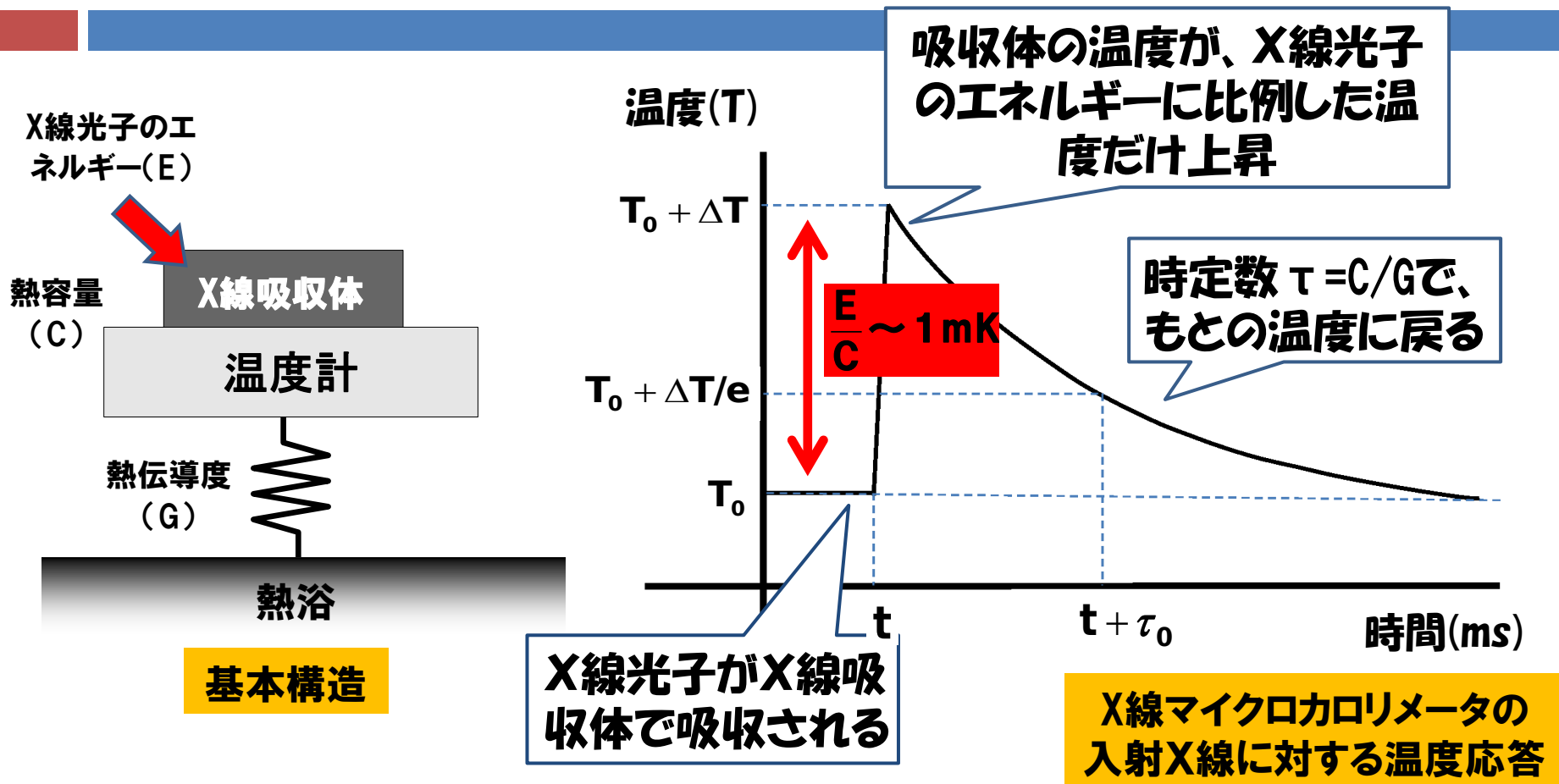


# 次世代X線マイクロカロリメータの ための最適微分フィルタの実装

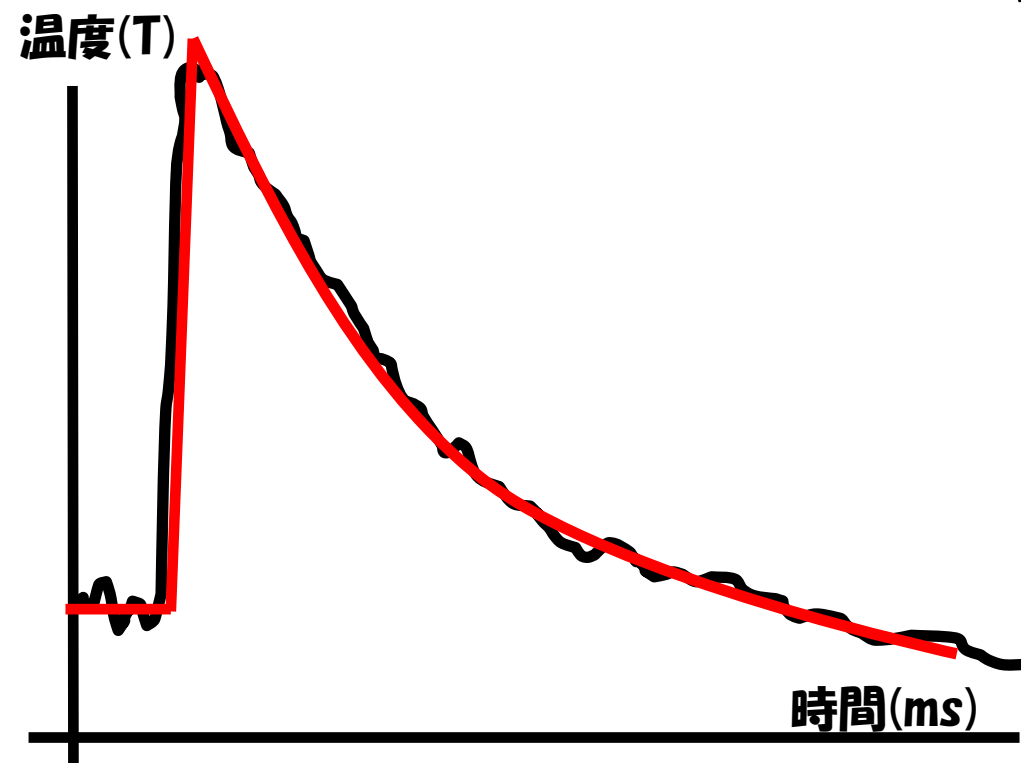
埼玉大学 修士1年 山口直

# X線マイクロカロリメータ



温度変化パルスの波高値をX線光子のエネルギーとして読みとることで高いエネルギー分解能を達成する。

# 最適フィルター処理

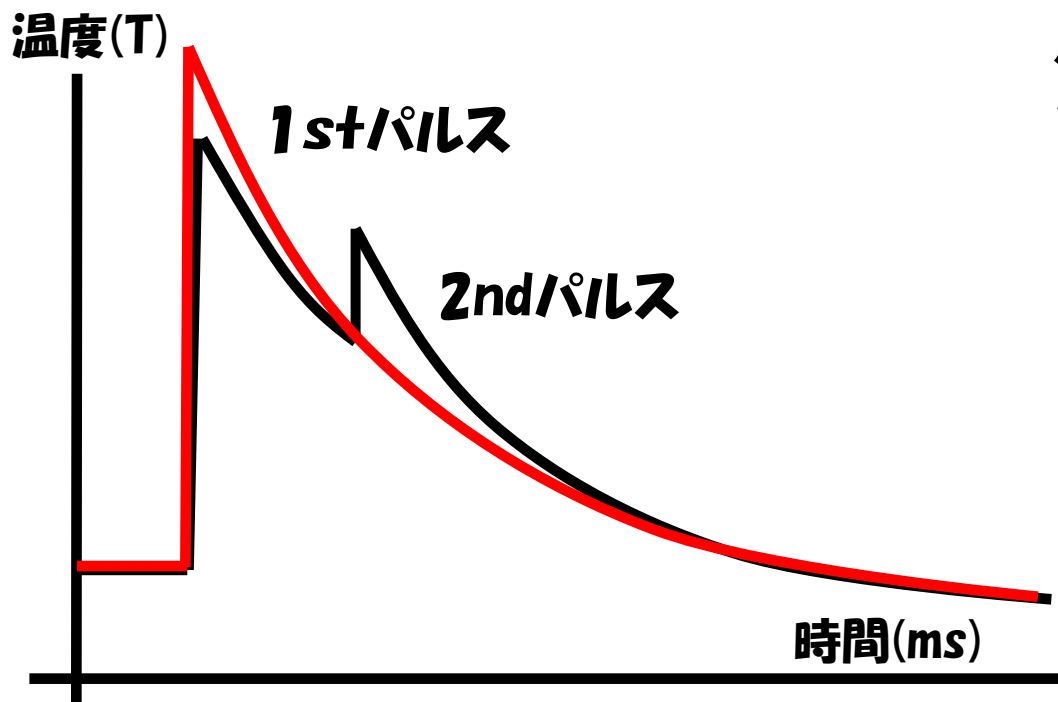


実際の波形はノイズによって変形されてしまうため、そのままの波高値を読んだだけでは正しいエネルギーを得られない。



波形全体をテンプレート波形(赤線)と比較して波高値を求める。(最適フィルター処理)

# 2ndパルスと最適フィルター処理



1stパルスが定常状態に戻る前に別の光子が入射してしまい、2ndパルスとして出でしまうことがある。そのような波形でも最適フィルター処理がうまくいかなくなる。(左図)

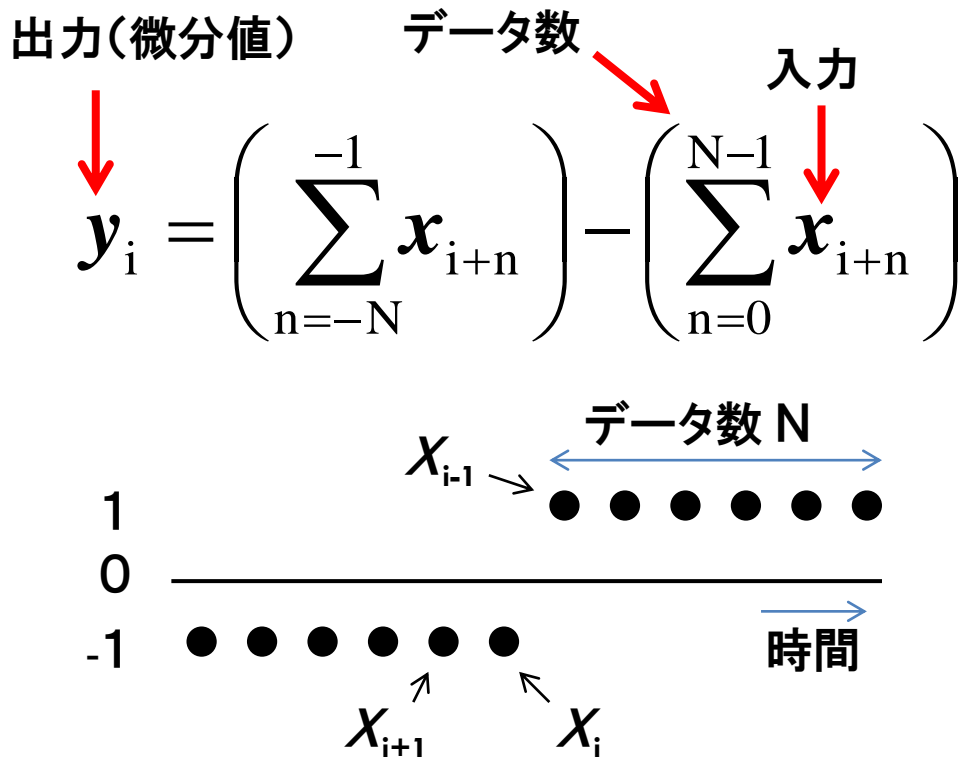


**高いエネルギー分解能の達成には、2ndパルスの検出・分離が必要不可欠である！**

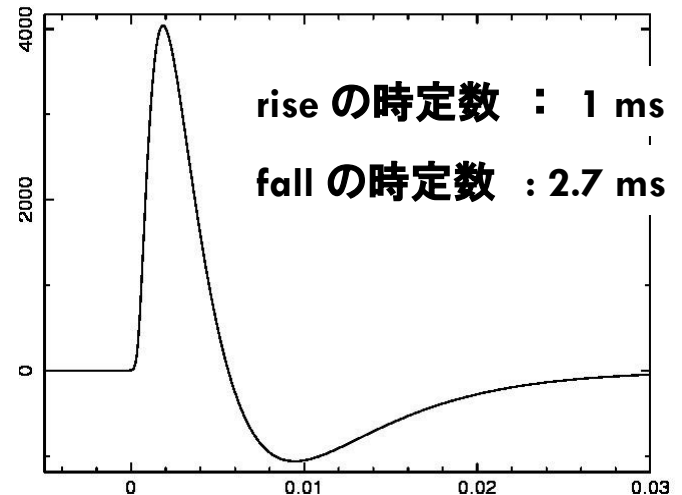
# X線イベントの検出

すざくやASTRO-Hなどの衛星では、X線到来イベントの検出にBoxcarフィルターで計算された**微分波形**を使用する。

## Boxcarフィルター

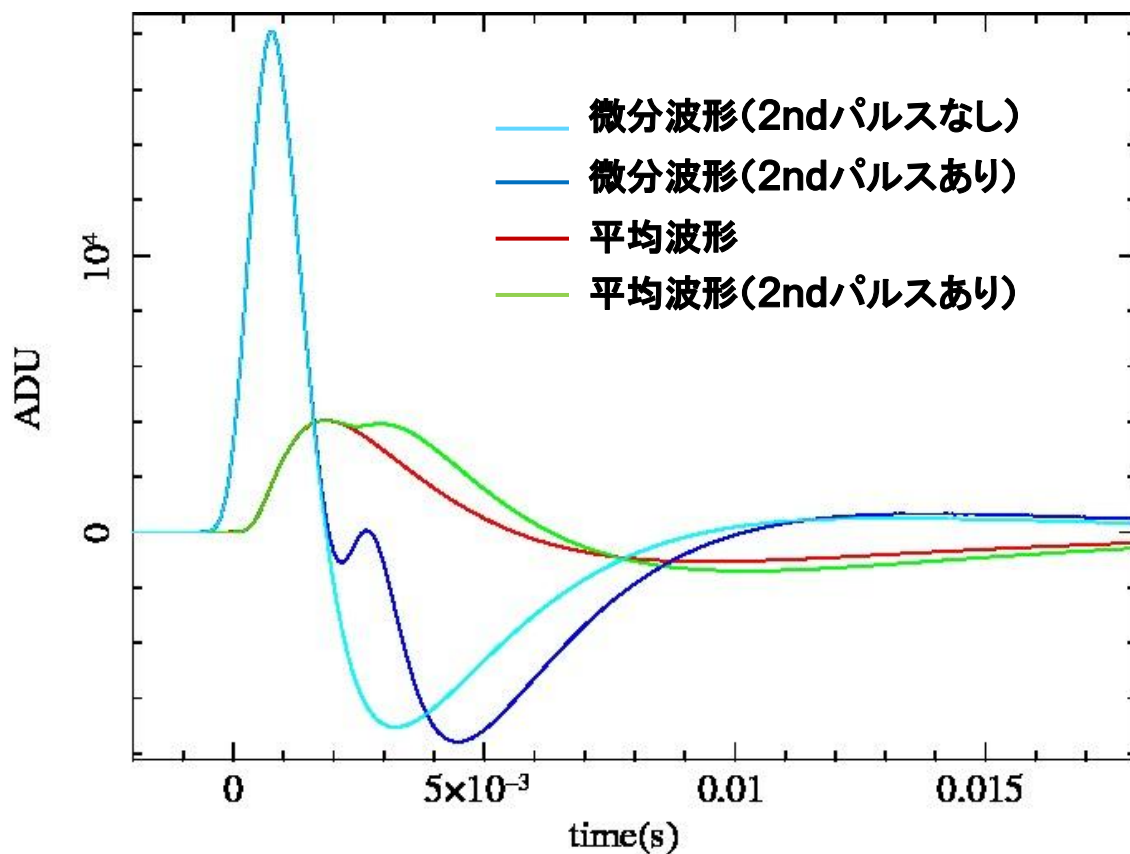


実際に使用した半導体マイクロカロリメータの平均波形



# 1stパルスの2ms後に2ndパルスが入射した波形をBoxcarフィルターに通すと、、、

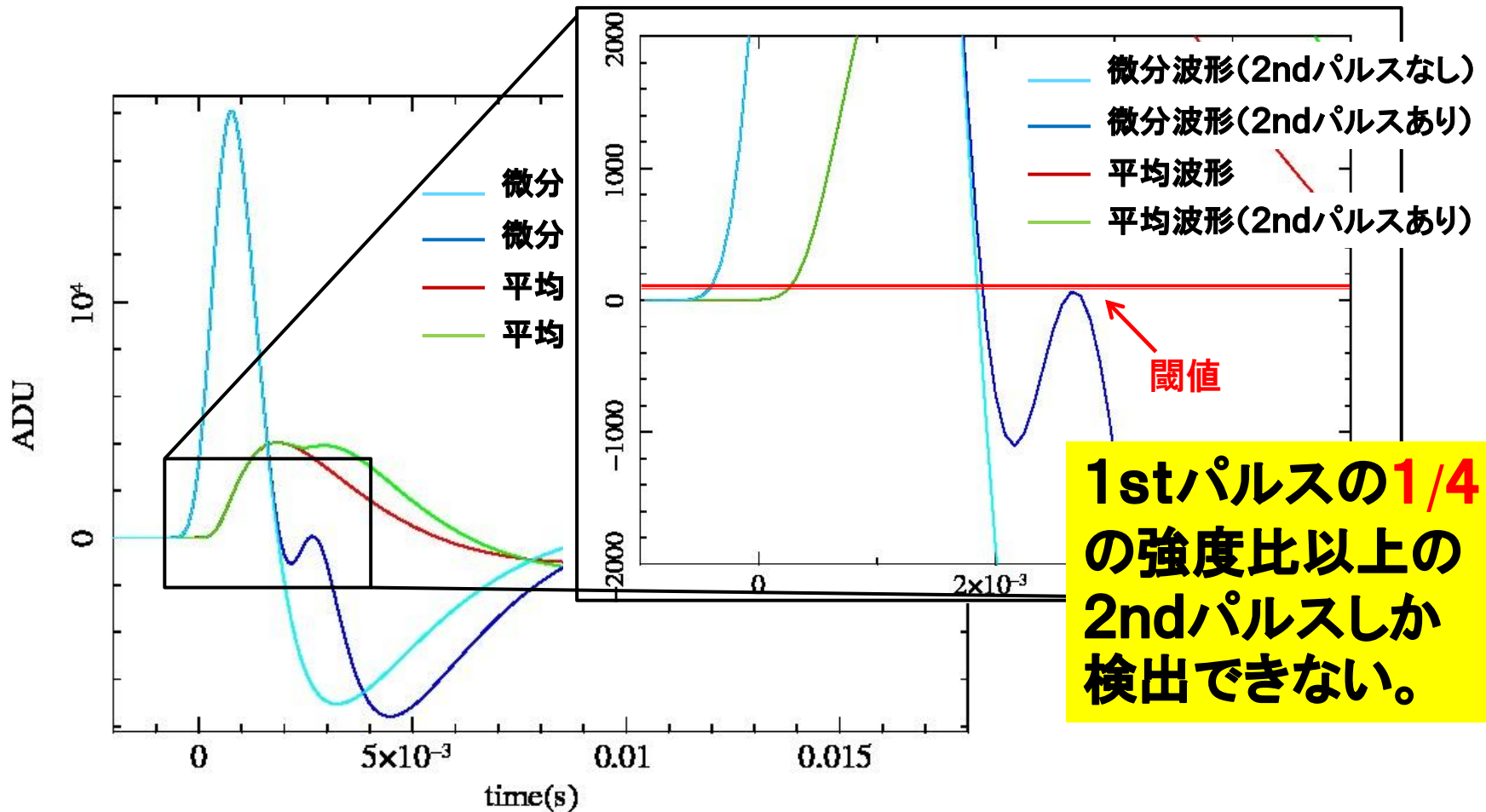
## Boxcarフィルターを用いた微分法で得られる微分波形



**微分値が、ある閾値を超えたものをX線イベントとする！！**

# Boxcarフィルターを用いての、2ndパルスの検出限界を見積もると、

## Boxcarフィルターを用いた微分法で得られる微分波形



# 最適微分フィルタの開発

ASTRO-H衛星では、FPGA上に実装されたBoxcarフィルタで2ndパルス検出を行うが、Boxcarフィルタだけでは十分な検出が行えない(前のページ参照)ため、FPGAでの処理の後にCPUを用いている。しかし、次世代のX線マイクロカロリメータでは多素子化が予想され、CPUの処理では間に合わなくなる。

→ そこで、我々は比較的安易に高速並列処理可能なFPGA上で実現可能な、Boxcarに代わる微分フィルタを開発している。

## FIRフィルタ

出力(微分値)  $y_i$

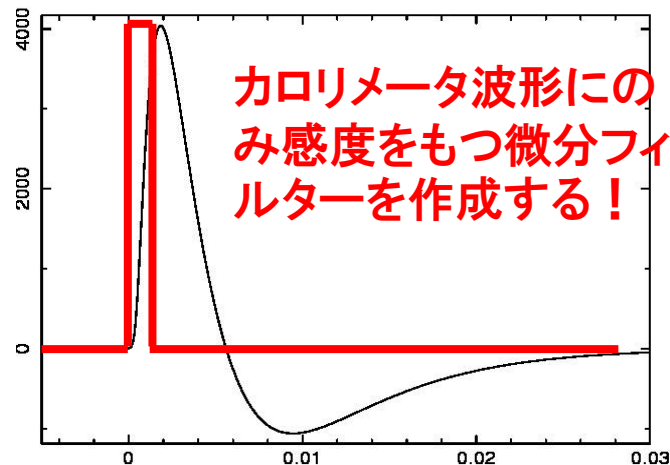
フィルター係数の数  $N-1$

フィルター係数  $h_n$

入力  $x_{i-n}$

$$y_i = \sum_{n=-N}^{N-1} h_n x_{i-n}$$

適当なフィルター係数を求めることで



FIRフィルタを用いた微分法で得られる理想の微分波形



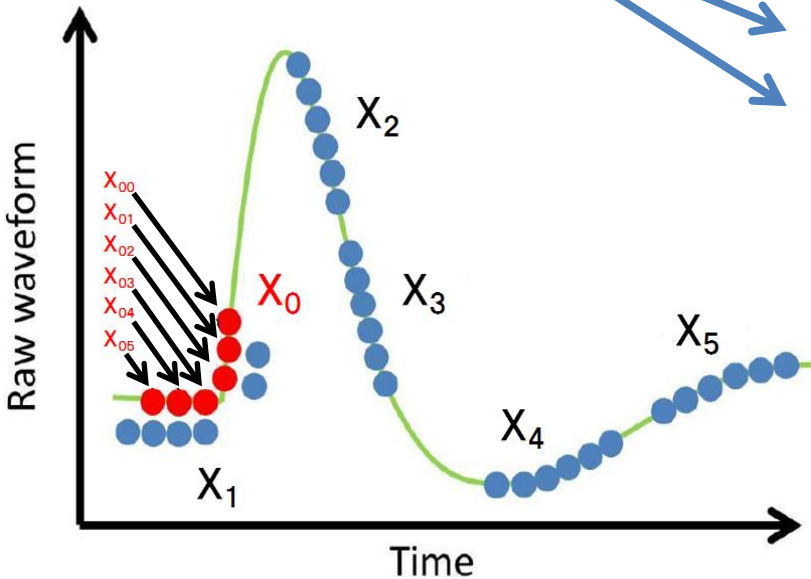
# フィルター係数の求め方

フィルター係数として  $h_0, h_1, h_2, h_3, h_4, h_5$  の6つを考える

パルスの立ち  
上がりなら1

それ以外なら0

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix} = \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} \end{pmatrix} \begin{pmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \end{pmatrix}$$



上の連立方程式を解いて、フィルター係数を求める

$$h_0 = -\frac{2}{16}, \quad h_1 = \frac{11}{16}, \quad h_2 = -\frac{16}{16},$$

$$h_3 = \frac{3}{16}, \quad h_4 = \frac{8}{16}, \quad h_5 = -\frac{4}{16}$$

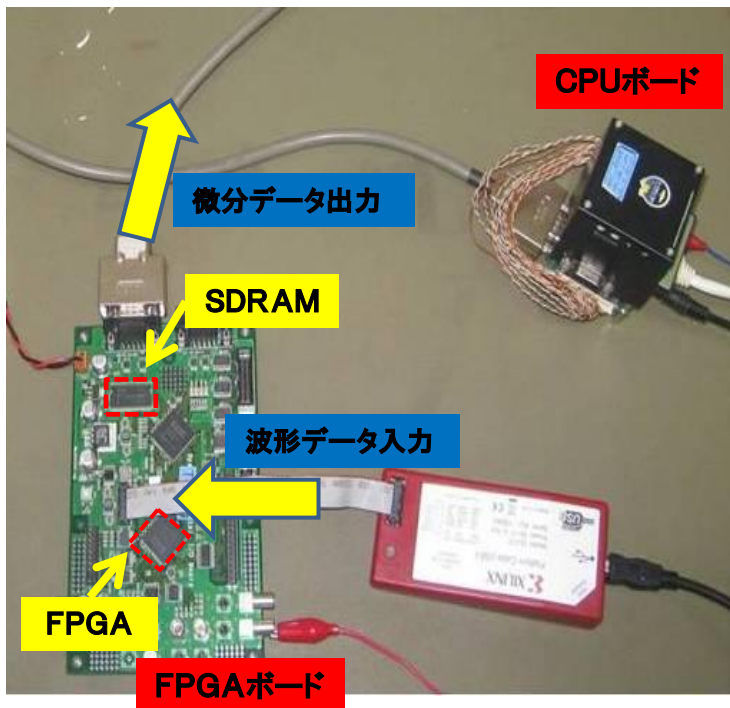
データ数は6(フィルター係数の数)×5(binまとめの数)で30とする

# FPGAロジックの開発・実装

実際に求めたフィルター係数を用いてVHDLコードを作成する

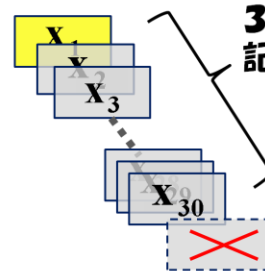


## セットアップ



波形入力

$X_1$  (入力データ)



30個分のデータ記憶領域を用意

$X_1$  が入ってくると一番古いデータが記憶領域から削除される

微分の計算

30個分のデータを微分計算に使用

VHDLコード

```

derivLong1 <= ((x1*h0)+(x2*h0)+(x3*h0)+(x4*h0)+(x5*h0));
derivLong2 <= ((x6*h1)+(x7*h1)+(x8*h1)+(x9*h1)+(x10*h1));
derivLong3 <= ((x11*h2)+(x12*h2)+(x13*h2)+(x14*h2)+(x15*h2));
derivLong4 <= ((x16*h3)+(x17*h3)+(x18*h3)+(x19*h3)+(x20*h3));
derivLong5 <= ((x21*h4)+(x22*h4)+(x23*h4)+(x24*h4)+(x25*h4));
derivLong6 <= ((x26*h5)+(x27*h5)+(x28*h5)+(x29*h5)+(x30*h5));
y <= derivLong1 + derivLong2 + derivLong3 + derivLong4 +
      derivLong5 + derivLong6;
    
```

微分結果

クリッピング

微分結果が以下の範囲に収まらない場合はその範囲に収まるようにクリッピングする

$$-32768 < y < 32767$$

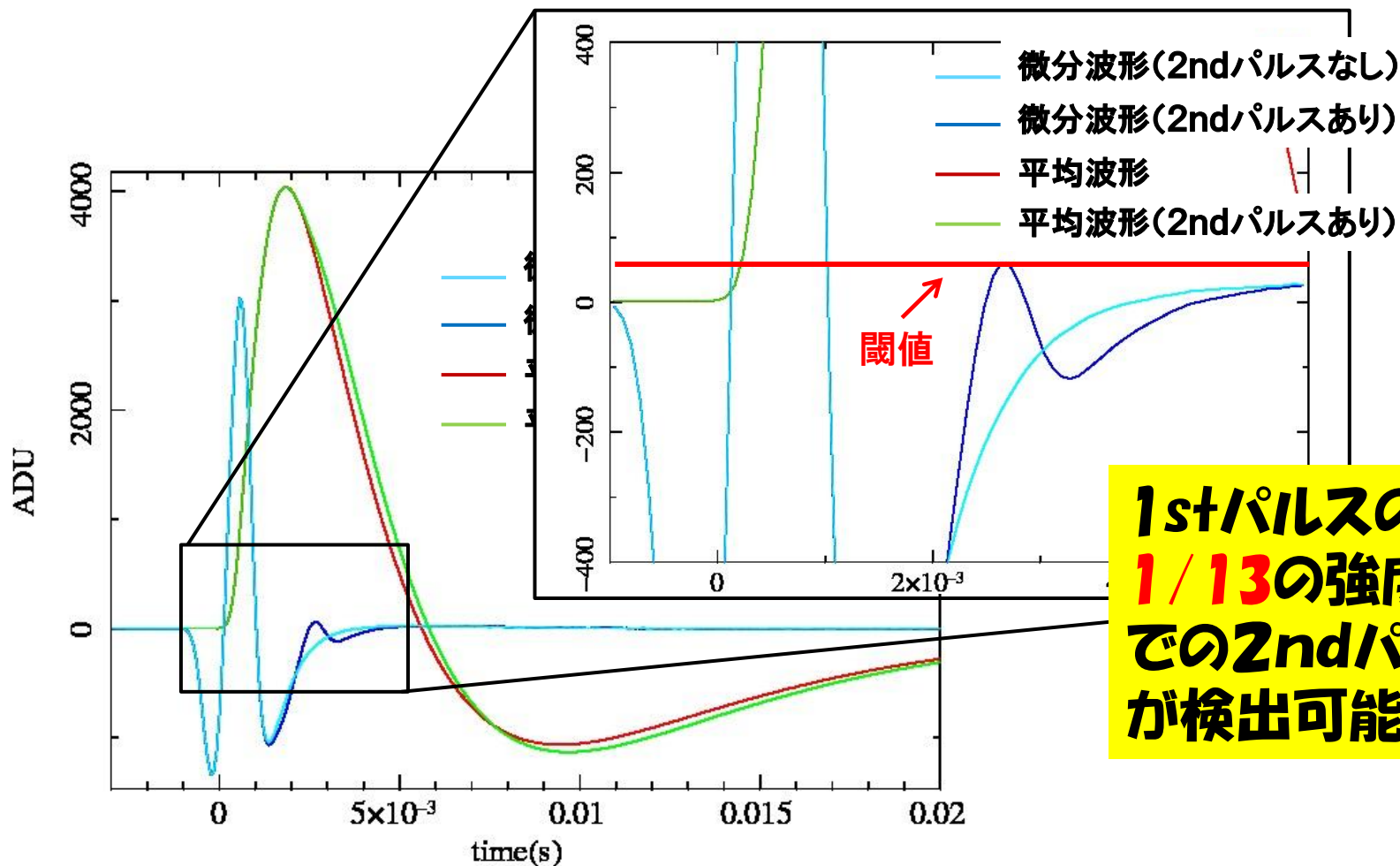
D10ボード上のSDRAMにFIR微分波形を保存

後段の処理へ

実装した微分ロジック

# 1stパルスの2ms後に2ndパルスがある波形を、実装したFPGA上のFIR微分フィルターに通すと、...

## 実装したFPGAで計算された微分波形



**1stパルスの  
1/13の強度比ま  
での2ndパルス  
が検出可能！！**

# まとめ

- FPGA上で、FIRフィルターを応用した微分フィルターの実装に成功した。
- rise と fallの時定数がそれぞれ 1 ms, 2.7 ms の半導体マイクロカロリメータの波形を用いて、1stパルスの 2 ms後に到来した**強度比1/13**の 2ndパルスの検出に成功した。これはBoxcarフィルターの同条件での限界である**強度比1/4**を大きく上回る性能である。