

ATLAS muon triggerにおける TCPを用いたDAQ

東京大学素粒子物理国際研究センター
大谷育生

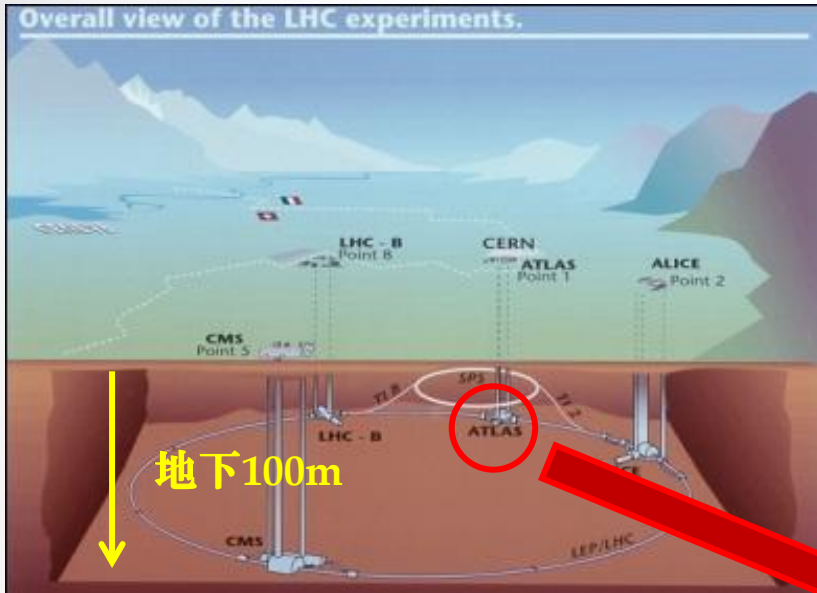
目次

- ATLASのアップグレードに関する読み出しプロトタイプ
の開発について
 - ミューオントリガーシステムのアップグレード
 - SiTCPおよびGTX Transceiverの検証
 - SLプロトタイプの開発について
- ATLAS ミューオントリガーにおけるTCPの導入例
 - 読み出し系における問題
 - Raw data takingのスキーム
 - 開発状況

ATLASアップグレードに関する 研究開発

...

ATLAS実験

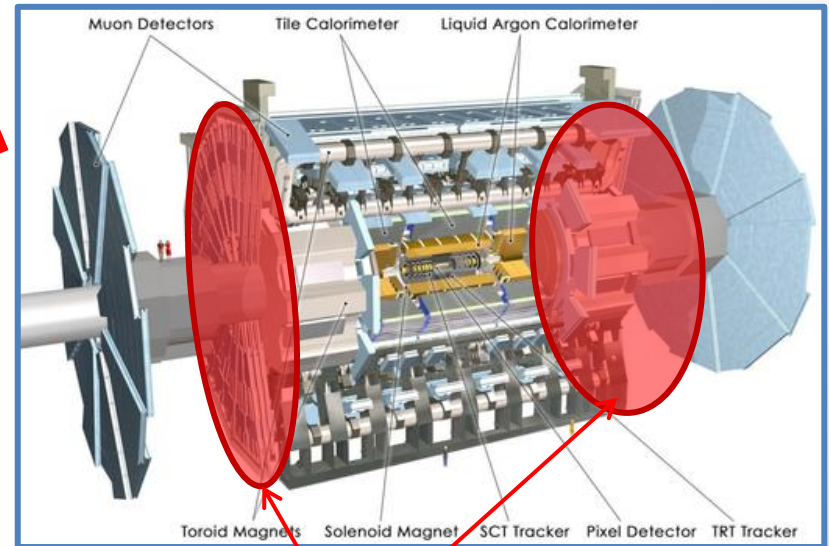


LHC

- 陽子・陽子衝突型加速器
- 周長 26.7km
- 衝突頻度 40MHz
- ビームエネルギー 7TeV
- ルミノシティ $10^{34}\text{cm}^{-2}\text{s}^{-1}$

ATLAS detector

- 大型汎用検出器
 - トラッカー (Pixel, SCT, TRT)
 - カロリメータ (EM, Hadronic)
 - ミューオンスペクトロメータ (MDT, CSC, RPC, TGC)
- ヒッグス粒子や未知の物理の探索

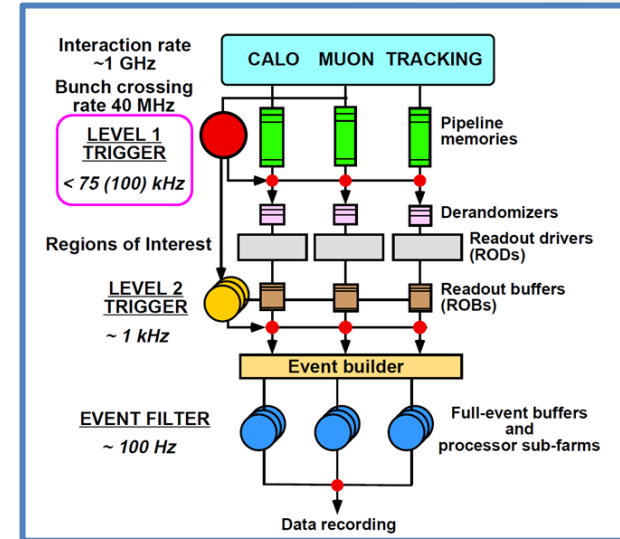
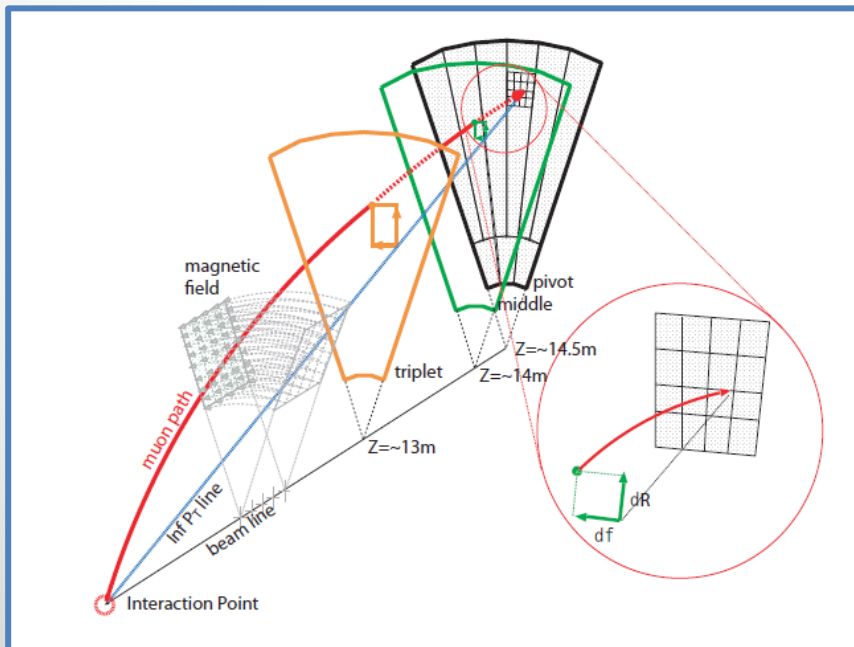


Thin Gap Chamber (TGC)

トリガーシステム

3段階のトリガー

- Level 1 ($\sim 100\text{kHz}$)
 - $2.5\mu\text{s}$ 以内に発行する必要がある
 - ゆえに全てハードウェアによるトリガー
 - TGCやカロリメータが担当
- Level 2 (\sim 数kHz)
- Event Filter (\sim 数100Hz)



TGC L1トリガー

- ミューオンは磁場で曲げられ、3枚のTGCにヒットを生じさせる
- 無限運動量のパス（直線）からのずれで運動量を算出し、高い運動量のミューオンを選び出す

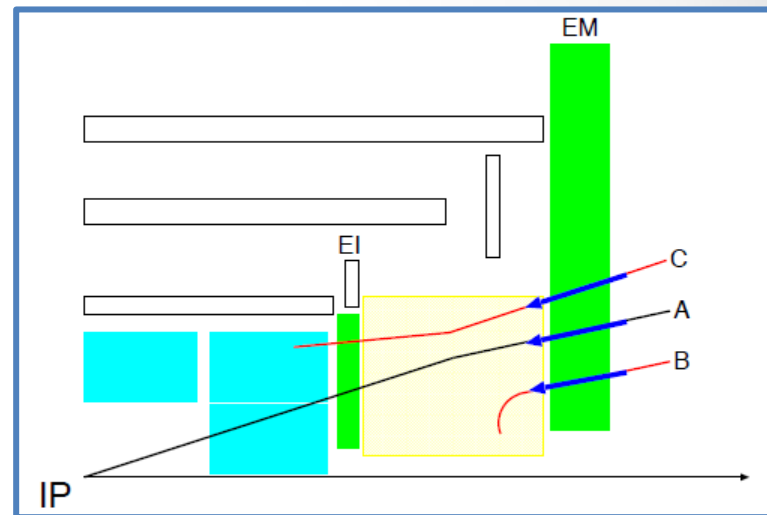
ATLASのアップグレード

高ルミノシティLHC (HL-LHC) へ

- 2020年代へ向け段階的アップグレード
- ルミノシティ： $0.8 \Rightarrow 5 \times 10^{34} \text{cm}^{-2}\text{s}^{-1}$

ATLAS検出器のアップグレード

- 放射線損傷した検出器の交換
- 新トリガーシステムの検討
 - 現行ではTGC L1トリガーの9割超がフェイクミューオン（衝突点以外からくるミューオン）でかかっている
 - 2018年に向けてインナーステーションの情報を用いた新しいトリガーシステムを構築する
→関連エレクトロニクスのアップグレードが必要

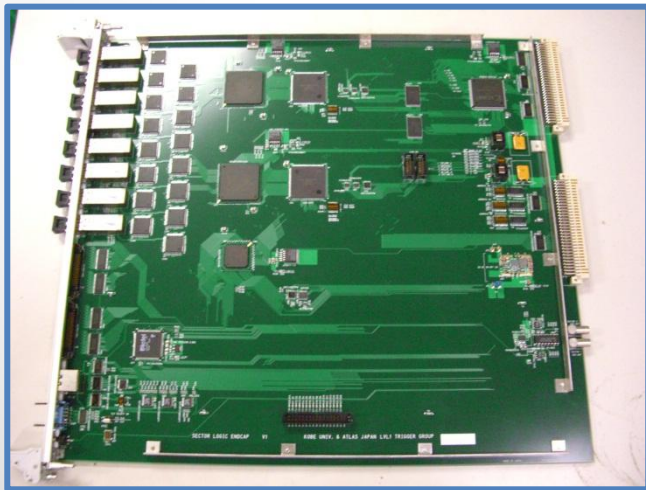


Aが衝突点由来のミューオン
B, Cは別由来だが衝突点から
来るように“見える”

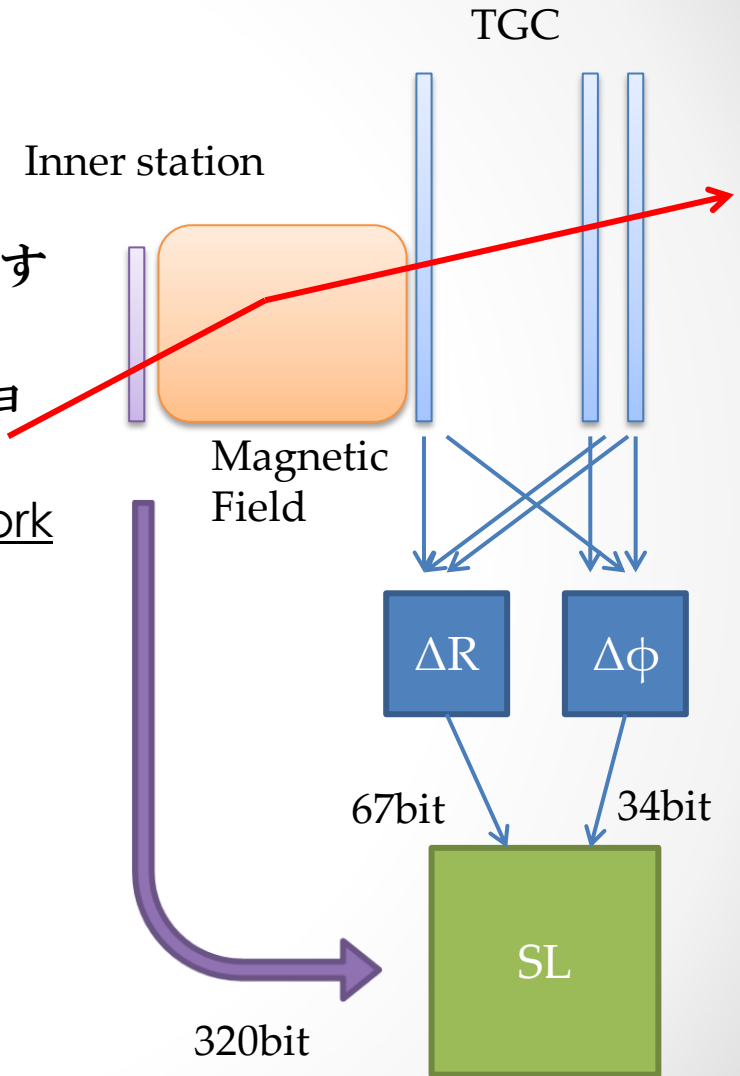
SLのアップグレード

SL (Sector Logic)

- R方向と ϕ 方向の情報を統合し運動量を割り出すモジュール
 - フェイクを落とすために、インナーステーションの情報を入れられる新SLを作る予定
- 新SL開発に向けたプロトタイプ的设计=my work



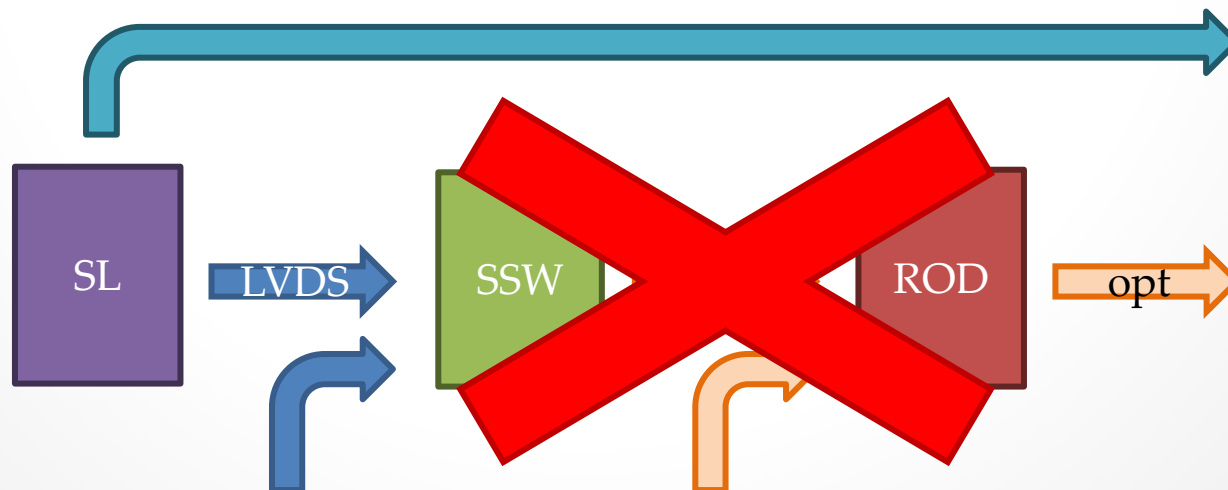
SL board



新SLにおける改善点

- インプットの増量
 - 従来の入力：optical 101bit (4Gbps)
 - インナーステーションからの追加入力：optical 320bit (12.8Gbps)→4倍のインプット

- 新しいリードアウトラインの構築
 - 現行ではリードアウトバッファは他モジュールのASICを流用
 - ASICの出力に合わせて後段のDAQも他のラインに組み込んでいる→バッファをFPGA内に作り、独立した経路を確立する



新SLプロトタイプに必要な技術

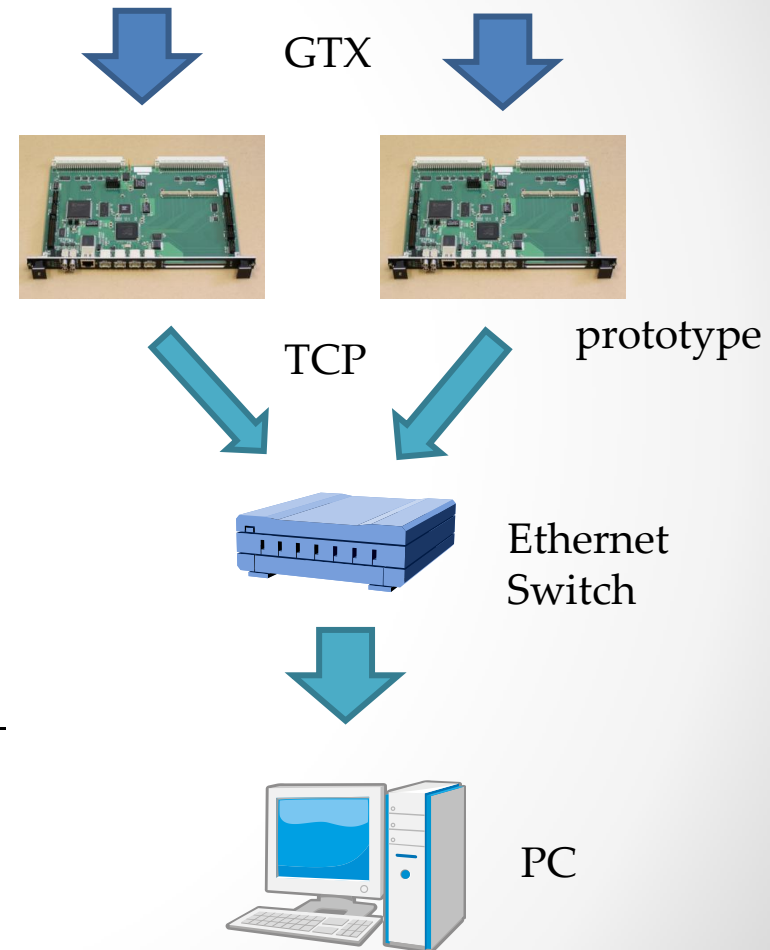
入力: GTX Transceiver

- Xilinx社のハイエンド/ミドルクラスFPGAに搭載されたギガビットトランシーバ
- 少ないリソースで新SLの入力を再現可能

出力: SiTCP

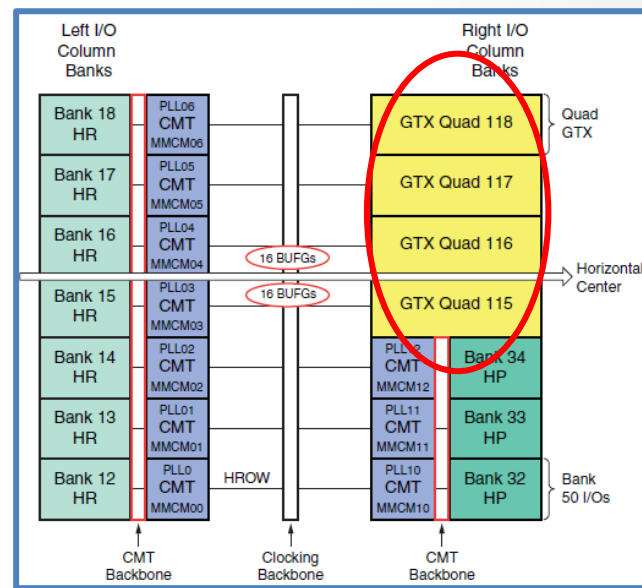
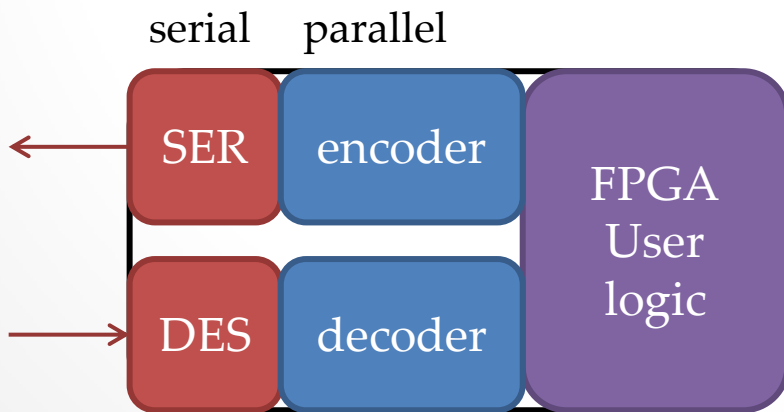
- Bee Beans Technologies社の提供するFPGAベースのネットワークプロセッサ
- TCPを用いることで後段のDAQが容易に

これらの技術を実装したプロトタイプを制作し、
テスト環境を構築する



GTX Transceiver

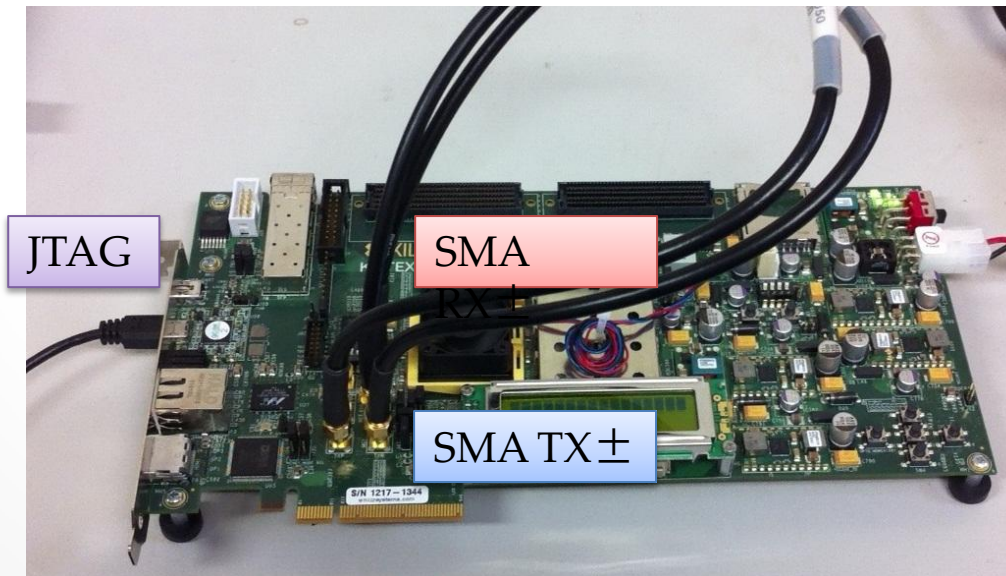
- 高速シリアル通信用ギガビットトランシーバ
 - Xilinx社のKintex7 FPGAなどに搭載されたハードマクロ
 - 1レーンで最大12.5Gbps
 - 全二重通信可
- 種々のプロトコルに対応
 - PCIe, 10GBASE-R, Serial RapidIO
 - 8b10bを用いた単純なコーディングも可能



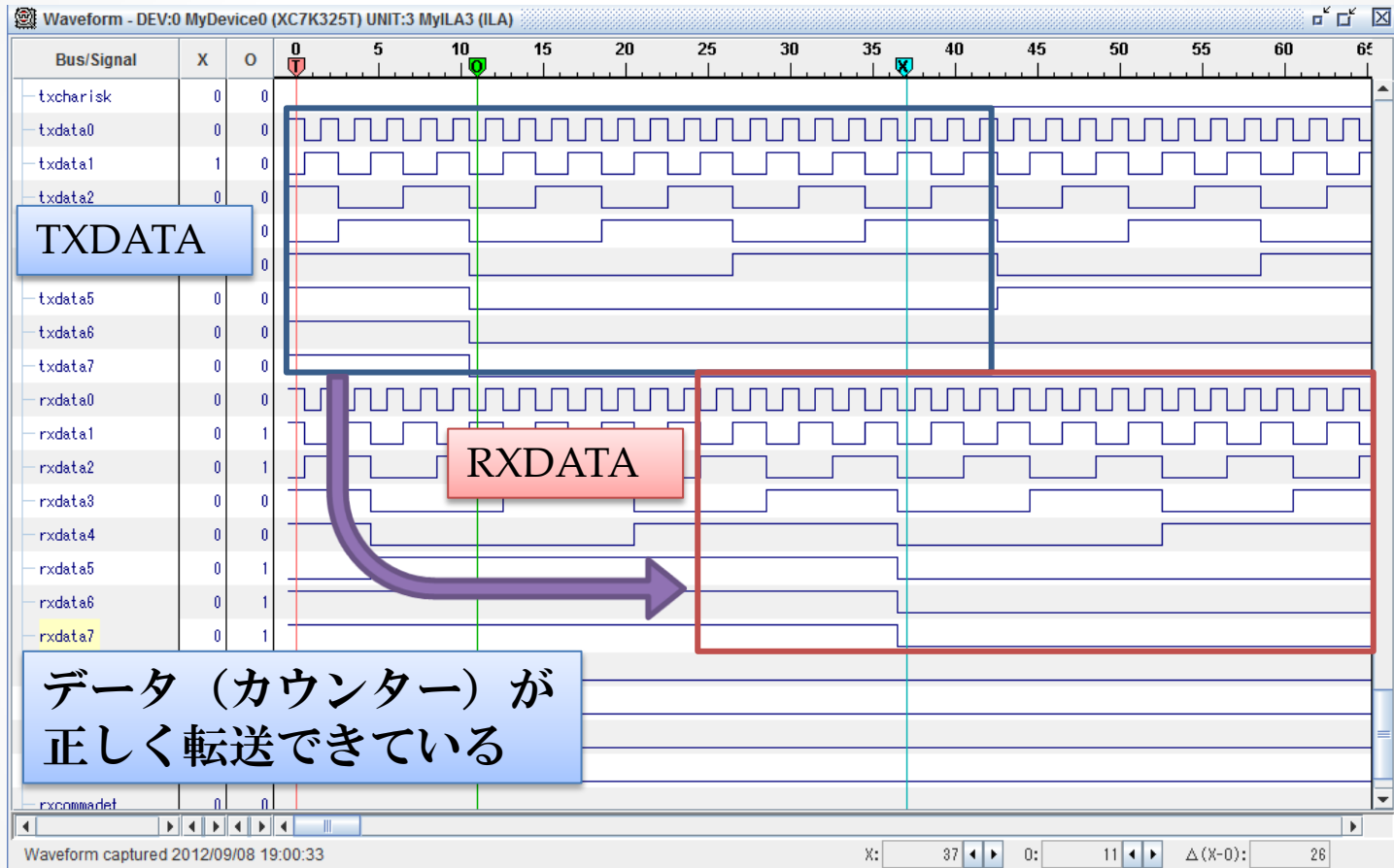
Kintex7 FPGA

GTX Transceiverの検証1

- Kintex7評価ボードKC705を用いる
- GTX Transceiver Wizardでデザインを生成する
 - 2.5Gbps = 20bit x 125MHz
 - 送信データは単純なカウンター
- SMAケーブルでtxから自身のrxへループバックさせる
- デバッグツールChipScope proを用いて検証:次ページ
 - JTAG経由で内部信号が見れるツール



GTX Transceiverの検証1



GTX Transceiverの検証2

- IBERT(Integrated Bit Error Ratio Test)を用いて検証
 - JTAG経由でBER (Bit Error Rate)を計測するツール

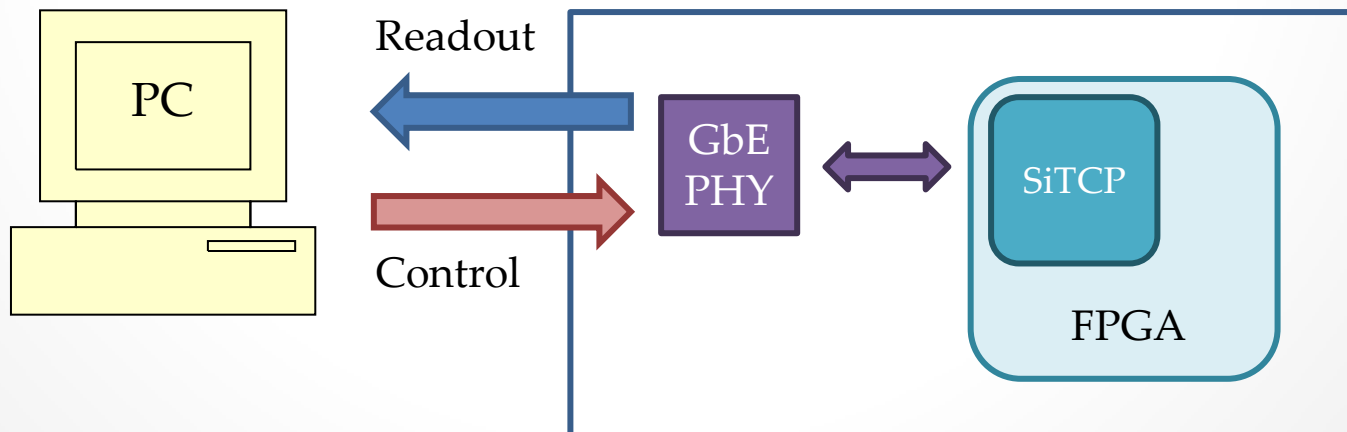
The screenshot shows the IBERT Console interface for a GTX transceiver. The window title is "IBERT Console - DEV:0 MyDevice0 (XC7K325T) UNIT:1_0 MyIBERT K7 GTX1_0 (IBERT K7 GTX)". The interface has four tabs: "MGT/BERT Settings", "DRP Settings", "Port Settings", and "RX Margin Analysis". The "MGT/BERT Settings" tab is active, showing a tree view with "GTX_X0Y8" selected. The "MGT Settings" section includes fields for MGT Alias (GTX0_117), Tile Location (GTX_X0Y8), MGT Link Status (2.5 Gbps), PLL Status (QPLL LOCKED), Loopback Mode (None), Channel Reset (Reset), TX Polarity Invert (checkbox), TX Error Inject (Inject), TX Diff Output Swing (850 mV (1100)), TX Pre-Cursor (1.67 dB (00111)), TX Post-Cursor (0.68 dB (00011)), RX Polarity Invert (checkbox), RX Termination Mode (Programmable), and RX Termination Voltage (800 mV). The "BERT Settings" section includes TX Data Pattern (PRBS 31-bit), RX Data Pattern (PRBS 31-bit), RX Bit Error Ratio (1.814E-014), RX Received Bit Count (5.512E013), RX Bit Error Count (0.000E000), and a BERT Reset (Reset) button. Two callout boxes are present: a blue one pointing to "2.5 Gbps" with the text "2.5Gbpsでリンクが取れている" and a red one pointing to "1.814E-014" with the text "BER 1.8E-14 5.5E13bit送ってエラーなし".

Category	Parameter	Value
MGT Settings	MGT Alias	GTX0_117
MGT Settings	Tile Location	GTX_X0Y8
MGT Settings	MGT Link Status	2.5 Gbps
MGT Settings	PLL Status	QPLL LOCKED
MGT Settings	Loopback Mode	None
MGT Settings	Channel Reset	Reset
MGT Settings	TX Polarity Invert	<input type="checkbox"/>
MGT Settings	TX Error Inject	Inject
MGT Settings	TX Diff Output Swing	850 mV (1100)
MGT Settings	TX Pre-Cursor	1.67 dB (00111)
MGT Settings	TX Post-Cursor	0.68 dB (00011)
MGT Settings	RX Polarity Invert	<input type="checkbox"/>
MGT Settings	RX Termination Mode	Programmable
MGT Settings	RX Termination Voltage	800 mV
BERT Settings	TX Data Pattern	PRBS 31-bit
BERT Settings	RX Data Pattern	PRBS 31-bit
BERT Settings	RX Bit Error Ratio	1.814E-014
BERT Settings	RX Received Bit Count	5.512E013
BERT Settings	RX Bit Error Count	0.000E000
BERT Settings	BERT Reset	Reset

SiTCP

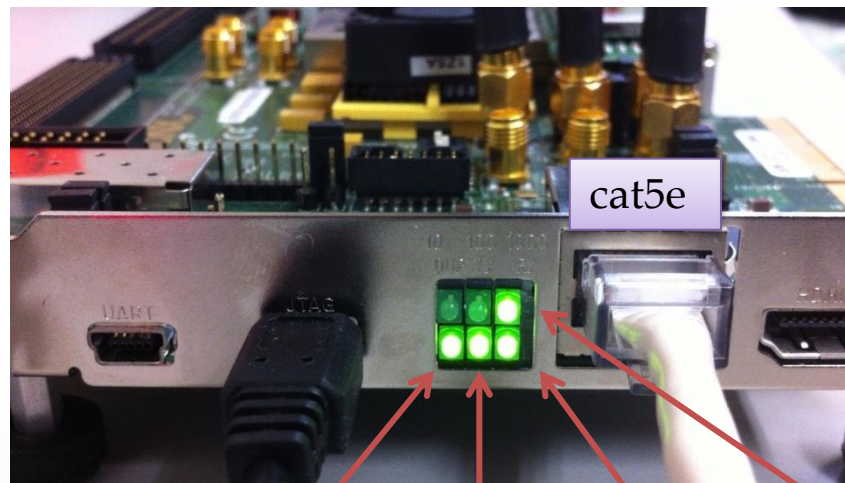


- ハードウェアによるネットワークプロセッサ
 - KEKの内田さんにより開発された技術
 - FPGAに組み込むことで手軽にギガビットイーサネットが使える
 - FPGA側からはFIFO、PC側からはサーバーに見える
- TCPによる高速リードアウト
 - 最大1Gbps
- UDPベースのプロトコルにより外部からのコントロールも可能



SiTCPの検証

- Kintex7評価ボードKC705を用いる
- Kintex7用SiTCP (ver50) をデザインに組み込む
 - 送信データは単純なカウンター
- Cat5eケーブルでPCとつなぐ
- Bee Beans Technologiesのデバッグツールを用いて検証:次ページ



Full duplex

Tx

Rx

1000M Link

SiTCPの検証

The screenshot shows the SiTCP Utility application window. The title bar reads "SiTcpユーティリティ". The main window contains the following information:

- Version: 0.9.2
- Copyright(C) Bee Beans Technologies, Inc.
- Logo: bee beans with an eye icon.
- Navigation tabs: 一般 (General), 制御(UDP) (Control/UDP), データ(TCP) (Data/TCP).
- Statistics: 受信サイズ=647,168 Bytes, 受信時間=31.17 sec, 受信レート=166.11 kbps.
- Buttons: 受信停止 (Stop Reception).
- File operations: ファイルに保存 (Save to File), ... (Browse), 開く (Open).
- Options: ファイルのテキスト変換 (Convert file text), データ表示(65535byteまで) (Data display up to 65535 bytes).
- Format field: フォーマット 11111111111111111111.
- Data dump table:

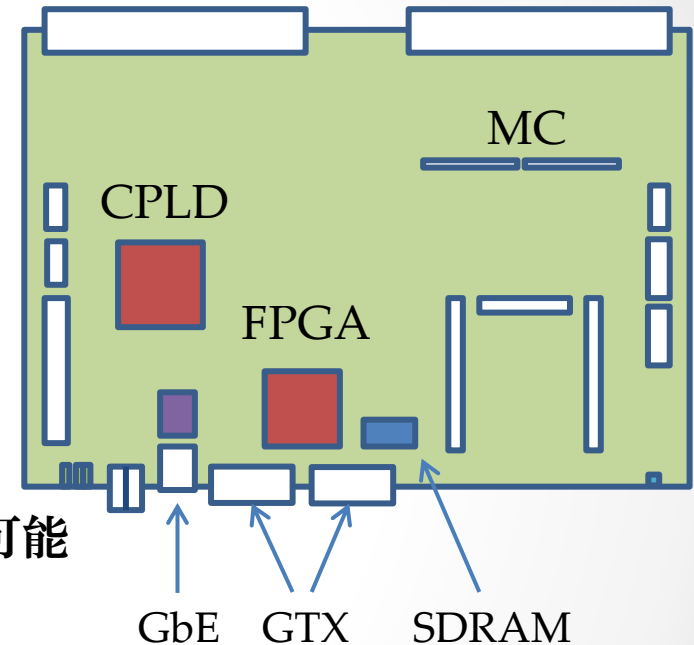
54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63
64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73
74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	80	81	82	83
84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93
94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	A0	A1	A2	A3
A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3
B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	C0	C1	C2	C3
C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3
D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF	E0	E1	E2	E3
E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0	F1	F2	F3
F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF	00	01	02	03

データ (カウンター) が
正しく転送できている

新プロトタイプの開発

- 名称：ProtoType7 (PT7)
 - 6UサイズのVMEモジュール
 - 汎用モジュールとしても使用可能
- Kintex7 FPGA (XC7K325T) を搭載
- GTX Transceiverを8レーン使用
 - コネクタはinfiniband 4xを2つ使用
- SiTCPによるギガビットイーサネット
- DDR3-1333 SDRAMを搭載
- Mezzanine Cardでインターフェイスの拡張可能
- MicroBlaze CPUを組み込むことを想定

PT7 概略図



部品の選定：FPGA, SDRAM

SDRAM = MT41J64M16JT-15E

- コンポーネントDDR3 SDRAM
 - 省スペース
- 16bit x 8Meg x 8banks = 1Gb
- バスクロック667MHz (DDR3-1333)
 - 1333MT/s x 16 = 21Gbps
 - 新SLの入力17Gbpsを受けられる

FPGA = XC7K325T-2FFG900

- デバイス：XC325T
 - 最大User IOが500と多い
 - 評価ボードKC705と同じ
- パッケージ：-2FFG900
 - DDR3: 最大1333MT/s
 - GTX: 最大10.3125Gbps

FPGA	Logic cell	Block RAM	GMT	CMT	DSP	User IO
6SLX150T	150,000	5Mb	8GTP	6	180	540
7K160T	160,000	12Mb	8GTX	8	600	400
7K325T	325,000	16Mb	16GTX	10	840	500
7K410T	410,000	29Mb	16GTX	10	1500	500

部品の選定：GTX

コネクタ：Infiniband 4x

- SMAではtx±とrx±の計4本のケーブルが必要になる
- Infinibandではこれら4本を1レーンにまとめて全二重通信を可能にする
- さらに4xタイプなら4レーンを1ケーブルでまかなう
- DDR (double data rate)対応のケーブルを用いることで $5\text{Gbps} \times 4 = 20\text{Gbps}$
- 1x-4xタイプのケーブルでPT6との互換性の確保



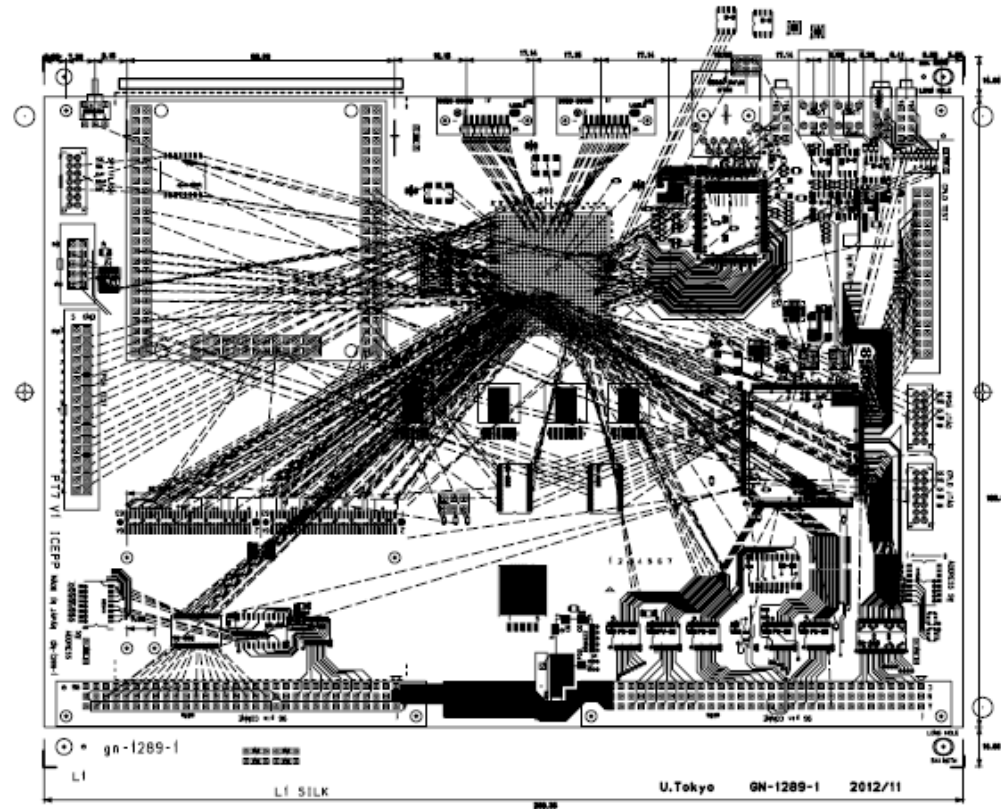
4x copper cable



4x receptacle

開発状況

- 回路図の作成と部品の選定は終了、現在は配線作業中
- 12月中旬ごろ試作機が完成予定
- できあがり次第年末にかけてテストを行う



ATLAS ミューオントリガーにおける TCPの導入例

...

TGC readout

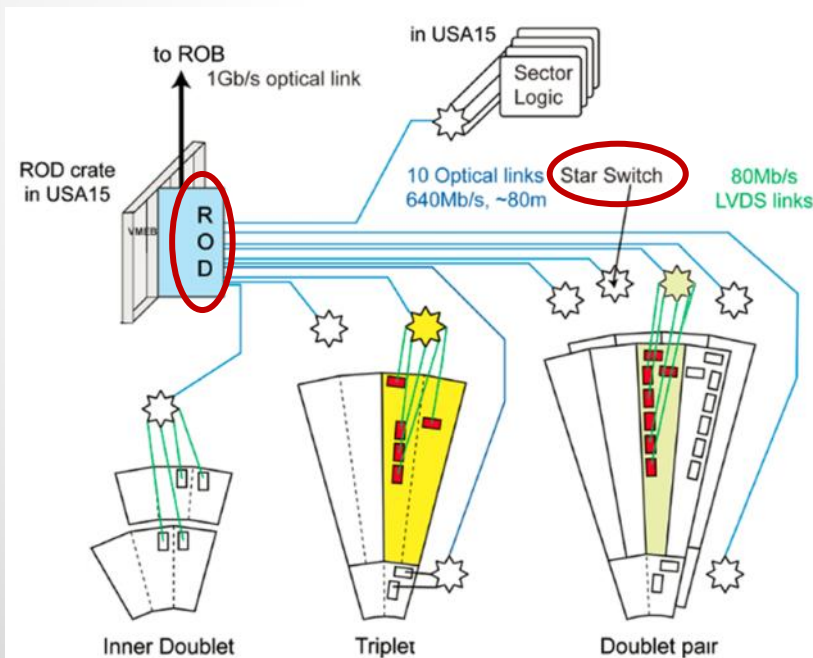
TGC読み出し系

- TGCのデータはStar Switch (SSW) に集められ、圧縮を受ける
- SSW10個分のデータがRead Out Driver (ROD)に集められる

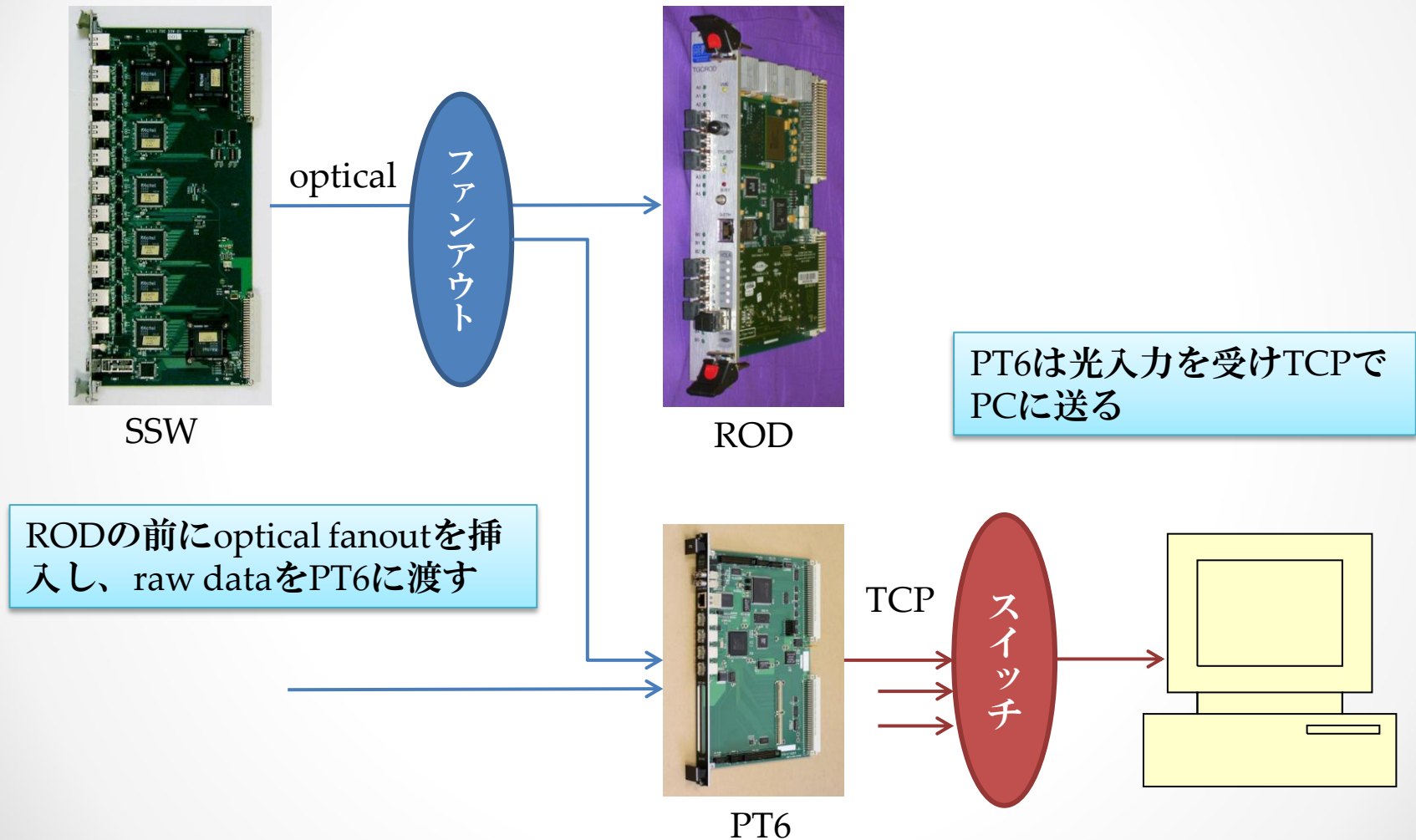
ROD (Read Out Driver)

- SSWからのデータをまとめてATLAS共通のフォーマットに変換する
 - よくbusyを起こしてrunがとまる
- 原因を究明するために

SSWからのraw dataを見る必要あり



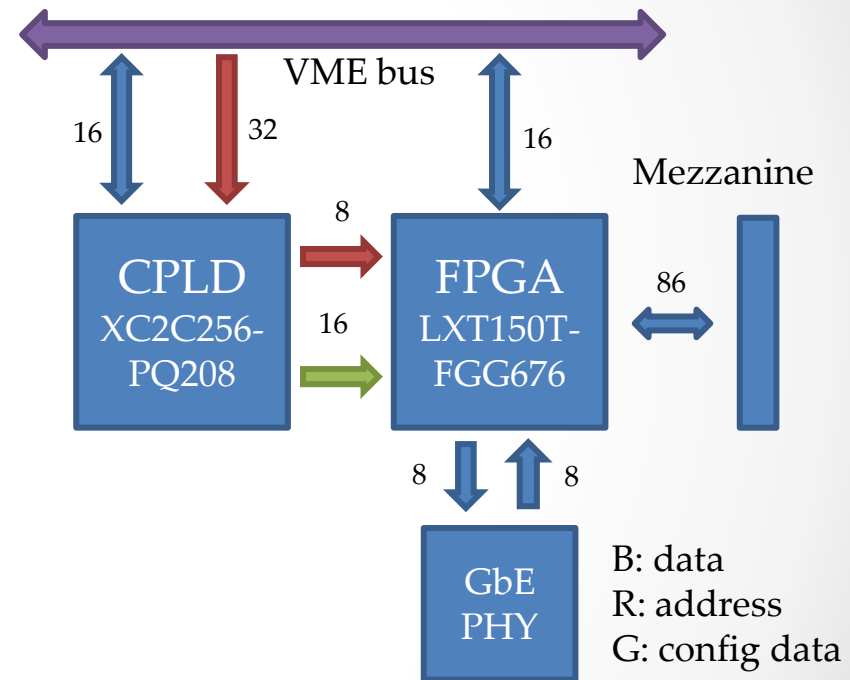
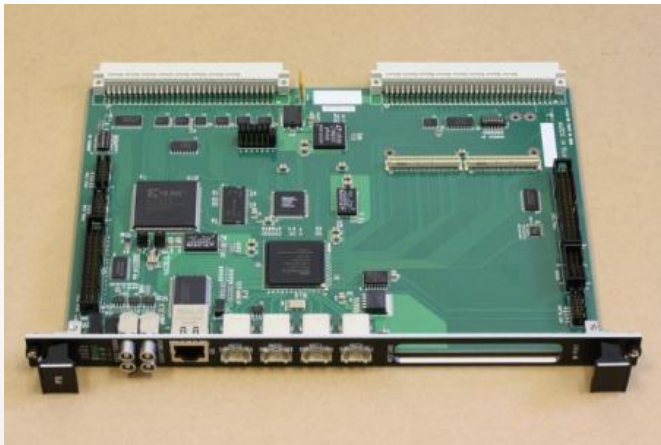
Raw data taking scheme



PT6ボード



- RODのアップグレードプロトタイプとして開発されたVMEモジュール
- Spartan6 FPGAにSiTCPを搭載してギガビットイーサネットを使用可能
- Mezzanine Cardを載せることでSSW2つからの光入力を受けられる



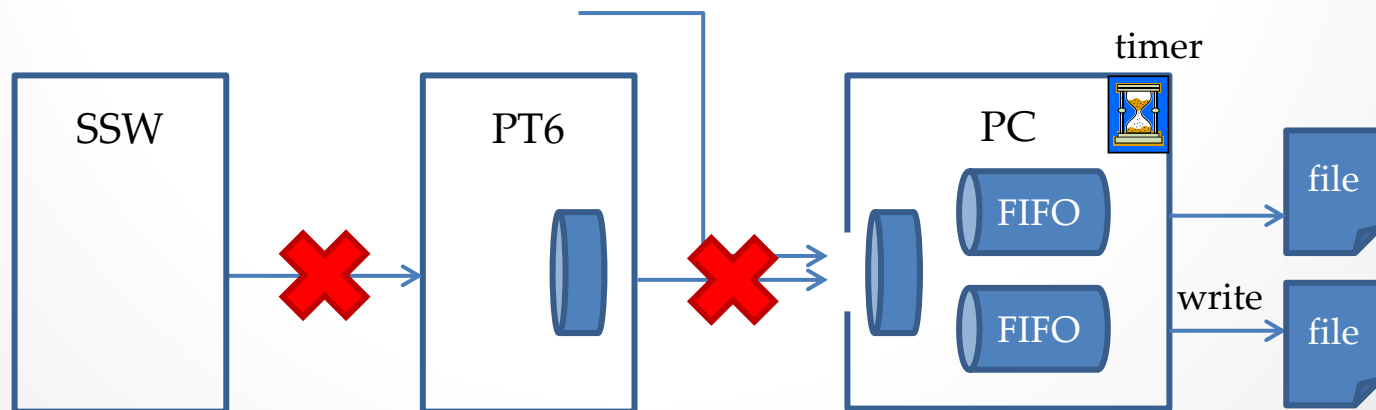
DAQの方法1：PCにバッファ

PT6側

- PT6はSSWから受けとった（中身のある）データを全てSiTCPに渡す

PC側

- PCは複数のPT6とTCP connectionを確立しておく
- Selectシステムコールでポーリングし、到着したデータをFIFOに格納する
- RODがbusyになるとSSWからのデータが止まり、PT6もデータを流せなくなる
- PCはデータが来ない時間をカウントして一定時間経ったらファイルにダンプする



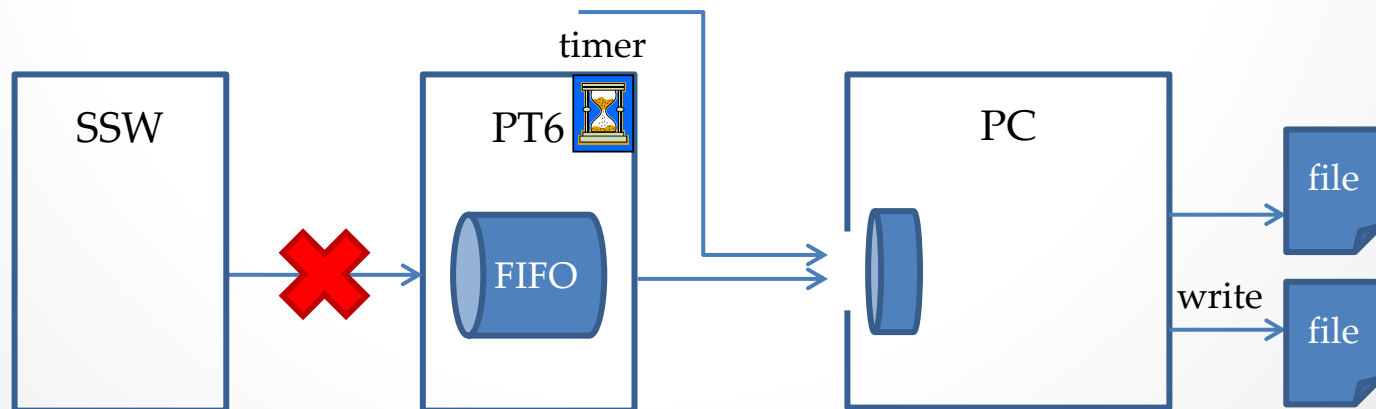
DAQの方法2：PT6にバッファ

PT6側

- PT6内のBlock RAMを用いてSSWからくるデータをバッファする
- RODがbusyになるとSSWからのデータが止まるので、ある一定期間データがこないときにbusyと判断してトリガ
- トリガがかかったらバッファにあるすべてのデータをSiTCPに渡す

PC側

- PCは複数のPT6とTCP connectionを確立しておく
- Selectシステムコールでポーリングし、データが到着していたらファイルにダンプ



長所・短所

PT6にバッファ

利点

- 帯域を節約できる（データは止まっているのでゆっくり投げればよい）

欠点

- HDLによる記述でデバッグしにくい
- バッファが限られる
- PT6のBlock RAMは最大600kbyte
→ $600k / 400 = 1.5k \text{ event} = \text{約}30\text{ms}$

PCにバッファ

利点

- バッファの大きさを大きくできる
- ソフトウェアによる処理なので開発やデバッグが容易

欠点

- 1つのPT6が占める帯域が大きい
- 2つのSSWから400byte x 50kHz
→160Mbps（せいぜいPT6は5つまで）

今回はリソースも限られているため、PT6にバッファする方針で行く
私が関連するSoftwareとFirmwareの開発を行った

Firmwareのテスト

コンフィギュレーション

- SSWの代わりにとなるPT6を2台、その間に試験対象のPT6を配置
- 中心のPT6はNIMでクロックの分配とTCPでデータ読み出し
- 両端のPT6はカウントデータを光で送る

結果

- データが届いていないときに正しくトリガし、バッファ内のデータを全てダンプできた
- しかし読み出したデータに誤りが見受けられ、どこで生じているか探している
- 11月下旬のLHC machine developmentまでにfixする予定



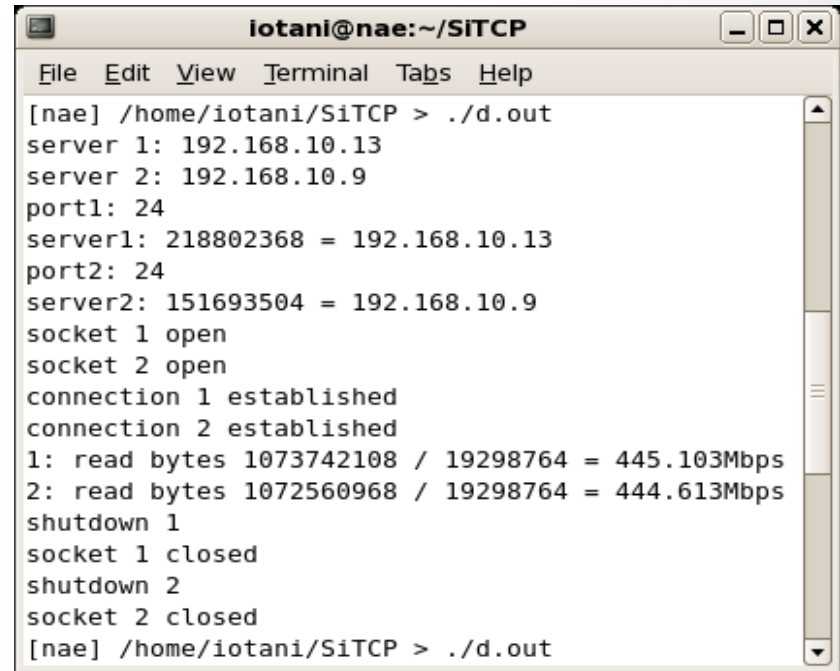
Softwareのテスト

コンフィギュレーション

- イーサネットスイッチ経由で2台のPT6をPCとつなぐ
- 最大限の速さでデータをプッシュする

結果

- 2台のPT6からデータを正しくダンプできた
- 速度はおよそ900Mbps
- 前述のFirmwareのデバッグが済み次第、統合試験をする予定

A terminal window titled 'iotani@nae:~/SiTCP' showing the execution of a script './d.out'. The output displays server and port information, socket opening, connection establishment, and data transfer statistics for two servers. The transfer rates are 445.103Mbps for server 1 and 444.613Mbps for server 2.

```
iotani@nae:~/SiTCP
File Edit View Terminal Tabs Help
[nae] /home/iotani/SiTCP > ./d.out
server 1: 192.168.10.13
server 2: 192.168.10.9
port1: 24
server1: 218802368 = 192.168.10.13
port2: 24
server2: 151693504 = 192.168.10.9
socket 1 open
socket 2 open
connection 1 established
connection 2 established
1: read bytes 1073742108 / 19298764 = 445.103Mbps
2: read bytes 1072560968 / 19298764 = 444.613Mbps
shutdown 1
socket 1 closed
shutdown 2
socket 2 closed
[nae] /home/iotani/SiTCP > ./d.out
```

まとめ

ATLASアップグレードに関して

- ATLASミュオン検出器の運動量判定モジュールSLのアップグレードプロトタイプとしてVMEモジュールPT7を開発中
- Kintex7にSiTCPを載せる予定で、評価ボードを用いて動作検証した
- 現在配線作業中で12月中に完成予定

TCPの導入例について

- データ読み出しモジュールRODのbusyの原因追及のためraw dataをファンアウトしてTCPで読み出すシステムを開発中
- PT6にバッファをおく方法でFirmwareとSoftwareが1通り完成した