

# MicroBlazeを用いたPT6のFPGA設計

東京大学大学院 素粒子物理国際研究センター  
二ノ宮 陽一

1. PT6について
2. MicroBlazeについて
3. 開発環境
4. MicroBlaze使用例
5. PT6のFPGAデザイン
6. まとめ
7. 要望など

PT6はATLAS実験のミュオントリガー検出器であるTGCの読み出しシステムのアップグレード研究のために開発された。

PT6を用いて、アップグレードの読み出しシステムのFPGA設計を行う。

アップグレードでは、L1トリガーレートが150kHzになるため、読み出しシステムは460MByte/sの処理速度が求められる。

安定した動作、強靱性を備えた読み出しシステム開発のために、MicroBlazeというソフトプロセッサコアの導入を検討。

Xilinx社製FPGAに搭載可能なソフトプロセッサコア。

- ・32bit RISC型
- ・ハーバードアーキテクチャ
- ・ビッグエンディアン
- ・OSを搭載可(μITRON、uClinux)

## ユーザ定義

- ・クロック周波数
  - Spartan6、230MHz程度
- ・パイプライン段数
  - 3段 or 5段
- ・周辺インターフェイス
- ・メモリ管理ユニット



# MicroBlazeの利点

---

- ・ソフトウェアによる処理を可能にする。

- CPU + ソフトウェアのメリット

- HDLをモジュール化(単機能化)し、HDL間接続の動的制御を行う。

- ・デバッグが容易。
      - ・拡張性が高い。

- HDLは条件分岐が多くなるような処理には適していないので、複雑な処理が必要な部分(エラー処理など)をソフトウェアが行う。

- ・FPGAがあれば使用可能。

- 既存のボードでもFPGAがあれば使える

- ・基板の広さやコストを節約

# 組み込みの利点

## 組み込みの利点

1. 空のfor loopを1000万回処理したときに要する時間

	時間	クロック数	1ループあたり
<b>MicroBlaze (50MHz)</b>	<b>2.6s</b>	<b>1億3000万</b>	<b>13</b>
<b>Pentium4 (2.4GHz)</b>	<b>16.8ms</b>	<b>4000万</b>	<b>4</b>

2. ボード上のメモリへのアクセスと、VME経由でスレイブボード上のメモリアクセス

	時間	クロック数
<b>MicroBlaze</b>	<b>0.94us</b>	<b>47</b>
<b>Pentium4</b>	<b>4.97us</b>	<b>(11,200)</b>



# MicroBlazeの開発環境

---

Xilinxが提供する開発ツール。GUI環境下で構築、設定。

Xilinx ISE(Integrated Software Environment)

FPGAおよびCPLDデバイスを設計するための総合デザイン開発環境

Xilinx EDK(Embedded Development Kit)

- ・XPS

Xilinx Platform Studio。MicroBlazeやPowerPCプロセッサを使用したエンベデッドプロセッサシステムのハードウェア、ソフトウェアを開発するための環境。

- ・SDK

ソフトウェア開発キット。ソフトウェアアプリケーションプロジェクトの開発環境。



The screenshot displays the Xilinx ISE Project Navigator interface. The main window shows the Design Summary for the project 'S3E\_SK\_SiTCP'. The Design Overview pane lists various reports such as Summary, IOB Properties, Module Level Utilization, Timing Constraints, Pinout Report, Clock Report, Static Timing, Errors and Warnings, and Detailed Reports. The Errors and Warnings pane shows a list of messages including Parser Messages, Synthesis Messages, Translation Messages, Map Messages, Place and Route Messages, Timing Messages, Bitgen Messages, and All Implementation Messages. The Design Properties pane shows options for enabling message filtering and displaying design summary contents.

The Device Utilization Summary table provides a detailed breakdown of logic utilization:

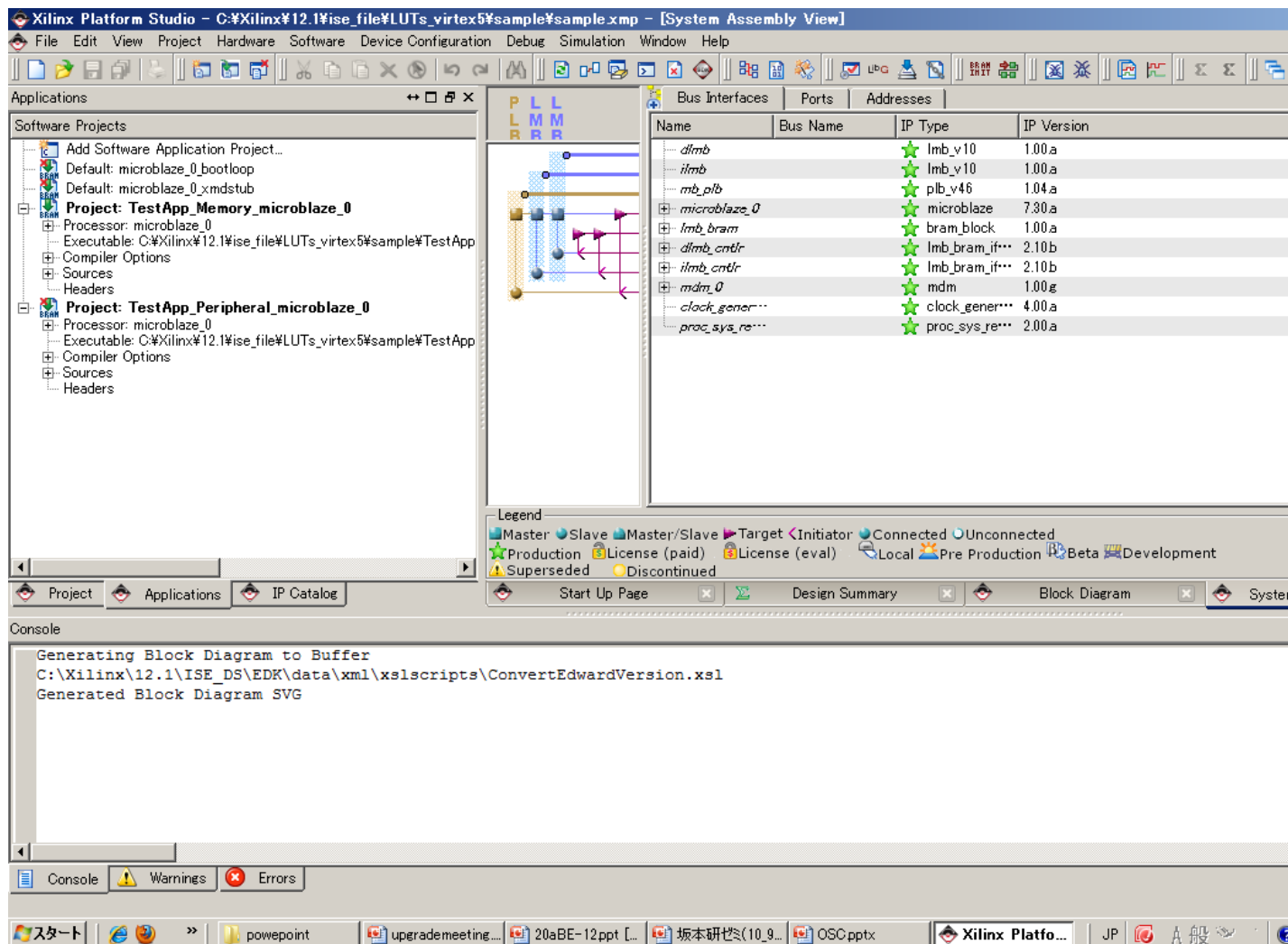
Device Utilization Summary		
Logic Utilization	Used	Avail
Number of Slice Flip Flops	4,334	
Number of 4 input LUTs	3,820	
Number of occupied Slices	3,314	
Number of Slices containing only related logic	3,314	
Number of Slices containing unrelated logic	0	
Total Number of 4 input LUTs	4,109	
Number used as logic	3,707	
Number used as a route-thru	289	
Number used as 16x1 RAMs	8	
Number used as Shift registers	105	
Number of bonded IOBs	31	
Number of RAMB16s	12	

The Console window at the bottom shows the following messages:

```

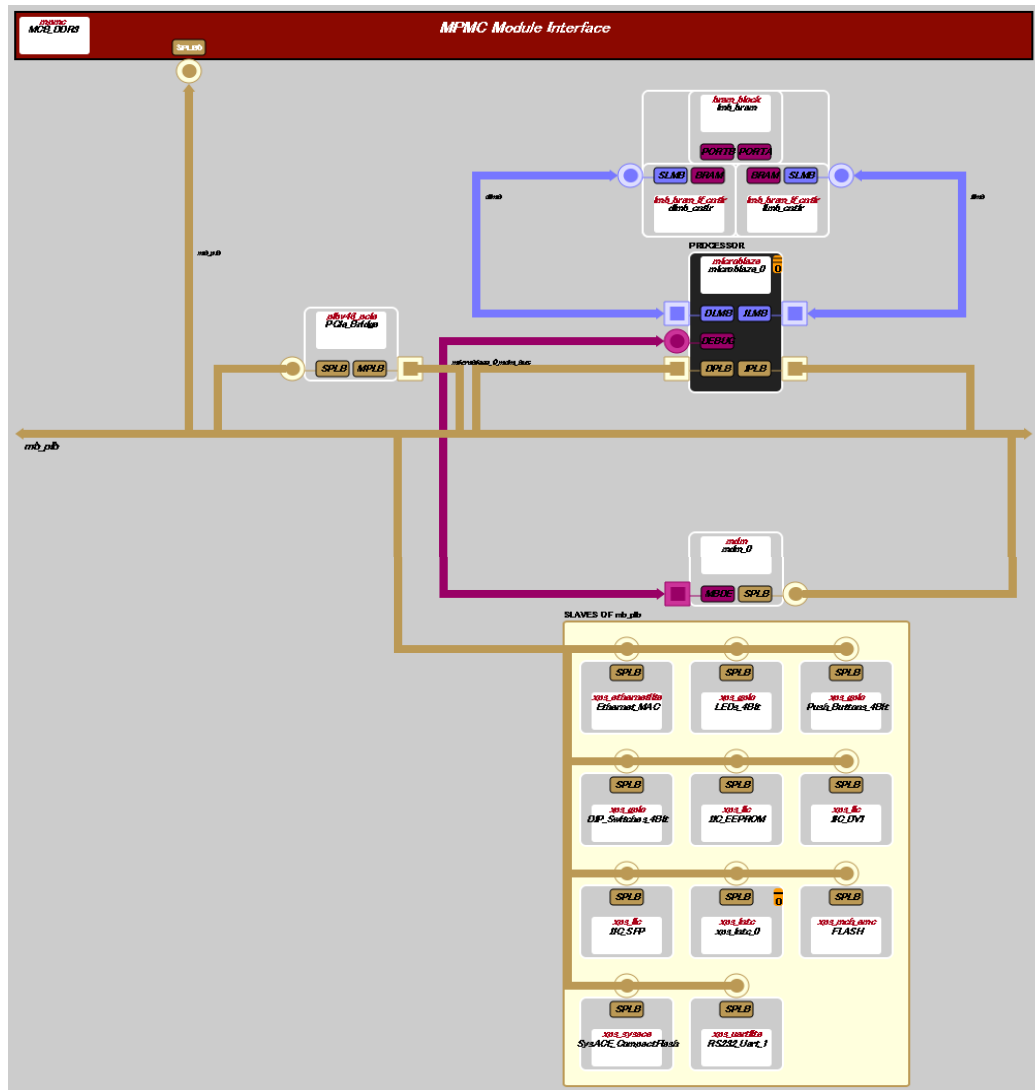
INFO:HDLCompiler:1574 - Analyzing Verilog file \"C:/Xilinx/12.1/ise_file/seminar_2010/SiTCP/sitcp_fifo.v\" into library work
INFO:ProjectMgmt:656 - Parsing design hierarchy completed successfully.
Launching Design Summary/Report Viewer...
    
```

## VHDL, Verilogで記述

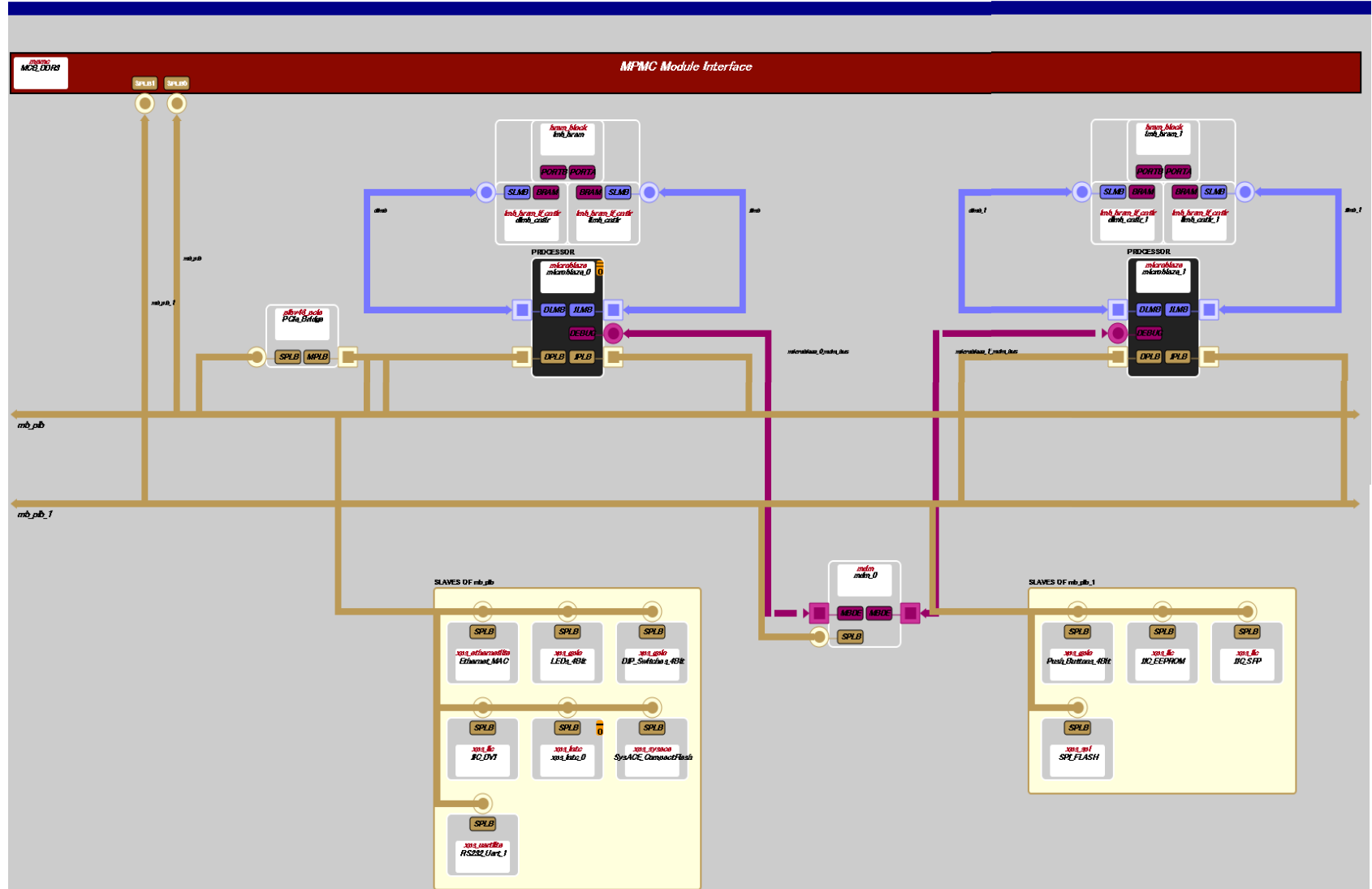


## C,C++で記述

# Single Core



# Dual Core



# MicroBlazeの開発環境

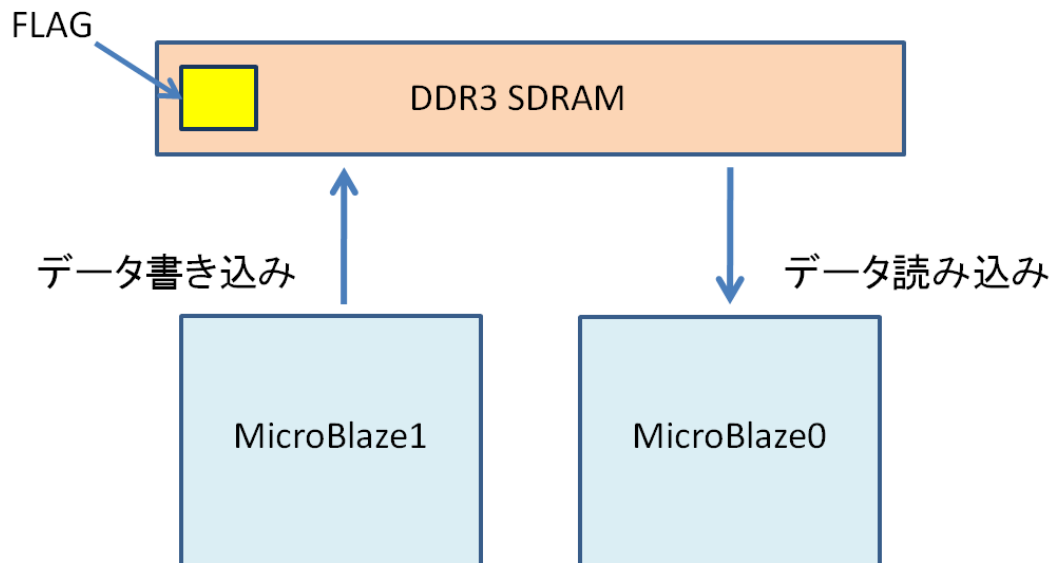
XPS Synthesis Summary (estimated values)				
Report	Generated	Flip Flops Used	LUTs Used	BRAMS Used
<a href="#">system</a>	水 1 26 01:02:54 2011	2127	2589	4
<a href="#">clock_generator_0_wrapper</a>	水 1 26 01:02:34 2011			
<a href="#">rs232_uart_1_wrapper</a>	水 1 26 00:45:19 2011	146	144	
<a href="#">microblaze_0_wrapper</a>	水 1 26 00:45:10 2011	1281	1573	
<a href="#">dataio_wrapper</a>	火 1 25 21:45:18 2011	270	263	
<a href="#">flagio_wrapper</a>	火 1 25 21:45:08 2011	77	43	
<a href="#">proc_sys_reset_0_wrapper</a>	火 1 25 21:44:58 2011	69	54	
<a href="#">mdm_0_wrapper</a>	火 1 25 21:44:54 2011	126	127	
<a href="#">lmb_bram_wrapper</a>	火 1 25 21:44:33 2011			4
<a href="#">ilmb_cntlr_wrapper</a>	火 1 25 21:44:29 2011	2	6	
<a href="#">dlmb_cntlr_wrapper</a>	火 1 25 21:44:25 2011	2	6	
<a href="#">dlmb_wrapper</a>	火 1 25 21:44:21 2011	1	1	
<a href="#">ilmb_wrapper</a>	火 1 25 21:44:15 2011	1	1	
<a href="#">mb_plb_wrapper</a>	火 1 25 21:44:11 2011	152	371	

MicroBlazeの生成には約1500LUTsが必要。

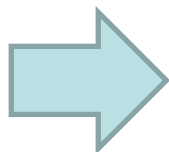
PT6に搭載されている、Spartan6 LX150Tは92152LUTs。

# ボード上のメモリへのアクセス

Multi-Port Memory ControllerというIPコアを用いることで簡単に操作できる。



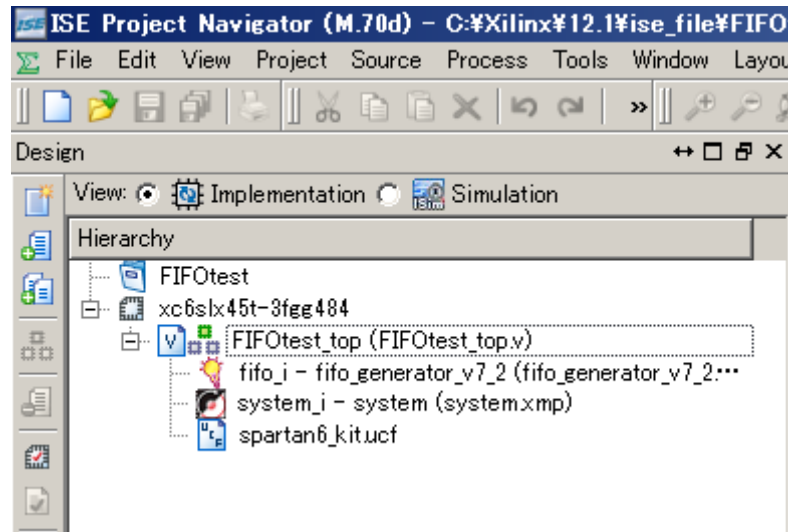
DDR3 SDRAMを2つのMicroBlazeの共用メモリとして使用。  
メモリ上にデータの書き込みと読み込みを制御するためにFLAGを設け、2つのMicroBlazeはこの値を参照することで、書き込みと読み込みが同時に行われないようにする。



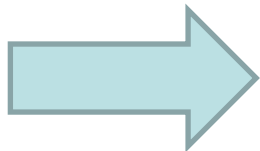
正しく作動していることを確認

# ISEとEDKの統括

EDKのデザインをISEのサブモジュールとして組み込むこともできる。

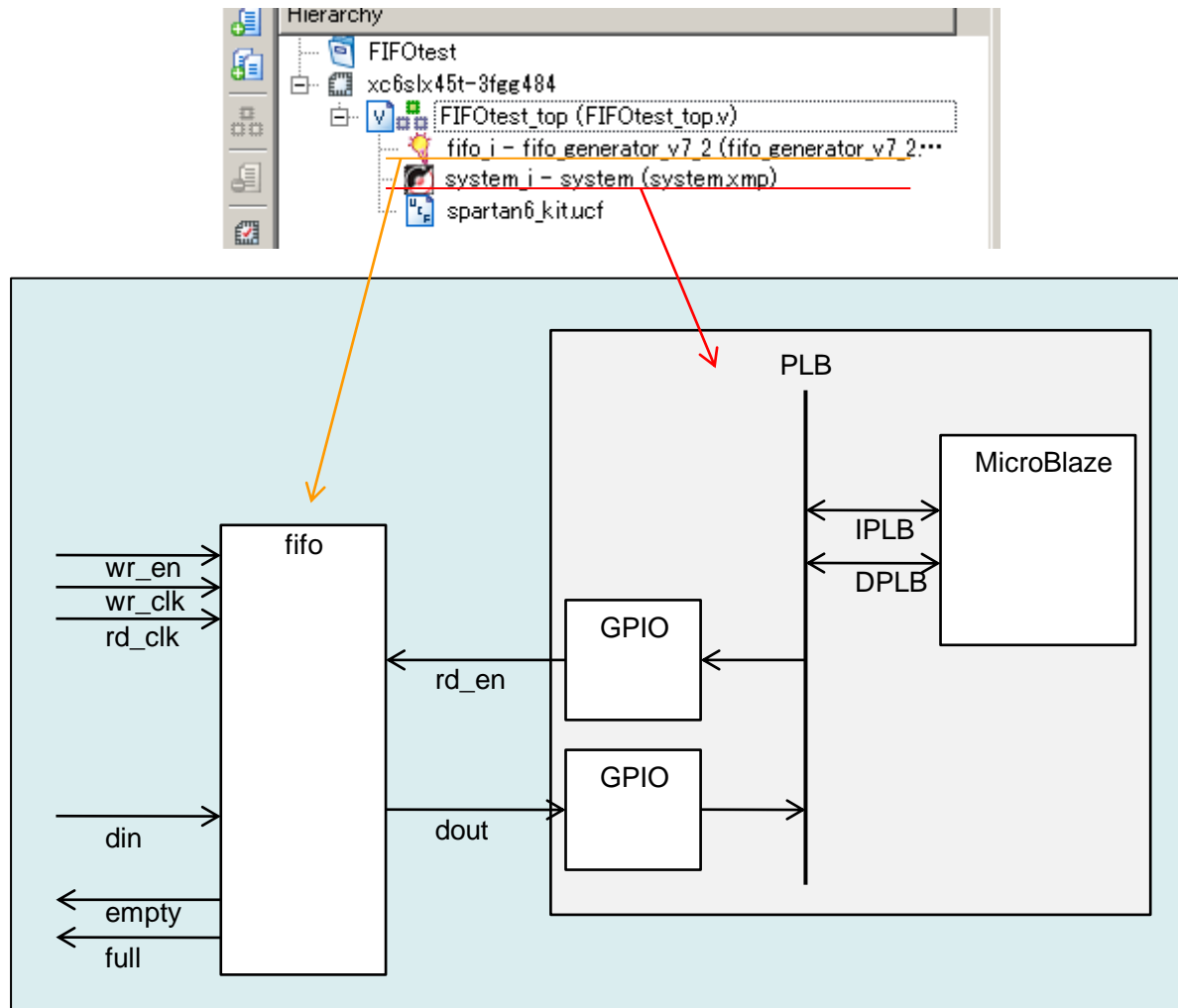


EDKでGPIO(General Purpose IO)を用意しておくことで、ISE側で入出力の信号線として使用することができる。



EDK側のソフトウェアによる処理結果をISE側に渡したり、その逆の処理などが可能になる。

# FIFOバッファのデータ取り出し



MicroBlazeがデータをプルする場合



# FIFOバッファのデータ取り出し

1. MicroBlazeがデータをプルする。

1つの関数を使用するのに95クロック必要。

for loopを用いて100回データ取得を繰り返したところ、12555クロックかかった。

従って、7.3MByte/s(230MHz)でデータを受信することができると言える。

```
XGpio_DiscreteWrite(&flagIO,1,0);
```

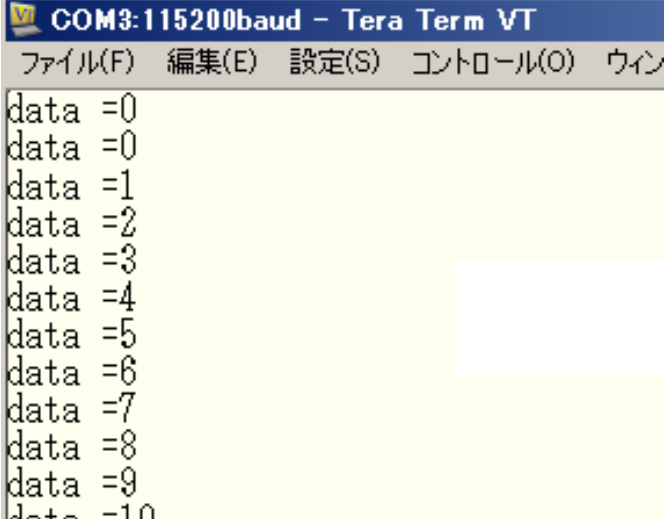
```
for(i=0;i<100;i++){
```

```
    XGpio_DiscreteWrite(&flagIO,1,1);
```

```
    data[i] = XGpio_DiscreteRead(&dataIO,1);
```

```
    XGpio_DiscreteWrite(&flagIO,1,0);
```

```
}
```



The screenshot shows a terminal window titled "COM3:115200baud - Tera Term VT". The window contains a list of data values: data =0, data =0, data =1, data =2, data =3, data =4, data =5, data =6, data =7, data =8, data =9, data =10. The text is displayed in a monospaced font on a light background.

# FIFOバッファのデータ取り出し

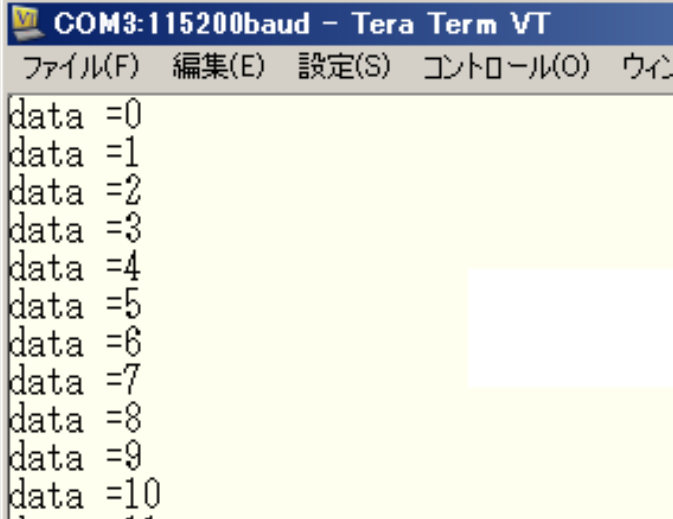
## 2. HDLデザイン側がデータをプッシュする。

関数のオーバーヘッドがあるので、ISE側と上手くタイミングが合うようにハンドシェイクしてあげる必要がある。

for loopで100データ分取得したところ、17256クロックかかっていることがわかった。

従って、5.3MByte/s(230MHz)でデータを受信できる。

```
for(i=0;i<100;i++){
  XGpio_DiscreteWrite(&flagIO,1,0);
  while(1){
    flag = XGpio_DiscreteRead(&flagIO,1);
    if(flag == 1){
      XGpio_DiscreteWrite(&flagIO,1,1);
      data[i] = XGpio_DiscreteRead(&dataIO,1);
      break;
    }
  }
}
```



The screenshot shows a terminal window titled "COM3:115200baud - Tera Term VT". The window contains a list of data values received from the device, ranging from 0 to 10. The text is as follows:

```
COM3:115200baud - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W)
data =0
data =1
data =2
data =3
data =4
data =5
data =6
data =7
data =8
data =9
data =10
1 . 11
```

# PT6のFPGAデザイン

## 新RODのデザイン

新RODでは、CPUコア (MicroBlaze) を搭載することを検討中。CPUコアを搭載することでソフトウェアを用いた動作が可能になる。

### CPUコアを用いた処理

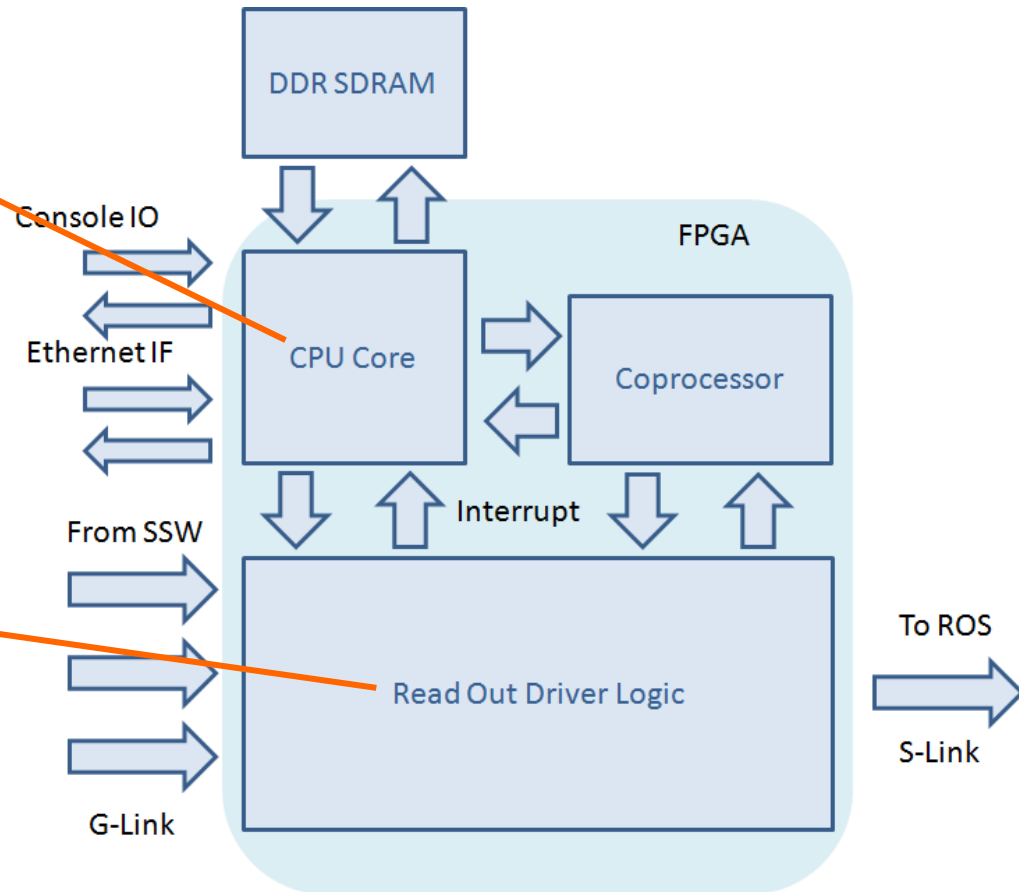
#### ・エラーハンドラー

- データ収集のエラーにより起動
- データフローの停止
- エラー発生時のデータの保管
- エラー内容の解析
- データフローシーケンスの初期化
- データフローの再起動

#### ・システム診断

### Read Out Driver Logic

- ・HDLデザイン
- ・SSWからのデータを集めて、イベントビルディングを行い、データを送る。



# HDLデザイン

## RODの機能

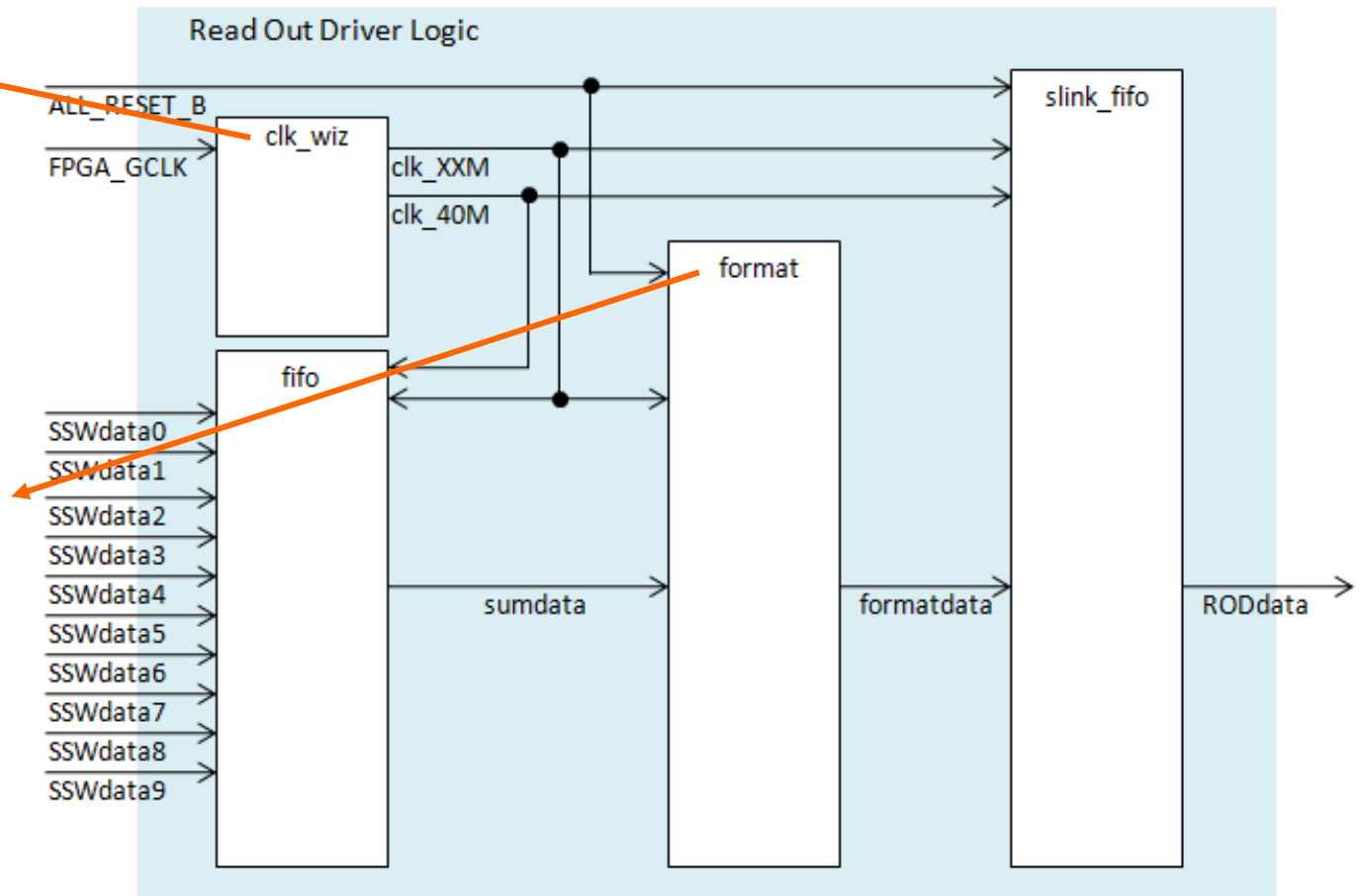
SSWからのデータをATLAS共通のフォーマットに整形してROSにデータを送信する。

### Clock Wizard

Clock Generator。  
外部から供給されるクロックを定数倍して内部の処理を高速化する。

### format

ATLASの共通フォーマットに従って、Headerとtrailerをつける





# まとめ

---

- ・MicroBlazeはXilinx社製FPGAに搭載可能なソフトプロセッサコア。
  - 専用のツールを用いることで簡単に生成、使用でき、これによりソフトウェアを用いた処理が可能となる。
  - FPGAの容量が許す限り、いくつでも生成可能。これにより処理速度を上げることができる。
- ・PT6の具体的なソフトウェアプログラムはこれから記述。

- ・共通のWikiなどがあると便利
  - ツールの使い方
  - Verilog、VHDLの記述例

Back-up



# パイプライン

・3段(フェッチ、デコード、実行)

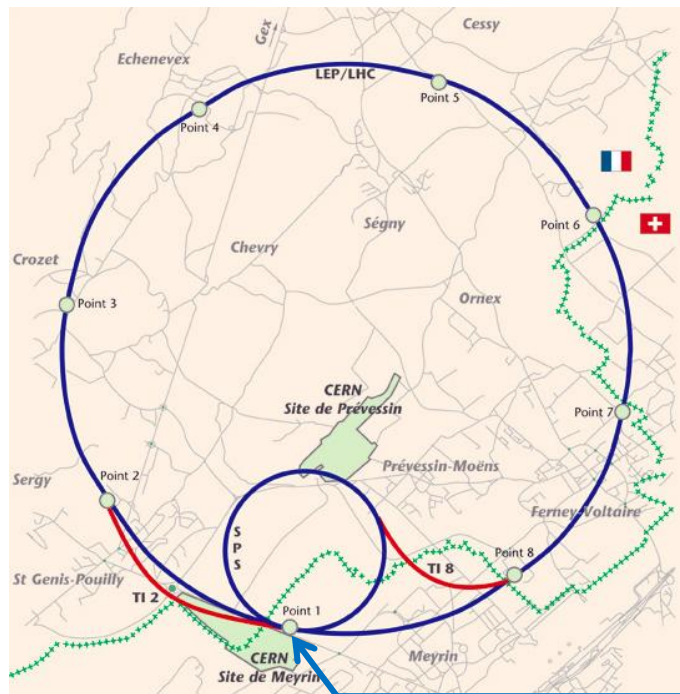
1つの命令を完了するのに1クロック。実行に数クロックかかるものもある。

	cycle 1	cycle 2	cycle 3	cycle4	cycle5	cycle6	cycle7
instruction 1	Fetch	Decode	Execute				
instruction 2		Fetch	Decode	Execute	Execute	Execute	
instruction 3			Fetch	Decode	Stall	Stall	Execute

・5段(フェッチ、デコード、実行、メモリアクセス、ライトバック)

	cycle 1	cycle 2	cycle 3	cycle4	cycle5	cycle6	cycle7	cycle8	cycle9
instruction 1	IF	OF	EX	MEM	WB				
instruction 2		IF	OF	EX	MEM	MEM	MEM	WB	
instruction 3			IF	OF	EX	Stall	Stall	MEM	WB

# LHC加速器とATLAS検出器



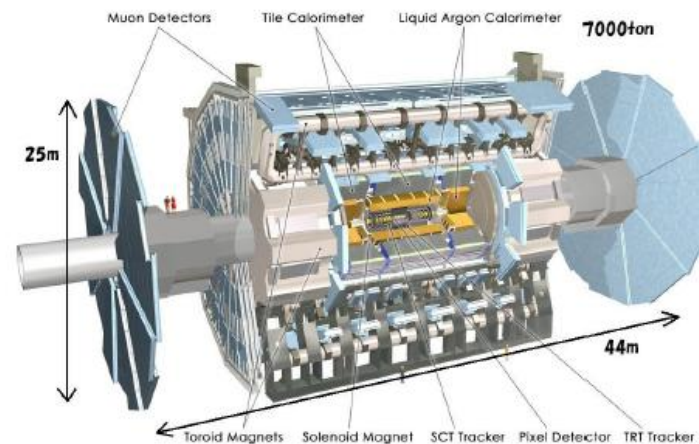
## ATLAS検出器

Higgs粒子や、標準模型を超える物理現象の探索を行う汎用検出器

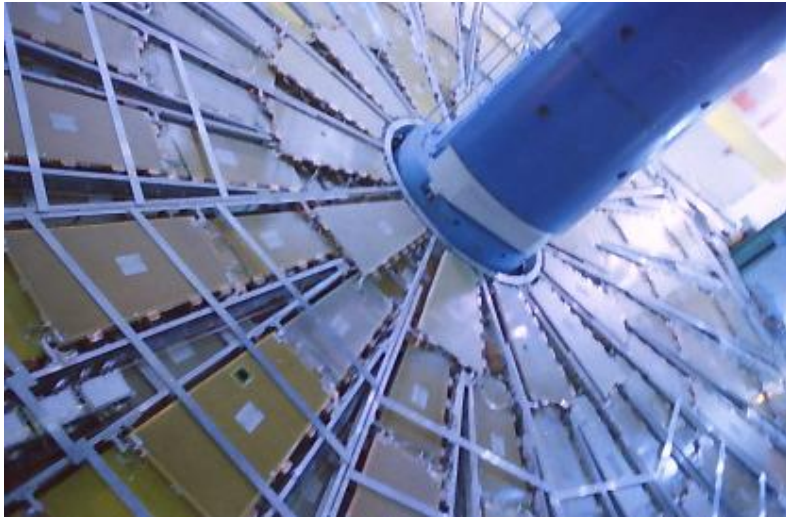
- Tracker
  - MDT
  - CSC
  - TGC
  - RPC
- Calorimeter
- Muon Spectrometer

## LHC加速器

- 主リング長 26.66km
- ビームエネルギー 7TeV
- デザインミノシティ  $10^{34}\text{cm}^{-2}\text{s}^{-1}$
- 衝突頻度 40.08MHz
- バンチ数 2808個



# TGC(Thin Gap Chamber)

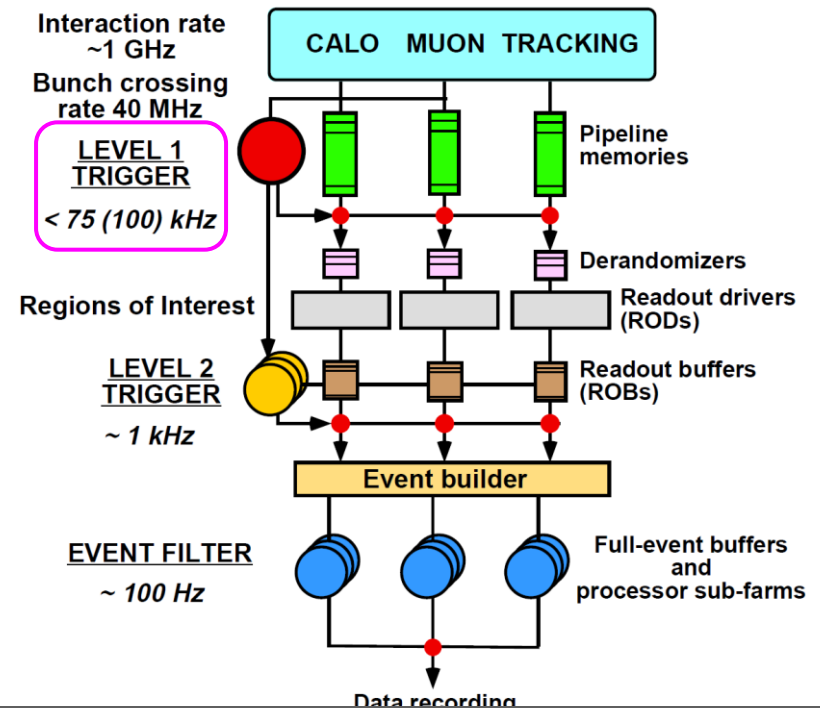


## TGC(Thin Gap Chamber)

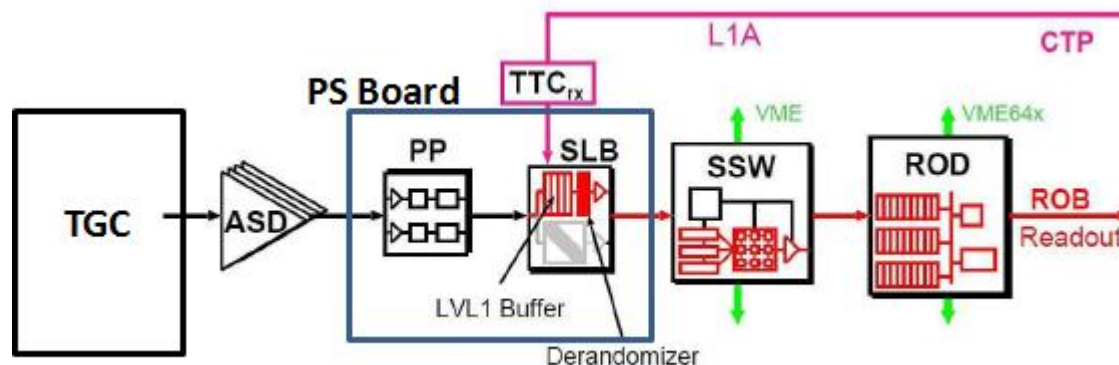
前後方部のトリガーを担う、MWPCの一種。wire間隔よりもwire-strip間の方が狭くなっている。25ns以下の時間分解能。wireがR方向、stripがφ方向の情報を出力。

## Level 1 トリガー

TGC、RPC、カロリメータの情報を基に2.5 $\mu$ s以内での高速なトリガー判定。L1A信号を発行。トリガーレート75kHz。



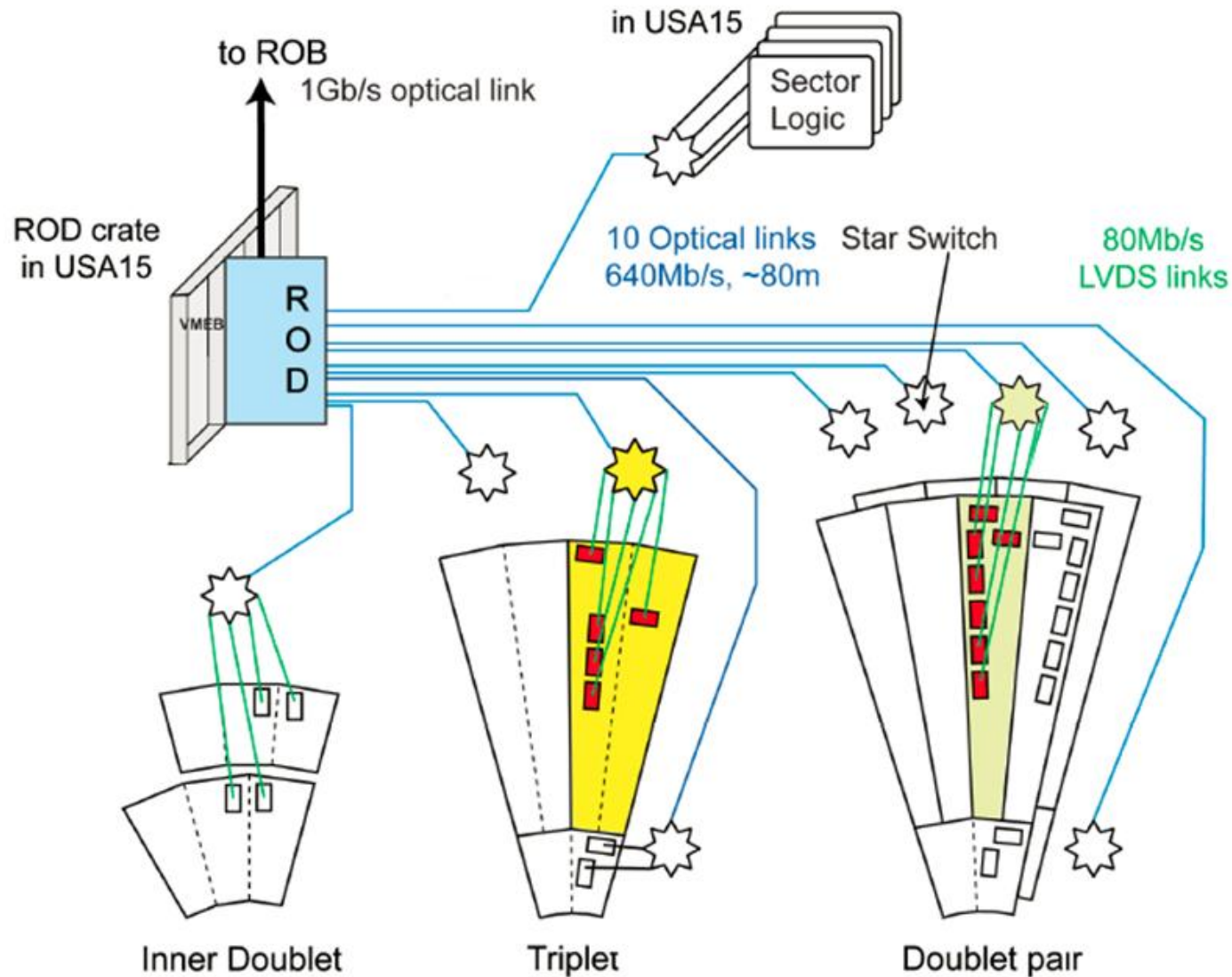
# TGCのリードアウトライン



1. TGCからの信号をASDで増幅、整形、デジタル処理。
2. PSボード上のPP ASICにて遅延を調整。レベル1バッファーに保存。
3. L1Aを受け取った前後3バンチ分のデータがSSWへと送られる。
4. SSWでデータを圧縮。
5. RODでデータの整形を行い、ROSへと送信。

	SSW0	SSW1	SSW2	SSW3	SSW4	SSW5	SSW6	SSW7	SSW8	SSW9
SLB数	18個	18個	10個	15個	15個	15個	15個	10個	11個 又は 12個	6個
担当領域	Endcap $\phi$ 0/1	Endcap $\phi$ 2/3	Forward	Endcap $\phi$ 0	Endcap $\phi$ 1	Endcap $\phi$ 2	Endcap $\phi$ 3	Forward	EI/FI (セクター 2/5/8/11)	SL
	TGC1			TGC2/3					EI/FI	SL

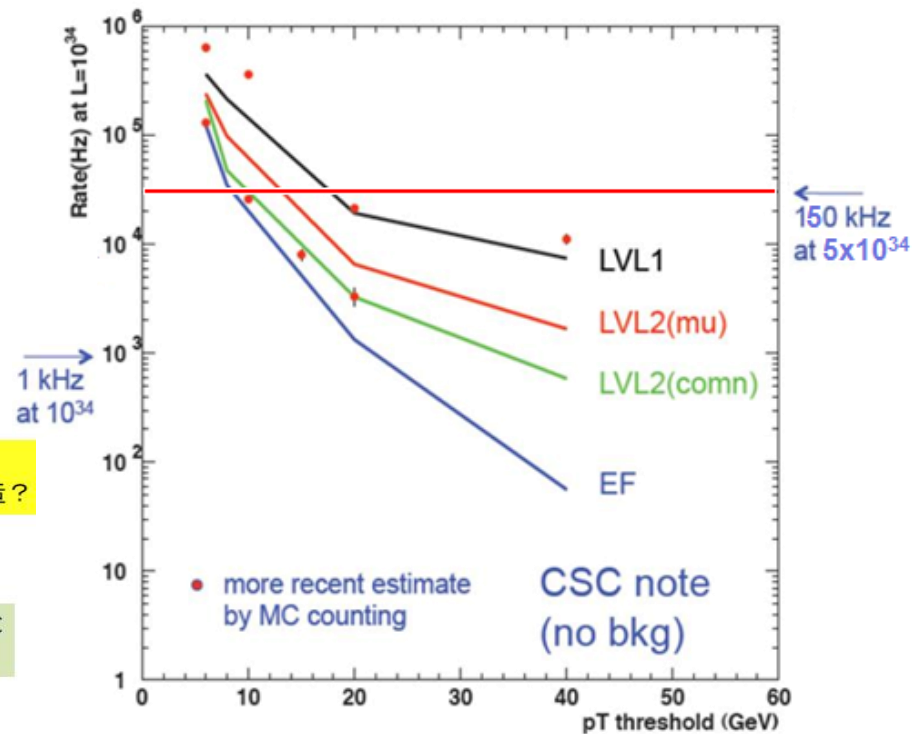
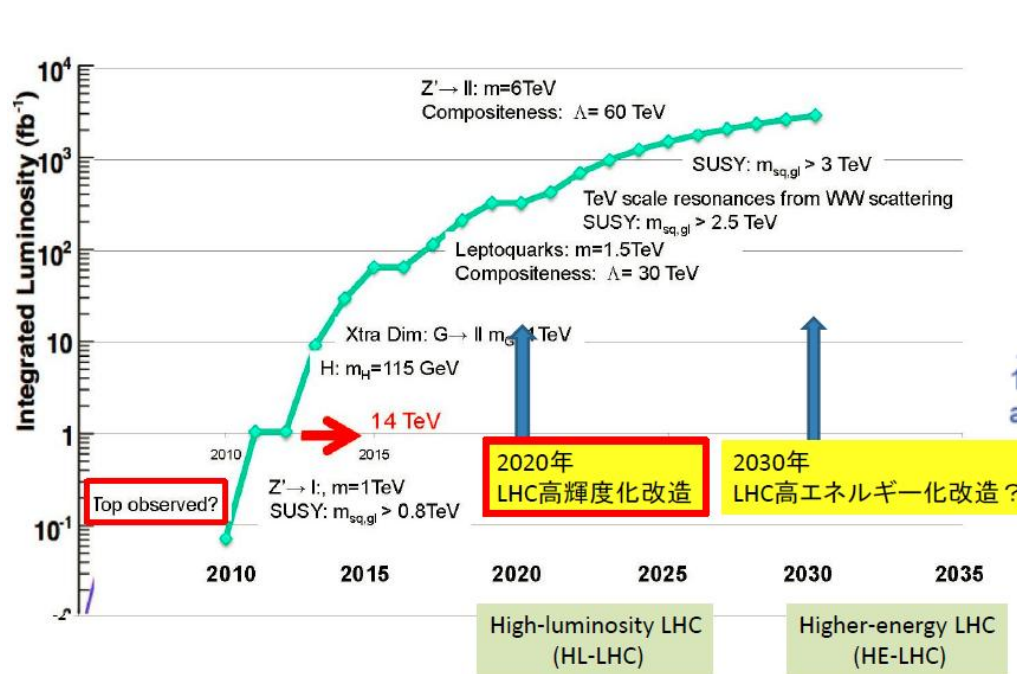
# TGCのリードアウトライン





# LHC加速器のアップグレード計画

LHCのルミノシティが $10^{34}\text{cm}^{-2}\text{s}^{-1}$ から $5 \times 10^{34}\text{cm}^{-2}\text{s}^{-1}$ に。  
 →Higgs粒子の精密測定、質量の大きい新粒子の発見  
 レベル1トリガーレートは、現状の75kHzから150kHzへ。  
 →処理速度の向上が必要



# RODのアップグレード

## データ量の考察

HeaderとTrailerだけで、1152Byte存在。1セクターで10トラックあると、ヒットデータは、80Byte。計1232Byte。アップグレード後単純にトラックの数が5倍になるとすると、ヒットデータは400Byte。計1552Byte。

➡ アップグレード後のデータサイズは、1.25倍

## 処理時間の考察

待ち行列理論で用いられる平均利用率 $\rho$ を用いて、処理時間を決める。

$\lambda$  = 平均到着率

(単位時間当たりの平均到着データサイズ)

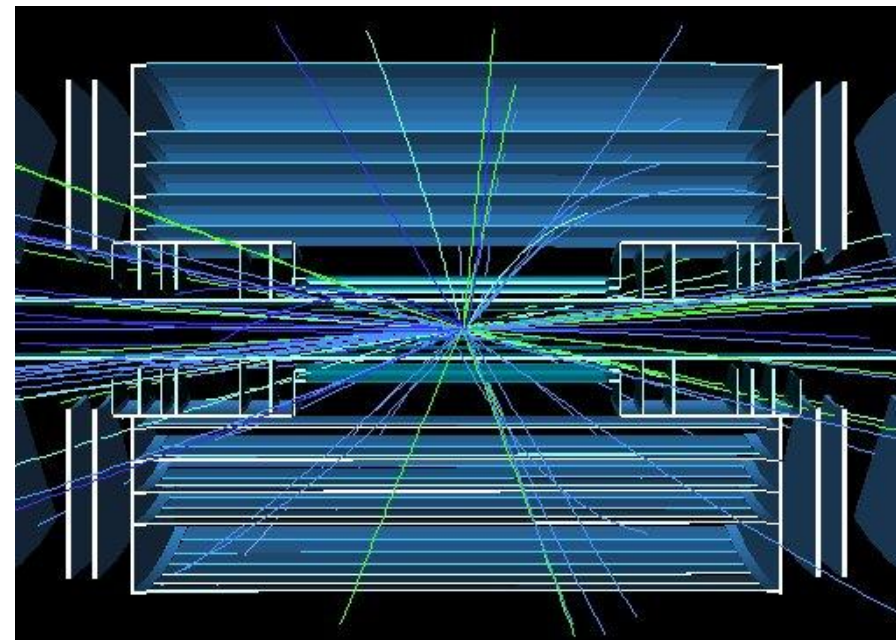
$\mu$  = 平均サービス率

(単位時間当たりの平均処理データサイズ)

を用いて、 $\rho$ は、

$$\rho = \lambda / \mu$$

と表すことができる。 $\rho = 0.5$ 以下とするためには、1イベント分のデータを $3.88\mu\text{s}$ 以内で処理しなければならない。



ルミノシティ  $0.2 \times 10^{34}$

# イスラエルROD

## TGCの読み出しシステム

現状のRODはイスラエルグループが担当。  
搭載チップが古く、高レートに対応できない  
可能性が示唆されている。



## 新ROD開発計画

現状の問題点を考慮し、高速で、より柔軟  
な処理を可能とするデザイン。

