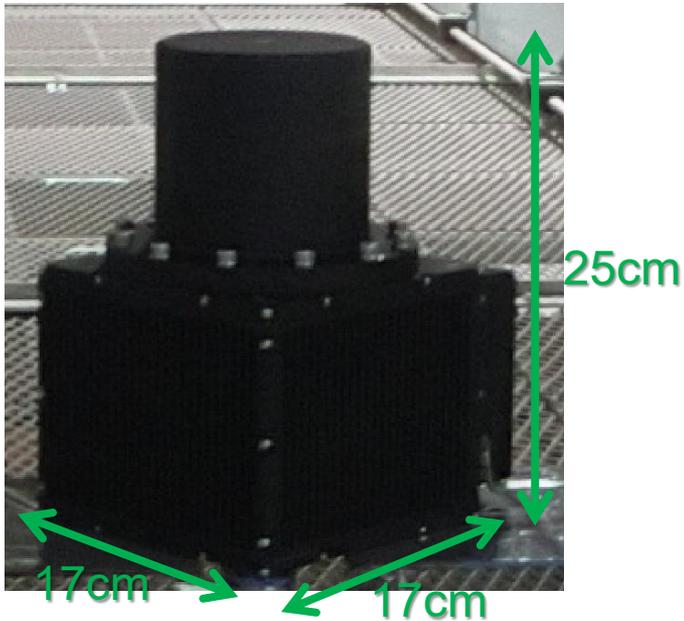


PS-TEPCの月周回有人拠点 搭載に向けたDAQ開発

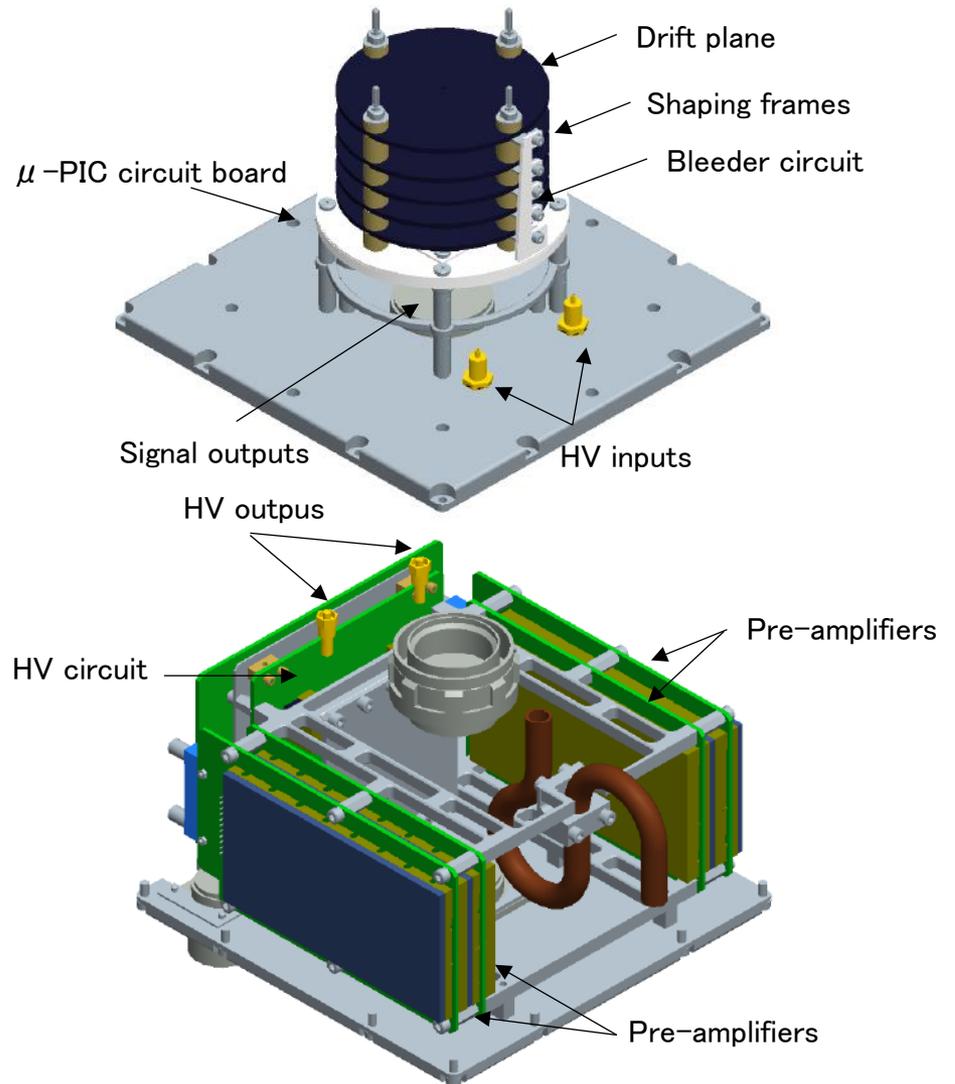
高エネルギー加速器研究機構

高橋一智

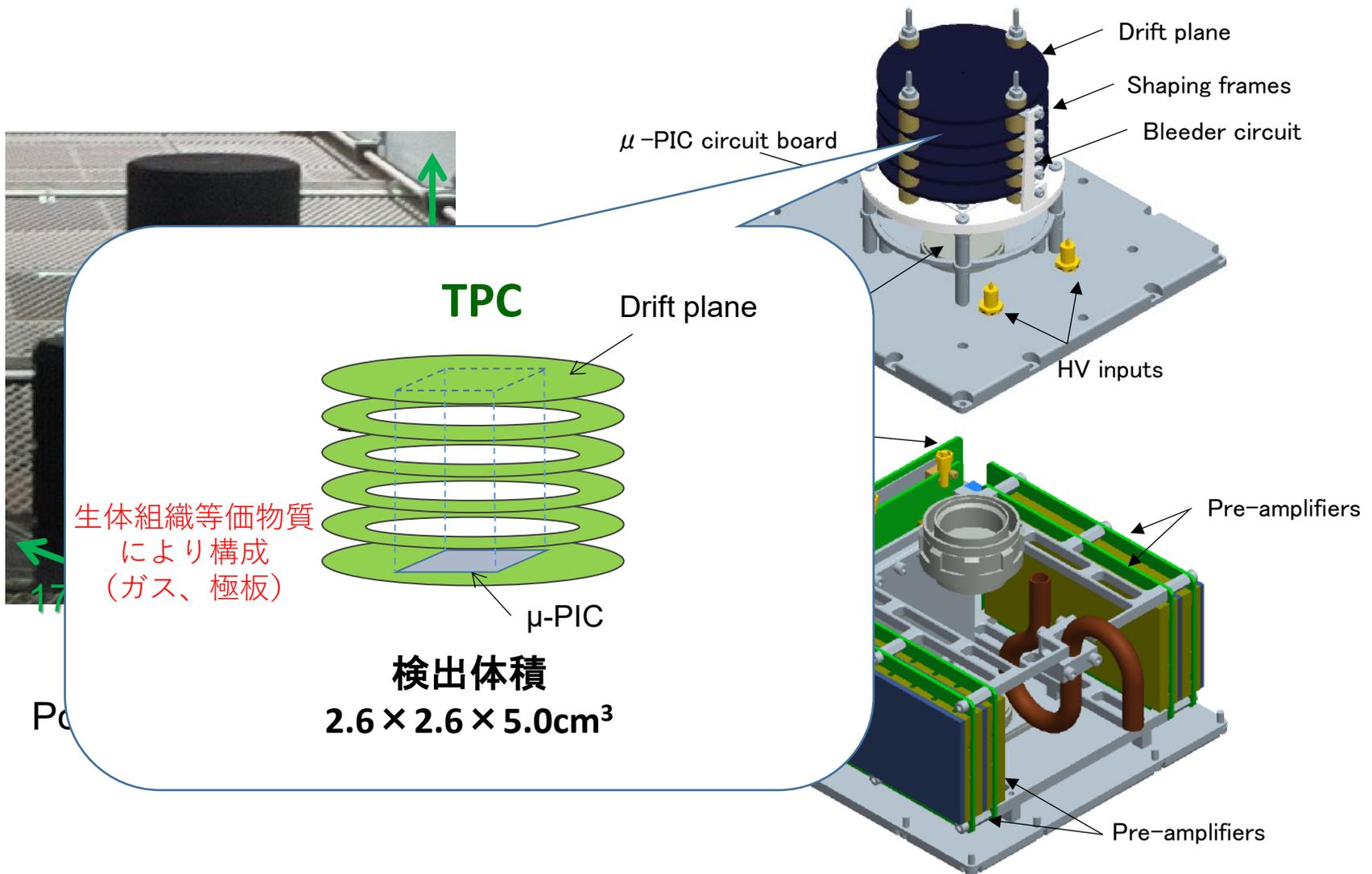
生体組織等価比例計数箱 (PS-TEPC)



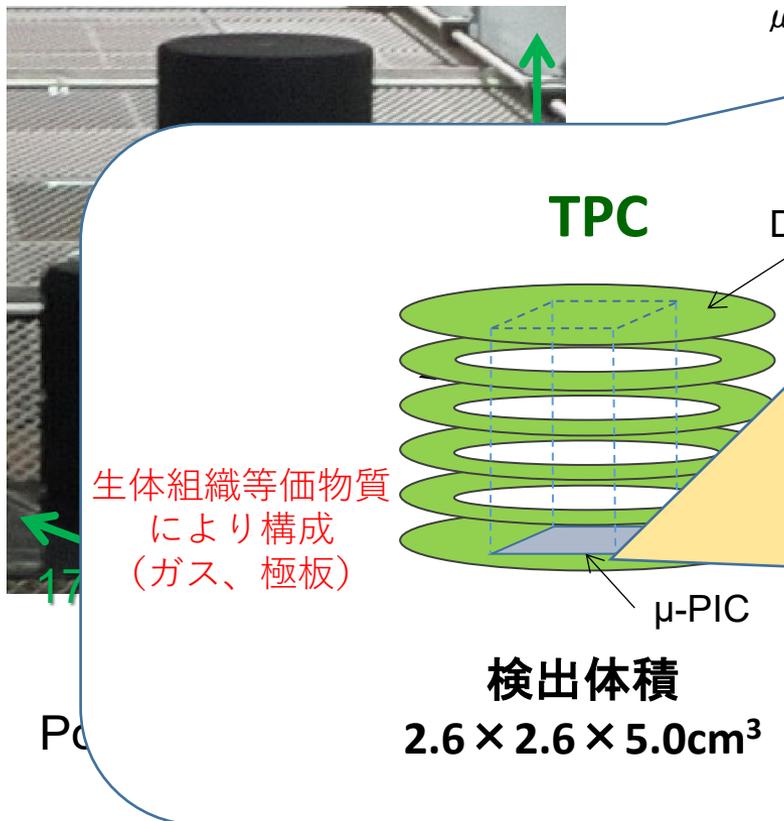
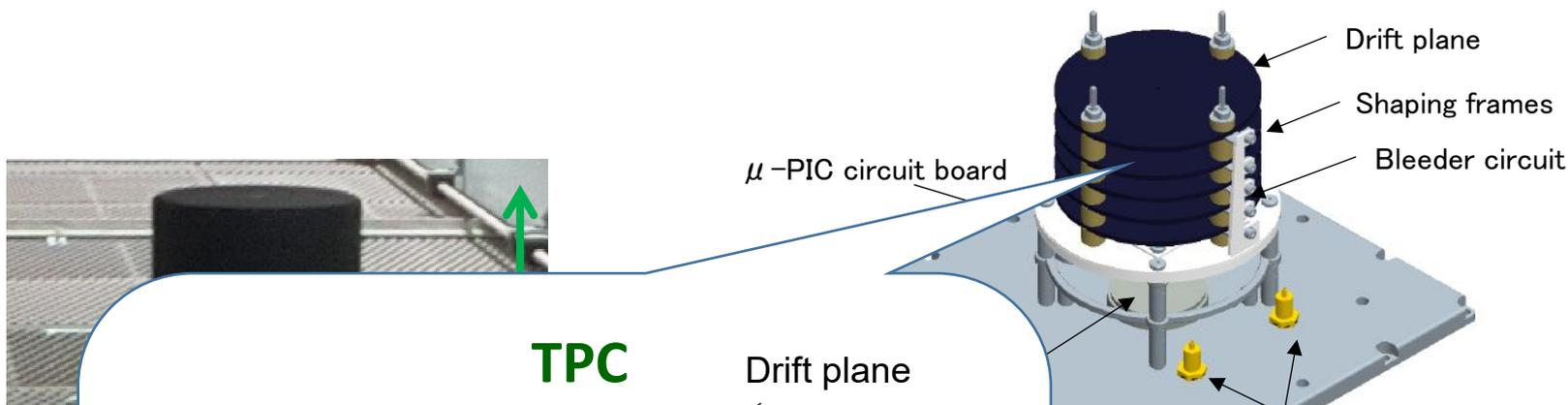
Power consumption : 8.99W



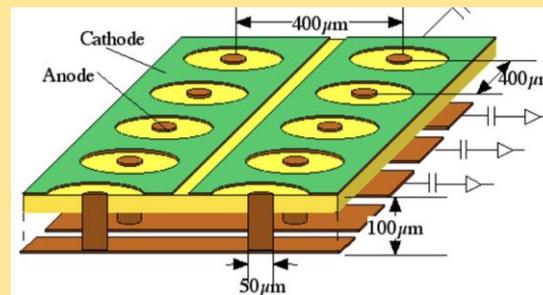
生体組織等価比例計数箱 (PS-TEPC)



生体組織等価比例計数箱 (PS-TEPC)



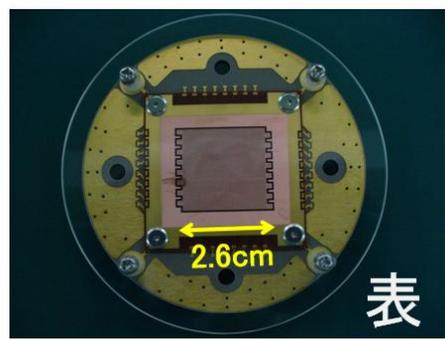
μ -PIC (Micro Pixel Chamber)



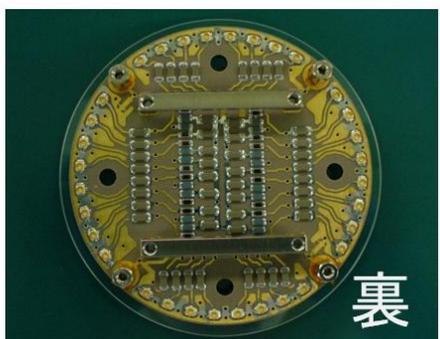
- 電子雲の電荷量と2次元位置が取得可能な電荷収集デバイス
- 有効面積 $2.6 \times 2.6 \text{ cm}^2$
- Cathode, Anode 各64strip

生体組織等価比例計数箱 (PS-TEPC)

μ-PIC基盤



表



裏

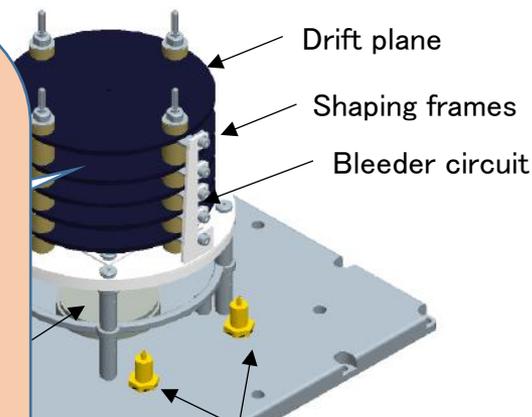
- 読みだしは4strip毎にまとめ、Cathode16ch+Anode 16ch

(ガス、極板)

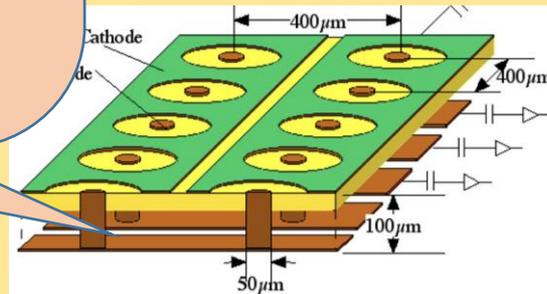
μ-PIC

検出体積

$$2.6 \times 2.6 \times 5.0 \text{ cm}^3$$



Micro Pixel Chamber)

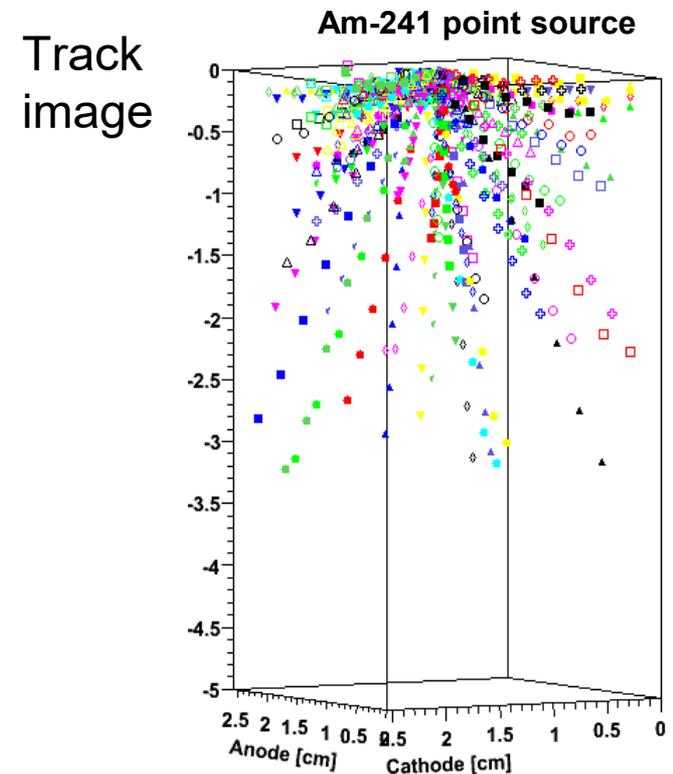


- 電子雲の電荷量と2次元位置が取得可能な電荷収集デバイス
- 有効面積 $2.6 \times 2.6 \text{ cm}^2$
- Cathode, Anode 各64strip

荷電粒子線LET計測

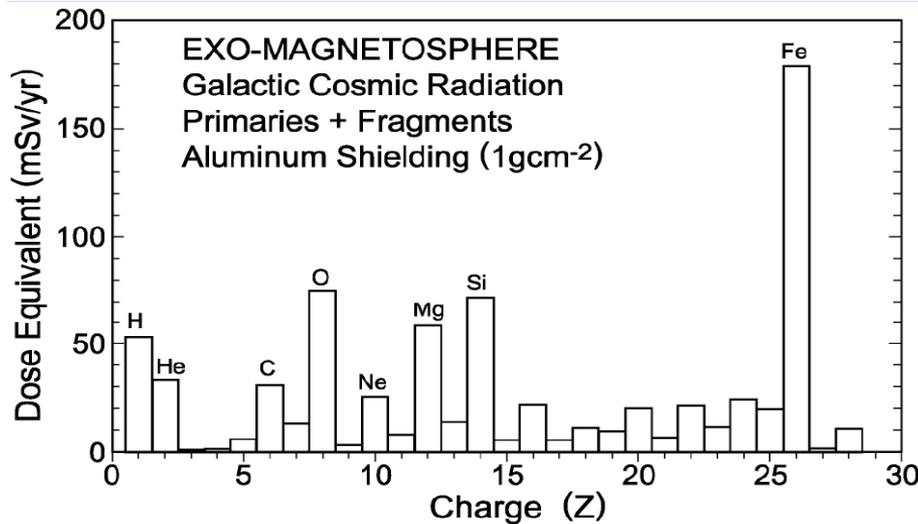
Alpha 5.5MeV point source

- 生体等価検出器で照射線量とLETを同時計測できる
- 荷電粒子線の放射線被ばくを高確度で測定可能
- 宇宙空間での被ばく線量の測定ができる

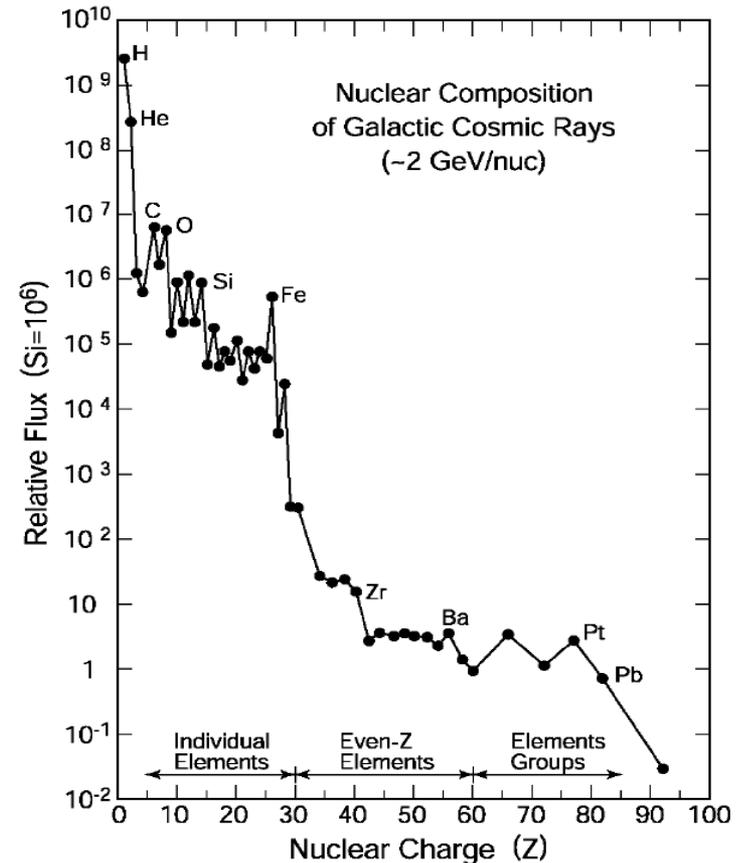


宇宙放射線環境

- 宇宙環境の放射線被ばく
 - 太陽からの陽子線
 - 銀河宇宙線
 - 2次中性子線



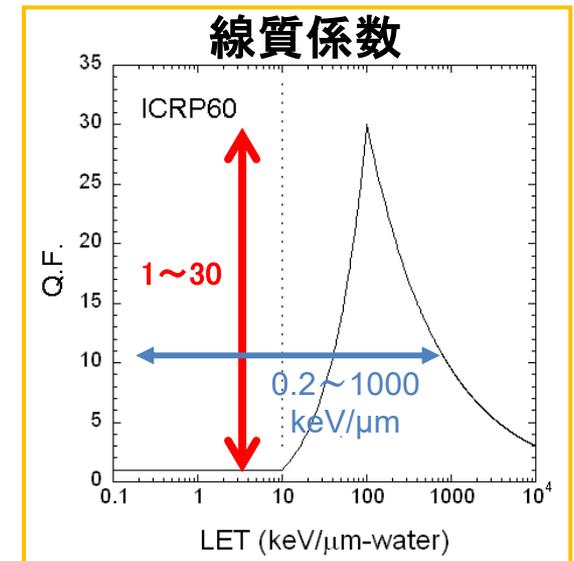
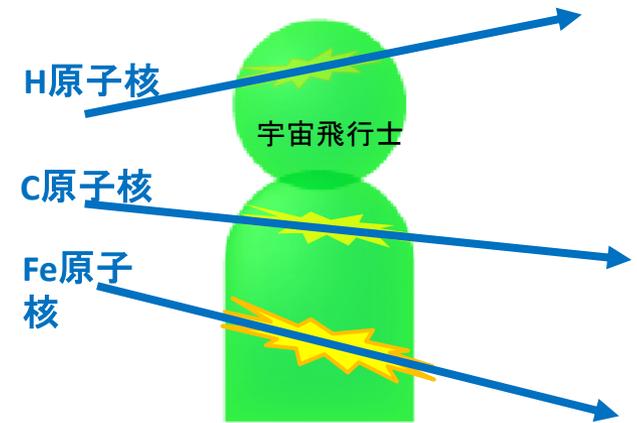
Adopted from Letaw, J.,R., Silberberg, R., Tsao, C.H.,
Adv. Sp. Res. 9, No.10, 285, 1989.



Adopted from Mewaldt, R.A., in *Interplanetary Particle Environment* JPL Publication 88-28, p.121, 1988.

宇宙放射線環境下での被ばく量

- 宇宙線の被ばく量は荷電粒子線のLETで決まる
- 宇宙線のLETは0.2～1000keV/μmの広範囲にあり、線質計数は1～30までの連続関数
- この線量計測におけるエラーは30%(1σ)以内であることが望ましい
- 生体等価検出器で吸収線量とLETの測定が可能な位置有感型生体組織等価比例計数箱(PS-TEPC)で測定する



線量当量 = 線質係数Q(LET) × 吸収線量

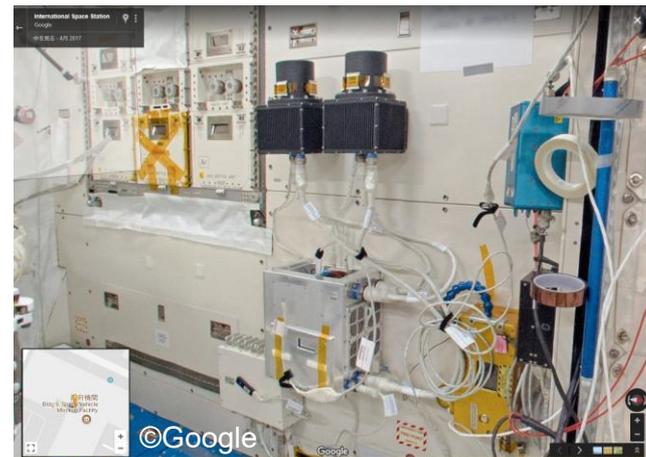
LET(Linear Energy Transfer, 線エネルギー付与)

= Energy deposition / Path length

国際宇宙ステーションでの動作実証試験

日時	イベント
2016.12.9	打ち上げ
2016.12.14	きぼうモジュールへ設置
2016.12.14～12.28	立ち上げ、初期チェック、パラメータ設定
2016.12.28～	定常運用開始

- 2台のディテクターユニット(DU)
- 線壁へ取り付け
- 受動型線量計PADLESの貼り付け
- TSFCでの運用・KEKへのリアルタイムデータ転送

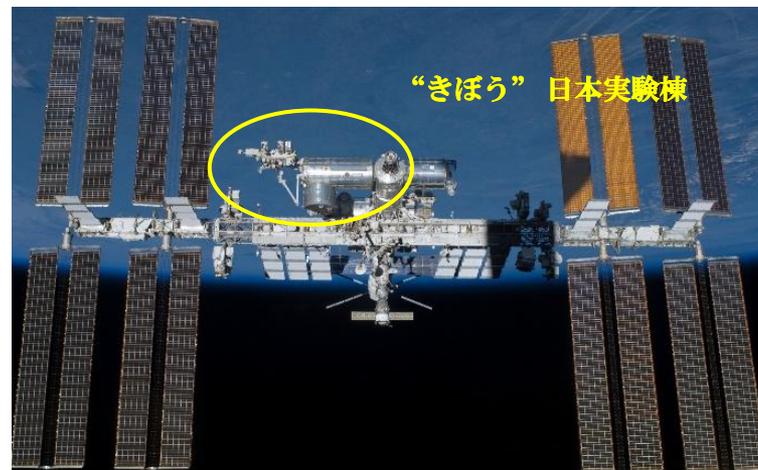


設置されたPS-TEPC



©JAXA

H-IIIBロケットによる打ち上げ
HTV(こうのとり)6号機に積載



国際宇宙ステーション

月周回有人拠点Gateway

2. 月周回有人拠点Gateway



月面及び火星に向けた中継基地として、米国の提案のもと開発中の有人拠点。

年間30日ほど宇宙飛行士が滞在し、それ以外の期間は無人運用される。

Gatewayの軌道



平均38万km

近月点
4000km

北極

南極

月長楕円極軌道

Near Rectilinear Halo Orbit (NRHO)

- 燃料消費が少なく済む月長楕円極軌道に設置予定。
- 軌道面は常に地球を向くため、地球との通信は常時確保される。
- 地球からの到達エネルギーは月低軌道までの70%程度と輸送コストが比較的小さい。
- 月の南極の可視時間が長く、南極探査の通信中継としても都合がよい。

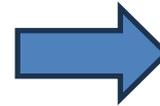
遠月点
75000km

©JAXA

Gateway搭載に向けて

目的

- 月周回軌道での線量計測
- 月面での線量計測



- 輸送コストが大きい
- 利用可能なリソースが小さい
- ポータブル化

ISSとGatewayの比較

	国際宇宙ステーション (ISS)	Gateway
大きさ	約108.5m×72.8m (サッカー場)	下図参照
質量	約420トン	約70トン
組立フライト回数	43回	7回
宇宙飛行士滞在日数 (年間)	365日 (常時)	10~30日
滞在宇宙飛行士人数	6人	4人
食料、消耗品 (年間)	2,190人日分	40~120人日分

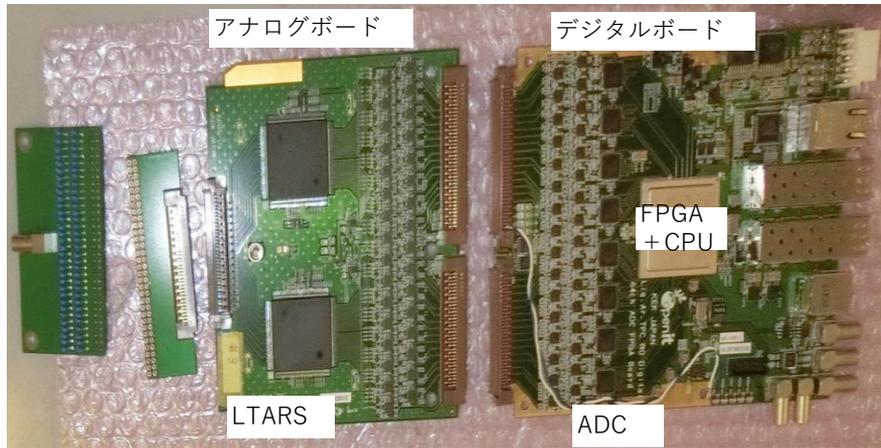


検出部のシステム
全体の小型・
軽量・省電力化
が課題

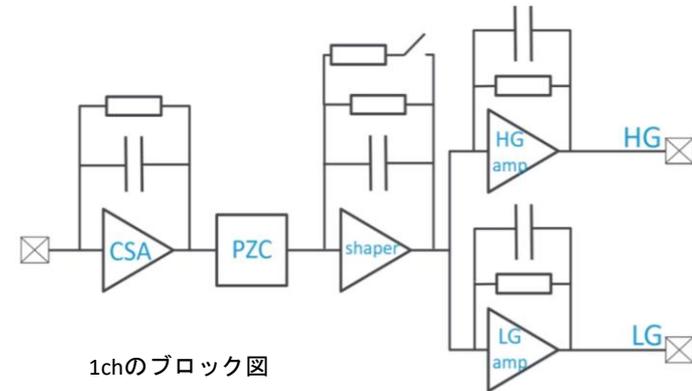


LTARS2018K06A

KEK Open-Itで開発された多チャンネル読み出しのASICの利用を決定

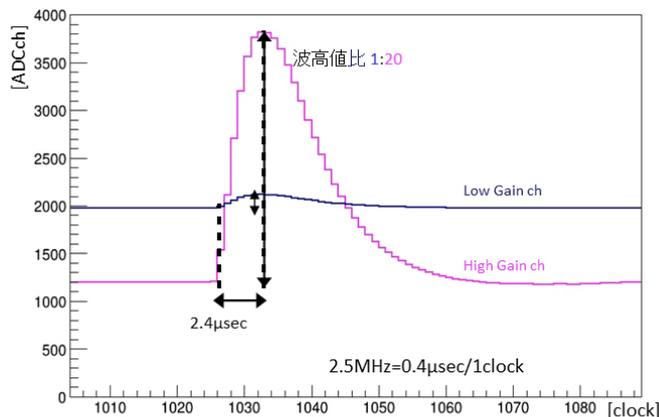


LTARS2018K06A評価基板



1chのブロック図

出典：中澤美紀, 修士論文, 2018年2月, 神戸大学.



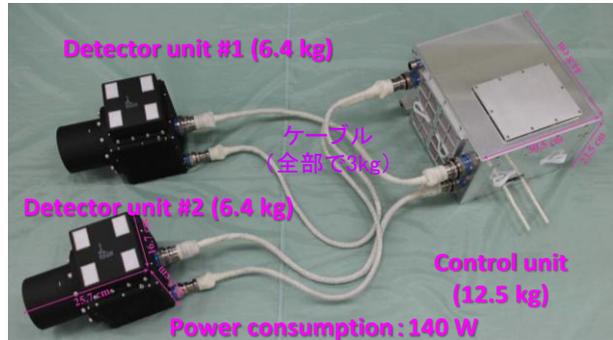
出力波形

- 液体アルゴン検出器の読み出し用に開発
- 暗黒物質探索実験の読み出し用にも使用
- 1入力に対して2ゲインの出力
 - 大幅な小型・軽量化
- μ -PICで動作実績が有る
 - 開発期間の短縮・堅実な実装

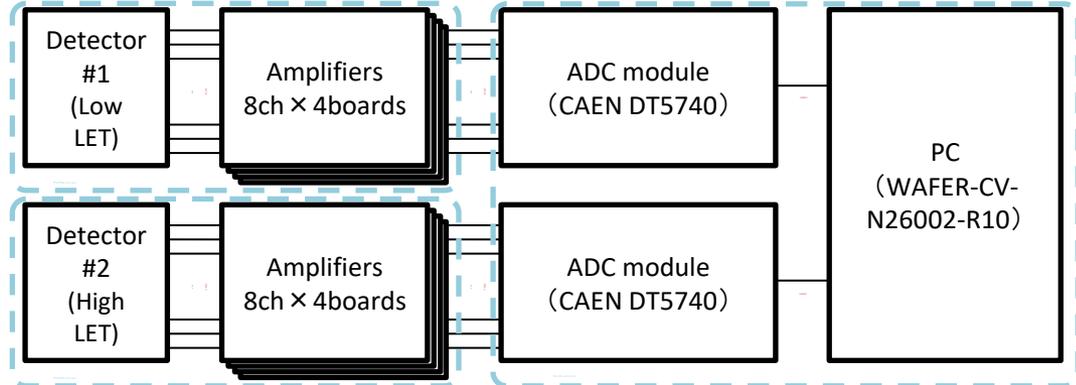
コンパクトモデルPS-TEPC

We are improving the ISS 2016 model of the PS-TEPC to make it more compact, lighter, and more power-saving by utilizing ASIC for the readout of the detector signals.

ISS model

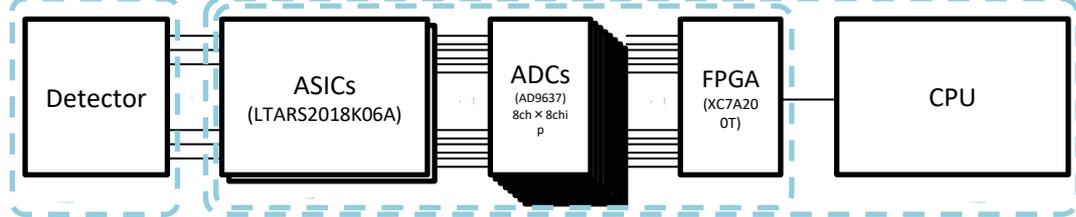


Overall picture of the ISS model



Block diagram of the ISS model

Compact model試作機

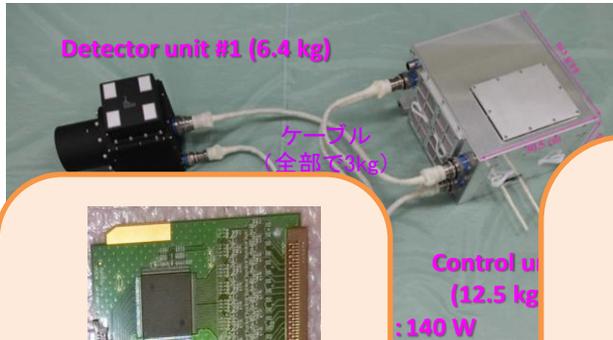


Block diagram of the compact model

コンパクトモデルPS-TEPC

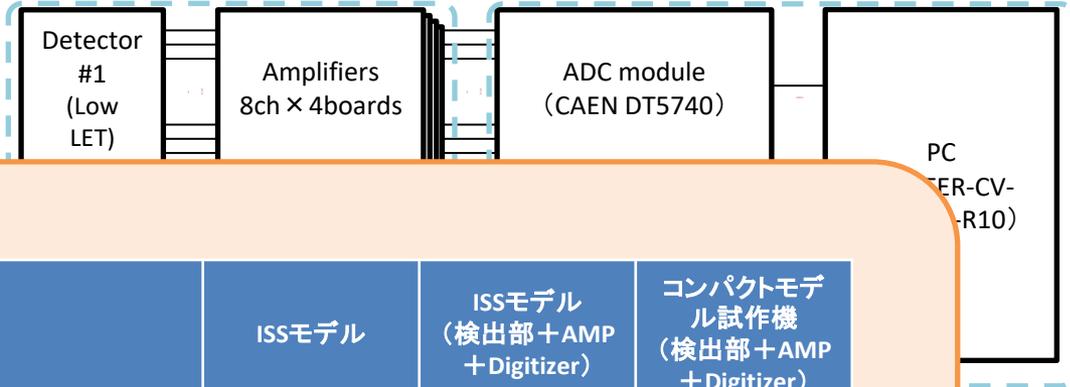
We are improving the ISS 2016 model of the PS-TEPC to make it more compact, lighter, and more power-saving by utilizing ASIC for the readout of the detector signals.

ISS model



ISS model

コンパクトモデル試作機



	ISSモデル	ISSモデル (検出部+AMP +Digitizer)	コンパクトモデル 試作機 (検出部+AMP +Digitizer)
消費電力	140W (DC/DC Conv.の 変換効率を考慮 -> 110W)	83W (9W+25W)*2 +15W	8.5W

消費電力の比較



Block diagram of the compact model

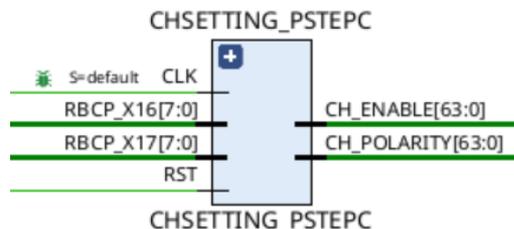
コンパクトモデルに向けた FPGAロジック開発

- Anode/Cathodeの両極シグナル対応
- 各ChのトリガEnable/Disable設定
- セルフトリガ発行
- リアルタイムの最大波高値・トリガータイミング決定
- 後段CPUへの高頻度データ送信

ADC_SiTCP_V20に追加機能

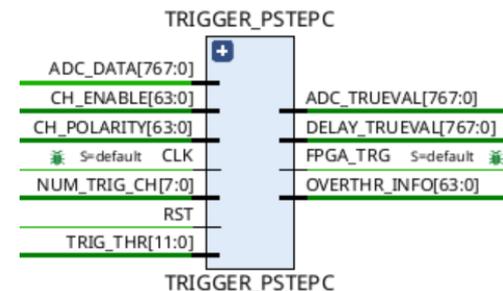
- チャンネル設定モジュール

- Enable
- Polarity



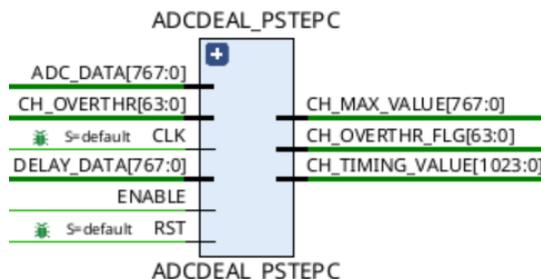
- セルフトリガモジュール

- CICフィルタベースライン抽出

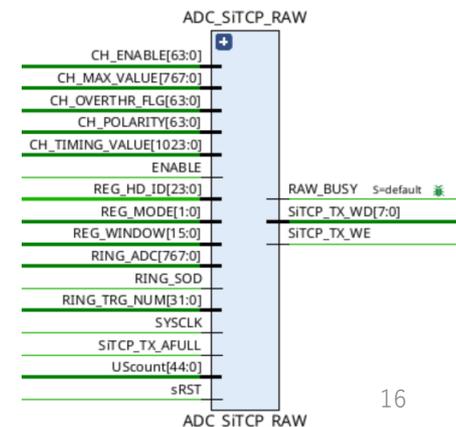


- 二値化モジュール

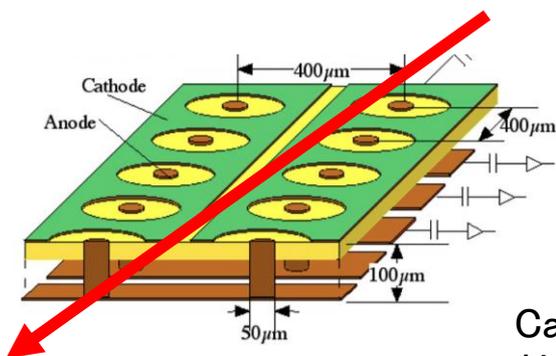
- MaxValue
- Timing



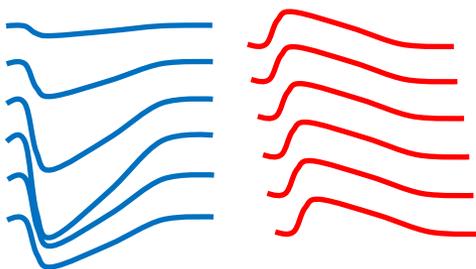
- 計算データ送信



Anode/Cathode・Enable/Disable CH設定

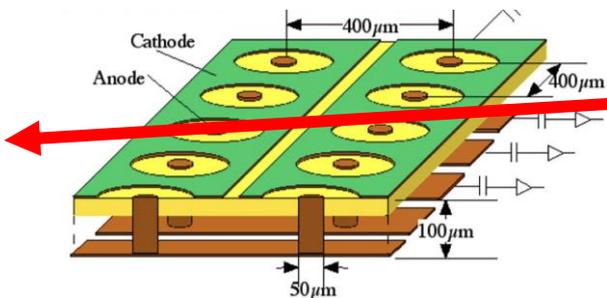


Cathodeストリップに平行方向から粒子が入射した場合のシグナル

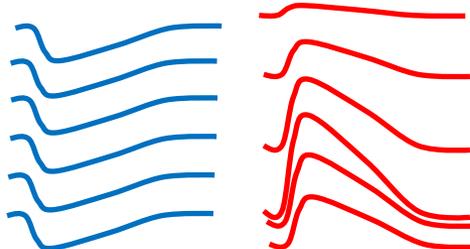


- PS-TEPCでは垂直に交差するμ-PICの全てのストリップがトリガ・時間情報の要素となるため、両極対応のロジックが必要。

➡ Anode/Cathode対応



Anodeストリップに平行方向から粒子が入射した場合のシグナル



- 軌道上で連続放電などの理由でトリガ要素から特定CHを除去する必要がある場合遠隔操作で除去。

➡ Enable/Dsiable設定

Ch設定のためレジスタ外部操作

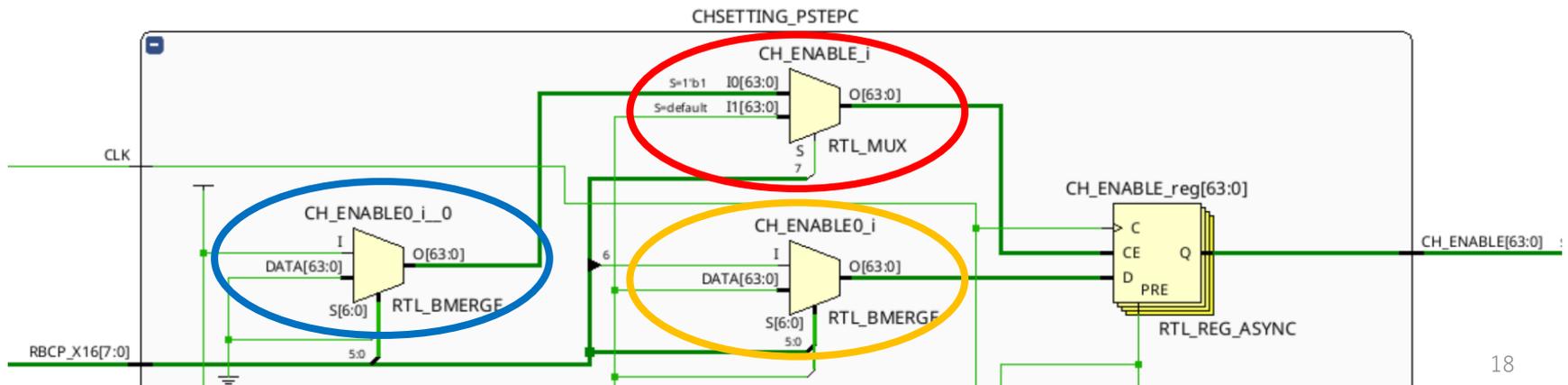
- 各チャンネルごとの設定には、64bit分のレジスタが必要
- SiTCP上のレジスタを外部から操作可能なRBCPで設定を考えたが、64bit×2が重くて断念
- RBCP上のレジスタ値を自作レジスタに読み込む方法を考えた
- 64chに対して各々Enable/DisableとAnode/Cathodeを個別に設定可能で開発要求を満たす

0x00000000

数値を変えるかどうか
Enable:1 Disable:0

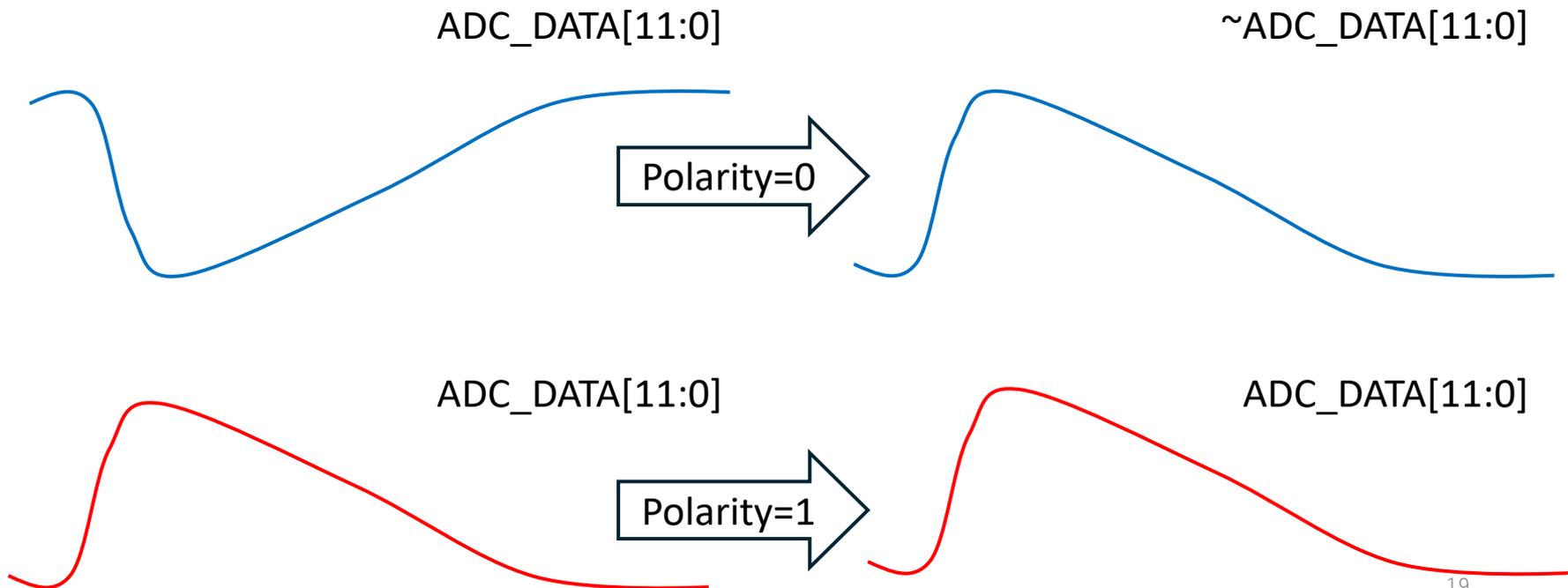
セットする値
0 or 1

値を変更する位置
6bit = 64ch設定可能



Cathodeシグナル反転

- CathodeのADCデータを処理の一番最初で補数を取って反転
- 以降の処理はCathode/Anode共通になる



CIC_Compilerでベースライン抽出

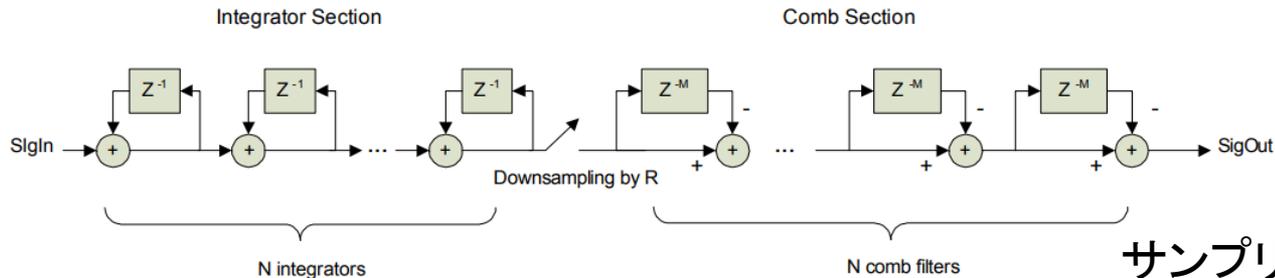
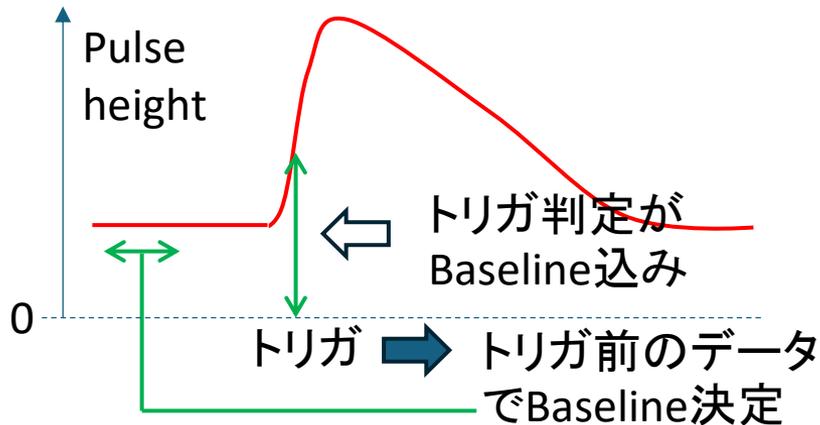


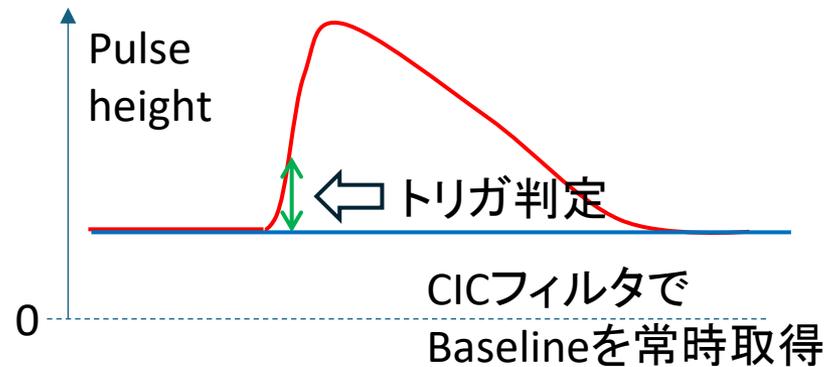
Figure 7: CIC Decimation Filter

サンプリングレート変換用
ローパスフィルタとして使用可能

従来

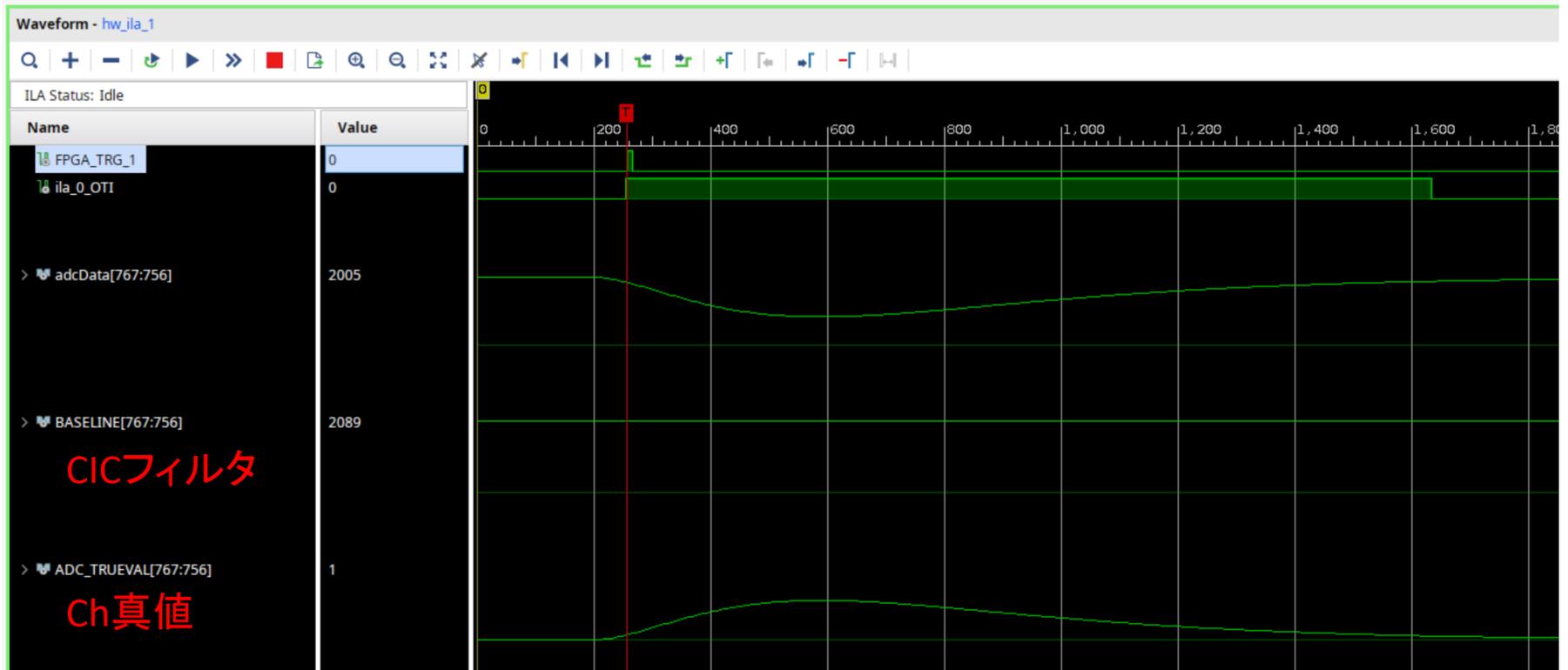


CICフィルタ採用



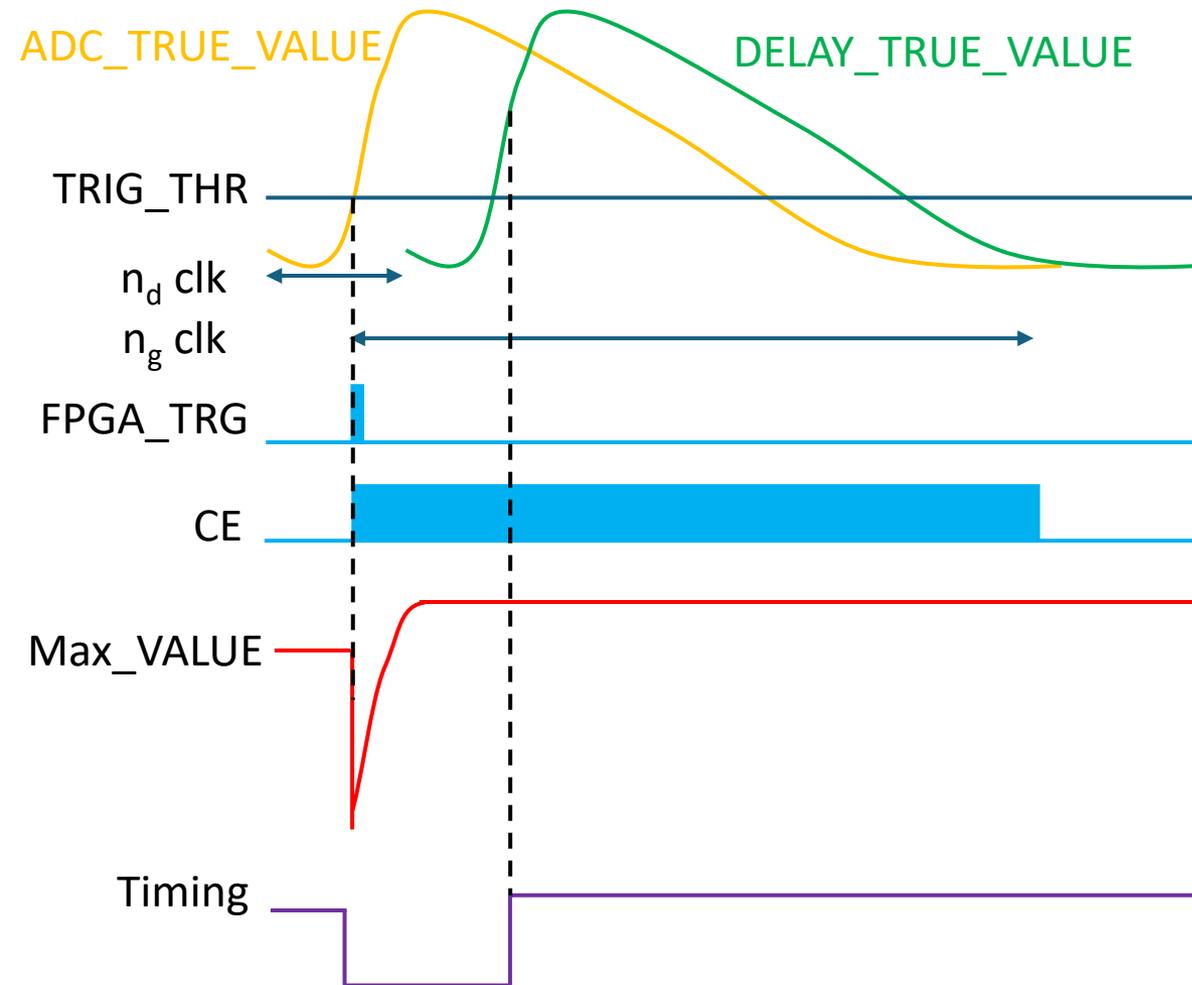
- ベースライン変動に強い
- リアルタイムデータ処理が可能

処理を行った実波形データ



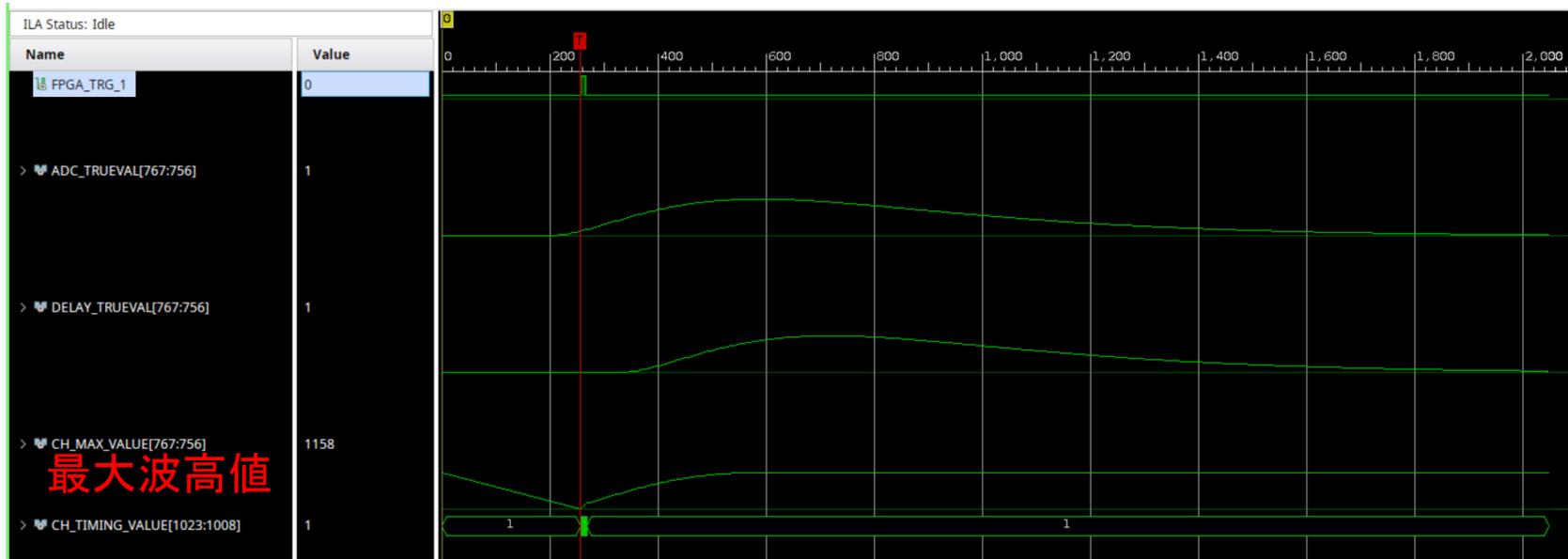
いずれかのchのTRUEVALが閾値を超えたらトリガ発行

最大波高・トリガタイミング決定



- ADC値をトリガから MAX_VALUEと比較、 n_g CLKの間数値を更新
- FIFOで n_d CLK分の遅延データ作成、MAX_VALUEの50%を遅延データが超えたクロック数をTimingとして決定
- 次のトリガまで MAX_VALUEとTimingを保持、SiTCPの波形データの最後に計算値を付与

処理中の実信号



トリガタイミング

最大波高値・トリガタイミングはレジスタに保持される

ADC_SiTCP_RAW改造

- SiTCP_TX_WDに合わせて、データを8bitのレジスタに格納
- ソースファイルから送信Window数を制御している部分を見つける → len[22:0]のうちlen[21:7]が該当
- case(~len[21:7])で分岐
 - default: ADCデータ
 - 16'd1: 最大波高値
 - 16'd0: トリガタイミング

```
reg [15:0] rdLen; // 20250919 ktaka add

always@ (posedge SYSCLK) begin
    rdHa[3:0]    <= ha[3:0];
    hdVal       <= ~ha[4] & hdEnb;

    rdRa[4:0]   <= ra[4:0];
    rdRa[6:0]   <= ra[6:0];
    rdVal[0]    <= ~len[21] & rdEnb;
    rdVal       <= ~len[22] & rdEnb;

    rdLen <= {1'b0, ~len[21:7]}; // 20250919 ktaka add
end
```

```
integer dataloop;
always@ (posedge SYSCLK) begin
    for (dataloop = 0; dataloop < 64; dataloop=dataloop+1) begin
        REG_OVERTHR [dataloop] <= CH_OVERTHR_FLG [dataloop];
        REG_MAXVAL_L [dataloop] <= CH_MAX_VALUE [dataloop*12 +: 8];
        REG_MAXVAL_H [dataloop] <= CH_MAX_VALUE [dataloop*12+8 +: 4];
        REG_TIMVAL_L [dataloop] <= CH_TIMING_VALUE [dataloop*16 +: 8];
        REG_TIMVAL_H [dataloop] <= CH_TIMING_VALUE [dataloop*16+8 +: 8];
        REG_ENABLE [dataloop] <= CH_ENABLE [dataloop];
        REG_POLARITY [dataloop] <= CH_POLARITY [dataloop];
    end
end
```

```
case (dlyLen[15:0])

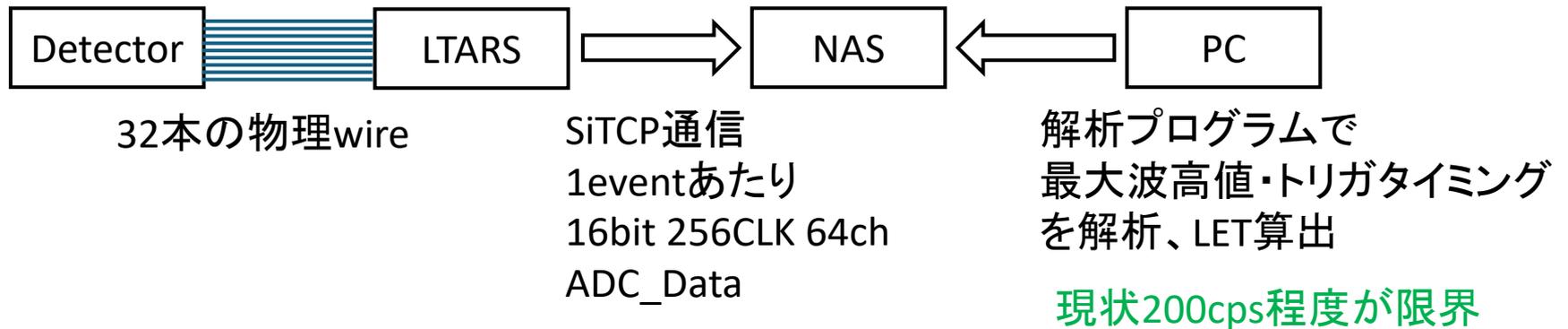
    16'd1: begin : CH_OVERTHR_AND_MAXVALUE_SEND
        case(dlyRdRa[0])

            1'd1: rdAdcData <= {REG_MAXVAL_L[dlyRdRa[6:1]}};
            default: rdAdcData <= {REG_OVERTHR[dlyRdRa[6:1]], 1'b0};
        endcase
    end
    16'd0: begin : TIMING_SEND
        case(dlyRdRa[0])
            1'd1: rdAdcData <= {REG_TIMVAL_L[dlyRdRa[6:1]}};
            default: rdAdcData <= REG_TIMVAL_H[dlyRdRa[6:1]};
        endcase
    end

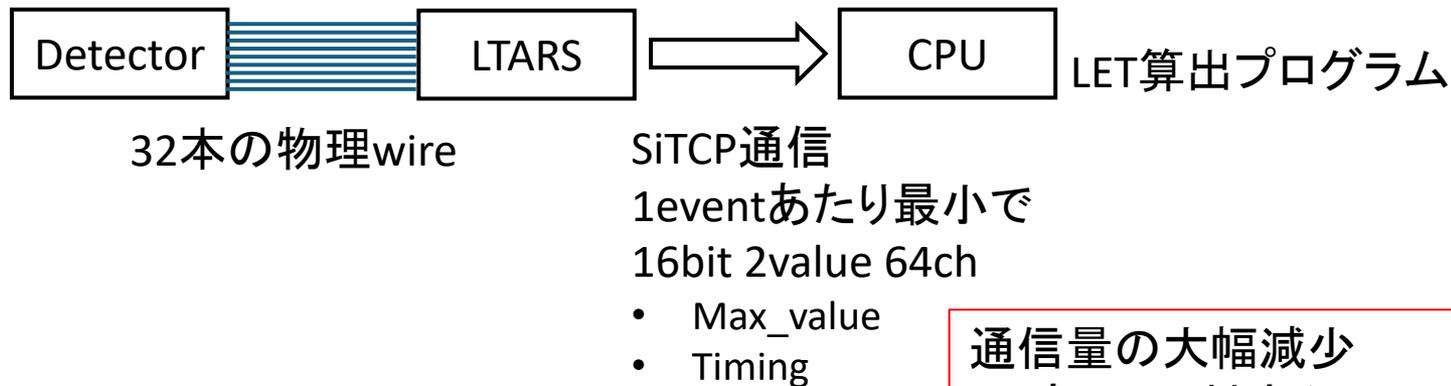
    default: begin : ADCDATASEND
```

送信されるデータ

従来



開発後



通信量の大幅減少

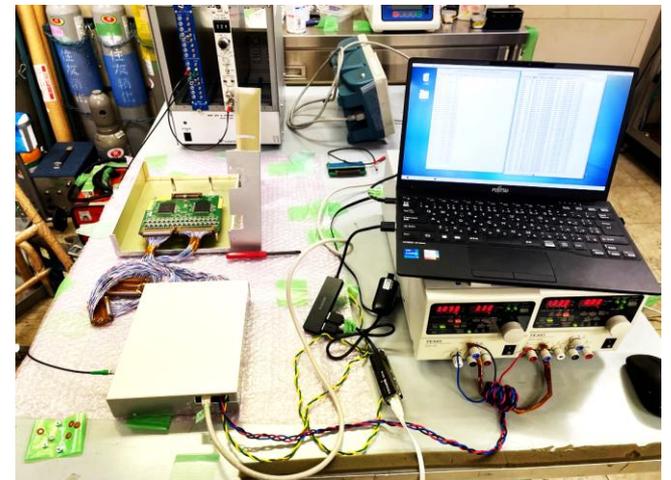
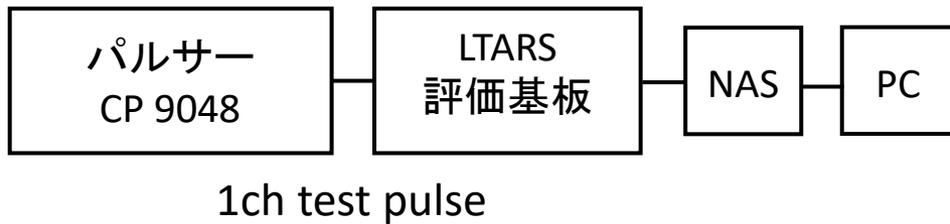
- 高レート対応 (~1000cps)

CPUの負荷軽減

- 省電力、実績あるCPUの利用

機能確認

- パルサーを使って1ch分のテストパルスを入力、動作を確認した



1event分抜き出し

- 1ch分のテストパルス
- Ch58にのみCh-Enableを設定してるため、anode側のノイズを無視してシグナルが取れている
- Timingはちゃんとしたパルスを入れないと正確な動作をしてるか検証が難しい
- MaxValueはちゃんと取れてそう

CH 01: MaxValue=520	OverTHR=1	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 02: MaxValue=28	OverTHR=0	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 03: MaxValue=351	OverTHR=1	Ch-Enable=0	Ch-Polarity=1	Timing=37
CH 04: MaxValue=19	OverTHR=0	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 05: MaxValue=411	OverTHR=1	Ch-Enable=0	Ch-Polarity=1	Timing=1
CH 06: MaxValue=22	OverTHR=0	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 07: MaxValue=311	OverTHR=1	Ch-Enable=0	Ch-Polarity=1	Timing=67
CH 08: MaxValue=17	OverTHR=0	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 09: MaxValue=363	OverTHR=1	Ch-Enable=0	Ch-Polarity=1	Timing=13
CH 10: MaxValue=20	OverTHR=0	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 11: MaxValue=284	OverTHR=1	Ch-Enable=0	Ch-Polarity=1	Timing=79
CH 12: MaxValue=15	OverTHR=0	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 13: MaxValue=352	OverTHR=1	Ch-Enable=0	Ch-Polarity=1	Timing=13
CH 14: MaxValue=20	OverTHR=0	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 15: MaxValue=270	OverTHR=1	Ch-Enable=0	Ch-Polarity=1	Timing=97
CH 16: MaxValue=14	OverTHR=0	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 17: MaxValue=319	OverTHR=1	Ch-Enable=0	Ch-Polarity=1	Timing=43
CH 18: MaxValue=18	OverTHR=0	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 19: MaxValue=246	OverTHR=1	Ch-Enable=0	Ch-Polarity=1	Timing=85
CH 20: MaxValue=14	OverTHR=0	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 21: MaxValue=295	OverTHR=1	Ch-Enable=0	Ch-Polarity=1	Timing=37
CH 22: MaxValue=16	OverTHR=0	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 23: MaxValue=237	OverTHR=1	Ch-Enable=0	Ch-Polarity=1	Timing=133
CH 24: MaxValue=13	OverTHR=0	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 25: MaxValue=280	OverTHR=1	Ch-Enable=0	Ch-Polarity=1	Timing=67
CH 26: MaxValue=16	OverTHR=0	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 27: MaxValue=226	OverTHR=1	Ch-Enable=0	Ch-Polarity=1	Timing=127
CH 28: MaxValue=13	OverTHR=0	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 29: MaxValue=276	OverTHR=1	Ch-Enable=0	Ch-Polarity=1	Timing=85
CH 30: MaxValue=16	OverTHR=0	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 31: MaxValue=220	OverTHR=1	Ch-Enable=0	Ch-Polarity=1	Timing=169
CH 32: MaxValue=12	OverTHR=0	Ch-Enable=0	Ch-Polarity=1	Timing=0
CH 33: MaxValue=0	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 34: MaxValue=0	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 35: MaxValue=0	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 36: MaxValue=0	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 37: MaxValue=0	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 38: MaxValue=0	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 39: MaxValue=0	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 40: MaxValue=0	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 41: MaxValue=0	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 42: MaxValue=0	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 43: MaxValue=0	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 44: MaxValue=0	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 45: MaxValue=0	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 46: MaxValue=0	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 47: MaxValue=0	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 48: MaxValue=0	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 49: MaxValue=229	OverTHR=1	Ch-Enable=0	Ch-Polarity=0	Timing=95
CH 50: MaxValue=13	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 51: MaxValue=250	OverTHR=1	Ch-Enable=0	Ch-Polarity=0	Timing=95
CH 52: MaxValue=14	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 53: MaxValue=578	OverTHR=1	Ch-Enable=0	Ch-Polarity=0	Timing=1
CH 54: MaxValue=78	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 55: MaxValue=613	OverTHR=1	Ch-Enable=0	Ch-Polarity=0	Timing=1
CH 56: MaxValue=101	OverTHR=0	Ch-Enable=0	Ch-Polarity=0	Timing=0
CH 57: MaxValue=498	OverTHR=1	Ch-Enable=0	Ch-Polarity=0	Timing=1
CH 58: MaxValue=802	OverTHR=1	Ch-Enable=1	Ch-Polarity=0	Timing=1
CH 59: MaxValue=1041	OverTHR=1	Ch-Enable=0	Ch-Polarity=0	Timing=1
CH 60: MaxValue=1481	OverTHR=1	Ch-Enable=0	Ch-Polarity=0	Timing=1
CH 61: MaxValue=2649	OverTHR=1	Ch-Enable=0	Ch-Polarity=0	Timing=103
CH 62: MaxValue=1702	OverTHR=1	Ch-Enable=0	Ch-Polarity=0	Timing=251
CH 63: MaxValue=718	OverTHR=1	Ch-Enable=0	Ch-Polarity=0	Timing=1
CH 64: MaxValue=1463	OverTHR=1	Ch-Enable=0	Ch-Polarity=0	Timing=1

まとめ

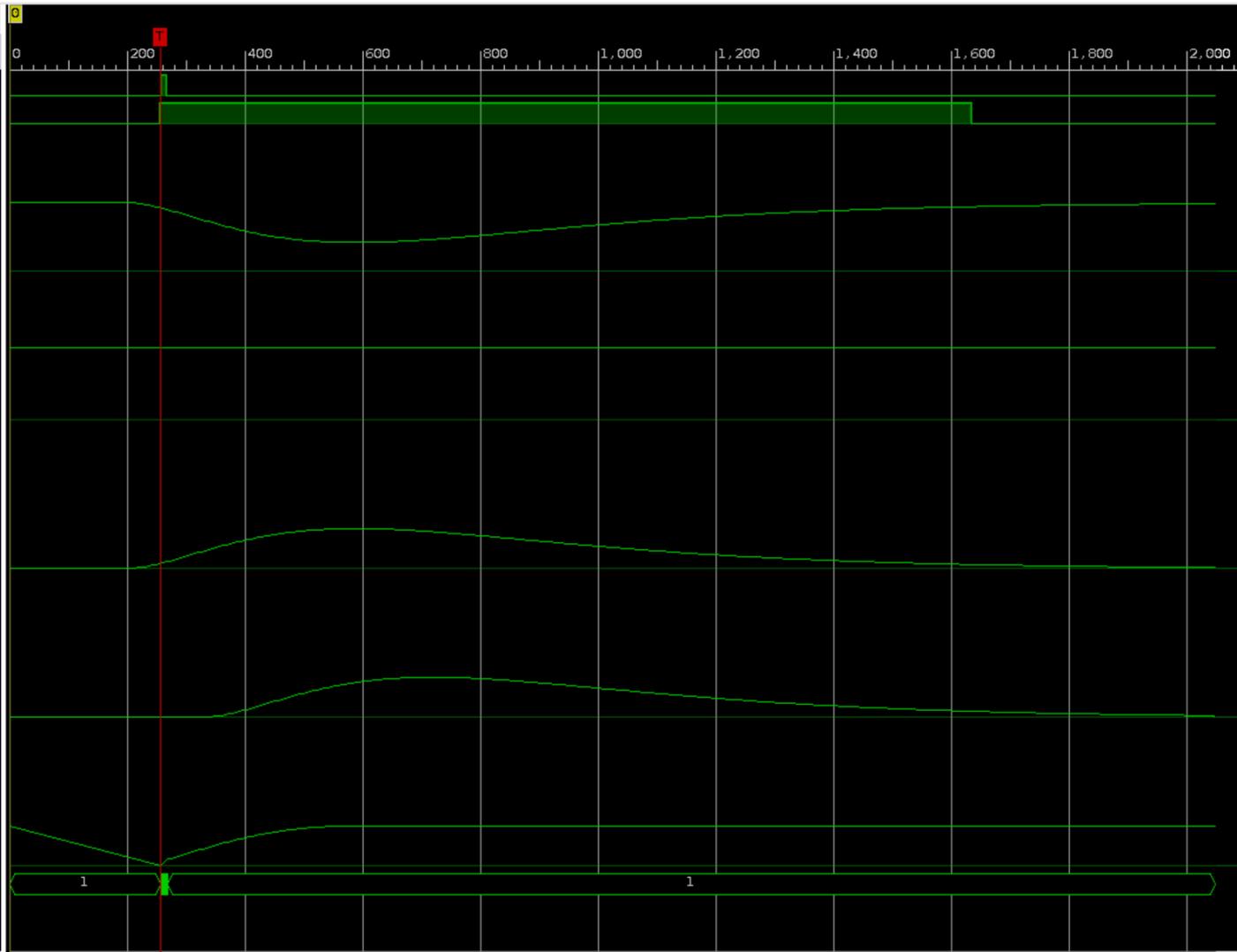
- PS-TEPCは μ -PICを利用し生体等価物質で構成された小型のタイムプロジェクションチェンバーである。宇宙船内で荷電粒子線のLETを高確度で測定することを目的とし開発を行っている。2016～2017年にて国際宇宙ステーションで動作実証がされている。
- 月周回友人拠点Gatewayへの搭載に向けて、PS-TEPC全系の小型・軽量・省電力化をLTARS2018の導入を核として実現することを目指している。
- 検出器からのデジタイズされた波形データを処理するFPGAにはGatewayでの動作を想定した際に必要となる以下の4つの機能を実現する必要がある。機能の開発を進め、動作確認を行っている。
 - 1. Anode、Cathode ch毎の正負極性を考慮したトリガー判定 → 機能開発済み、動作確認済
 - 2. トリガーをch毎にEnable/Disableする → 機能開発済み、動作確認済
 - 3. 後段へのデータ転送軽量化のための最大波高値とトリガータイミングの抽出による2値化 --> 機能開発済。一部動作未確認
 - 4. リアルタイムベースラインレベル抽出 --> VivadoIP利用、動作確認済
- 今後の予定
 - まだ動作確認が出来ていないものの確認。2値化によりトリガーレートがどこまで改善するかや、実際の検出器波形を用いた試験により、上記の機能が想定通り動作することを確認していく。

- backup



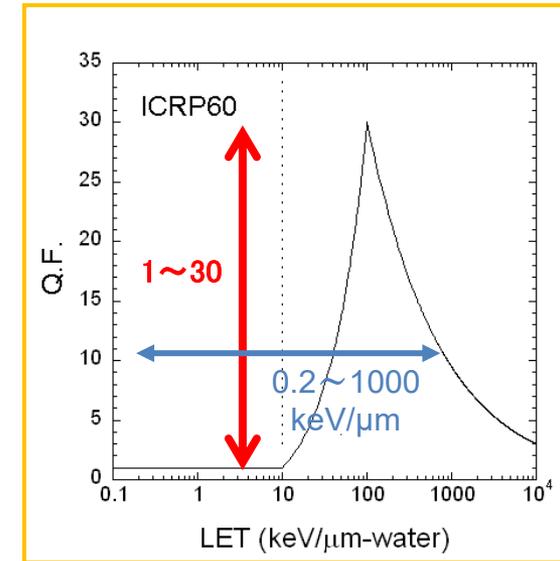
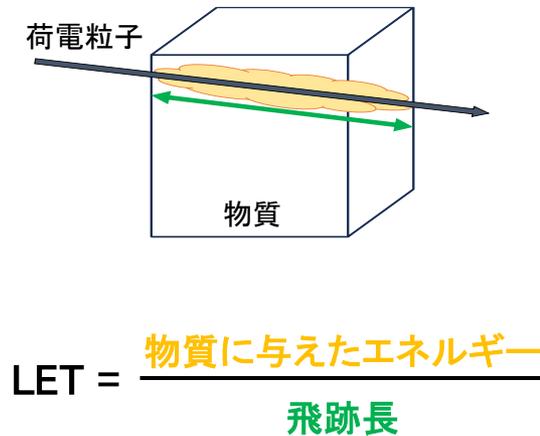
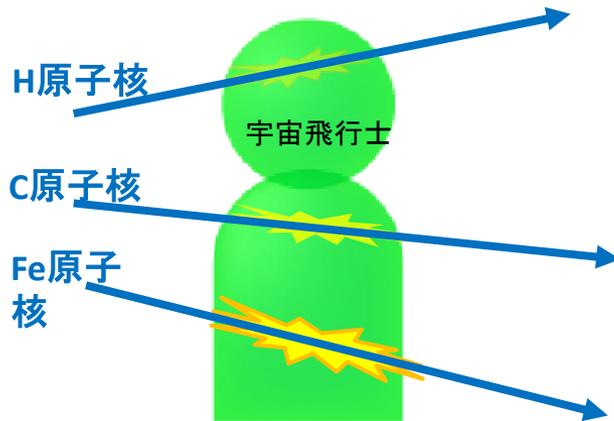
ILA Status: Idle

Name	Value
FPGA_TRG_1	0
ila_0_OTI	0
> adcData[767:756]	2005
> BASELINE[767:756]	2089
> ADC_TRUEVAL[767:756]	1
> DELAY_TRUEVAL[767:756]	1
> CH_MAX_VALUE[767:756]	1158
> CH_TIMING_VALUE[1023:1008]	1



Updated at: 2025-Dec-18 19:28:13

宇宙放射線環境下での線量評価方法



線質係数

宇宙放射線環境下での線量は以下の線量当量によって評価されるのが一般的

線量当量 = 線質係数 $Q(\text{LET})$ x 吸収線量

LET (Linear Energy Transfer, 線エネルギー付与)

= Energy deposition / Path length

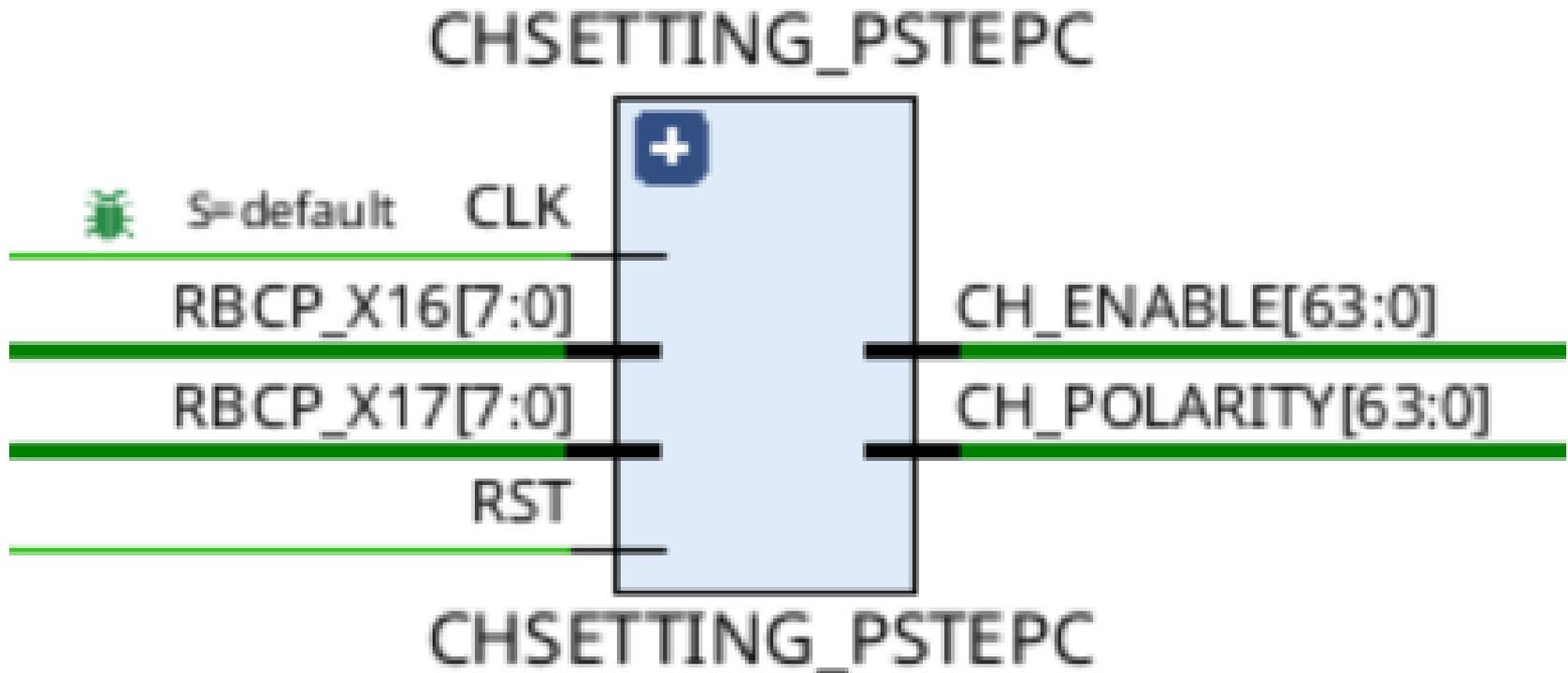


吸収線量だけでなく、**LET**も同時に測定できる線量計

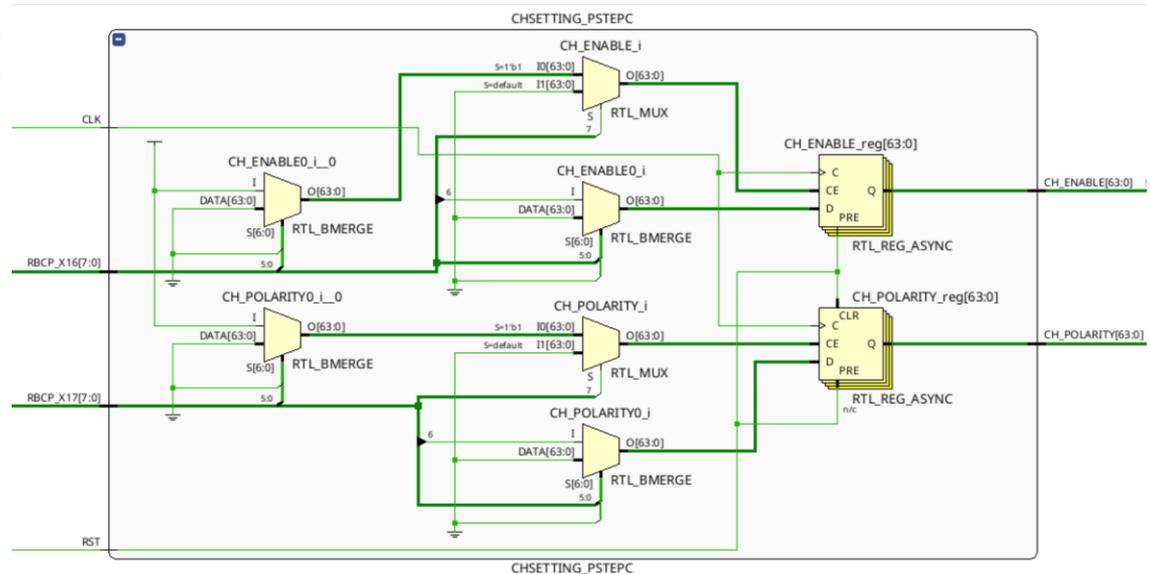
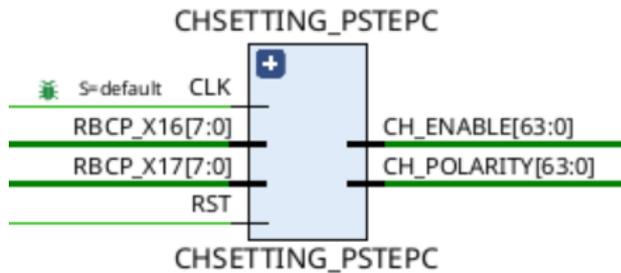
線量計測におけるエラーが **30% (1σ)** 以内であることが望ましい

National Council of Radiological Protection Report No. 142 (NCRP-142) 32

Ch設定モジュール



チャンネル設定



- RBCPのレジスタで直接設定しようとしたが、64bit × 2が意外と重くて断念
- 自前でレジスタを用意して6bitのアドレスを指定し、ON/OFFする
- 操作の8bitデータは { 1'b[set_Enable], 1'b [On/Off], 6'b [Ch_number] }

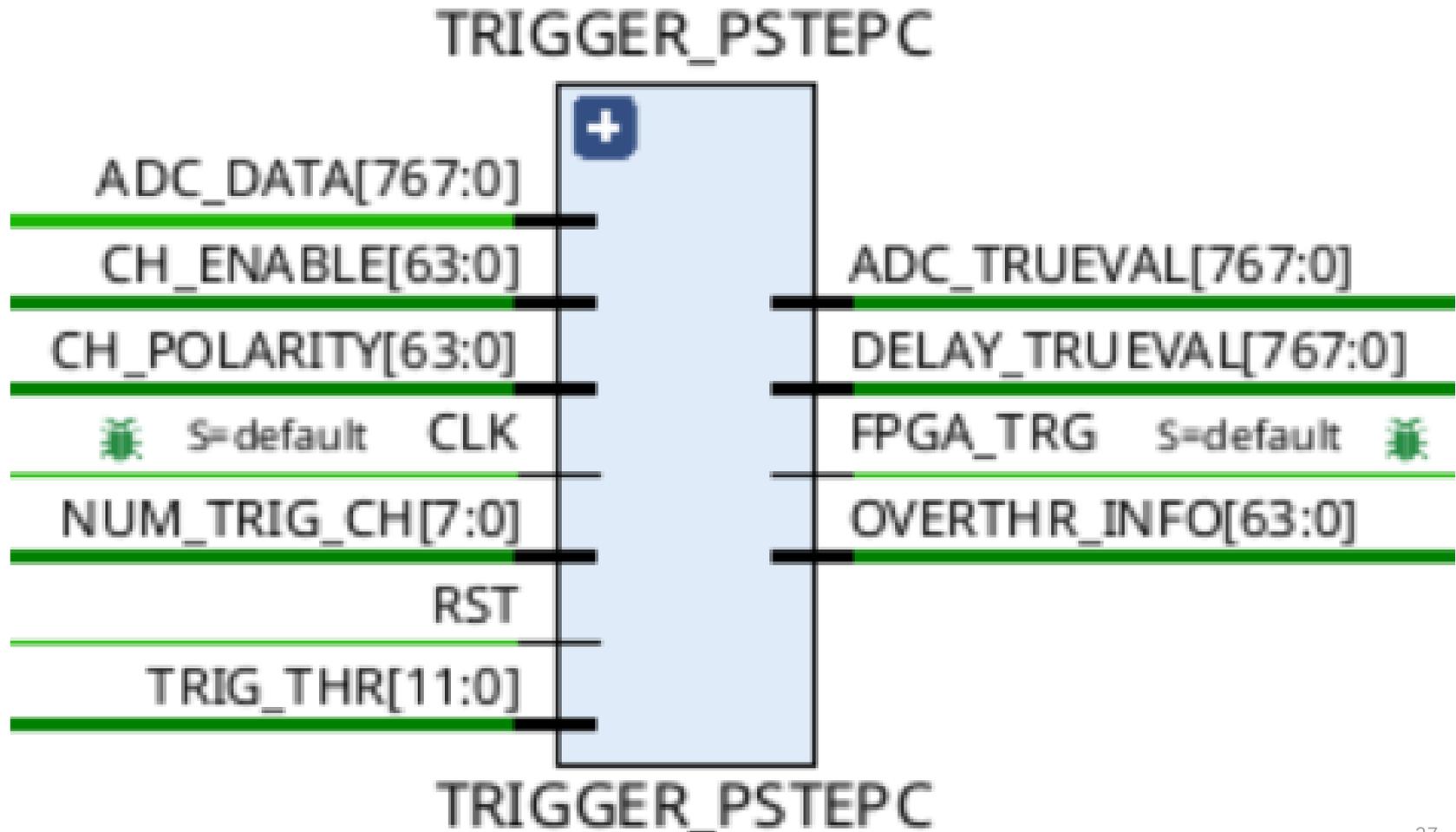
チャンネル設定

- 各チャンネルごとの設定には、64bit分のレジスタが必要
- 自前でレジスタを用意して6bitのアドレスを指定し、ON/OFFする
- 操作の8bitデータは { 1'b[set_Enable], 1'b [On/Off], 6'b [Ch_number] }

ILA Status: Idle	
Name	Value
> CH_ENABLE_1[15:0]	ffff
> CH_ENABLE_2[31:16]	ffff
> CH_ENABLE_3[47:32]	ffff
> CH_ENABLE[63:48]	ffff
> CH_POLARITY_1[15:0]	ffff
> CH_POLARITY_2[31:16]	ffff
> CH_POLARITY_3[47:32]	0000
> CH_POLARITY[63:48]	0000

ILA Status: Idle	
Name	Value
> CH_ENABLE_1[15:0]	ffff
> CH_ENABLE_2[31:16]	ffff
> CH_ENABLE_3[47:32]	ffff
> CH_ENABLE[63:48]	ffff
> CH_POLARITY_1[15:0]	ffff
> CH_POLARITY_2[31:16]	ffff
> CH_POLARITY_3[47:32]	0000
> CH_POLARITY[63:48]	1000

セルフトリガモジュール



CIC_Compilerでベースライン抽出

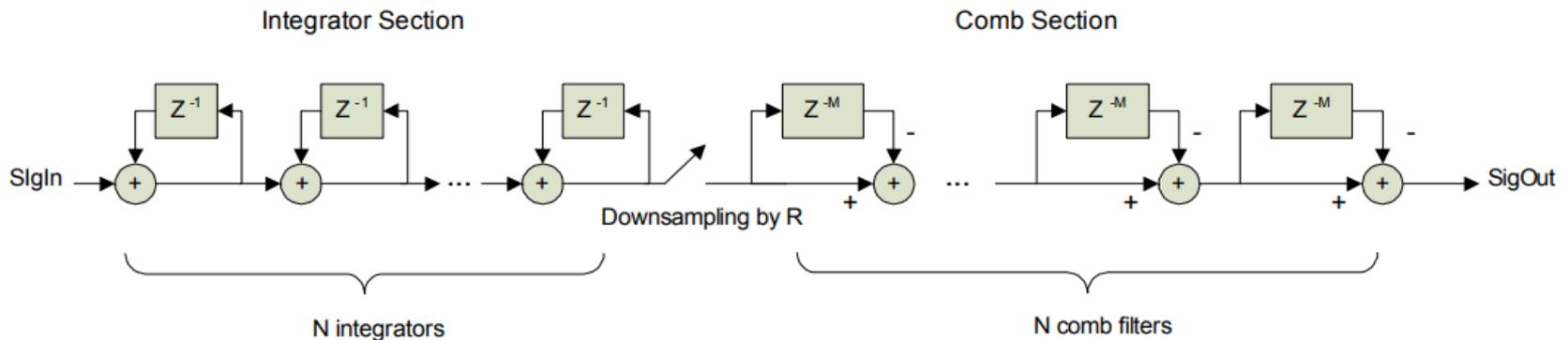
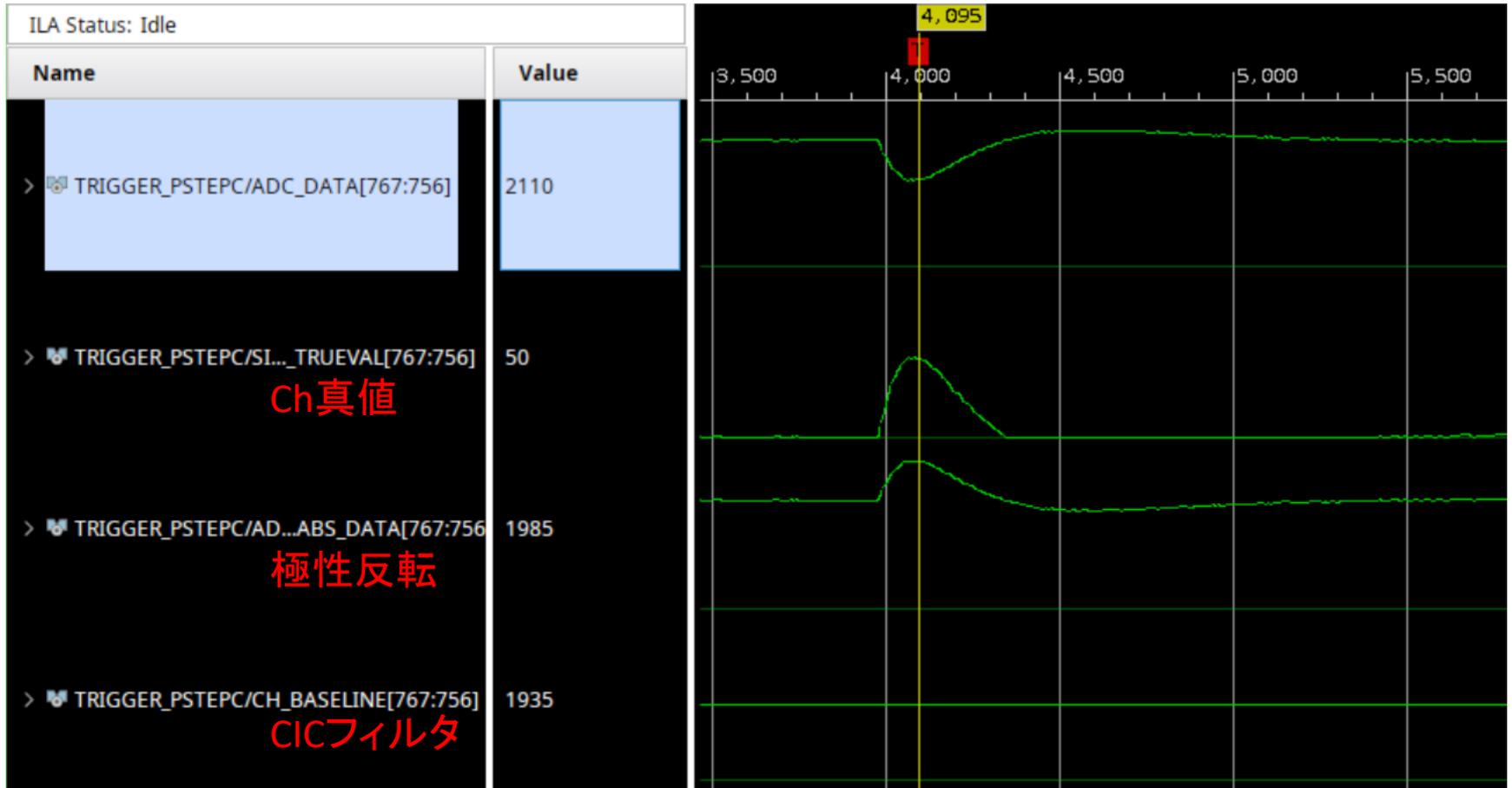


Figure 7: CIC Decimation Filter

- cic_compiler設定
 - Decimation, Stage:3, Sample:64
 - 30bit幅のoutput
 - 18bit右シフトしてBaselineが求まる
 - RSTから1024CLKでBaseline安定

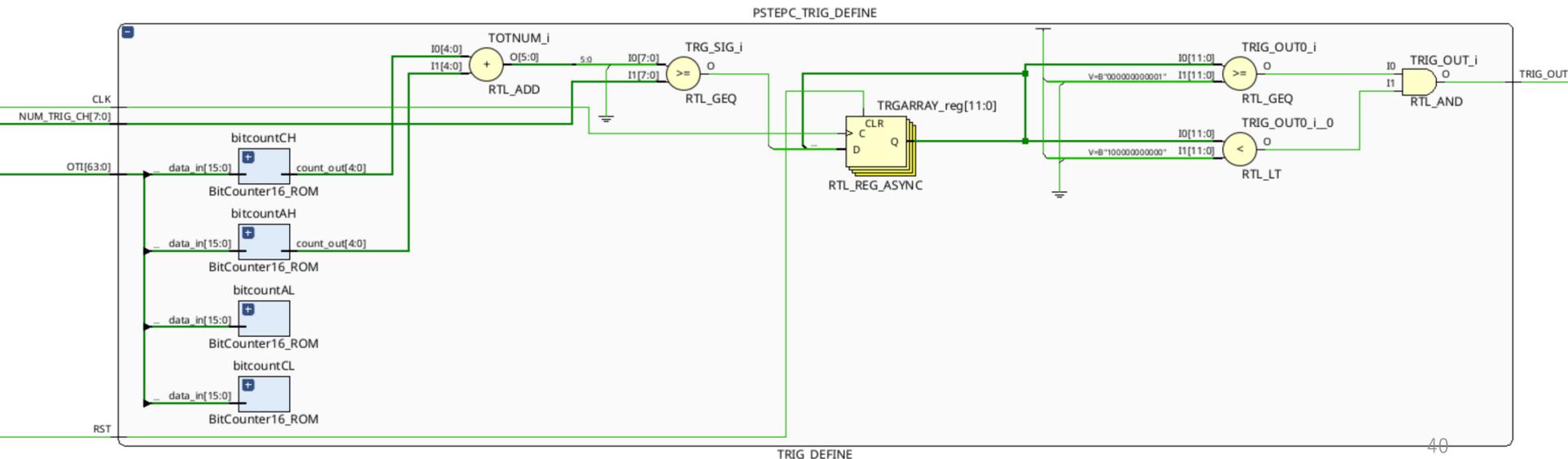
処理を行った実波形データ



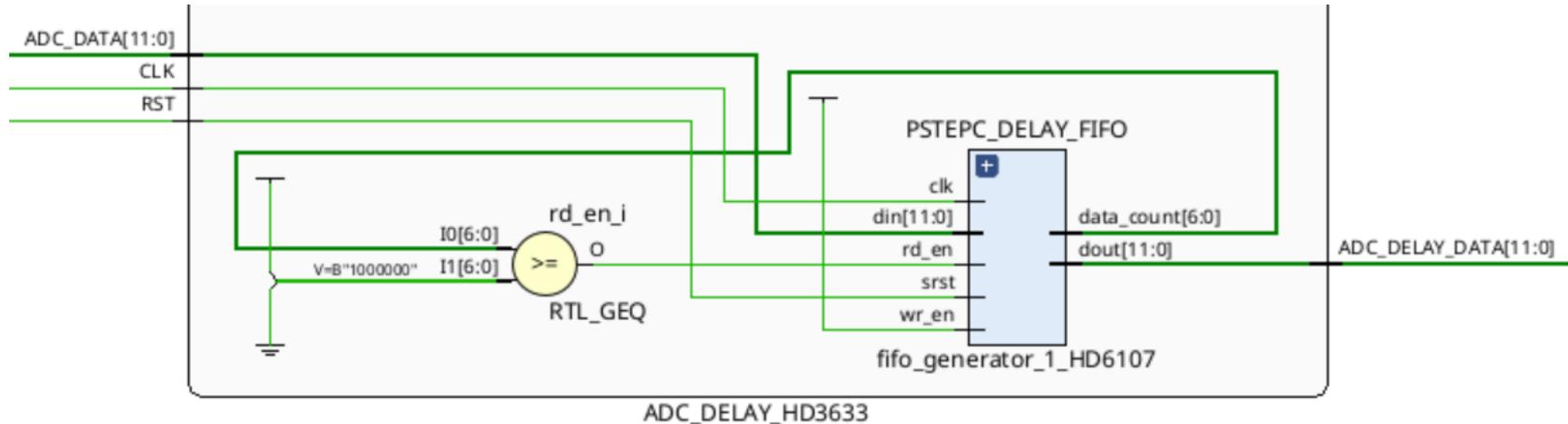
いずれかのchのTRUEVALが閾値を超えたらトリガ発行

トリガ発行

- 16chのBitCounterを作成して、Cathode/AnodeのHigh/Low gainを分けてカウント可能にする
- 今回はHighGainの32chの合計が設定値 (NUM_TRIG_CH \geq 1) を超えた場合に12CLKのトリガを出す

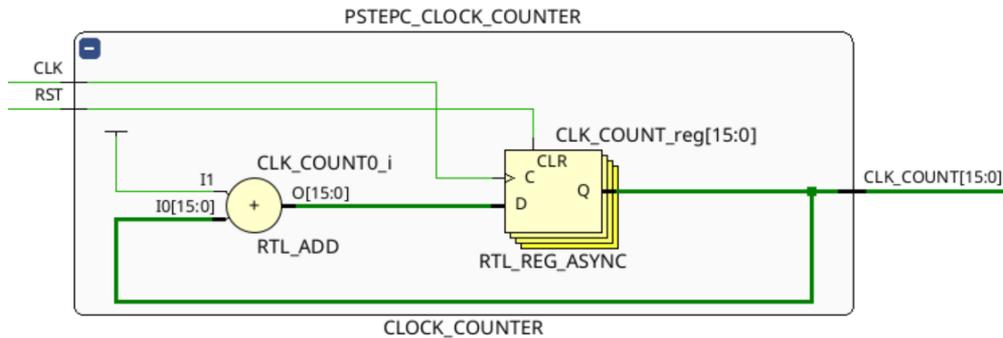


遅延データ作成



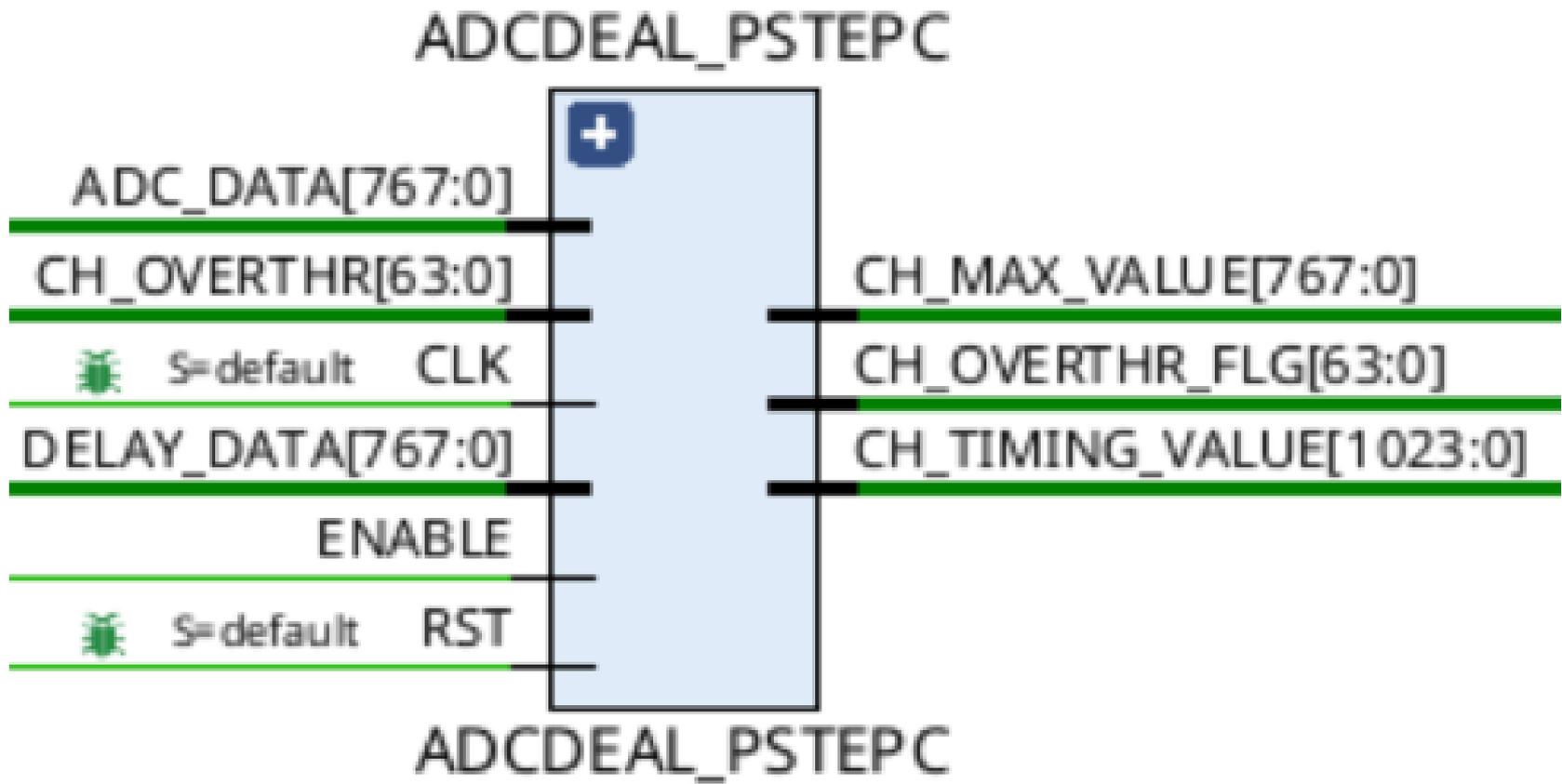
- FIFOで n_d CLK分の遅延データ(n_d は現在は固定パラメータ)を生成してTimingの決定に使う
- RSTタイミングが異なるため二値化モジュールからは外に出してある

カウンタ

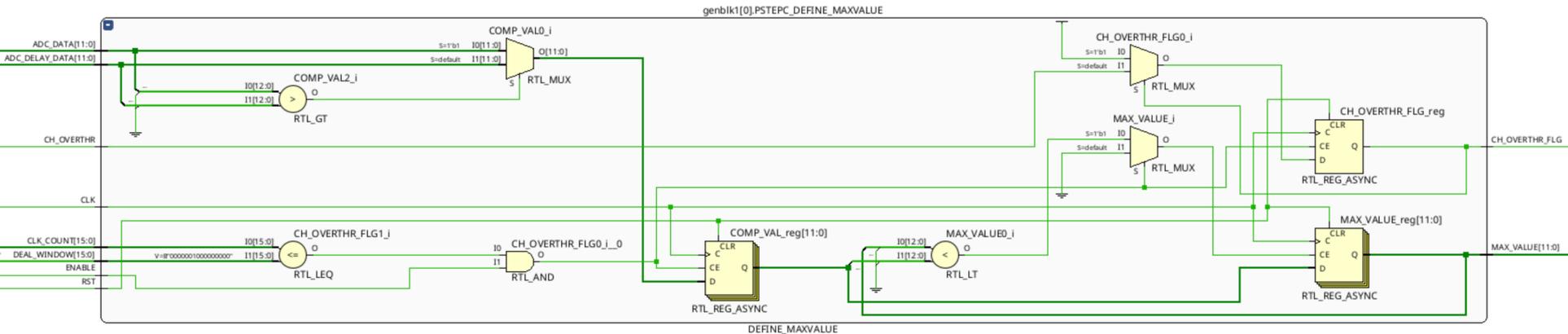


- タイミング決定用のクロックカウンタ、トリガでRST

最大波高・トリガタイミング決定 モジュール

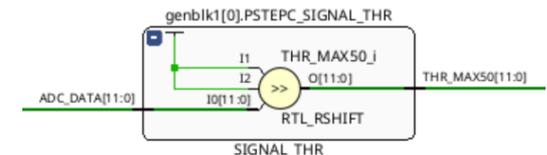
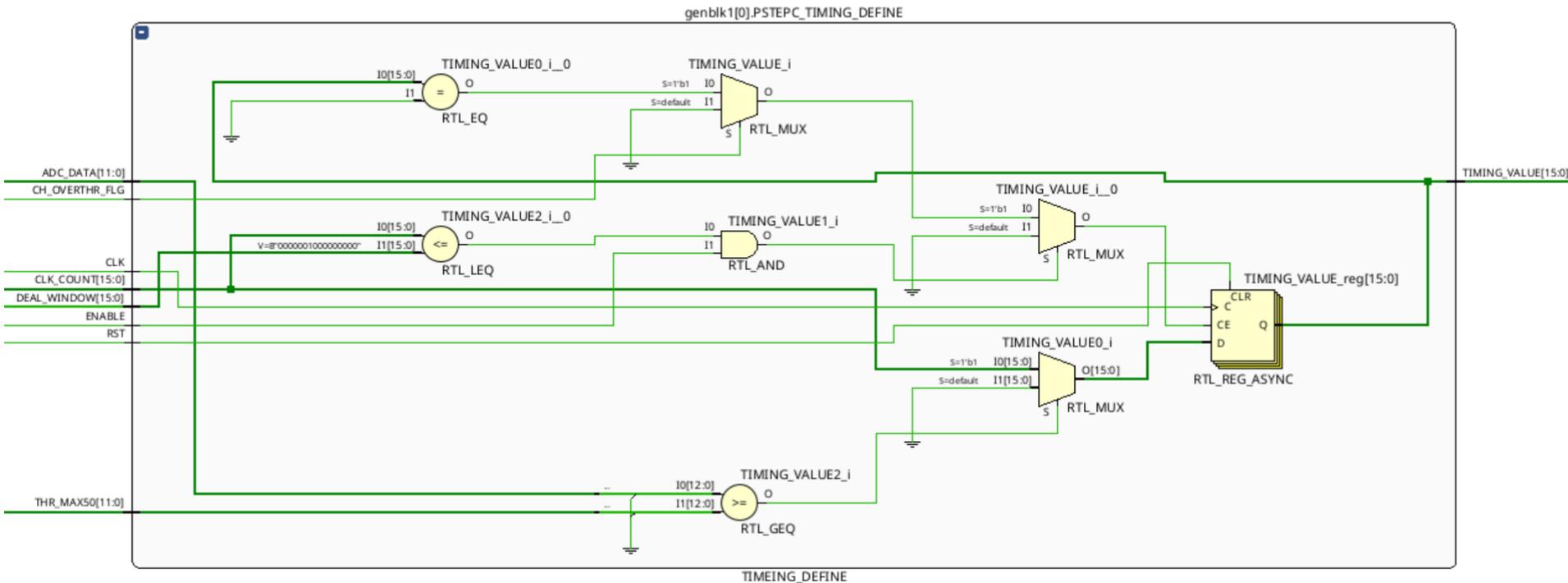


MaxValue決定



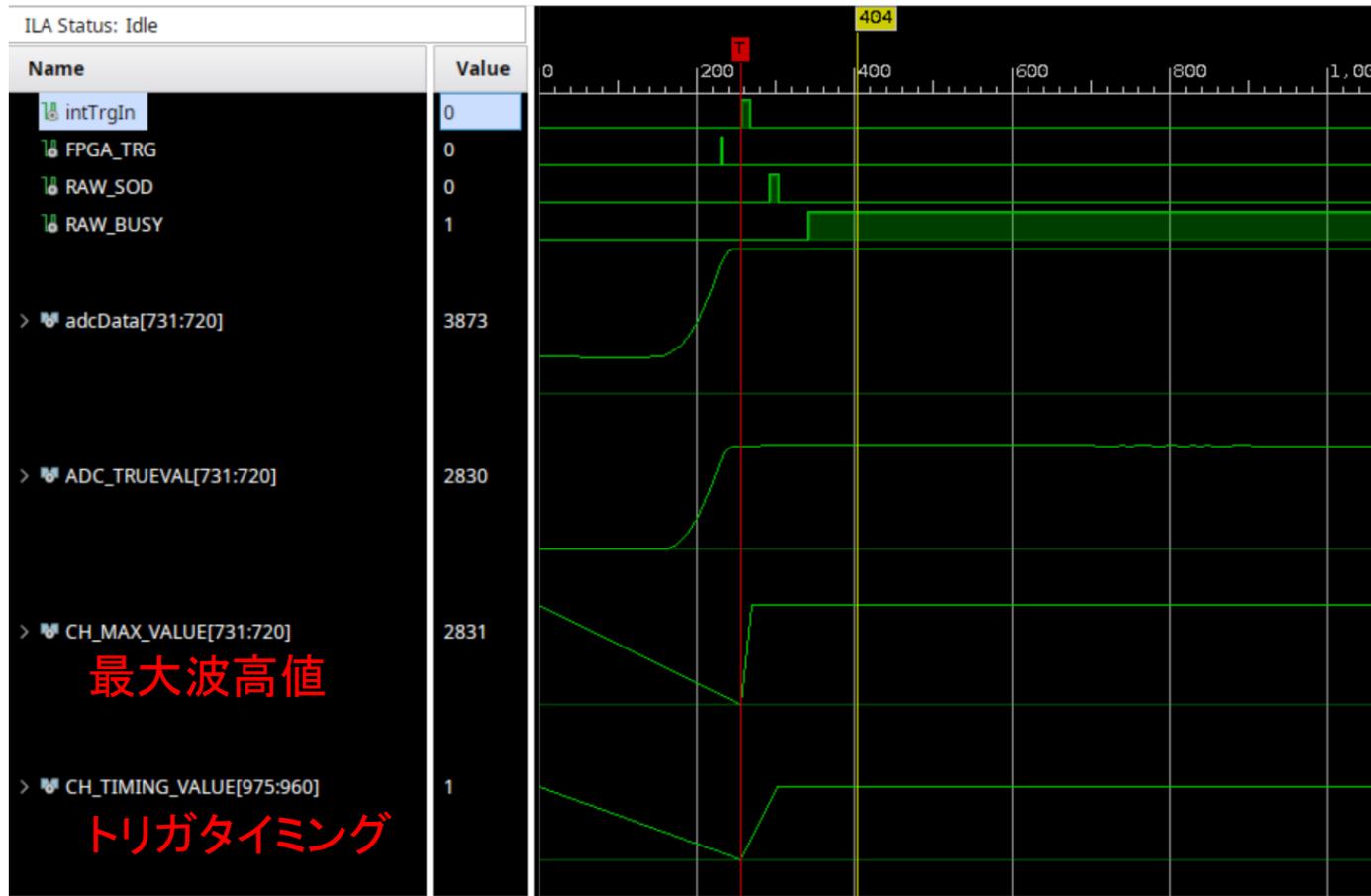
- トリガでRST
- カウントが n_g CLK (n_g は現在固定パラメータ) より小さい間のみCEをHに
- Wireで入力されるADC_DATAを順次比較して大きければレジスタ更新
- ChごとのOverThresholdもフラグとして保持

Timing決定



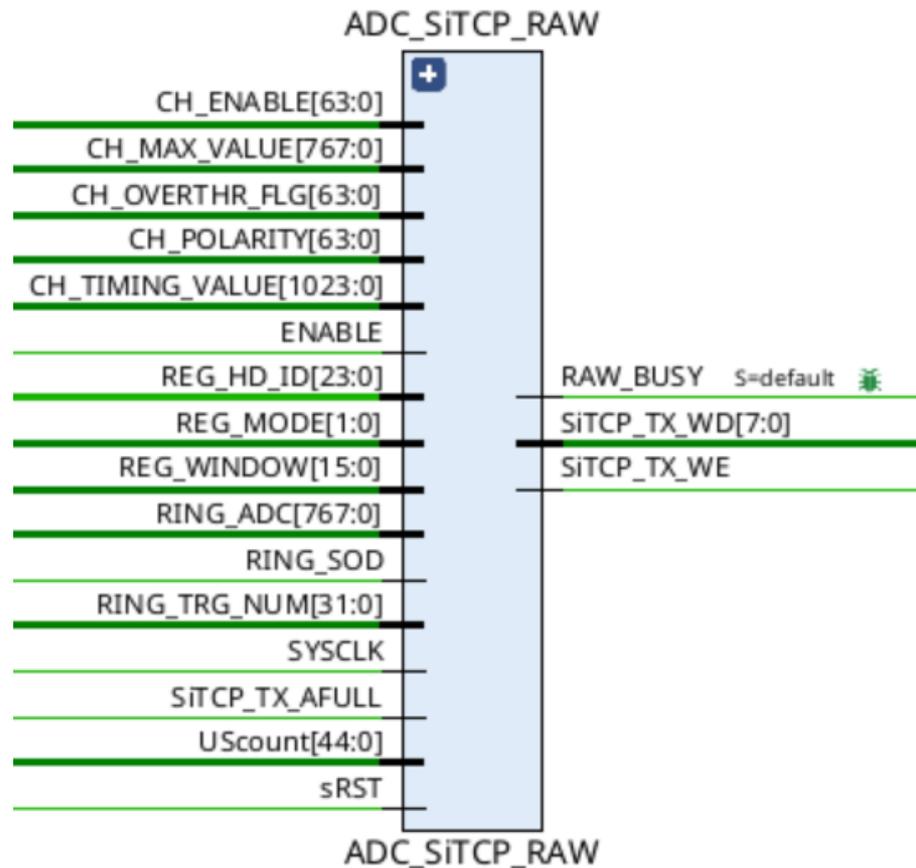
- トリガでRST
- 遅延データとMaxValueの右1bitシフトを比較して、大きくなったCLK_CLOUNTを保持

処理中の実信号



最大波高値・トリガタイミングはレジスタに保持される

計算データ送信 (ADC_SiTCP_RAW改造)



SiTCPデータ送信位置の決定

```
845 |         reg [15:0] rdLen; // 20250919 ktaka add
846 |
847 |     always@ (posedge SYSCLK) begin
848 |         rdHa[3:0]   <= ha[3:0];
849 |         hdVal       <= ~ha[4] & hdEnb;
850 |
851 |         // rdRa[4:0]   <= ra[4:0];
852 |         rdRa[6:0]   <= ra[6:0];
853 |         // rdVal[0]   <= ~len[21] & rdEnb;
854 |         rdVal       <= ~len[22] & rdEnb;
855 |
856 |         rdLen <= {1'b0, ~len[21:7]}; // 20250919 ktaka add
857 |
858 |     end

866 |         reg [15:0] dlyLen; // 20250919 ktaka add
867 |
868 |     always@ (posedge SYSCLK) begin
869 |         dlyRdHa[3:0] <= rdHa[3:0];
870 |         dlyHdVal     <= hdVal;
871 |
872 |         // dlyRdRa[4:0] <= rdRa[4:0];
873 |         dlyRdRa[6:0] <= rdRa[6:0];
874 |         dlyRdVal     <= rdVal;
875 |
876 |         dlyLen[15:0] <= rdLen[15:0]; // 20250919 ktaka add
877 |     end
```

- Lenの[21:7]が送信データの位置を示す
- データ末尾に計算データを付与するためわかりやすく補数にする、~len[21:7]が1なら末尾から2clk目、0なら1clk目
- Window幅をそのまま読むと8bitデータが6bit分(64個)なので12bitデータ64個分(8bit × 2 × 64)に半分足りない
- 通常データと合わせて1clkずらす

SiTCP_TX_WD用レジスタ設定

```
888 // 20250919 ktaka add start
889     reg REG_OVERTHR[63:0];
890     reg [7:0] REG_MAXVAL_L[63:0];
891     reg [3:0] REG_MAXVAL_H[63:0];
892     reg [7:0] REG_TIMVAL_L[63:0];
893     reg [7:0] REG_TIMVAL_H[63:0];
894     reg REG_ENABLE[63:0]; // 20250925 ktaka add
895     reg REG_POLARITY[63:0]; // 20250925 ktaka add
896
897     integer dataloop;
898     always@ (posedge SYSCLK) begin
899         for (dataloop = 0; dataloop < 64; dataloop=dataloop+1) begin
900             REG_OVERTHR [dataloop] <= CH_OVERTHR_FLG [dataloop];
901             REG_MAXVAL_L [dataloop] <= CH_MAX_VALUE [dataloop*12 +: 8];
902             REG_MAXVAL_H [dataloop] <= CH_MAX_VALUE [dataloop*12+8 +: 4];
903             REG_TIMVAL_L [dataloop] <= CH_TIMING_VALUE [dataloop*16 +: 8];
904             REG_TIMVAL_H [dataloop] <= CH_TIMING_VALUE [dataloop*16+8 +: 8];
905             REG_ENABLE [dataloop] <= CH_ENABLE [dataloop];
906             REG_POLARITY [dataloop] <= CH_POLARITY [dataloop];
907         end
908     end
909 // 20250919 ktaka add end
```

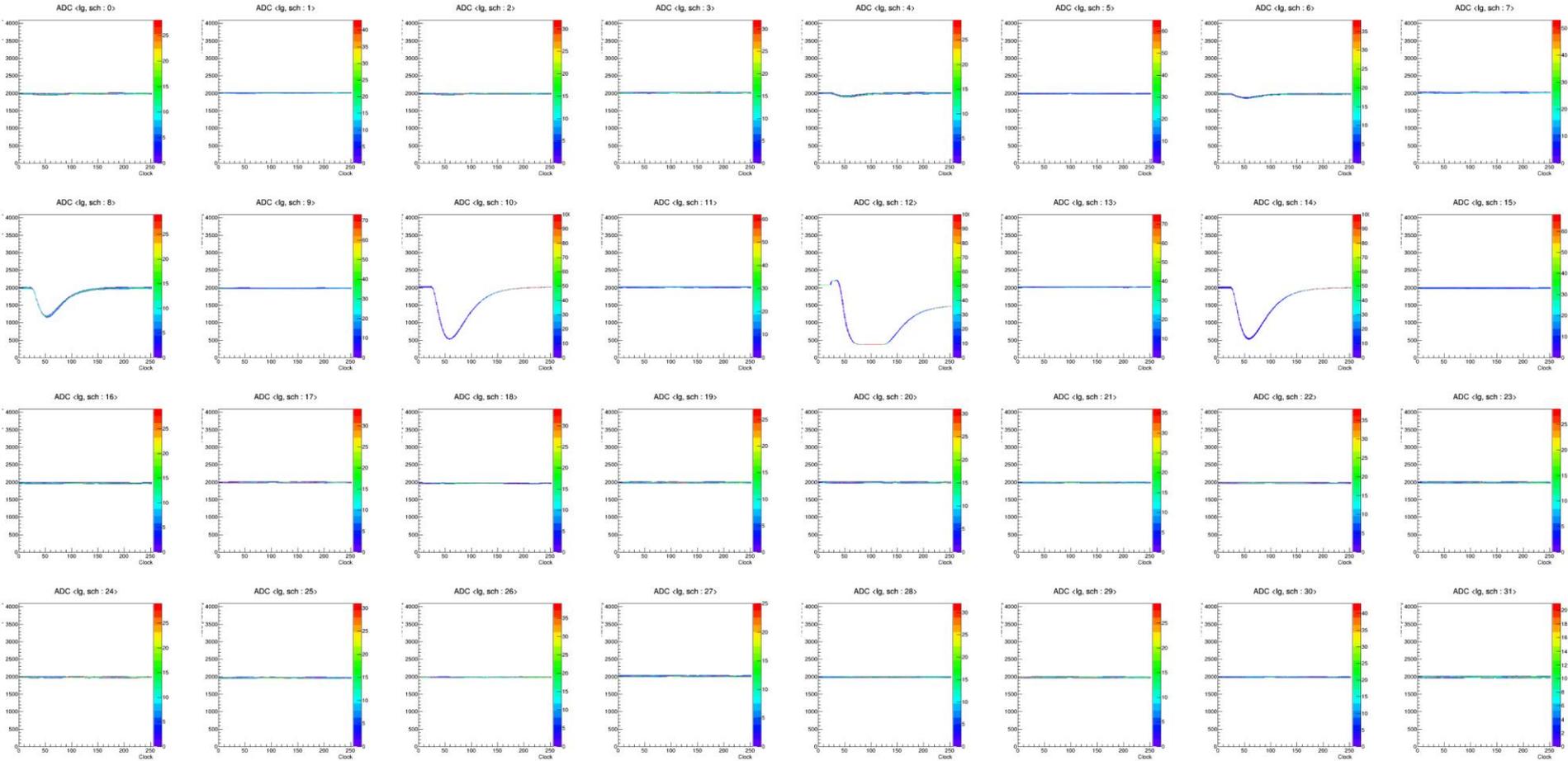
- 8bitに収まるように64ch分の配列レジスタに分割

SiTCP送信

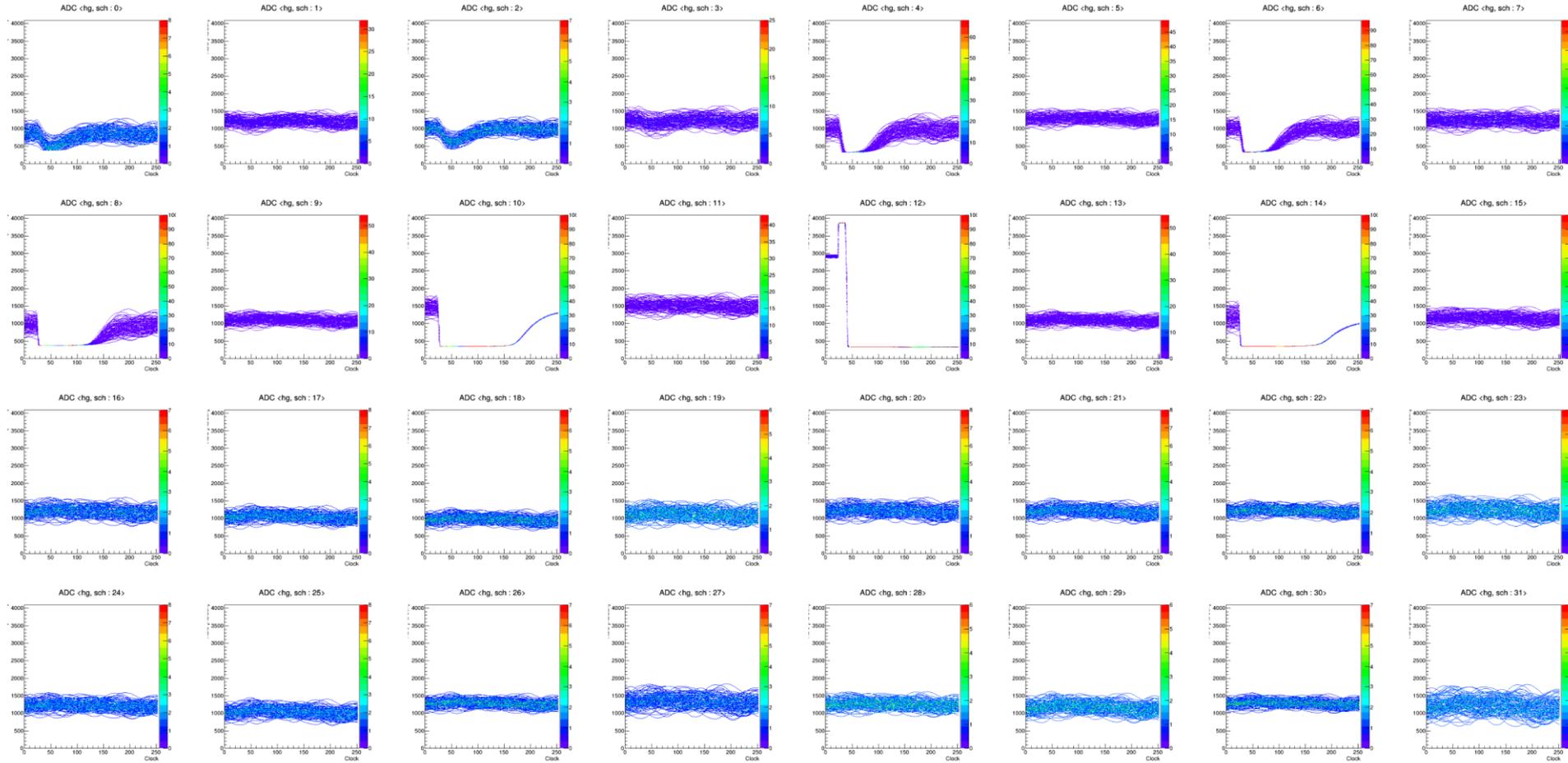
```
1038 // 20250919 ktaka add start
1039 case (dlyLen[15:0])
1040
1041     16'd1: begin : CH_OVERTHR_AND_MAXVALUE_SEND
1042         case(dlyRdRa[0])
1043
1044             1'd1: rdAdcData <= {REG_MAXVAL_L[dlyRdRa[6:1]]};
1045             default: rdAdcData <= {REG_OVERTHR[dlyRdRa[6:1]],1'b0, REG_ENABLE[dlyRdRa[6:1]], REG_POLARITY[dlyRdRa[6:1]], REG_MAXVAL_H[dlyRdRa[6:1]]};
1046         endcase
1047     end
1048     16'd0: begin : TIMING_SEND
1049         case(dlyRdRa[0])
1050             1'd1: rdAdcData <= {REG_TIMVAL_L[dlyRdRa[6:1]]};
1051             default: rdAdcData <= REG_TIMVAL_H[dlyRdRa[6:1]];
1052         endcase
1053     end
1054
1055     default:begin : ADCDATASEND
1056
1057 // 20250919 ktaka add end
1058
1059     case(dlyRdRa[6:0]) // [6:0]
1060         7'd0: rdAdcData[7:0] <= {4'd0, rd[12*0+11:12*0+8]};
1061         7'd1: rdAdcData[7:0] <= rd[12*0+7:12*0];
1062         7'd2: rdAdcData[7:0] <= {4'd0, rd[12*1+11:12*1+8]};
1063         7'd3: rdAdcData[7:0] <= rd[12*1+7:12*1];
1064         7'd4: rdAdcData[7:0] <= {4'd0, rd[12*2+11:12*2+8]};
1065         7'd5: rdAdcData[7:0] <= rd[12*2+7:12*2];
```

- dlyLen[15:0]で分岐 (case:0(Timing), 1(Maxvalue), 他(ADC_DATA)) ← 並びは逆
- DlyRdRaが奇数の場合下位8bit、偶数の場合上位8bit
- MaxValueは12bitなので4bit空いてる分に { OverThreshold, 0, Ch_Enable, Ch_Polarity } を詰める

LG側の実シグナル



HG側の実シグナル



まとめ

- PS-TEPCは生体等価物質で構成された、TPC + μ -PICによる放射線検出器である。
- 生体等価検出器で荷電粒子線のLETを高確度で測定可能で、宇宙放射線量計としての実績を持つ。
- Gatewayへの搭載に向けて、小型・軽量・省電力化の要請を、LTARS2018の採用により実現する。
- ADC_SiTCP_V20にセルフトリガ、二値化などのモジュールを実装し、SiTCPによりデータ送信を可能とした。