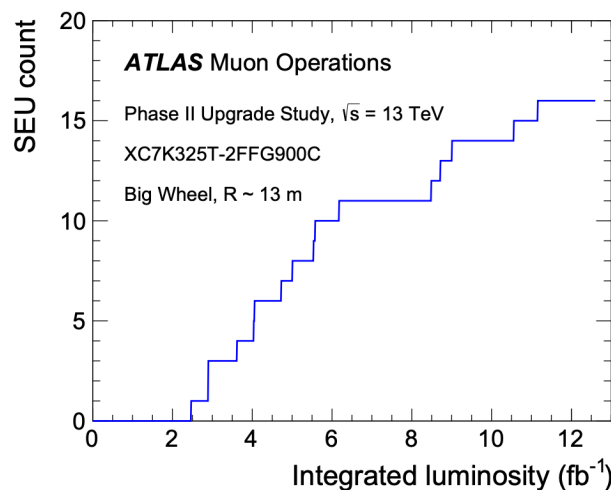


図 5.16: PS board に搭載されている FPGA における、積分ルミノシティと SEU が起こる回数の関係 [25]。

最後に高輝度 LHC-ATLAS 実験における PS board の FPGA 再コンフィギュレーションに伴う読み出し回路への影響について評価する。PS board において回復不可能な SEU はボードごとに起きうるが、そのたびに個々のボードが個別に再コンフィギュレーション、パラメータの再設定、リンクの再確立を行い、読み出しシステムへ復

旧するため、全体の DAQ システムを止めることはない。よって今から述べるように、1 台の PS board の挙動に着目して、データロスの評価すれば良い。

まず、PS board に搭載されている FPGA における、積分ルミノシティと SEU が起こる回数の関係は、図 5.16 で示される。これより、

$$\frac{\text{SEU count}}{\text{Integrated Luminosity}} \approx \frac{16}{12} \text{ fb} \quad (5.3)$$

と見積もることができる。この関係性は、ルミノシティによらず成立すると仮定する。また、参考文献 [26] から、300 回 SEU が発生すると、そのうち大まかに、1 回が回復不可能な SEU であると推定できる。よって、回復不可能な SEU に対しては、PS board はリンクリセットも含めた FPGA の再コンフィギュレーションを行うので、

$$\frac{\text{Unrecoverable SEU count}}{\text{Integrated Luminosity}} \approx \frac{16}{12} \times \frac{1}{300} \text{ fb} \quad (5.4)$$

となる。

高輝度 LHC-ATLAS 実験では、瞬間最高ルミノシティは $7.5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1} = 7.5 \times 10^{-5} \text{ fb}^{-1} \text{ s}^{-1}$ となる予定である。仮に、この瞬間最高ルミノシティの元で、高輝度 LHC-ATLAS 実験が行われるとすると、(5.4) 式から、

$$\frac{\text{Unrecoverable SEU count}}{\text{second}} \approx \frac{16}{12} \times \frac{1}{300} \times 7.5 \times 10^{-5} \text{ s}^{-1} \quad (5.5)$$

となる。よって、1 回、回復不可能な SEU が生じるまでの平均時間 $\Delta t_{\text{Unrecoverable SEU}}$ は、

$$\Delta t_{\text{Unrecoverable SEU}} = 1 / (\text{Unrecoverable SEU count/second}) \quad (5.6)$$

$$\approx 1 / \left(\frac{16}{12} \times \frac{1}{300} \times 7.5 \times 10^{-5} \text{ s} \right) \quad (5.7)$$

$$= 3.0 \times 10^6 \text{ s} \quad (5.8)$$

となる。これと (5.2) 式より、読み出し回路における回復不可能な SEU に由来するデータロスは、

$$\frac{\Delta t_{\text{FPGA reprogramming}}}{\Delta t_{\text{Unrecoverable SEU}}} < \frac{5.44}{3.0 \times 10^6} (5\sigma \text{ excluded}) \quad (5.9)$$

$$< 1.82 \times 10^{-4} \% \quad (5.10)$$

となり、これは十分小さい値である。

第 6 章

結論と今後の展望

本研究では、2029 年から運転を開始する予定の高輝度 LHC-ATLAS 実験において、ミューオンのトリガーを行う TGC 検出器エレクトロニクスシステムのアップグレードを通して、大規模なエレクトロニクスシステムのコミッショニングやオペレーションモデルの研究を行なった。高輝度 LHC-ATLAS 実験では、TGC 検出器は高帯域化とトリガーの高度化を目指して、一部を除いた全てのエレクトロニクスの刷新を行う。具体的には、TGC 検出器からの信号に BCID を行い、ヒットの有無に関わらず全てのヒットビットマップを後段 (SL) へ転送する PS board、TGC 検出器と磁場内部の検出器からの信号を用いてミューオンの飛跡・運動量の再構成を行う SL、PS board が再構成したクロックの位相測定と、PS board FPGA (TAM FPGA) における修復不可能な SEU への回復手続きを行う JATHub、JATHub へ位相測定のための基準クロックを分配するとともに、VME マスターとして JATHub との VME 通信を行う TAM がある。

TGC 検出器エレクトロニクスシステムは、タイミング信号分配・位相合わせ、読み出し、放射線損傷による FPGA の SEU への対処といった機能が、これらの複数のモジュールに渡って横断的かつ相補的に実装されている。よって、個々のモジュールの試験を行うだけでは不十分であり、TGC 検出器エレクトロニクスを全て含んだテストベンチを立ち上げ、システムレベルの統合試験を行う必要がある。本研究では、KEK に、ASD ボード (176 台)、PS board (11 台)、JATHub、TAM、SL からなるテストベンチを立ち上げ、TGC 検出器エレクトロニクスの統合試験を行なった。この統合試験において、

- ブート・コンフィギュレーションやモニター・コントロールシステムを遠隔から、安定して運用できていること
- 固定位相でのタイミング信号分配と位相合わせが十分な精度 (~ 50 ps) で達成されていること及び、システムレベル試験での経験から位相合わせに関わる誤差を評価すること
- ASD テストパルスを用いた読み出し回路の包括的な検証と、PS board FPGA テストパルスを用いた読み出し回路の 1 MHz のストレスチェックを行い、適切な実装がなされていること
- 自律型制御機構の評価を行い、FPGA に SEU が起こった際読み出しシステムへの影響が十分少ないこと

を確かめた。このテストベンチでは、大規模エレクトロニクスシステムへの応用を踏まえて、制御の並列化・自動化を行っており、1434 台の PS board を有する大規模なエレクトロニクスシステムにも耐えうるような安定性・拡張性を有したシステム、運用方法を確立することができた。また、配線等を含めて実際のオペレーションと全く同様のセットアップを組むことによって、コミッショニングやオペレーションへの移行が率直である点も重要である。本研究で開発を行なった、ファームウェアやソフトウェアは、コミッショニングの際にケーブリングやハードウェアの不具合をチェックするスタンドアロンテストベンチとして利用し、さらにその先にオペレーションのプロトタイプとしても利用するため、本研究の重要度は高い。また、高輝度 LHC-ATLAS 実験に限らず、大規模化するエレクトロニクスシステムに対して、一つのオペレーション・コミッショニングシステムを与えるものであり、

個々の技術も含めて、大規模エレクトロニクスシステムのスマート化を行うものと言える。

今後は、まず、SLのトリガーロジックの実装を行い、統合試験の中でトリガーロジックの検証を行う必要がある。その上で、PS boardの数を31台まで増やし、1/24セクター全体を模倣するようなテストベンチを作成することで、オペレーションモデルをさらに洗練させることができる。加えて、現在PS boardとTAMは量産に向けて進んでいるが、これらの量産機に対して、個体差の測定を含めた全数試験を行い、本番測定の際に校正をかけることができる様な準備をする必要がある。特にPS boardに関しては、TGC検出器全体で1434台あるため、ここでも並列化・自動化を意識して、QAQCの枠組みを整備していく必要がある。

謝辞

本研究を行うにあたって、多くの方々にお世話になりました。この場を借りて、お礼を申し上げたいと思います。まず、指導教員である奥村恭幸准教授には大変お世話になりました。学部生の頃から気にかけてくださり、素粒子物理及び ATLAS 実験に関する知識を深めることができました。修士課程に進学してからも、研究方針から技術的な助言まで、具に相談に乗ってくださいました。また、石野雅也教授にも大変お世話になりました。特に、石野さんには鋭い助言をいただき、研究を深化させることができました。感謝いたします。

また、研究室ミーティング等で、日頃から研究に関する助言をくださったスタッフの皆様には感謝いたします。増渕達也助教、斎藤智之助教には、現行のシステムの視点に立って、様々な助言をしてくださいました。他の ICEPP の先生方にも大変お世話になりました。浅井祥仁センター長には組織の長として、このような研究テーマを与えてくださったことに感謝いたします。田中純一教授、澤田龍准教授には、ICEPP 夏の学校等で大変お世話になりました。特に、澤田龍准教授が学部3年生に向けて開講しているゼミがきっかけで、自分は ATLAS 実験に興味を持つようになりました。他にもセミナー等で、ICEPP の先生、学生の皆様には大変お世話になりました。また、秘書の皆様のお陰様で自分は研究に専念できていると思っています。ありがとうございます。

KEK でのテストベンチの立ち上げには非常にたくさんの方にお世話になりました。高エネルギー加速器研究機構素粒子原子核研究所の佐々木修氏には、特にハードウェア面での助言をいただき、大変助かりました。また、共同で高輝度 LHC-ATLAS 実験に向けたアップグレードの研究を行っている、TGC グループの皆様にもお世話になりました。戸本誠氏には TAM の制作も含めて、大変お世話になりました。堀井泰之氏には、PS board に関する助言をたくさん頂いたとともに、学会やセミナー等でも大変お世話になりました。前田順平氏には、ミーティングを通して、研究に様々なフィードバックを下さり、研究を深めることができました。青木雅人氏には、KEK のネットワーク設定等でお世話になりました。ありがとうございます。

奥村研究室の同期である成川さんには、お世話になっているとともに、良き競争相手として一緒に研究を行うことで、研究により没頭することができました。また、奥村・石野研究室の先輩方である田中さん、杉崎さん、林さん、青木さん、山下さん、三島さんには大変お世話になりました。エレクトロニクスに関して全くの無知であった自分に、辛抱強く教えてくださったと思います。また、名古屋大の橋本さん、和田さんには、KEK でのテストベンチや学会等を通して、お互いの研究に忌憚のない意見を言い合うことができました。後輩である、牧田さん、近藤さんにはテストベンチの立ち上げや TAM の QAQC を手伝っていただきました。感謝申し上げます。

以上に述べた方の他にも、多くの方々にお世話になりました。本研究に協力していただいたことに感謝いたします。そして最後に、家族に感謝します。

付録

A 固定位相でのタイミング信号分配・受信に必要な実装

A.1 受信側 (RX) に必要な実装

まずは受信側 (RX) に必要な実装について述べる。以下の議論では特に PS board FPGA の kintex-7 series GTX トランシーバ [7] について述べるが、本議論はそのまま TAM FPGA の kintex-7 series GTX [7] や SL FPGA の Ultrascale GTY トランシーバ [8] に対しても有効である。よって SL、PS board、TAM 全てに対して、固定位相での実装が必要な箇所は以下の方法で実装を行なっている。

図 A.1 に GTX RX のブロック図を示す。受信部のクロックドメインは RX Serial Clock, RX PMA Parallel Clock (RX XCLK), RX PCS Parallel Clock (RXUSRCLK), FPGA RX Parallel Clock (RXUSRCLK2) の 4 つに分かれている。

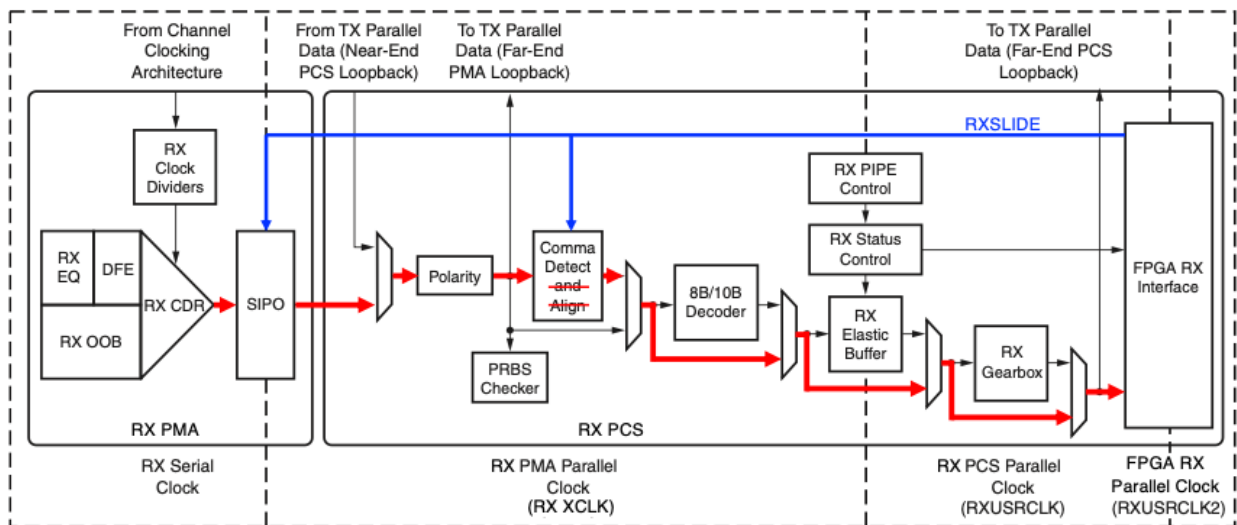


図 A.1: GT トランシーバのデータ受信部分 (RX) のブロック図 [10]。

まず、RX Serial Clock の生成の仕方について述べる。RX Serial Clock は図 A.1 の RX CDR というモジュールから生成され、SIPO のシリアル部分に供給される。RX CDR における RX Serial Clock の再構成の仕方を図 A.2 に示す。RX CDR では RXREFCLK として、オンボードから受け取ったクロックを逡倍して、RX PLLCLK を生成する。PS board の場合は RXREFCLK は 40.079 MHz、RX PLLCLK は 4 GHz である。こうして生成した RX PLLCLK を基準として、図 A.2 のように、シリアルデータのエッジと RX PLLCLK の立ち上がりもしくは立ち下がりが同期するように、RX PLLCLK の位相と周波数を微調整する。この時、図 A.2 に示すように、2 種類の RX Serial Clock が再構成され得る。後で述べるように、RXSLIDE の回数の偶奇にこの 2 種類が対応する。よって

RXSLIDE の偶奇をどちらかに定めて、それが達成されるまで、GTX RX のリセットを繰り返す実装を行なった。こうすることによりデータに対して、一意に RX Serial Clock を再構成することができる。こうして再構成された RX Serial Clock を 1/20 に分周されて、RXOUTCLK (200.395 MHz) が生成される。RX Serial Clock, RX PMA Parallel Clock (RX XCLK) の境界部分は SIPO というモジュールを用いてデータの受け渡しが行われる。ここでは、RX Serial Clock に同期して入力されたシリアルデータが、RX PMA Parallel Clock に同期した形で、40 ビットの平行データとして取り出される。この際、RX PMA Parallel Clock (RX XCLK) として、RXOUTCLK を用い、RX Serial Clock, RX PMA Parallel Clock (RX XCLK) の位相を一意に定めた。こうすることで RX Serial Clock, RX PMA Parallel Clock (RX XCLK) の境界部分で位相の不定性が生じないような実装になる。

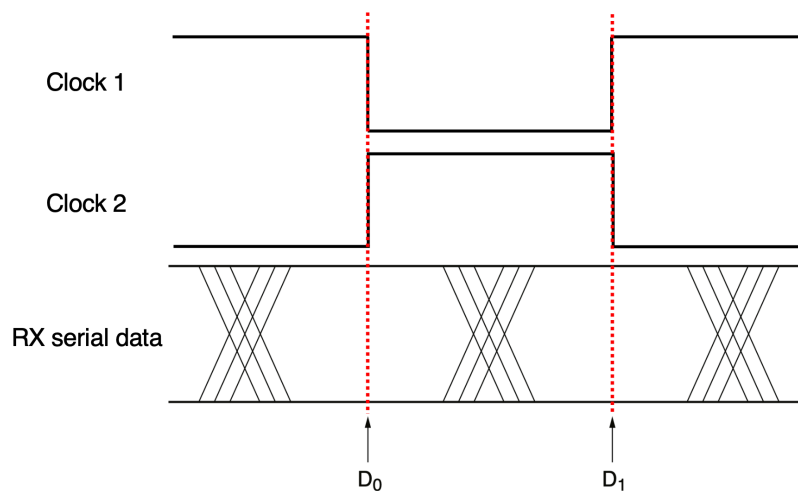


図 A.2: RX CDR におけるクロックの再構成の仕方 [10]。

RXOUTCLK は 2 種類の RX Serial Clock のどちらを取るか、及び RX Serial Clock の分周 1/20 の不定性の合計 40 種類の位相の不定性を持つ。よって、固定位相でのタイミング信号受信のためには、何らかの方法で、40 種類の位相の中から、1 種類を選び出すような仕組みが必要である。本実装では、これから述べるように、一般に GT トランシーバの受信部に備わっている、8b/10b decoder の機能を除き、手動で 8b/10b decode を行うことによって、位相を一意に定めている。

GTX RX 部には、図 A.1 に示すように Comma Align という機能が備わっている。これは、RX PMA Parallel Clock (RX XCLK) で駆動する。Comma Align を有効にすると、RX PMA Parallel Clock (RX XCLK) は固定して、シリアルデータを 1 ビットずつ動かすことによって、8b/10b の予約語である Comma を見つけ、後に繋がる 8b/10b Decoder というモジュールにデータを受け渡す。こうして 8b/10b Decoder に渡された 40 ビットの平行データは decode され 32 ビットのデータとなる。しかし、この方法だと、RX PMA Parallel Clock (RX XCLK) の位相が Comma の位置に対して一意に定まらない。よって、本実装では、Comma Align と 8b/10b Decoder の機能を除き、40 ビットの平行データをそのまま GTX RX の出力側 (図 3.10) へと受け渡し手動で 8b/10b decode を行う。この際、GTX RX にある RXSLIDE という機能を用いた。RXSLIDE は打ち込んだパルスの分だけシリアルデータと RXOUTCLK のどちらもシフトする機能である。具体的には、RXSLIDE に 1 回パルスを打ち込むと、シリアルデータが 1 ビットシフトする。その際に、RXOUTCLK の位相は変化しない。もう一度 RXSLIDE に 1 回パルスを打ち込むと、シリアルデータがさらに 1 ビットシフトして、かつ RXOUTCLK が 2 UI (シリアルデータ 2 ビット分) だけシフトする。このように RXSLIDE ごとのシリアルデータは 1 ビットシフトするが、RXOUTCLK は RXSLIDE が 2 回ごとに位相が 2 UI だけ変化する。よって RXSLIDE 機能を用いて、40 ビット

の平行データの下位 10 ビット (表 3.1) から Comma を見つけるまで RXSLIDE を打つという実装を行う。こうすることで、RXOUTCLK の位相は 2 種類に定まる。この 2 種類は RXSLIDE ごとにシリアルデータは 1 ビットシフトしていくのに対して、RXOUTCLK の位相は 2 回ごとに 2 UI だけシフトすることから起因する。よってこの 2 種類は RXSLIDE の偶奇に対応するが、RXSLIDE の偶奇は、一度シリアルデータの再構成が完了すると一意に定まっているが、この偶奇をユーザーロジック側から一意に設定することはできない。本実装では RXSLIDE の回数が奇数となるまで、GTX RX 部をリセットし 8b/10b decode の手続きを繰り返すという実装を行なった。こうすることで、40 種類ある位相の中からただ 1 つの位相を選択することができる。ただし、例えば同一のボードでのループバック等で、送信側と受信側の reference clock が同期している場合はリセットを行なっても RXSLIDE の偶奇は変化しないため、偶奇のどちらを選択するかを一意的に選ぶ必要がある。

RX PMA Parallel Clock (RX XCLK), RX PCS Parallel Clock (RXUSRCLK) は一般には位相関係定まっていないので、間に RX Elastic Buffer を挟んでデータの受け渡しを行う。この場合、そもそも、RX PMA Parallel Clock (RX XCLK), RX PCS Parallel Clock (RXUSRCLK) をどちらも同じソース (RXOUTCLK) から駆動するように変更する。こうすることで、2 者の位相関係は一意に定まる。また、図 A.1 に示すように、RX Elastic Buffer をバイパスするような実装を行うことで、固定位相でのデータ送信が達成される。RX PCS Parallel Clock (RXUSRCLK) と FPGA RX Parallel Clock (RXUSRCLK2) は同一のソースからなるクロックであるので、位相関係は固定されているので、特別な実装は必要ない。このようにして、固定位相での RXUSRCLK2(RXOUTCLK) とデータ再構成が可能になる。

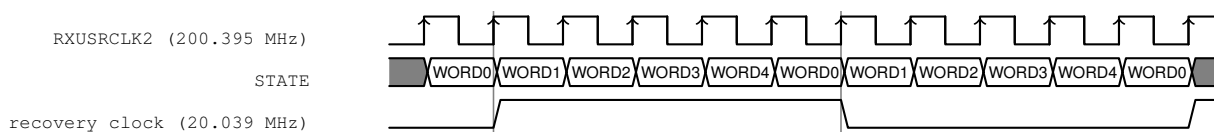


表 A.1: HEADER からのクロック再構成手法 (RX packet deformer 内部)。

こうして固定位相で再構成された RXUSRCLK2(200.395 MHz) から LHC CLK (40.079 MHz) を生成する際にも特別な措置が必要である。それを行なっているのが、図 3.10 の RX packet deformer である。単純に Xilinx 社の提供する clocking wizard IP コア [10] の、MMCM や PLL を用いて、200.395 MHz クロックを 1/5 に分周すると、5 種類の位相の不定性が生じる。よって、表 A.1 に示すように、再構成したデータの WORD0 で立ち上がりもしくは立ち下がりが来るようなクロックを生成する。このクロックの周波数は 20.039 MHz であるので、これを clocking wizard IP コア [10] の、MMCM を用いて 40.079 MHz へ通倍する。通倍の際は位相の不定性は生じないため、こうして生成された 40.079 MHz クロックは、データに完全に同期した形で固定位相で再構成される。この 40.079 MHz クロックは、MMCM に通しているため、ある程度低ジッターではあり、FPGA の内部のロジックに使う際には問題にならない。しかし GTX の reference clock として用いる場合には、ジッターが問題になる。よって、図 3.10 のように PS board のオンボードのクロックジッタークリーナーに入力して、FPGA には同じ周波数 40.079 MHz、GTX TX の reference clock には 200.395 MHz に通倍してから供給する。この際に、クロックジッタークリーナーのコンフィギュレーションとして、Zero delay mode を選択する。こうすることによって、入力クロックと出力クロックが同じ位相であることが保証される。

A.2 送信側 (TX) に必要な実装

次に送信側 (TX) に必要な実装について述べる。

まずは GTX の TX 部分に必要な実装について述べる。図 A.3 に GT トランシーバのデータ送信部 (TX) のブ

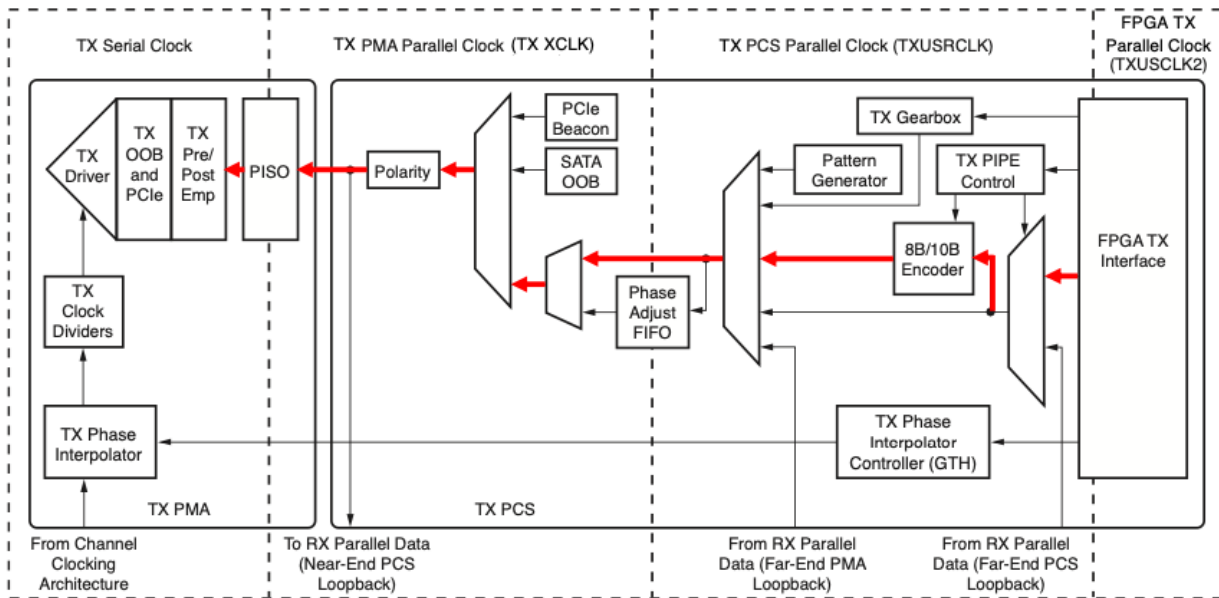


図 A.3: GT トランシーバのデータ送信部分 (TX) のブロック図 [10]。TX には 8b/10b Encoder、Phase Adjust FIFO、PISO 等のデータ送信に必要な機能揃っている。赤線が PS board, SL の TX 部分に実装されたデータの送信パスである。

ブロック図を示す。送信部のクロックドメインは、TX Serial Clock, TX PMA Parallel Clock (TX XCLK), TX PCS Parallel Clock (TXUSRCLK), FPGA TX Parallel Clock (TXUSRCLK2) の4つに分かれている。このうち、TX PMA Parallel Clock (TX XCLK) と TX PCS Parallel Clock (TXUSRCLK) は一般には位相関係が定まっていないので、境界部分に Phase Adjust FIFO を挟んでデータの受け渡しを行なっている。この場合、そもそも TX PMA Parallel Clock (TX XCLK) と TX PCS Parallel Clock (TXUSRCLK) の位相関係に不定性が生じていることに加えて、Phase Adjust FIFO は位相の吸収の仕方に任意性をもつため、固定位相でのデータ送信は達成されない。よって、まず TX PMA Parallel Clock (TX XCLK) と TX PCS Parallel Clock (TXUSRCLK) をどちらも同じソース (TXOUTCLK) から駆動するように変更する。こうすることで、2者の位相関係は一意に定まる。また、図 A.3 に示すように、Phase Adjust FIFO をバイパスするような実装を行うことで、固定位相でのデータ送信が達成される。TXOUTCLK は、GTX 送信部のロジックのクロック周波数と、FPGA の GTX reference clock のピンに供給されるクロックの周波数が一致していない場合は、GTX reference clock を分周して、適切な TXOUTCLK を作り出す。したがって、TXOUTCLK の位相を GTX reference clock に対して一意に定めるためには、GTX 送信部のロジックのクロック周波数と、FPGA の GTX reference clock のピンに供給されるクロックの周波数を一致させる必要がある。具体的には、表 3.1 と表 3.2 で示した通り、SL と PS board 間の通信では、ラインレート 8.016 Gbps であり、40 ビットの平行データを 200.395 MHz で扱うことに対応する。したがって、GTX 送信部のロジックのクロック周波数は 200.395 MHz となる。これに対して、GTX reference clock 周波数が、240.474 MHz であったりと 200.395 MHz を生成する際の分周で位相の不定性が生じる。よって、GTX reference clock 周波数は 200.395 MHz にする必要がある。また、図 3.10 に示すように固定位相で再構成した LHC CLK (40.079 MHz) をクロックジッタークリーナーに入れて、逡倍した 200.395 MHz クロックを GTX TX reference クロックとして用いている。こうすることによって再構成した LHC CLK (40.079 MHz) に対して、TXUSRCLK2 の位相関係を一意に定めることが可能になる。

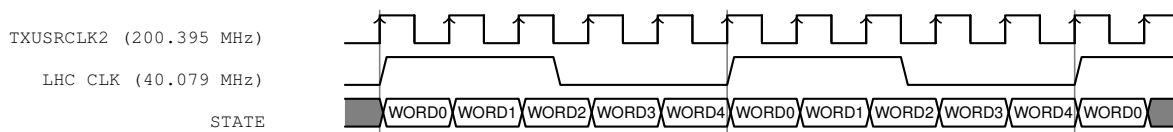


表 A.2: LHC CLK の立ち上がりとデータパッキングの関係 (TX packet former 内部)。

さらに、固定位相でのデータ送信を行うためには、TXUSRCLK2 の位相に対して、どのようにデータをパッキングするかを一意に定める必要がある。これを行うモジュールが、図 3.10 の、TX packet former である。表 A.2 に TX packet former 内部のデータパッキングの様子を示す。一般には、TXUSRCLK2 に対して特別な措置を行わないと、LHC CLK の立ち上がりと、WORD 0, WORD 1, WORD 2, WORD 3, WORD 4 の位相関係は一意に定まらない。したがって、必ず LHC CLK の立ち上がりと、WORD 0 の位相関係が表のように固定されているようなパッキングの仕方を実装した。これにより、データパッキングの仕方が LHC CLK と固定の位相関係を持つようになる。

B 統合試験における全 11 台の ASD テストパルスを用いた delay curve の結果

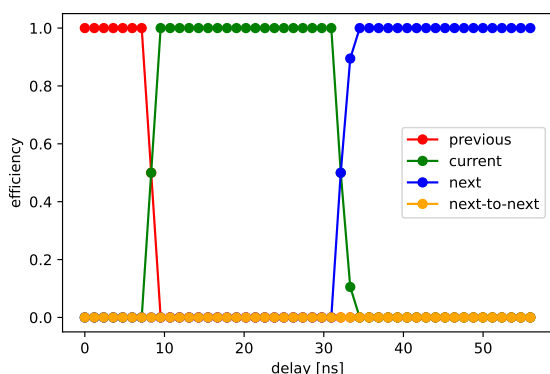


図 B.4: PS board 1 の ASD1

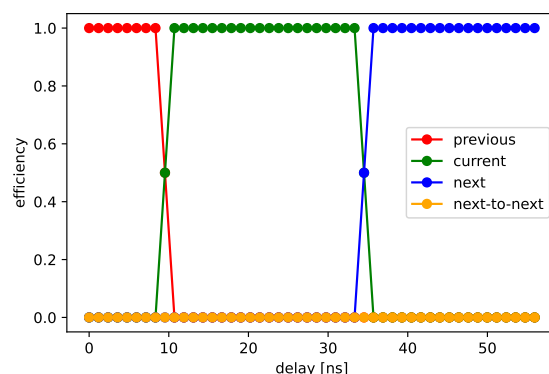


図 B.5: PS board 1 の ASD2

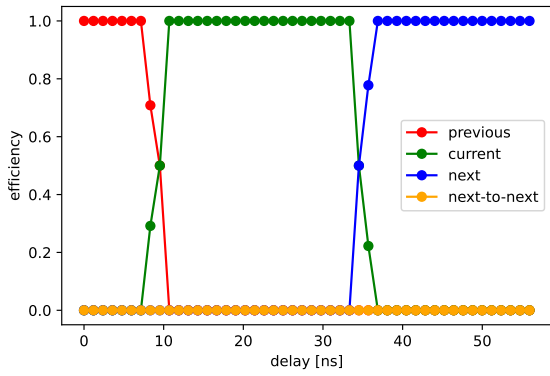


図 B.6: PS board 1 の ASD3

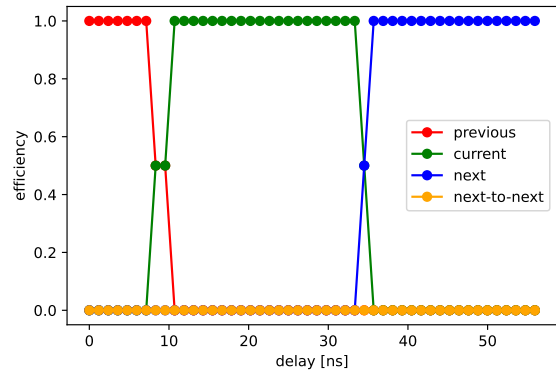


図 B.7: PS board 1 の ASD4

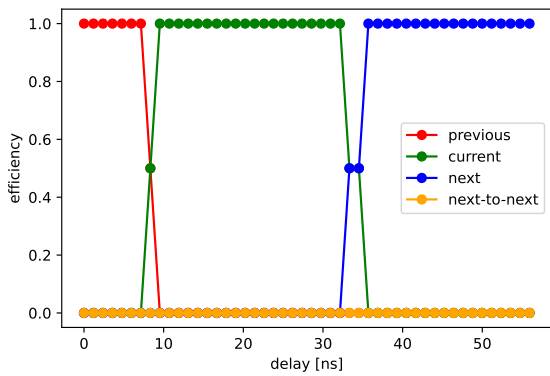


図 B.8: PS board 1 の ASD5

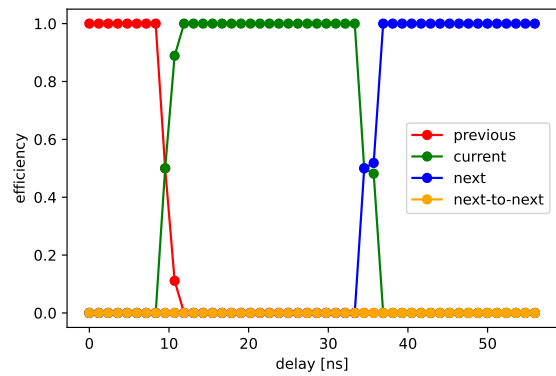


図 B.9: PS board 1 の ASD6

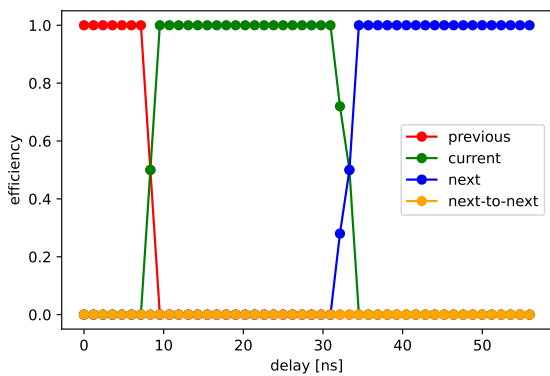


図 B.10: PS board 1 の ASD7

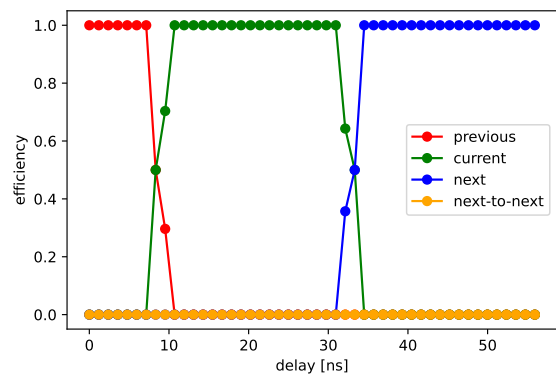


図 B.11: PS board 1 の ASD8

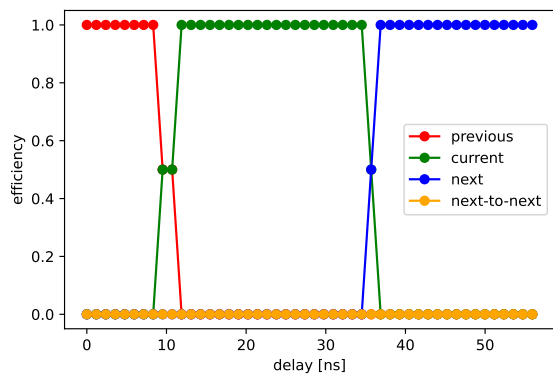


図 B.12: PS board 1 の ASD9

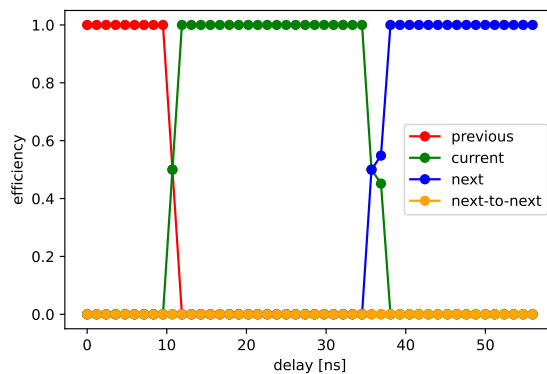


図 B.13: PS board 1 の ASD10

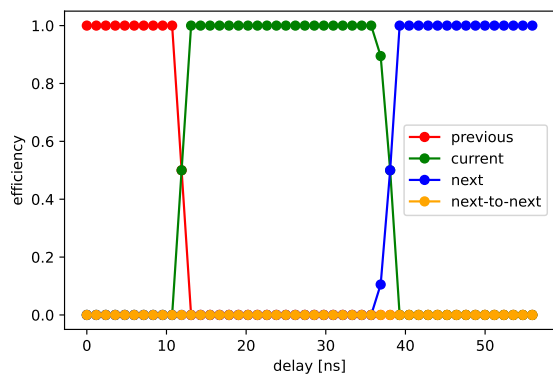


図 B.14: PS board 1 の ASD11

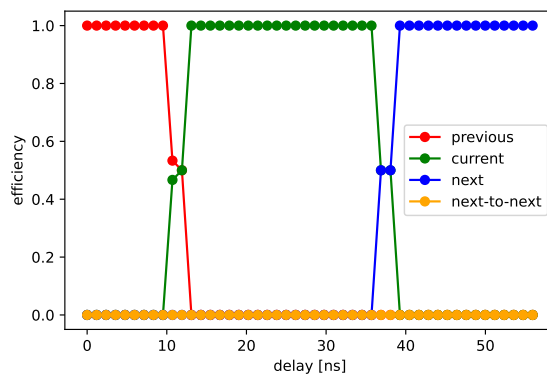


図 B.15: PS board 1 の ASD12

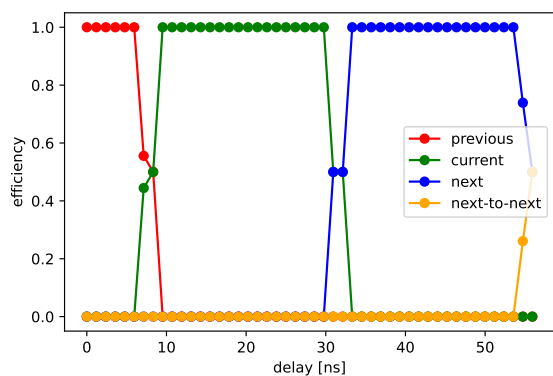


図 B.16: PS board 1 の ASD13

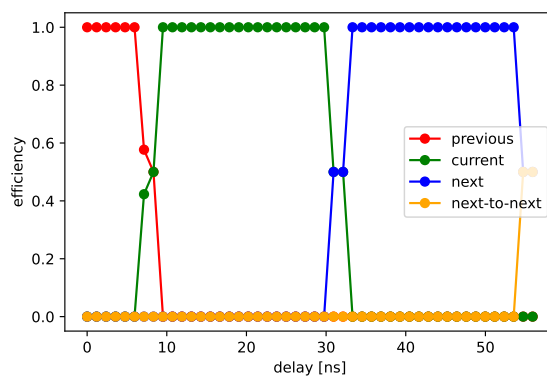


図 B.17: PS board 1 の ASD14

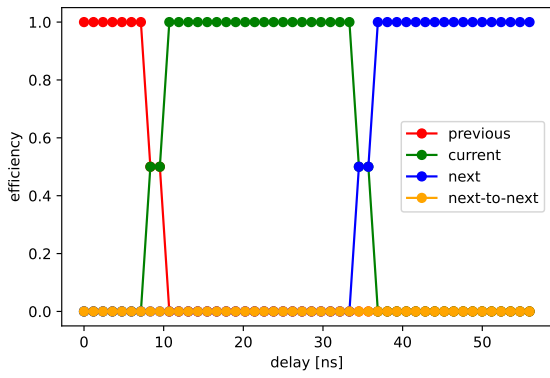


図 B.18: PS board 1 の ASD15

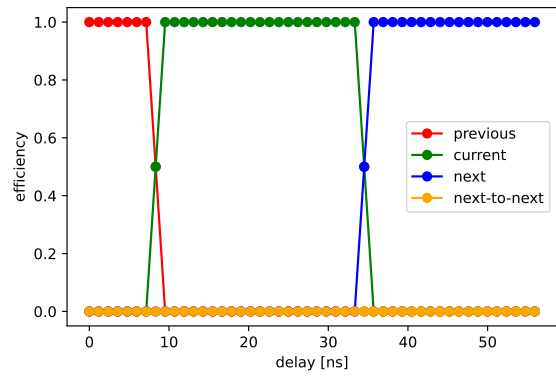


図 B.19: PS board 1 の ASD16

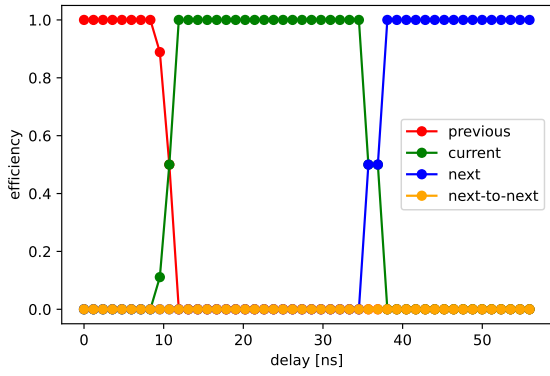


図 B.20: PS board 2 の ASD1

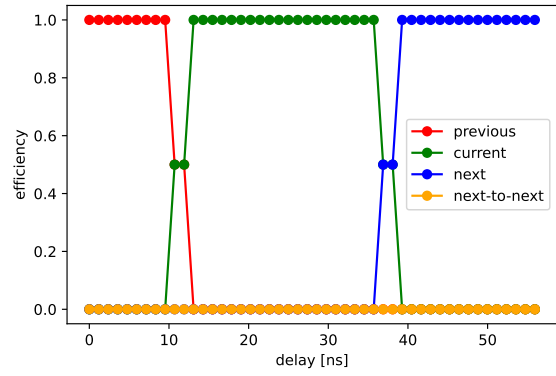


図 B.21: PS board 2 の ASD2

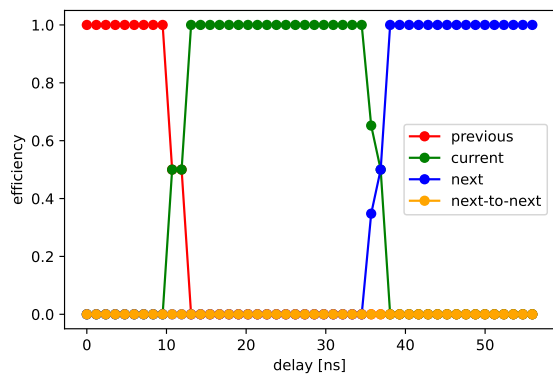


図 B.22: PS board 2 の ASD3

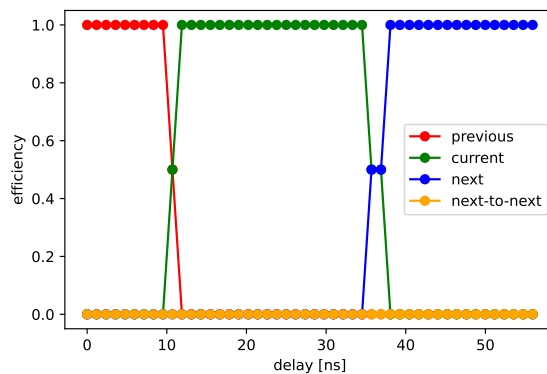


図 B.23: PS board 2 の ASD4

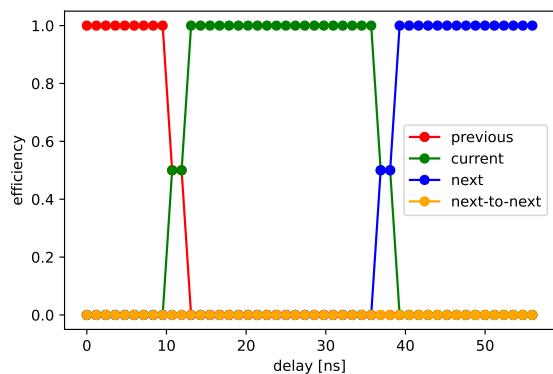


図 B.24: PS board 2 の ASD5

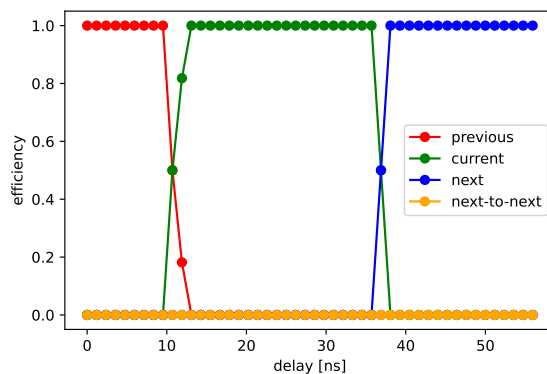


図 B.25: PS board 2 の ASD6

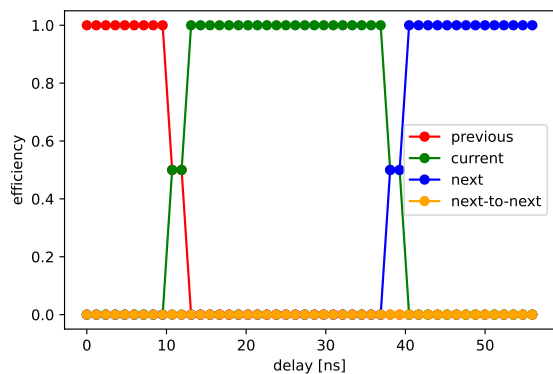


図 B.26: PS board 2 の ASD7

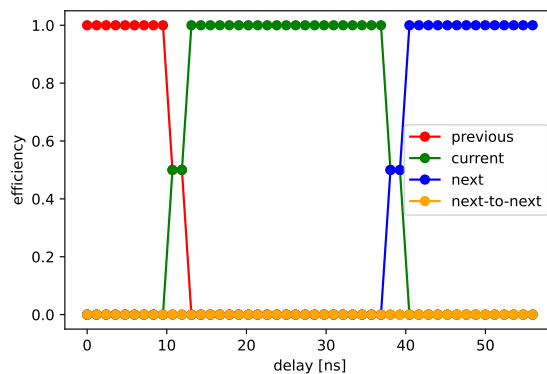


図 B.27: PS board 2 の ASD8

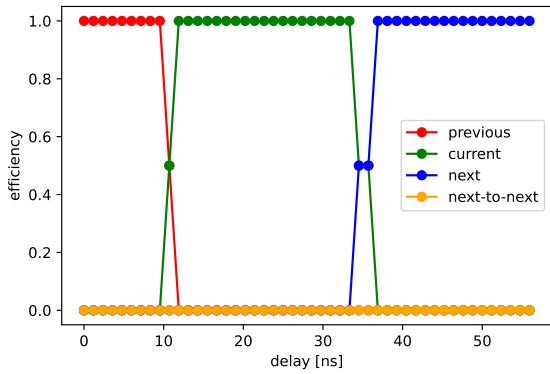


図 B.28: PS board 2 の ASD9

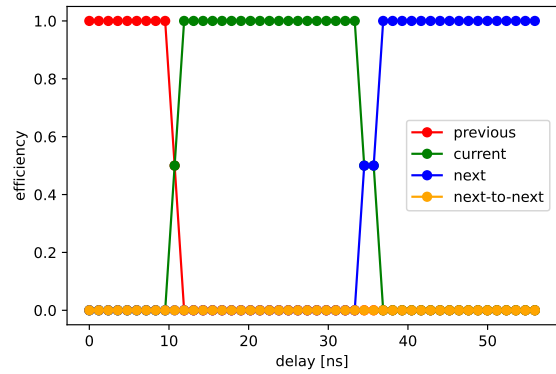


図 B.29: PS board 2 の ASD10

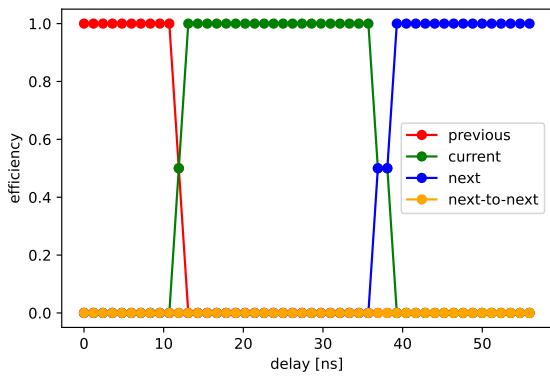


図 B.30: PS board 2 の ASD11

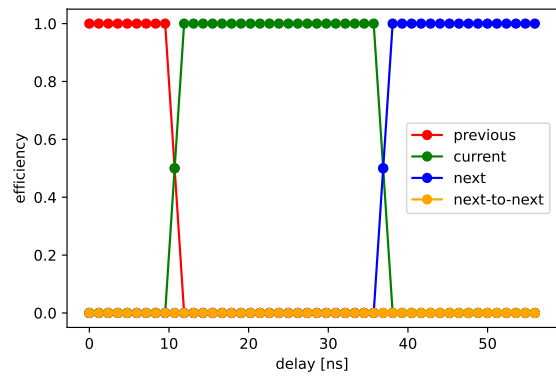


図 B.31: PS board 2 の ASD12

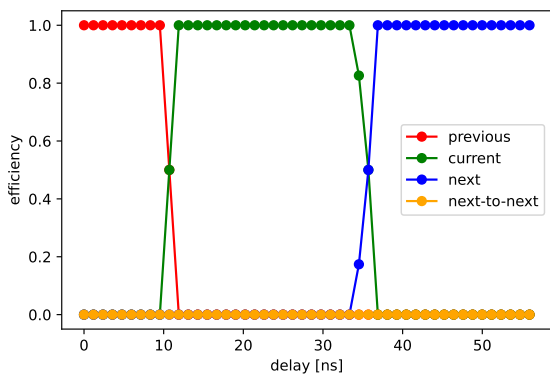


図 B.32: PS board 2 の ASD13

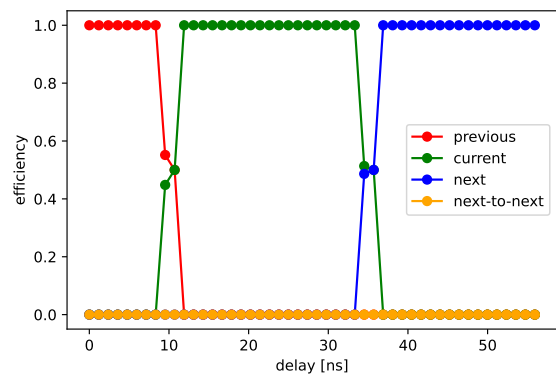


図 B.33: PS board 2 の ASD14

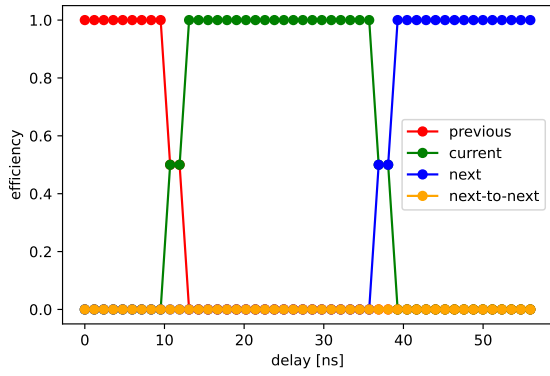


図 B.34: PS board 2 の ASD15

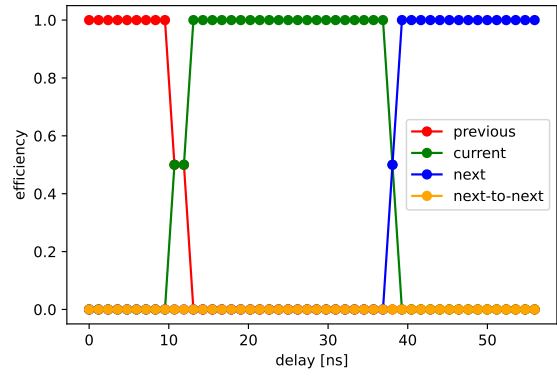


図 B.35: PS board 2 の ASD16

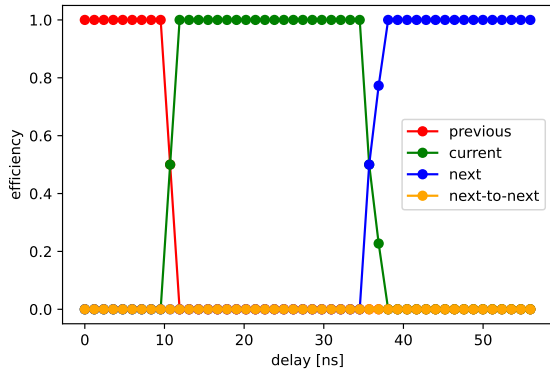


図 B.36: PS board 3 の ASD1

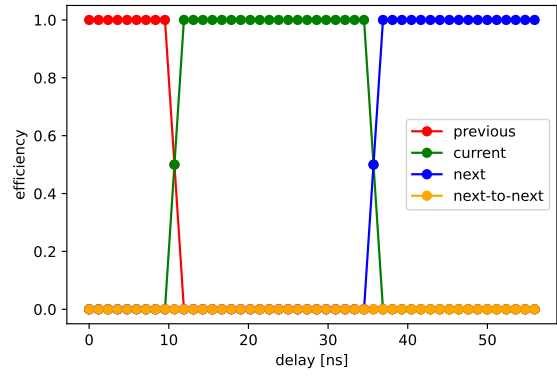


図 B.37: PS board 3 の ASD2

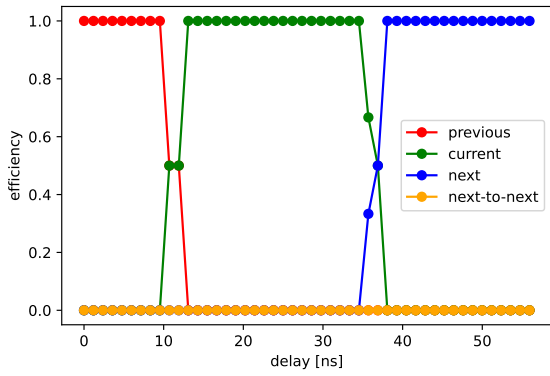


図 B.38: PS board 3 の ASD3

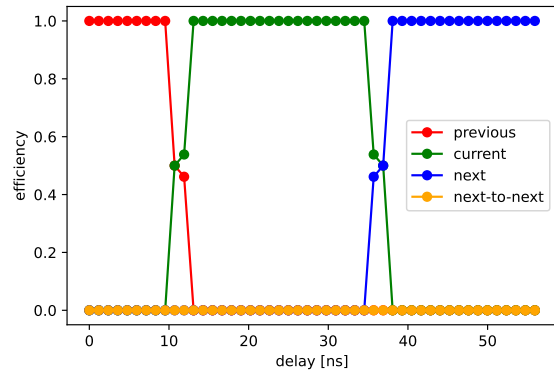


図 B.39: PS board 3 の ASD4

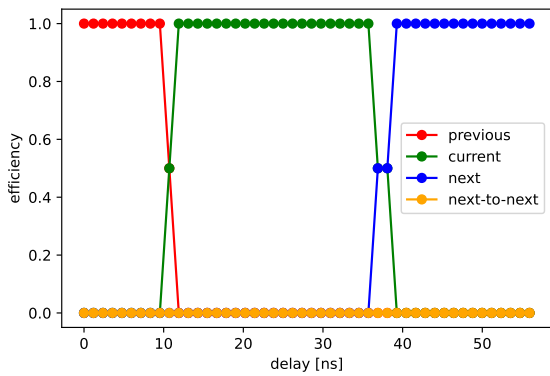


図 B.40: PS board 3 の ASD5

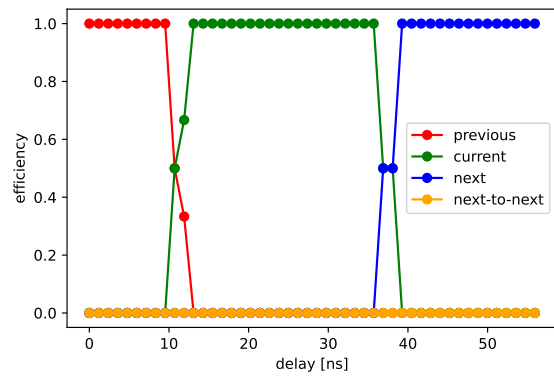


図 B.41: PS board 3 の ASD6

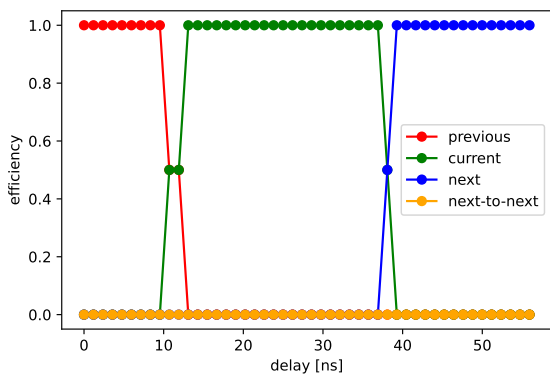


図 B.42: PS board 3 の ASD7

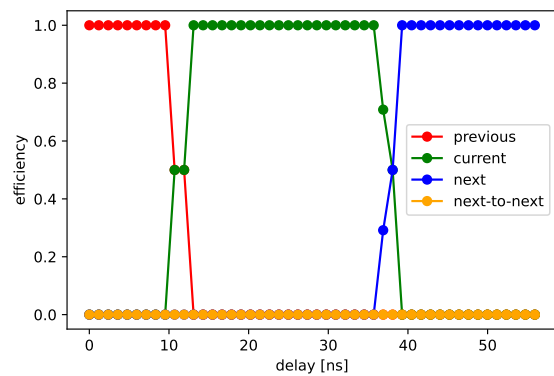


図 B.43: PS board 3 の ASD8

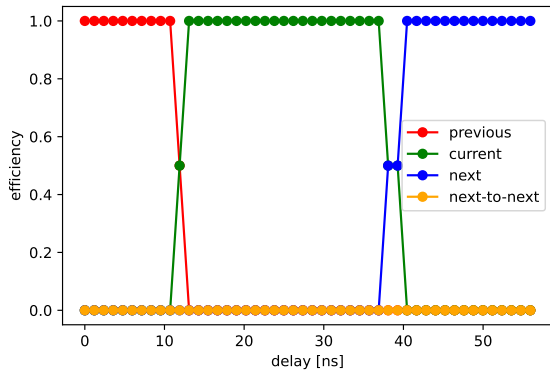


図 B.44: PS board 3 の ASD9

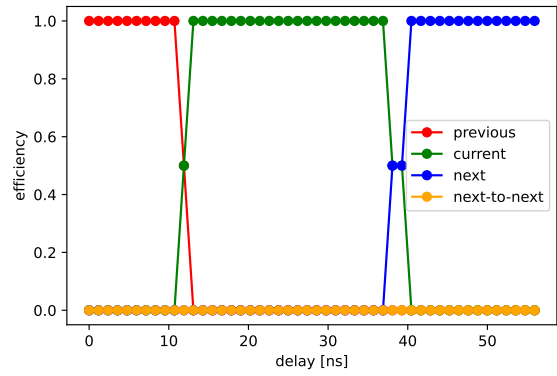


図 B.45: PS board 3 の ASD10

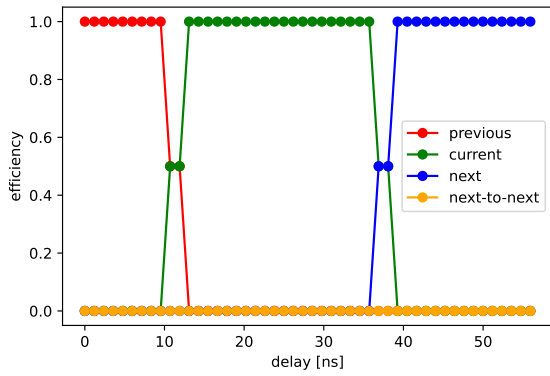


図 B.46: PS board 3 の ASD11

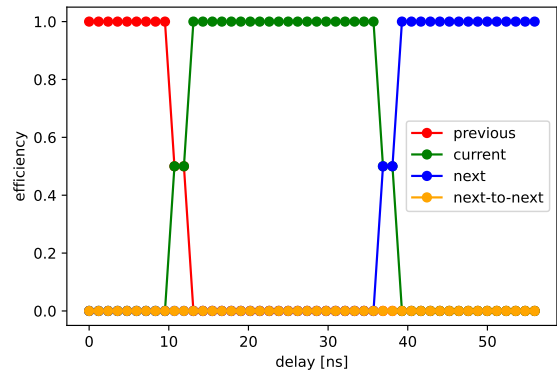


図 B.47: PS board 3 の ASD12

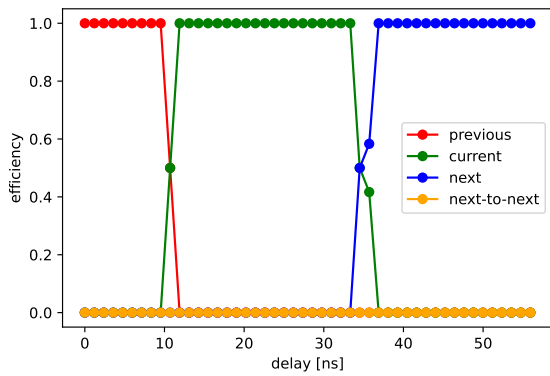


図 B.48: PS board 3 の ASD13

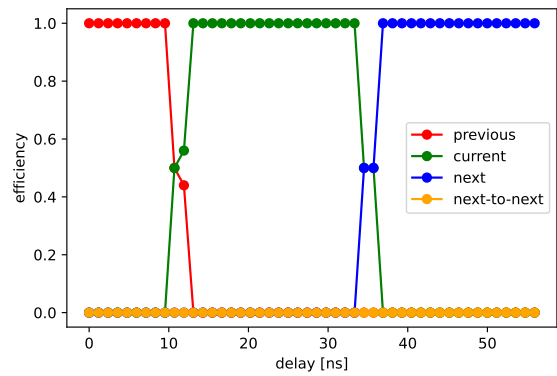


図 B.49: PS board 3 の ASD14

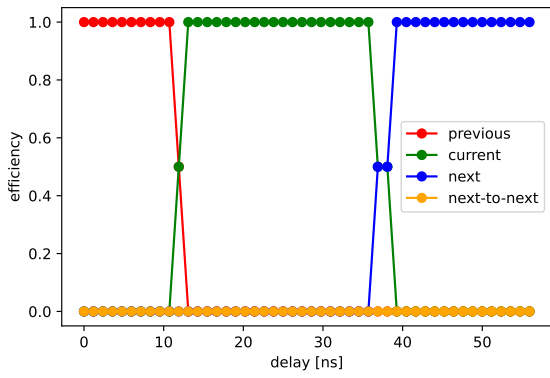


図 B.50: PS board 3 の ASD15

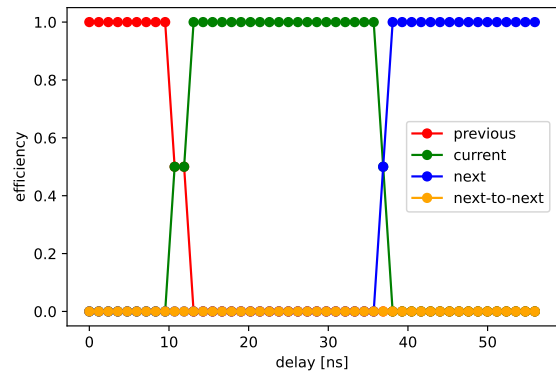


図 B.51: PS board 3 の ASD16

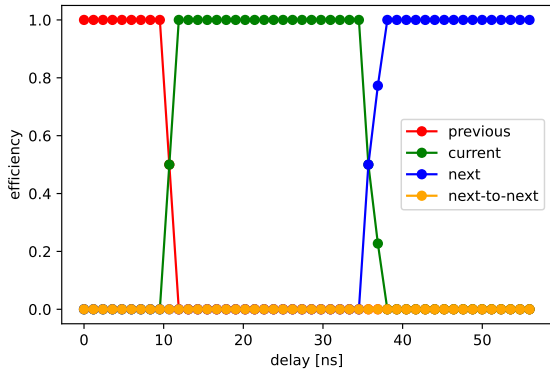


図 B.52: PS board 4 の ASD1

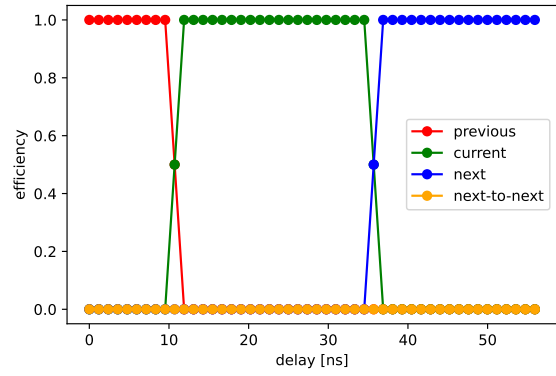


図 B.53: PS board 4 の ASD2

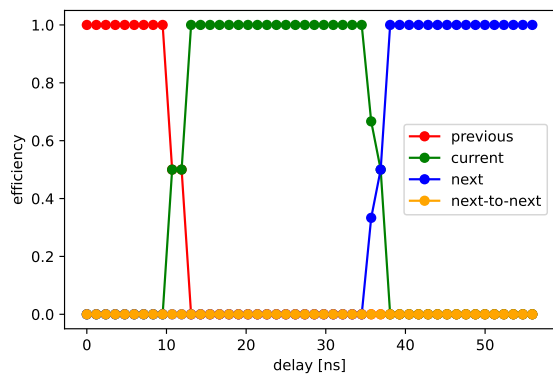


図 B.54: PS board 4 の ASD3

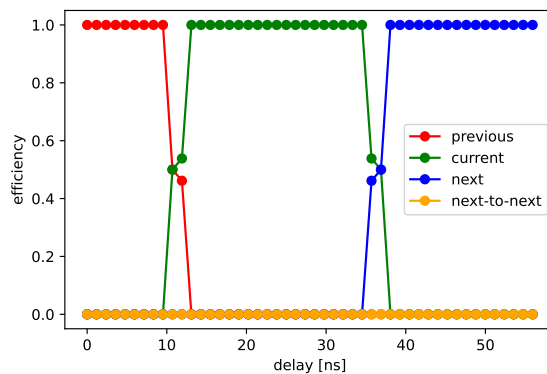


図 B.55: PS board 4 の ASD4

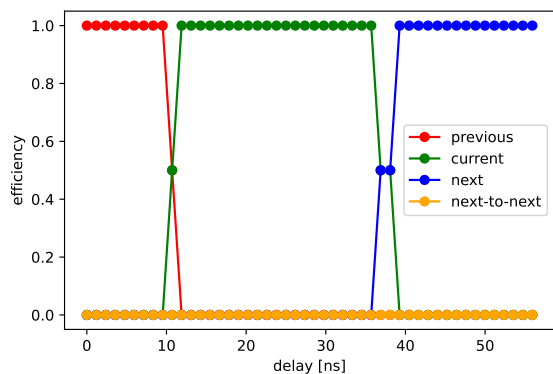


図 B.56: PS board 4 の ASD5

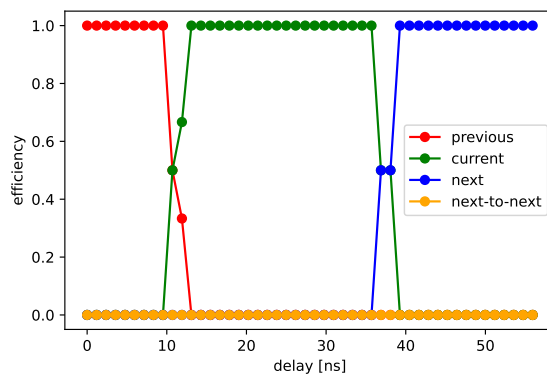


図 B.57: PS board 4 の ASD6

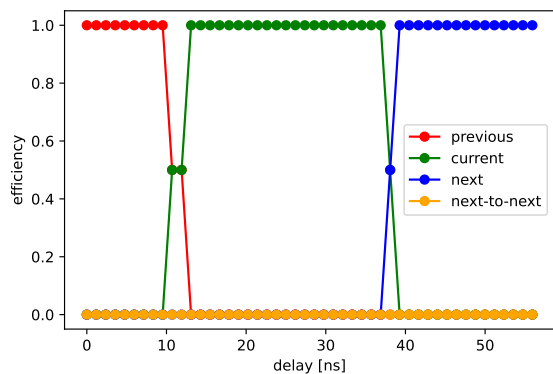


図 B.58: PS board 4 の ASD7

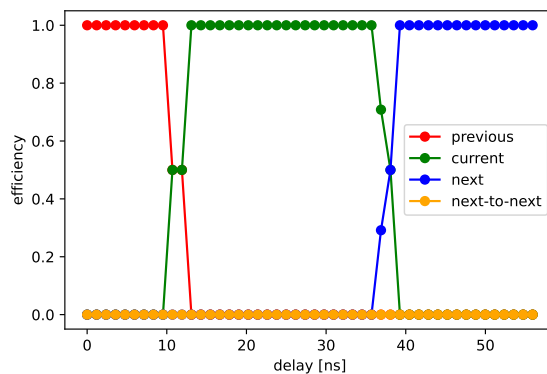


図 B.59: PS board 4 の ASD8

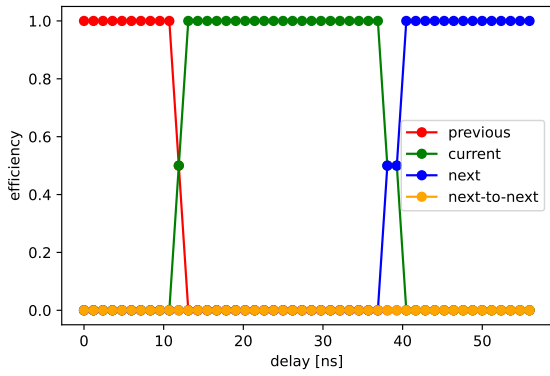


図 B.60: PS board 4 の ASD9

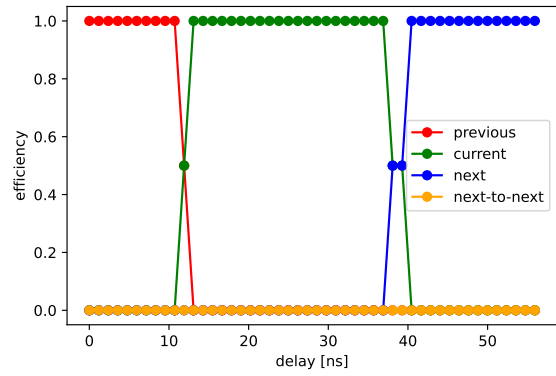


図 B.61: PS board 4 の ASD10

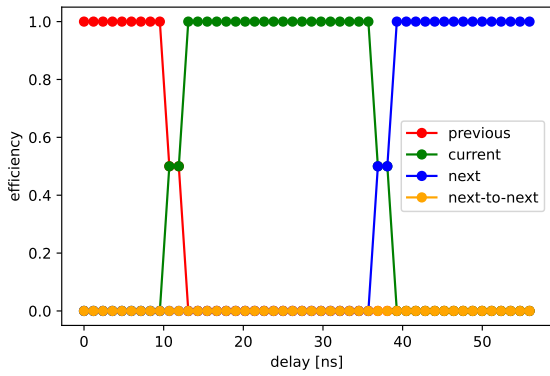


図 B.62: PS board 4 の ASD11

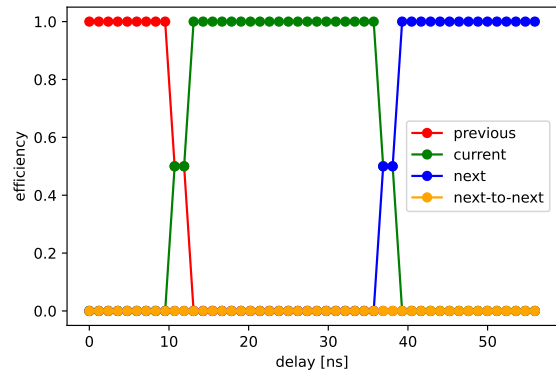


図 B.63: PS board 4 の ASD12

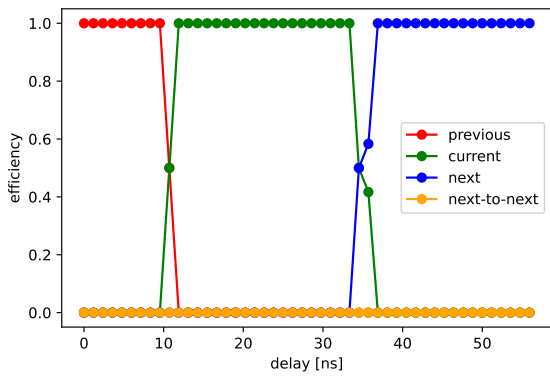


図 B.64: PS board 4 の ASD13

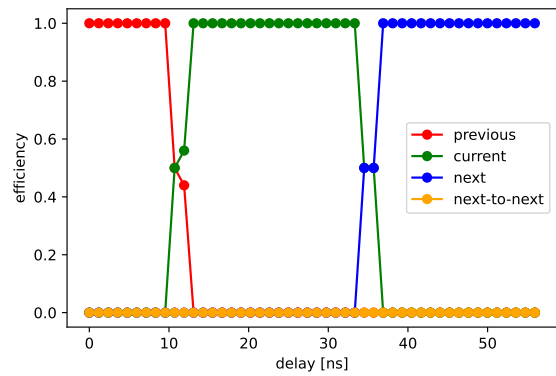


図 B.65: PS board 4 の ASD14

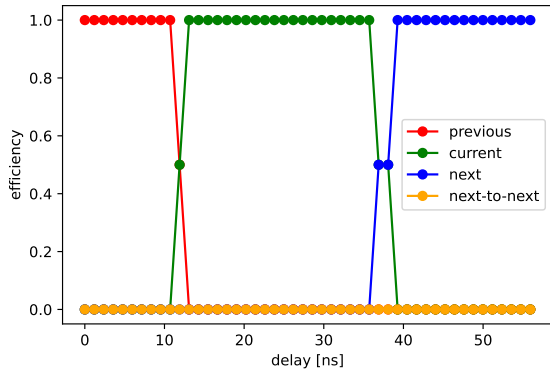


図 B.66: PS board 4 の ASD15

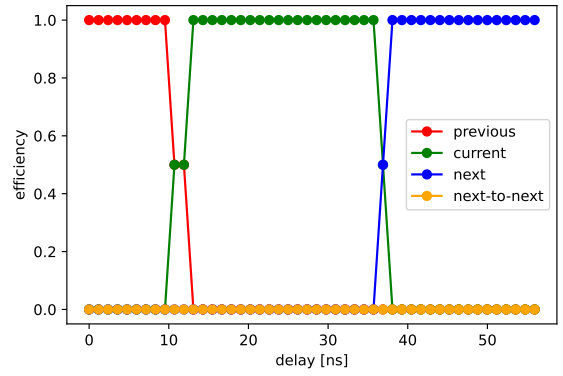


図 B.67: PS board 4 の ASD16

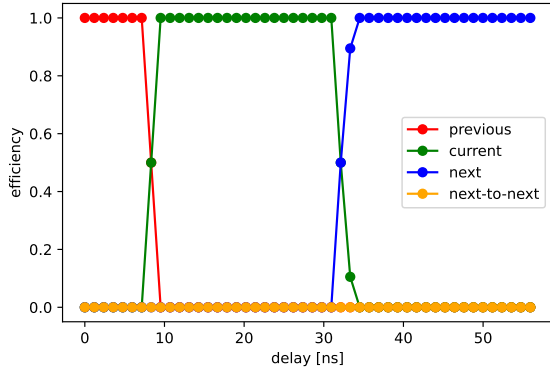


図 B.68: PS board 5 の ASD1

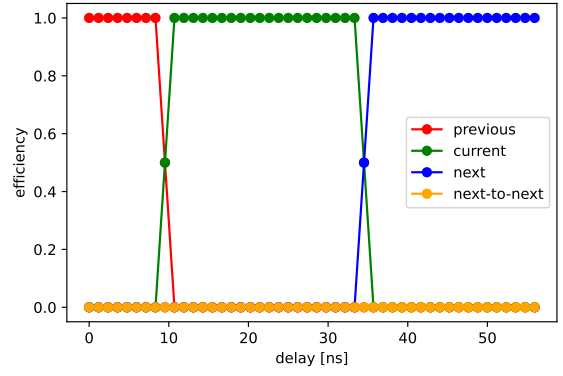


図 B.69: PS board 5 の ASD2

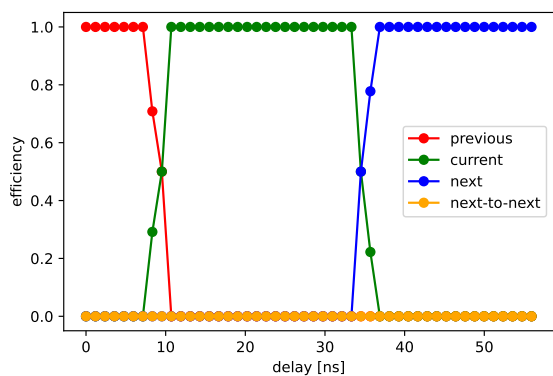


図 B.70: PS board 5 の ASD3

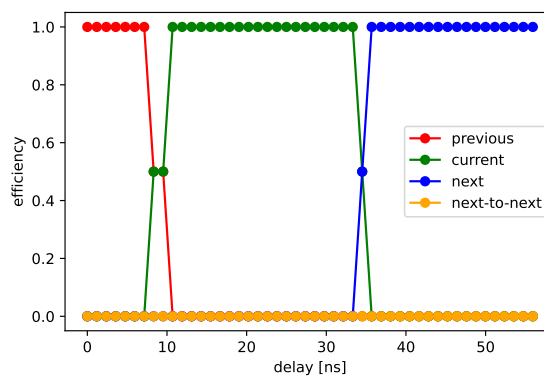


図 B.71: PS board 5 の ASD4

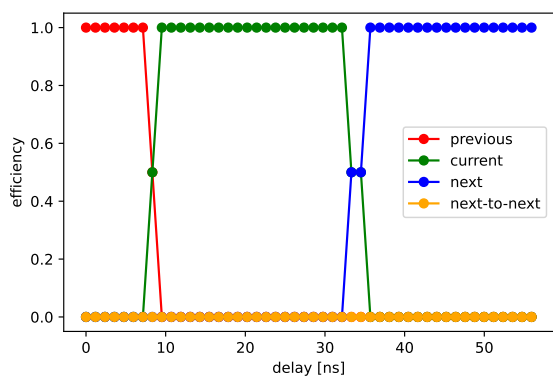


図 B.72: PS board 5 の ASD5

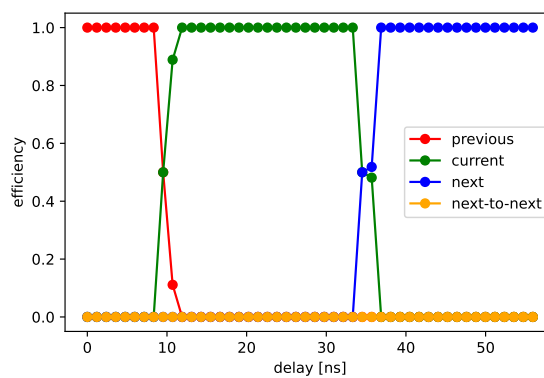


図 B.73: PS board 5 の ASD6

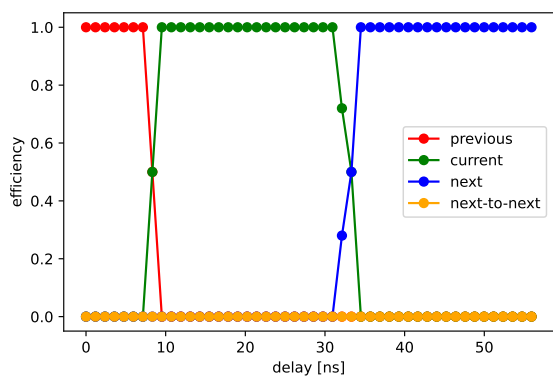


図 B.74: PS board 5 の ASD7

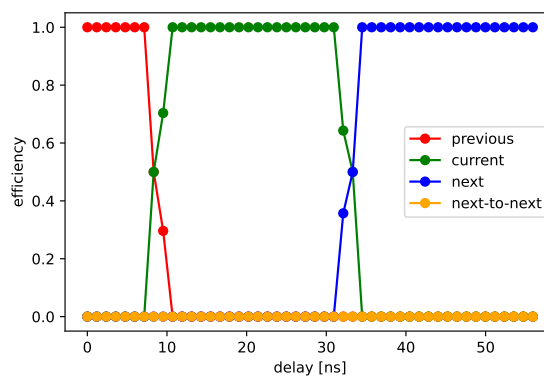


図 B.75: PS board 5 の ASD8

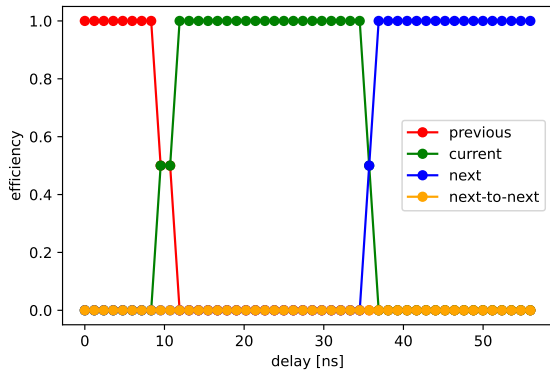


図 B.76: PS board 5 の ASD9

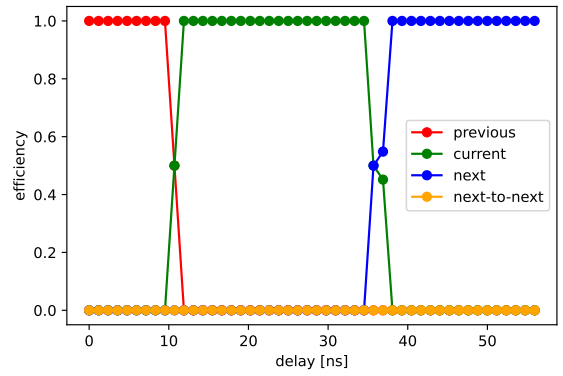


図 B.77: PS board 5 の ASD10

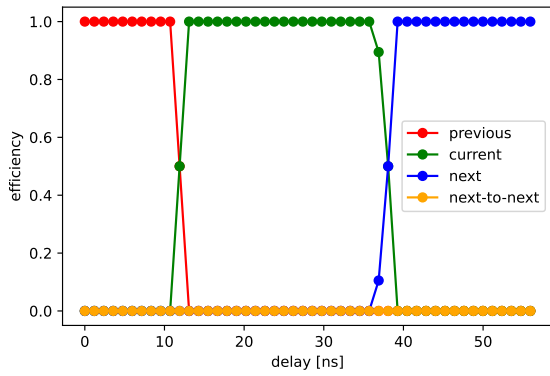


図 B.78: PS board 5 の ASD11

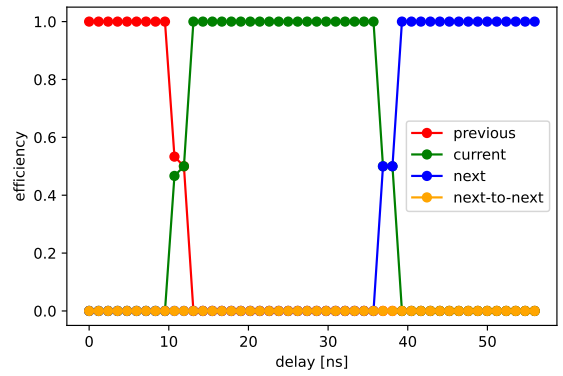


図 B.79: PS board 5 の ASD12

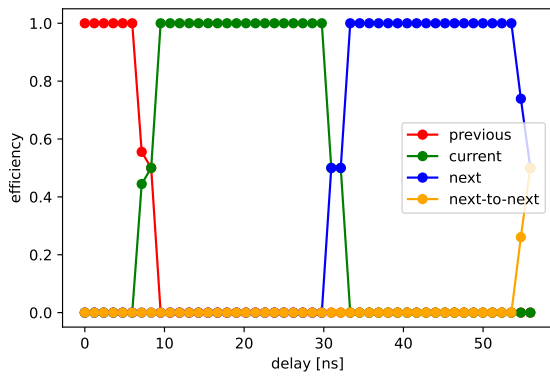


図 B.80: PS board 5 の ASD13

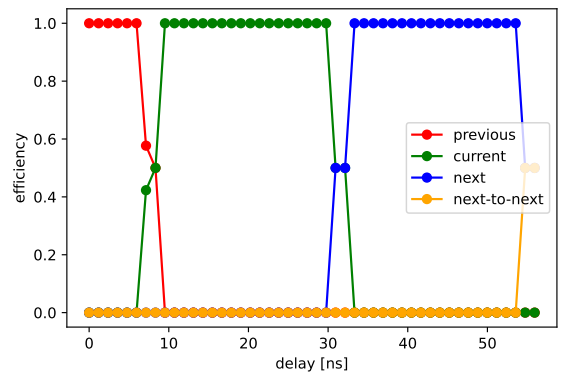


図 B.81: PS board 5 の ASD14

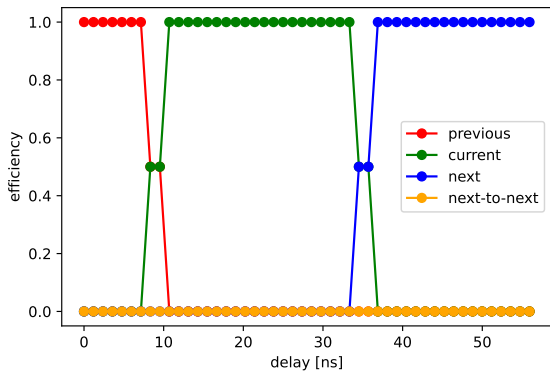


図 B.82: PS board 5 の ASD15

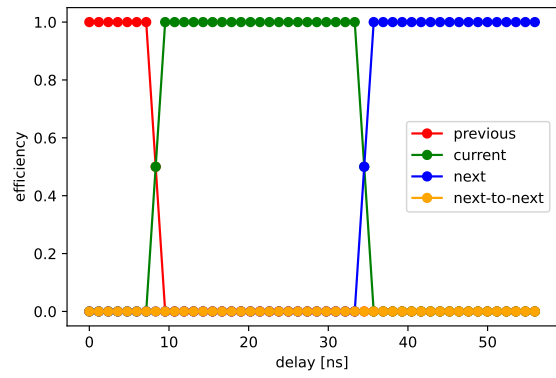


図 B.83: PS board 5 の ASD16

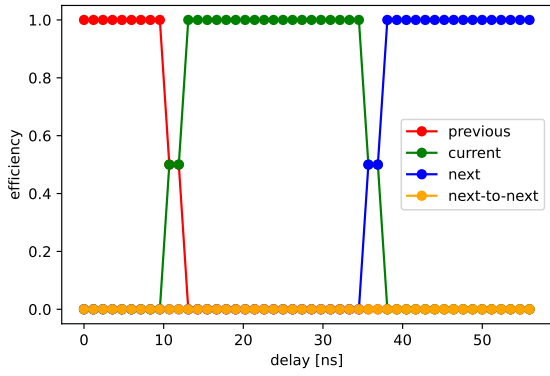


図 B.84: PS board 6 の ASD1

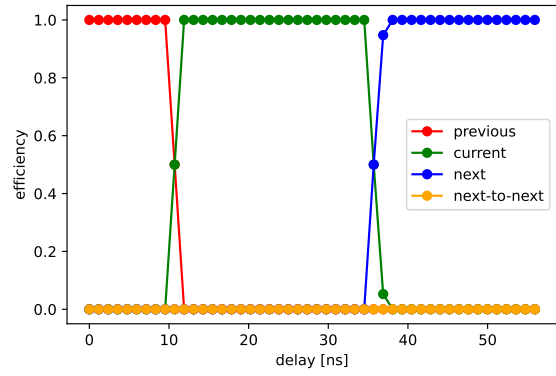


図 B.85: PS board 6 の ASD2

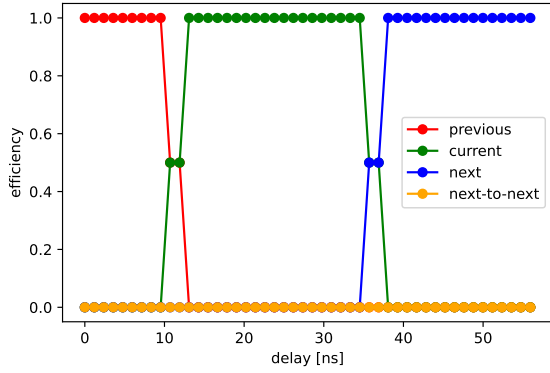


図 B.86: PS board 6 の ASD3

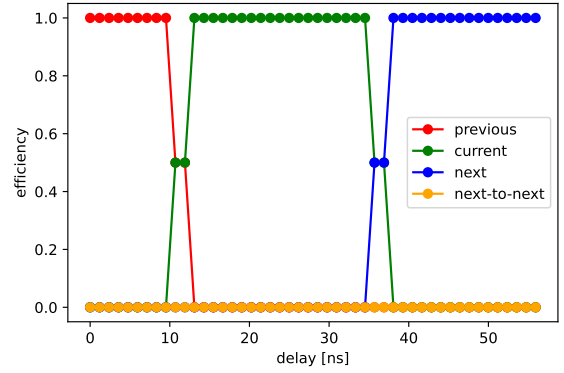


図 B.87: PS board 6 の ASD4

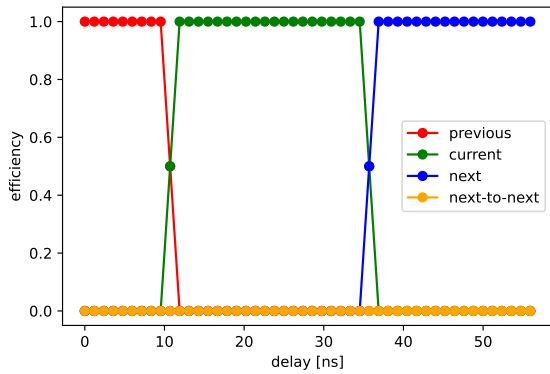


図 B.88: PS board 6 の ASD5

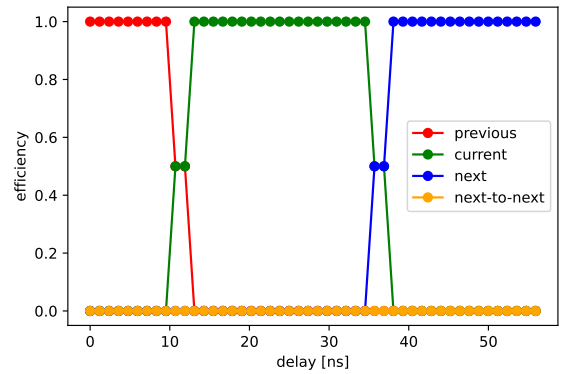


図 B.89: PS board 6 の ASD6

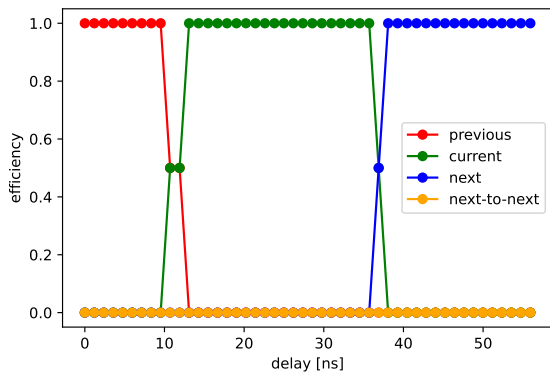


図 B.90: PS board 6 の ASD7

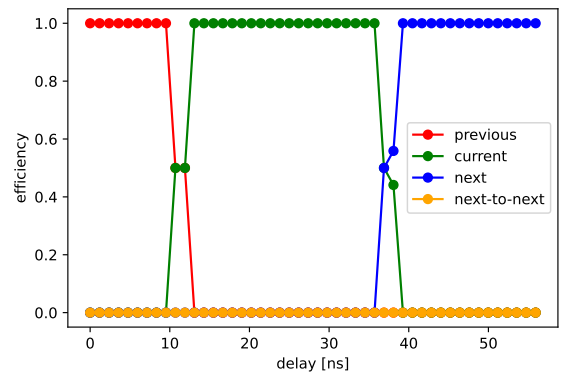


図 B.91: PS board 6 の ASD8

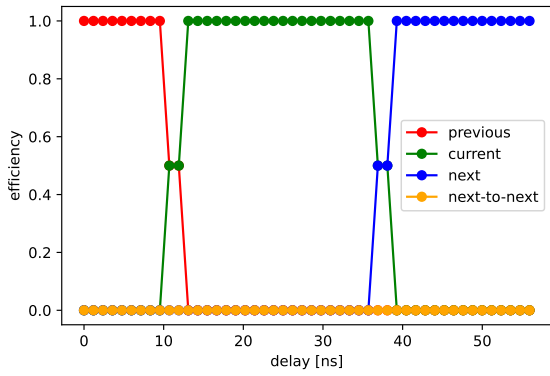


図 B.92: PS board 6 の ASD9

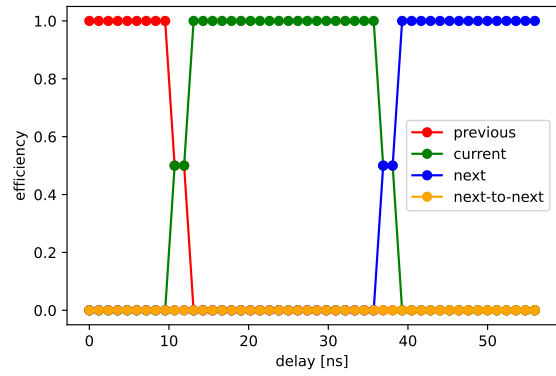


図 B.93: PS board 6 の ASD10

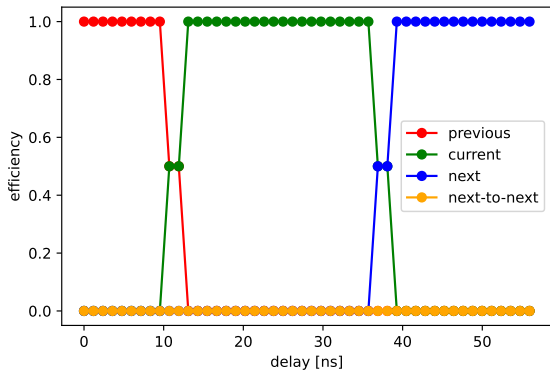


図 B.94: PS board 6 の ASD11

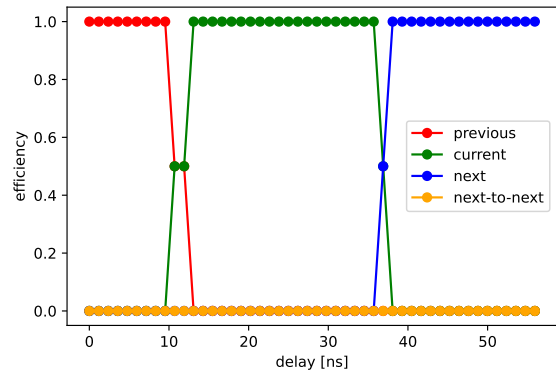


図 B.95: PS board 6 の ASD12

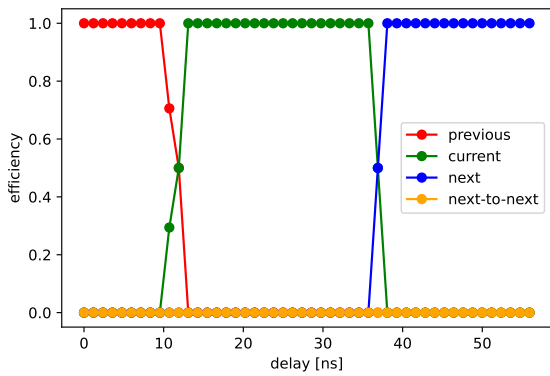


図 B.96: PS board 6 の ASD13

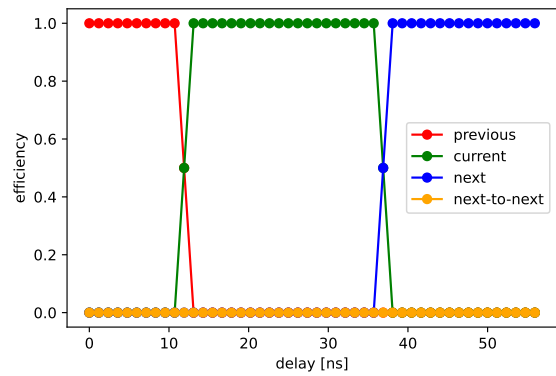


図 B.97: PS board 6 の ASD14

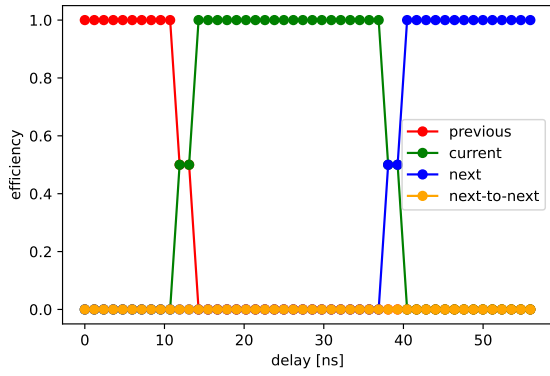


図 B.98: PS board 6 の ASD15

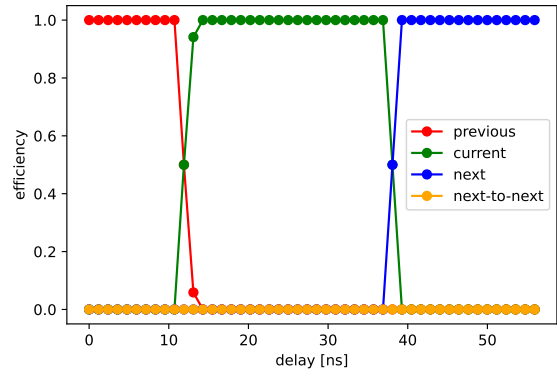


図 B.99: PS board 6 の ASD16

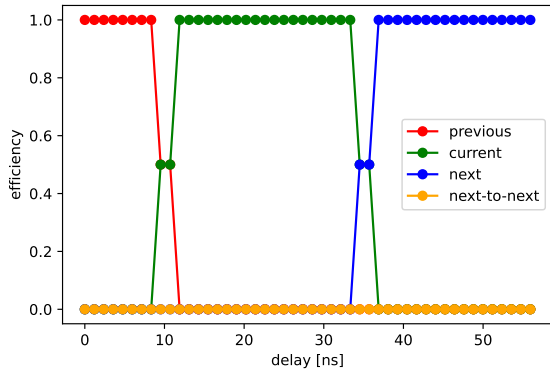


図 B.100: PS board 7 の ASD1

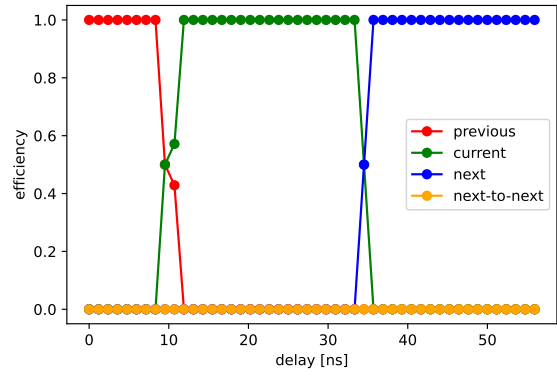


図 B.101: PS board 7 の ASD2

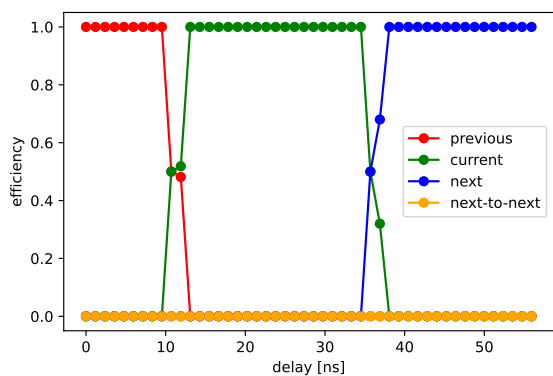


図 B.102: PS board 7 の ASD3

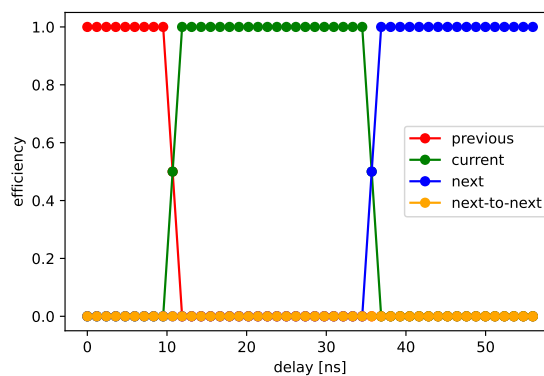


図 B.103: PS board 7 の ASD4

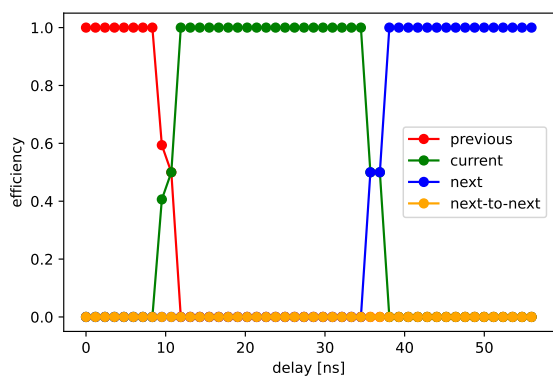


図 B.104: PS board 7 の ASD5

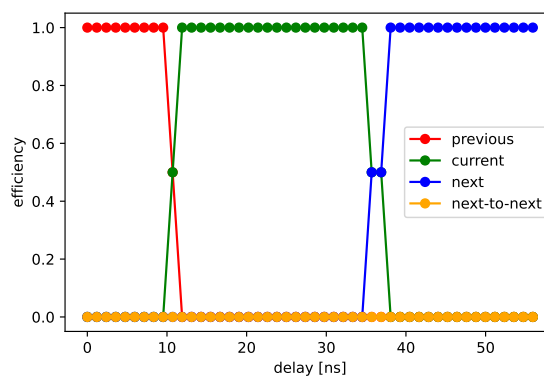


図 B.105: PS board 7 の ASD6

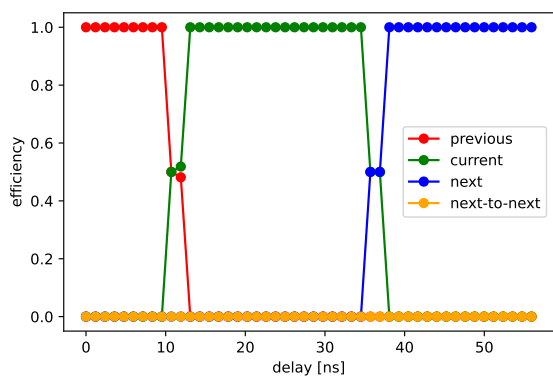


図 B.106: PS board 7 の ASD7

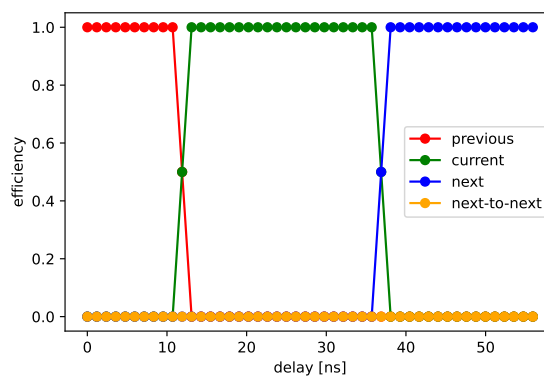


図 B.107: PS board 7 の ASD8

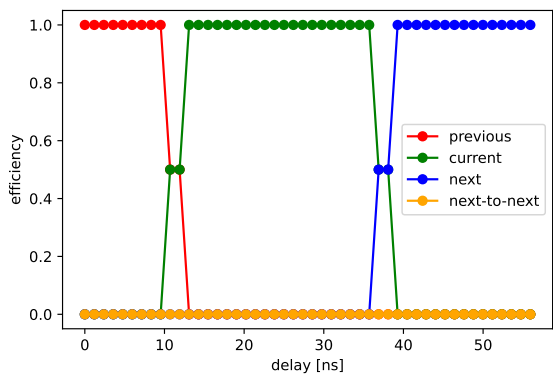


図 B.108: PS board 7 の ASD9

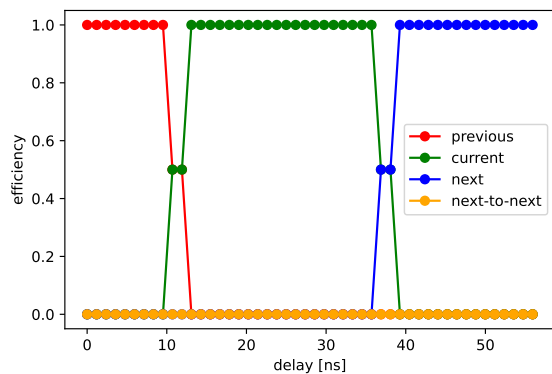


図 B.109: PS board 7 の ASD10

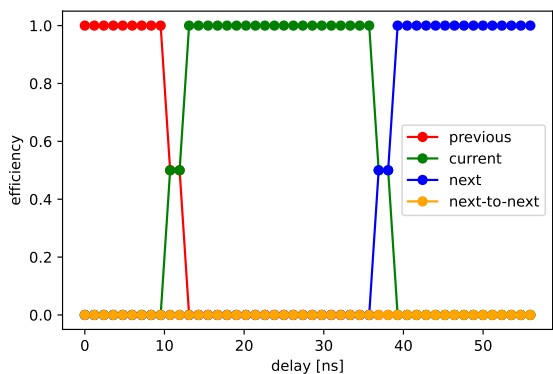


図 B.110: PS board 7 の ASD11

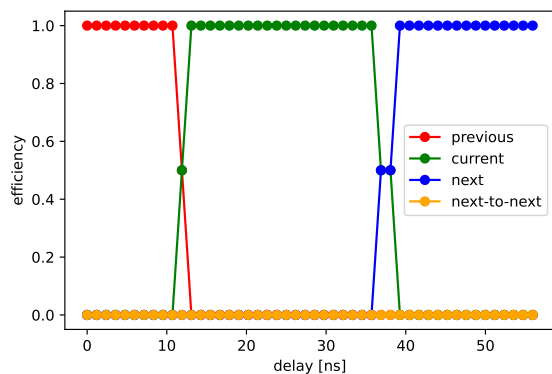


図 B.111: PS board 7 の ASD12

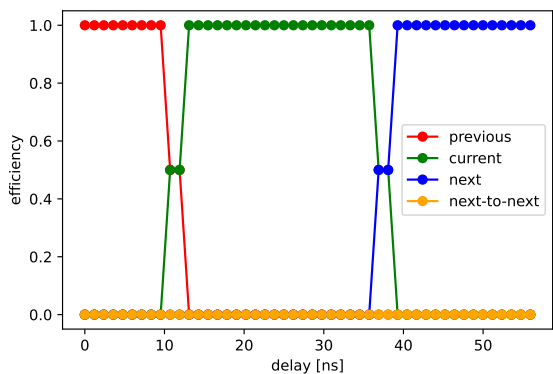


図 B.112: PS board 7 の ASD13

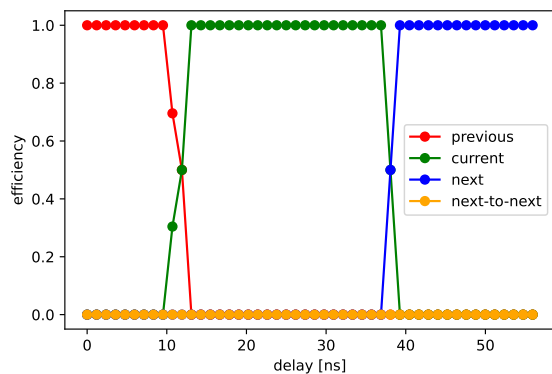


図 B.113: PS board 7 の ASD14

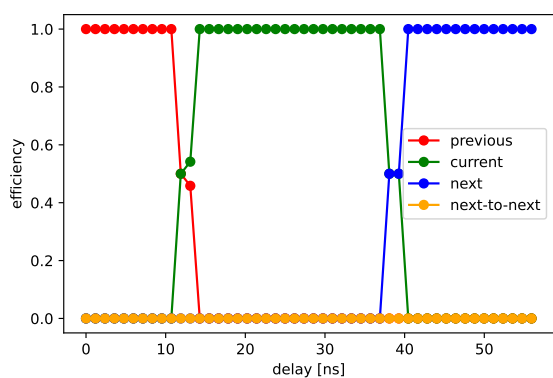


図 B.114: PS board 7 の ASD15

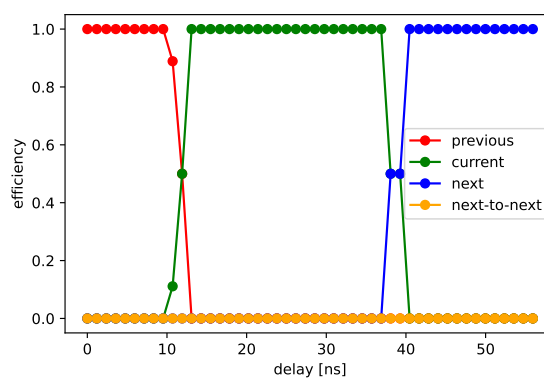


図 B.115: PS board 7 の ASD16

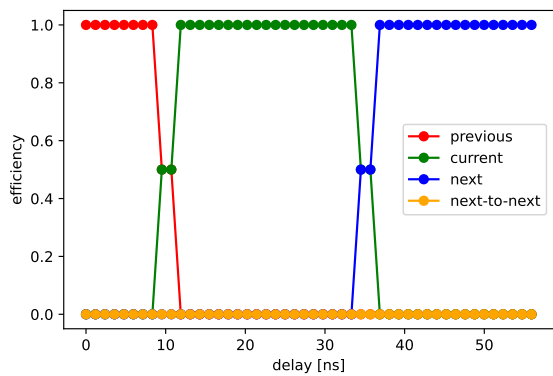


図 B.116: PS board 8 の ASD1

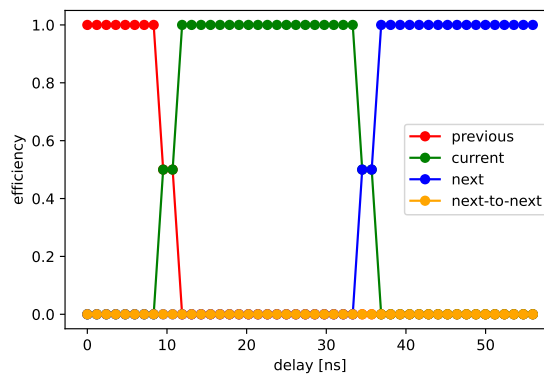


図 B.117: PS board 8 の ASD2

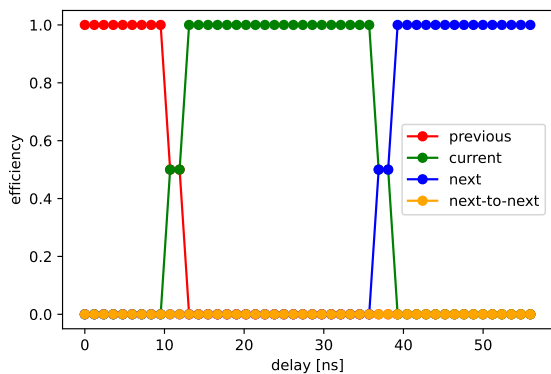


図 B.118: PS board 8 の ASD3

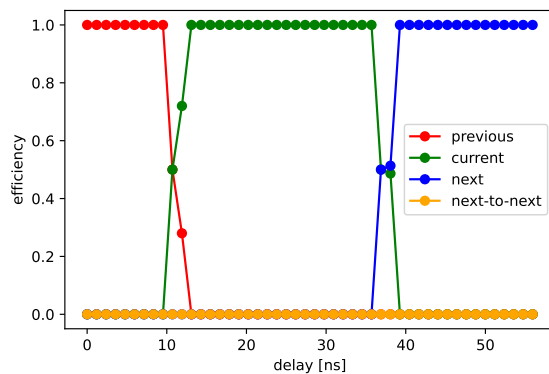


図 B.119: PS board 8 の ASD4

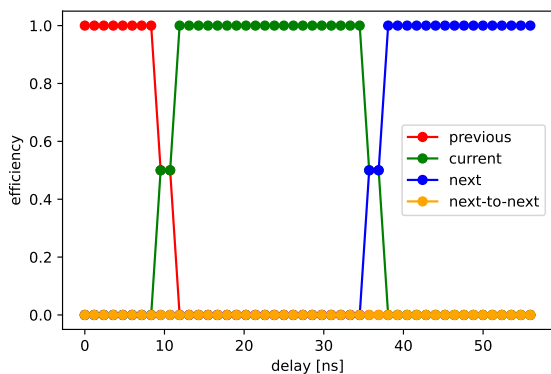


図 B.120: PS board 8 の ASD5

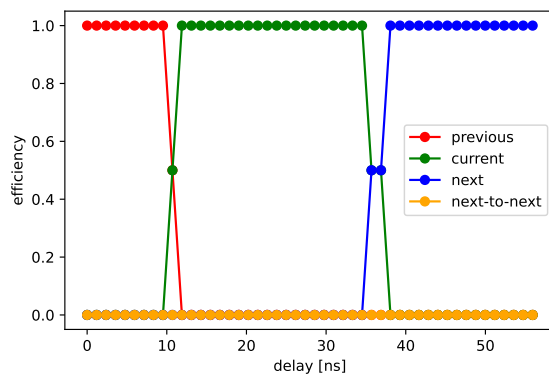


図 B.121: PS board 8 の ASD6

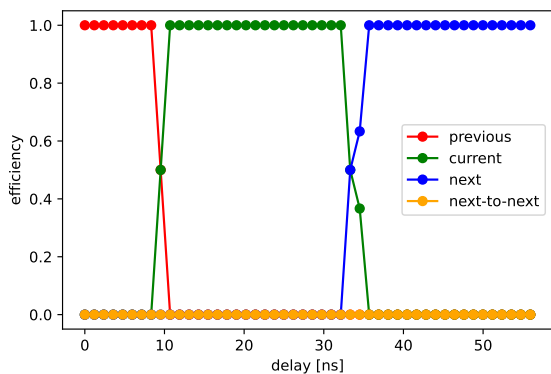


図 B.122: PS board 8 の ASD7

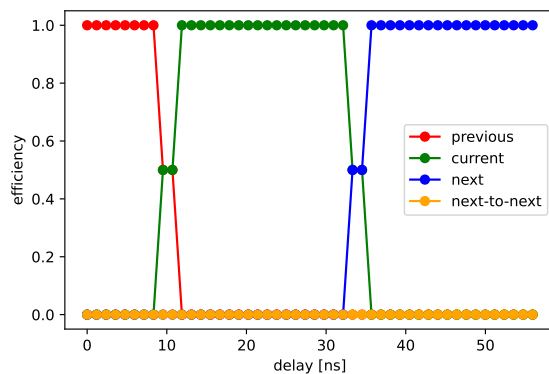


図 B.123: PS board 8 の ASD8

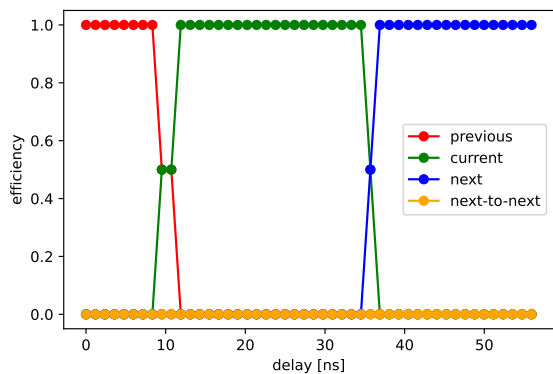


図 B.124: PS board 8 の ASD9

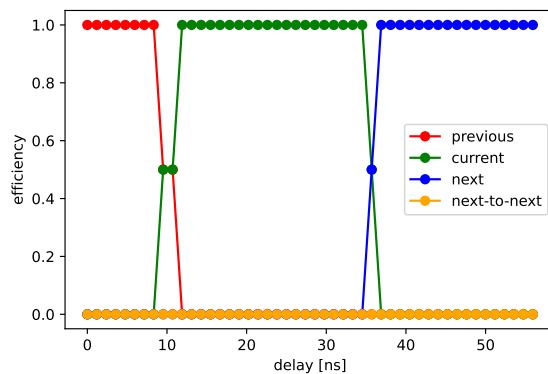


図 B.125: PS board 8 の ASD10

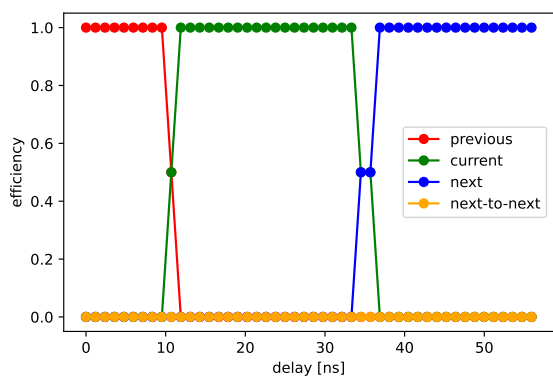


図 B.126: PS board 8 の ASD11

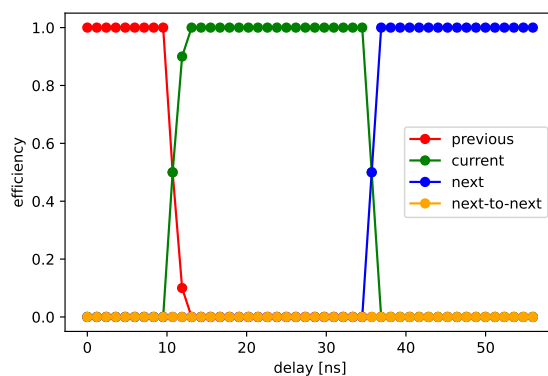


図 B.127: PS board 8 の ASD12

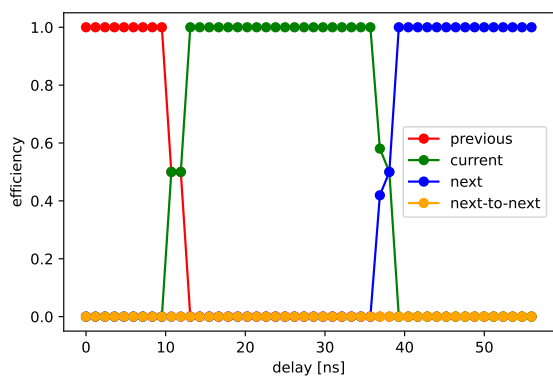


図 B.128: PS board 8 の ASD13

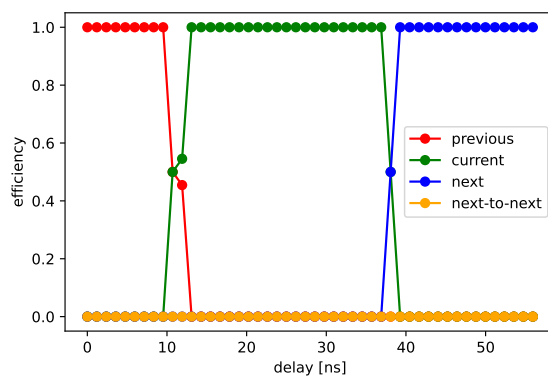


図 B.129: PS board 8 の ASD14

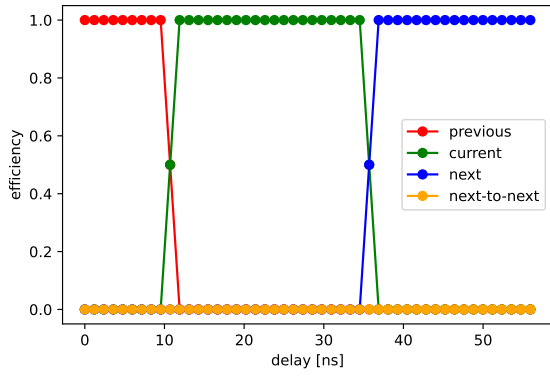


図 B.130: PS board 8 の ASD15

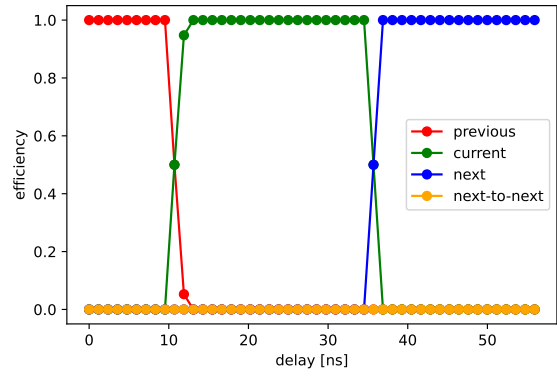


図 B.131: PS board 8 の ASD16

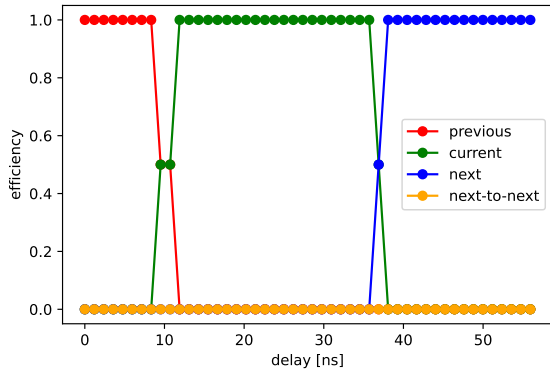


図 B.132: PS board 9 の ASD1

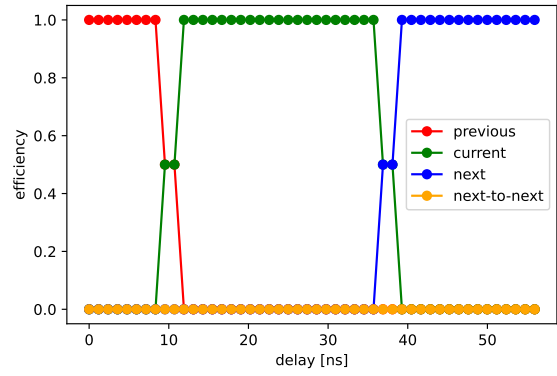


図 B.133: PS board 9 の ASD2

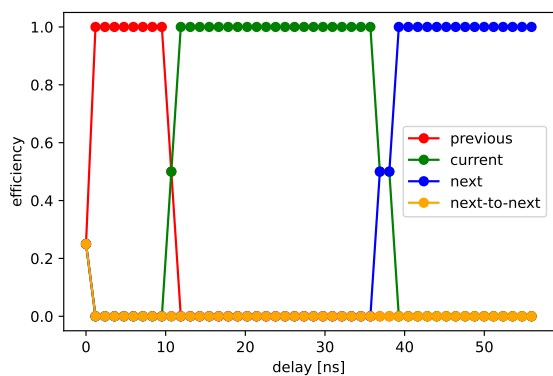


図 B.134: PS board 9 の ASD3

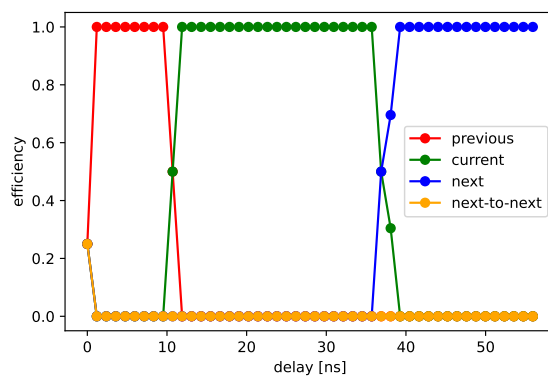


図 B.135: PS board 9 の ASD4

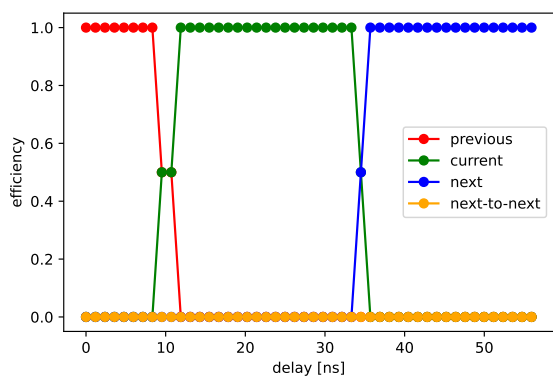


図 B.136: PS board 9 の ASD5

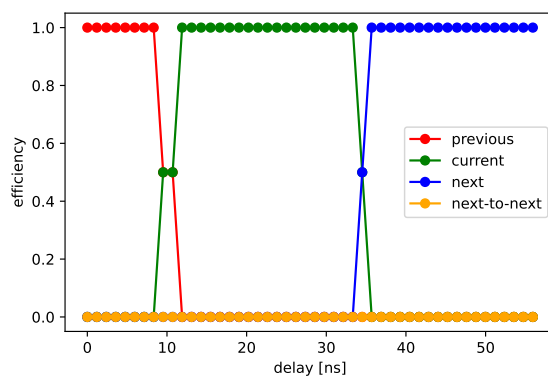


図 B.137: PS board 9 の ASD6

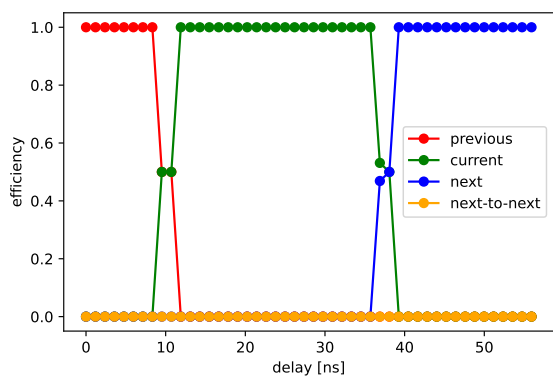


図 B.138: PS board 9 の ASD7

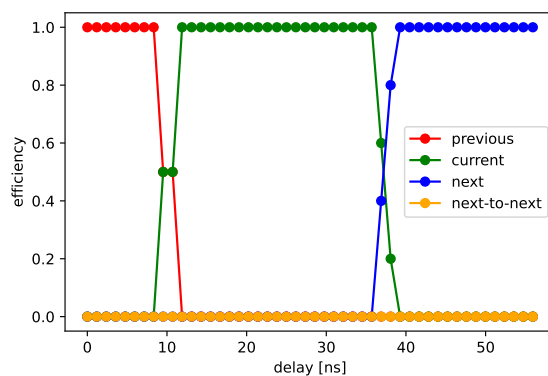


図 B.139: PS board 9 の ASD8

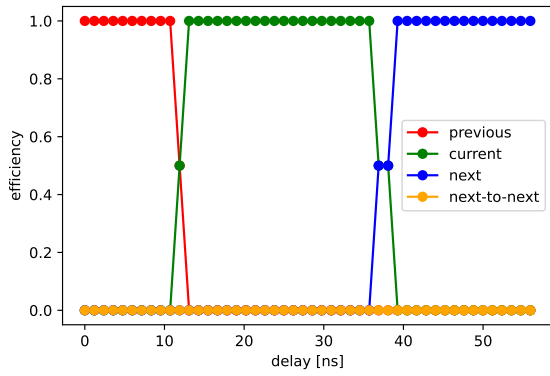


図 B.140: PS board 9 の ASD9

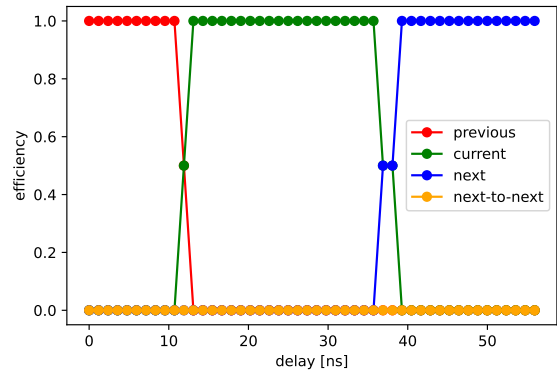


図 B.141: PS board 9 の ASD10

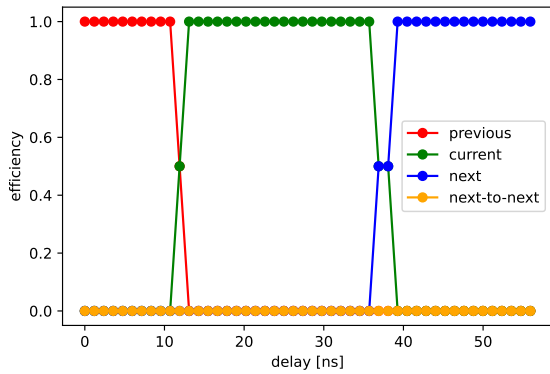


図 B.142: PS board 9 の ASD11

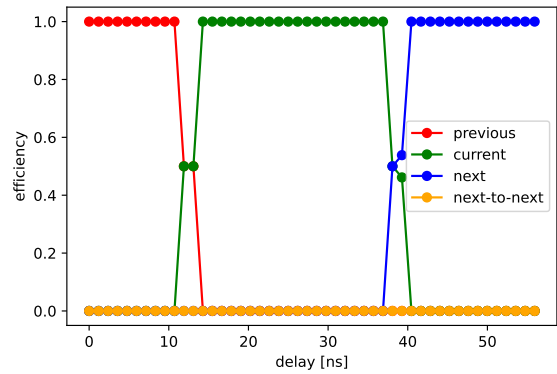


図 B.143: PS board 9 の ASD12

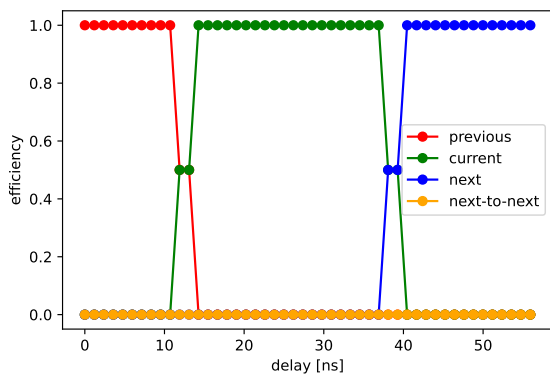


図 B.144: PS board 9 の ASD13

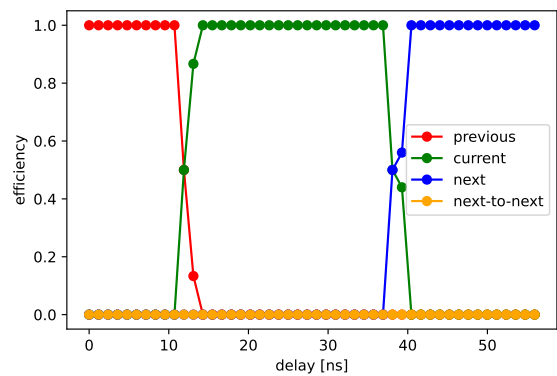


図 B.145: PS board 9 の ASD14

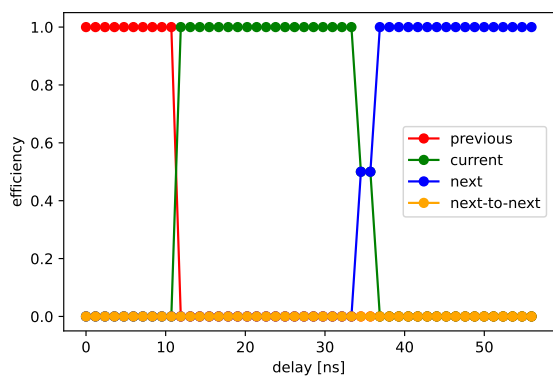


図 B.146: PS board 9 の ASD15

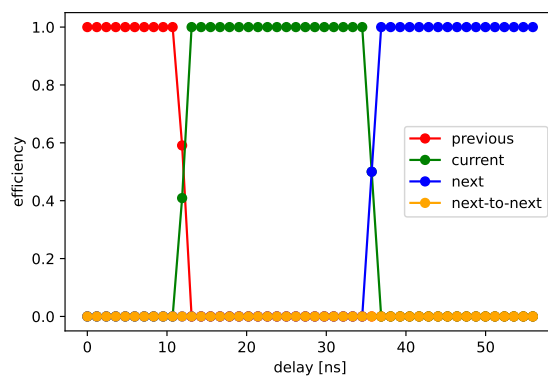


図 B.147: PS board 9 の ASD16

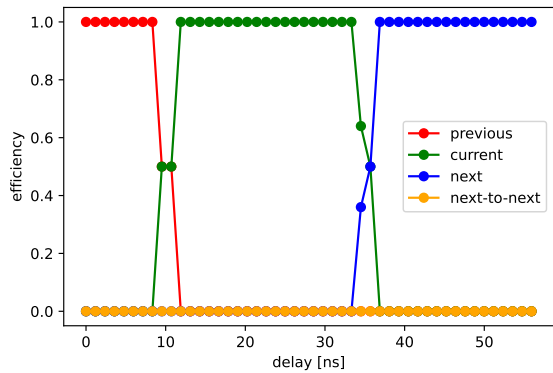


図 B.148: PS board 10 の ASD1

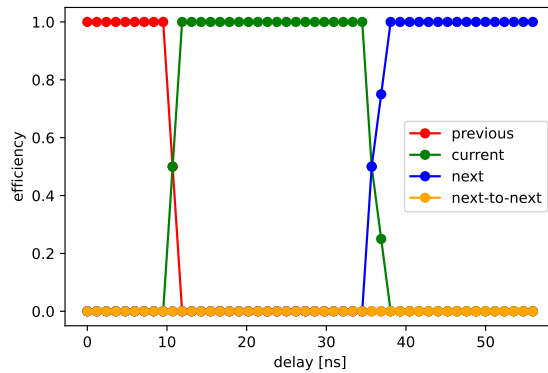


図 B.149: PS board 10 の ASD2

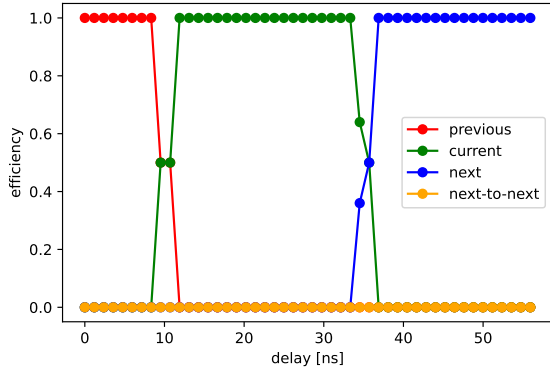


図 B.150: PS board 10 の ASD3

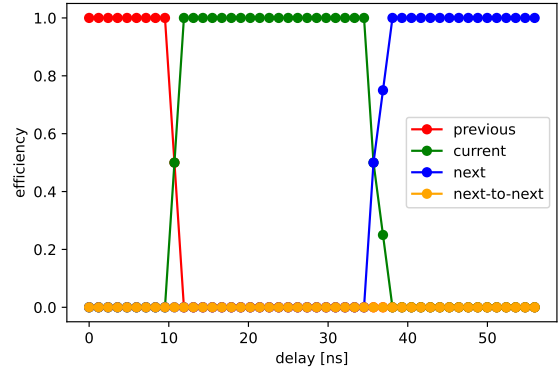


図 B.151: PS board 10 の ASD4

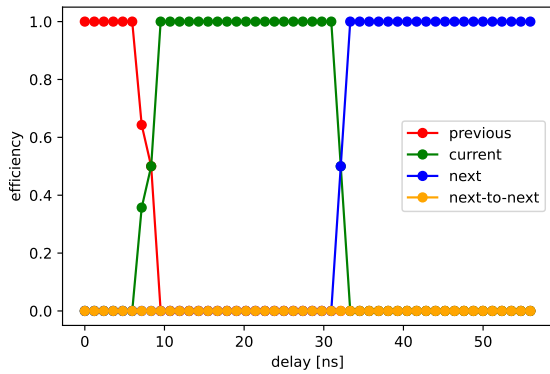


図 B.152: PS board 10 の ASD5

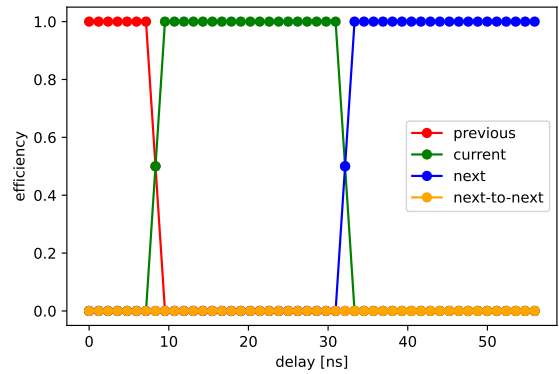


図 B.153: PS board 10 の ASD6

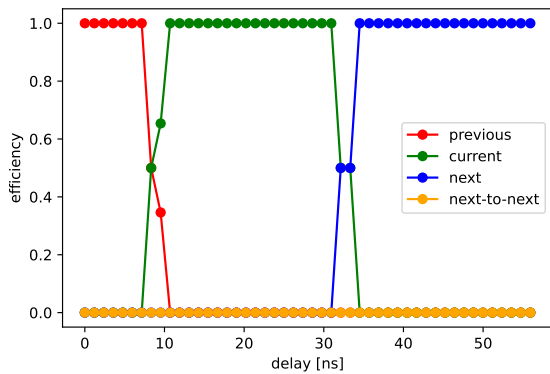


図 B.154: PS board 10 の ASD7

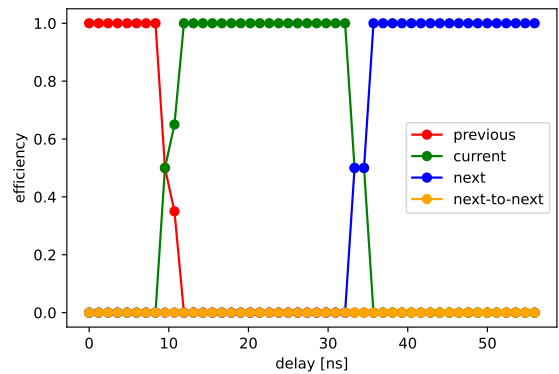


図 B.155: PS board 10 の ASD8

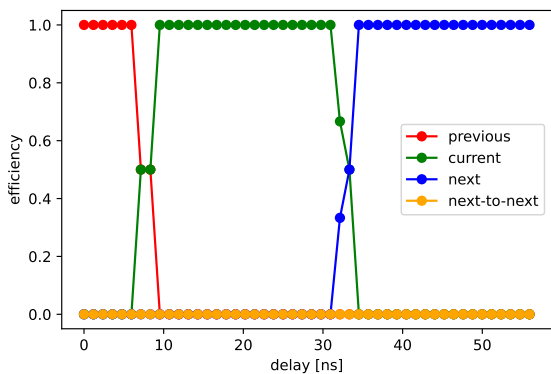


図 B.156: PS board 10 の ASD9

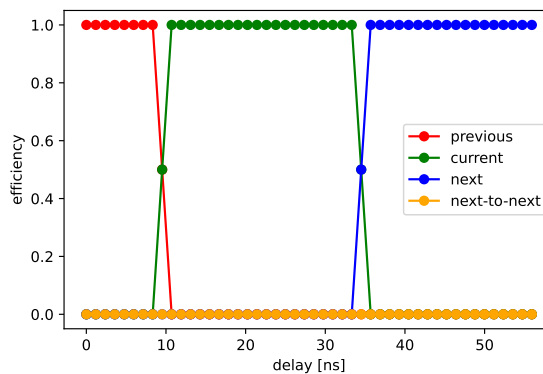


図 B.157: PS board 10 の ASD10

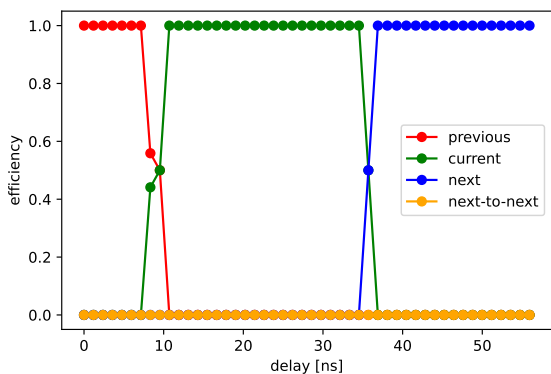


図 B.158: PS board 10 の ASD11

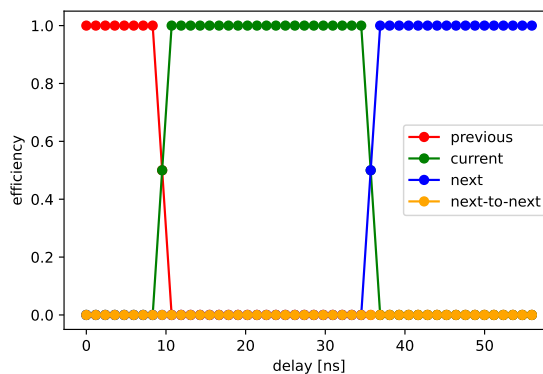


図 B.159: PS board 10 の ASD12

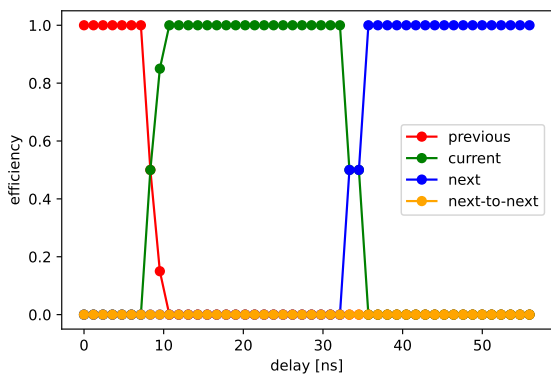


図 B.160: PS board 10 の ASD13

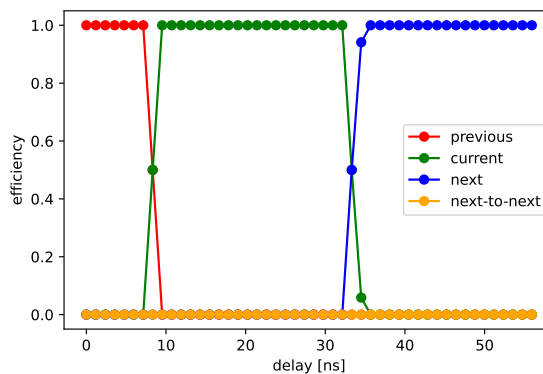


図 B.161: PS board 10 の ASD14

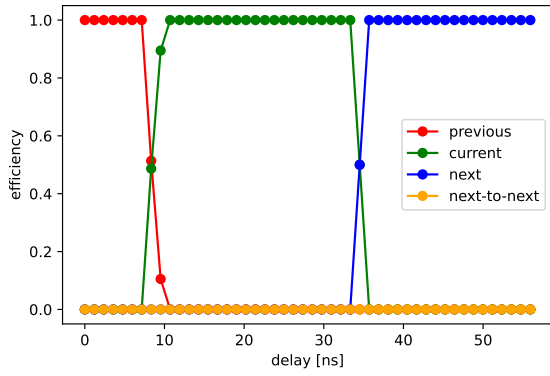


図 B.162: PS board 10 の ASD15

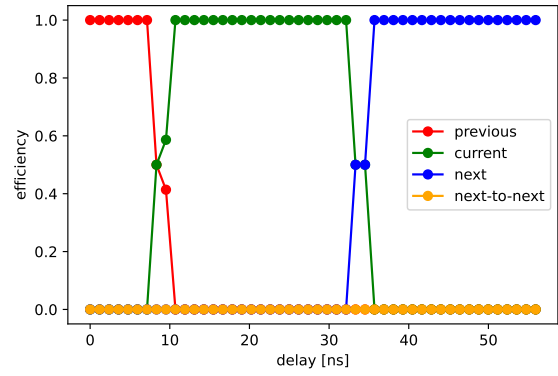


図 B.163: PS board 10 の ASD16

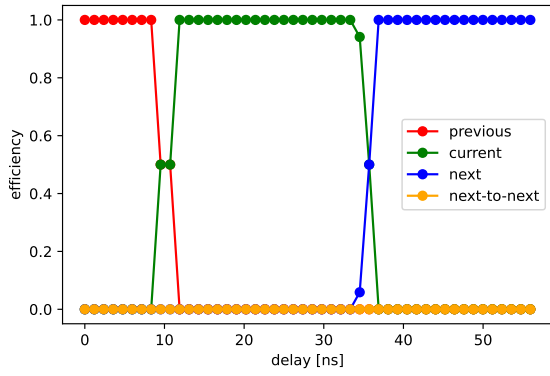


図 B.164: PS board 11 の ASD1

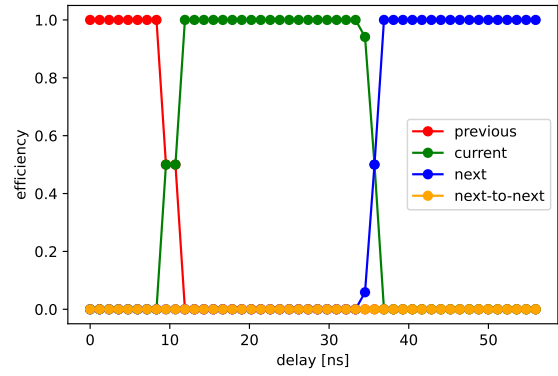


図 B.165: PS board 11 の ASD2

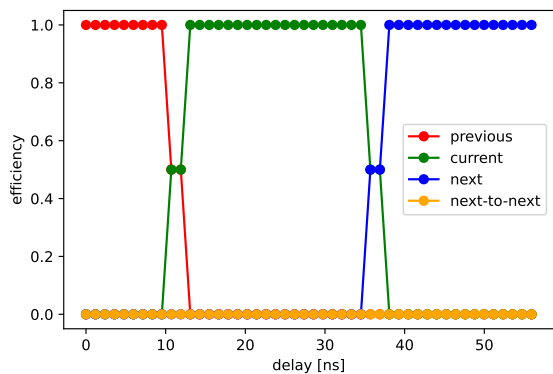


図 B.166: PS board 11 の ASD3

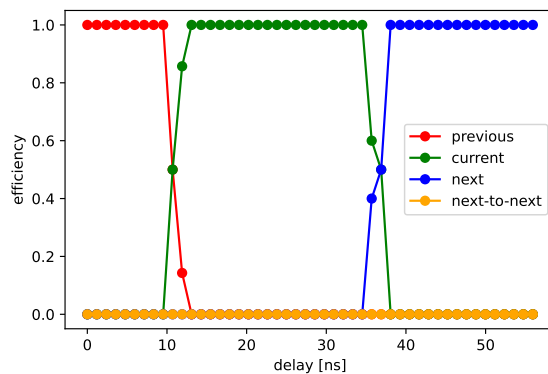


図 B.167: PS board 11 の ASD4

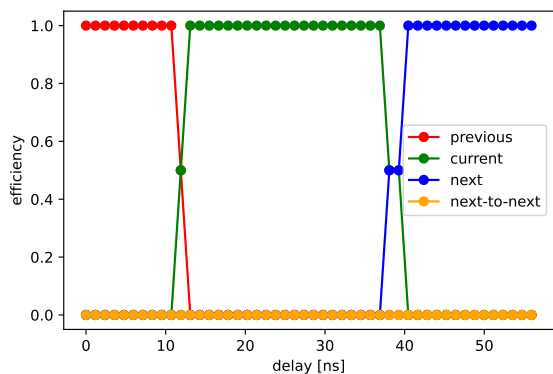


図 B.168: PS board 11 の ASD5

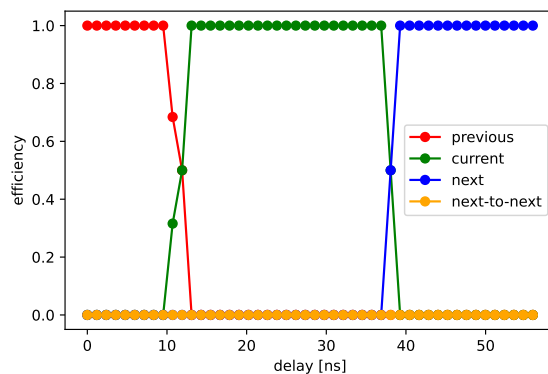


図 B.169: PS board 11 の ASD6

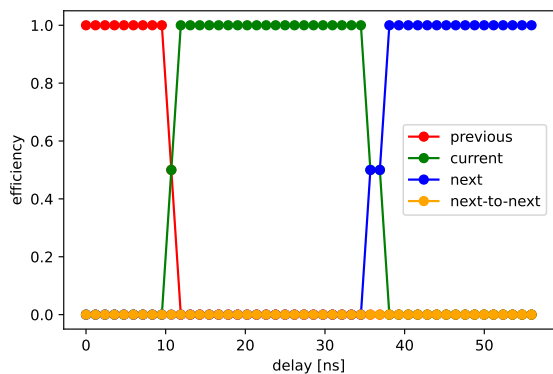


図 B.170: PS board 11 の ASD7

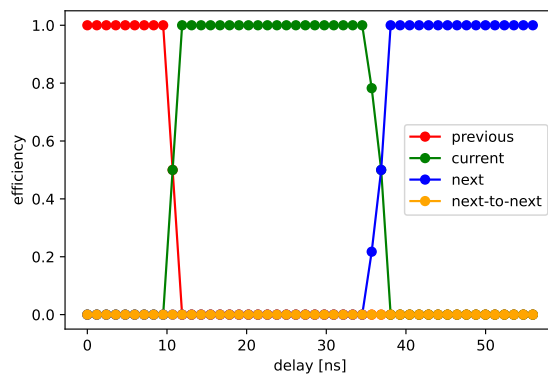


図 B.171: PS board 11 の ASD8

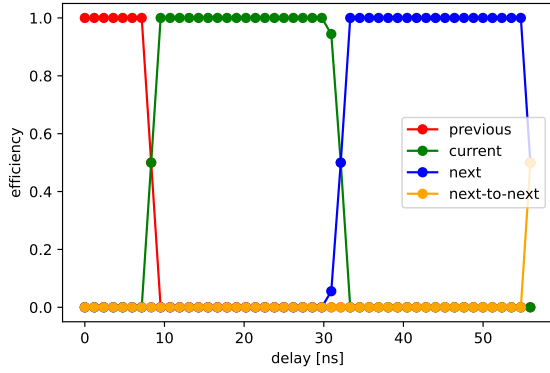


図 B.172: PS board 11 の ASD9

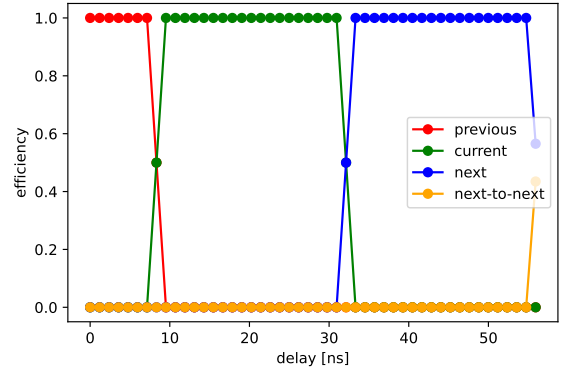


図 B.173: PS board 11 の ASD10

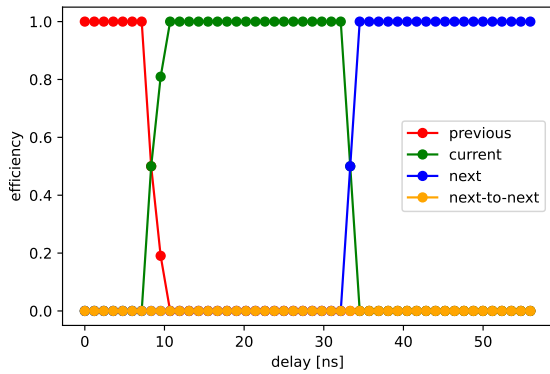


図 B.174: PS board 11 の ASD11

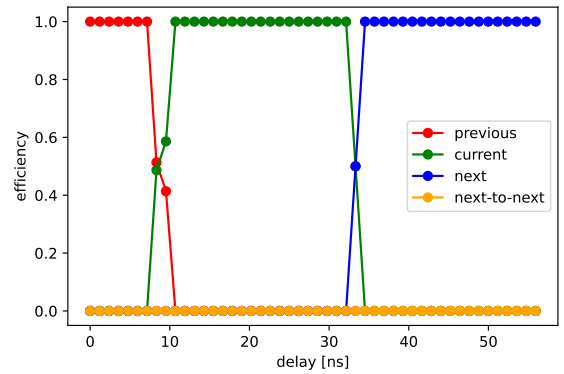


図 B.175: PS board 11 の ASD12

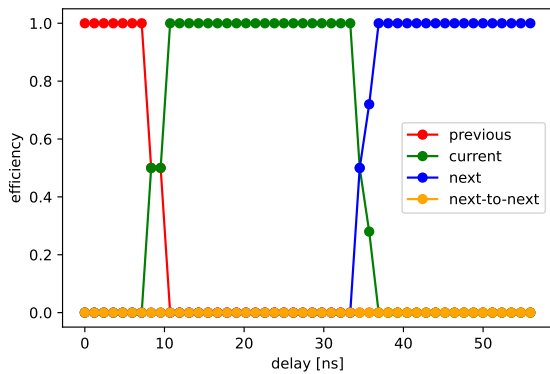


図 B.176: PS board 11 の ASD13

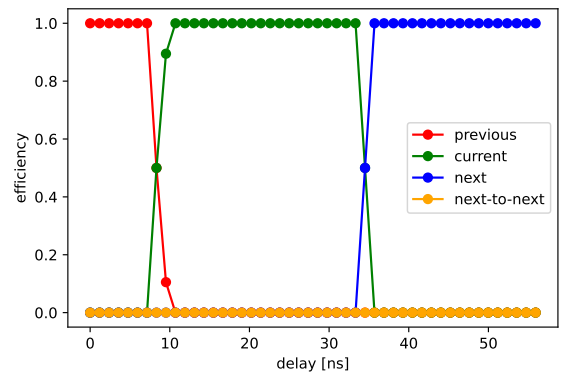


図 B.177: PS board 11 の ASD14

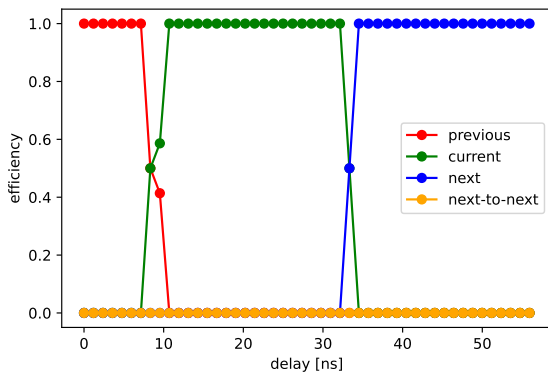


図 B.178: PS board 11 の ASD15

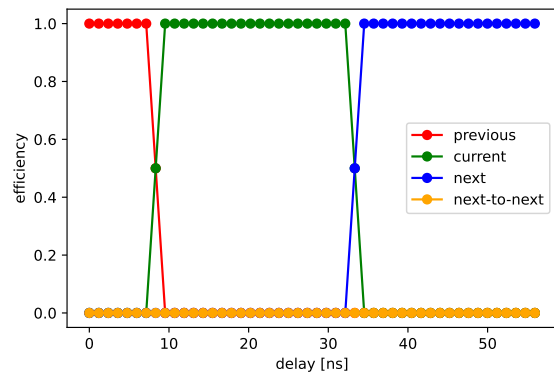


図 B.179: PS board 11 の ASD16

参考文献

- [1] Xilinx Inc, Soft Error Mitigation (SEM) Core, 2021, <https://www.xilinx.com/products/intellectual-property/sem.html#overview>.
- [2] 青木匠, 大規模エレクトロニクスシステムにおける次世代型オペレーションモデルの研究, 修士論文, 2022.
- [3] 三島章熙, 高輝度 LHC-ATLAS 実験に向けた ミューオントリガー論理回路の開発・実装・試験, 修士論文, 2023.
- [4] 竹本享史, ATLAS 実験 ミューオントリガーシステム用 LSI の開発と統合評価テスト, 修士論文, 2004.
- [5] ATLAS TGC Collaboration, Amplifier-Shaper-Discriminator ICs and ASD Boards, 1999, https://twiki.cern.ch/twiki/pub/Atlas/TgcDocument/ASD-PRR_v19991001.pdf.
- [6] Xilinx Inc, Vivado Design Suite 7 Series FPGA Libraries Guide, 2012, <https://docs.xilinx.com/v/u/2012.2-English/ug953-vivado-7series-libraries>.
- [7] Xilinx Inc, 7 Series FPGA GTX/GTH Transceiver, 2018, https://docs.xilinx.com/v/u/en-US/ug476_7Series_Transceivers.
- [8] Xilinx Inc, UltraScale Architecture GTY Transceiver, 2021, <https://docs.xilinx.com/v/u/en-US/ug578-ultrascale-gty-transceivers>.
- [9] Xilinx Inc, 7 Series FPGAs Memory Resources, 2019, https://docs.xilinx.com/v/u/en-US/ug473_7Series_Memory_Resources.
- [10] Xilinx Inc, Clocking Wizard v6.0 LogicCORE IP Product Guide (PG065), 2021, <https://docs.xilinx.com/r/en-US/pg065-clk-wiz/Overview>.
- [11] The ATLAS Collaboration, Technical Design Report for the Phase-II Upgrade of the ATLAS Muon Spectrometer, CERN-LHCC-2017-017, ATLAS-TDR-026, 2017, <https://cds.cern.ch/record/2285580>.
- [12] The ATLAS Collaboration, Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System, CERN-LHCC-2017-020, ATLAS-TDR-029, 2017, <https://cds.cern.ch/record/2285584>.
- [13] Skyworks Solutions Inc, Si5395/94/92 Reference Manual, 2021, <https://www.skyworksinc.com/-/media/Skyworks/SL/documents/public/reference-manuals/si5395-94-92-family.pdf>.
- [14] P.A. Zyla et al. (Particle Data Group), Review of Particle Physics (2020), Prog. Theor. Exp. Phys. 2020, 083C01 (2020).
- [15] Service graphique, CERN, Overall view of the LHC. Vue d'ensemble du LHC, 2014, <https://cds.cern.ch/record/1708849>.
- [16] Brüning et al, LHC Design Report, 2004, <https://cds.cern.ch/record/782076>.
- [17] M. Miyahara et al. (TGC Electronics Group), TGC Patch-Panel ASIC Design Report for Production Readiness Review, 2021.
- [18] National Semiconductor Inc, DS90LV018A 3V LVDS Single CMOS Dif-

- ferential Line Receiver, 2003, <https://datasheet.octopart.com/DS90LV018ATM-National-Semiconductor-datasheet-111584.pdf>.
- [19] CERN, CERN-IPMC: Technical overview, 2017, https://espace.cern.ch/ph-dep-ESE-BE-ATCAEvaluationProject/PP_IPMC/Public%20documents/CERN-IPMC%20-%20Technical%20overview.pdf.
- [20] The ATLAS Collaboration, ATLAS muon spectrometer: Technical Design Report, 1997, <https://cds.cern.ch/record/331068>.
- [21] Dzemaili, Nekija, A reliable booting system for Zynq Ultrascale+ MPSoC devices, 2021, <https://cds.cern.ch/record/2763095>.
- [22] The ATLAS Collaboration, The ATLAS Experiment at the CERN Large Hadron Collider, 2008, <https://dx.doi.org/10.1088/1748-0221/3/08/S08003>.
- [23] Mehlhase, Sascha, ATLAS detector slice (and particle visualisations), 2021, <https://cds.cern.ch/record/2770815>.
- [24] HL-LHC industry, Project Schedule, <https://voisins.web.cern.ch/en/high-luminosity-lhc-hl-lhc>.
- [25] The ATLAS Collaboration, Approved Muon Spectrometer Plots, <https://cds.cern.ch/record/2668478/files/ATL-COM-MUON-2019-012.pdf>.
- [26] 山田千尋, COMET Phase-I CyDet トリガーシステムの性能評価, 2023, <https://openit.kek.jp/workshop/2023/dsys/program>.