

小型科学衛星LiteBIRDの データ収集システムの開発(1)

岡山大,高工研^A

三澤尚典,石野宏和,服部香里

樹林敦子,美馬覚

松村知岳^A,羽澄昌史^A,

他 LiteBIRD ワーキンググループ

アウトライン

- 1.研究背景
- 2.実験セットアップ
- 3.圧縮アルゴリズム
- 4.まとめ

1.研究背景

データ圧縮の必要性

- 衛星上で取得したデータを地上まで送信する際、バンド幅の制限があるため、衛星上の生データを一度圧縮して地上まで送らなければならない。

検出器からのデータ : 7Mbps ~ 16bit × 2000ch × 200Hz

テレメトリで転送可能なデータ : 1 Mbps

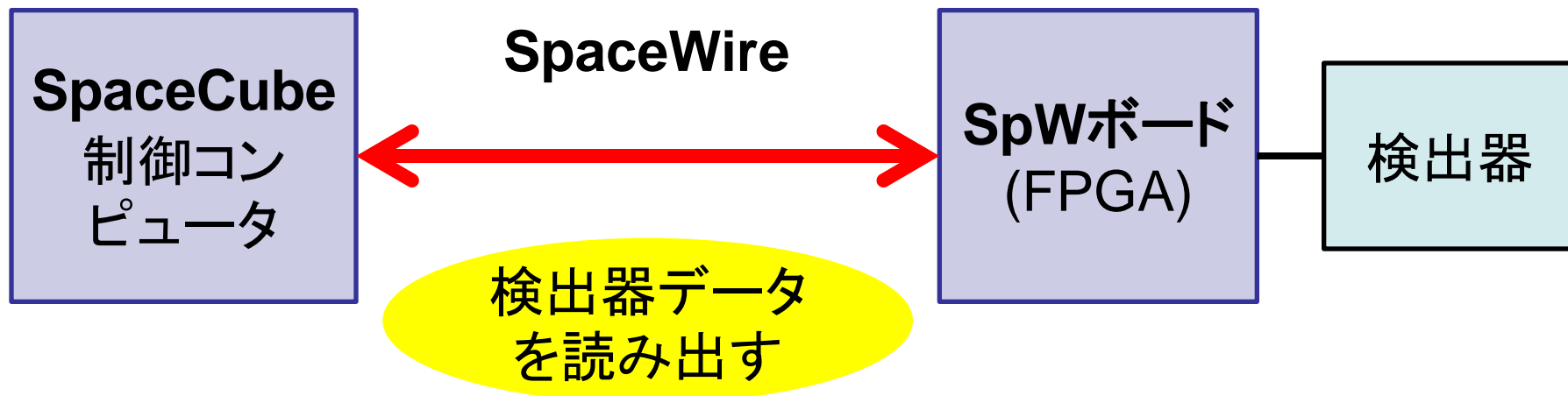
必要な圧縮率 : $1/7 = 14.3\%$

Space Wireを用いたDAQ

- 近年衛星バスシステムはSpaceWire (SpW)に統一されつつあり、この規格を使った読み出しシステムの構築をおこなう必要がある。

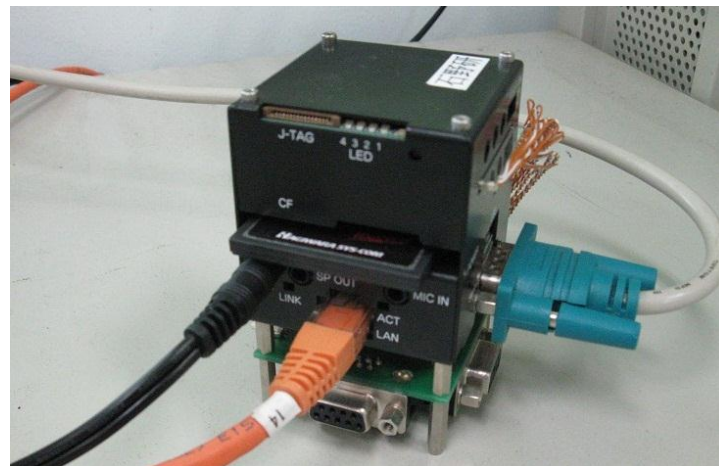
Space Wire

- 現在標準化が進められている衛星内通信規格。機器間の通信を共通化することで作業の効率化と簡素化ができる。
- SpaceCube 側から見れば、自身のメモリにアクセスするのと同じように、SpaceWire の接続先の機器(SpWボード上のメモリ)にアクセスできる。



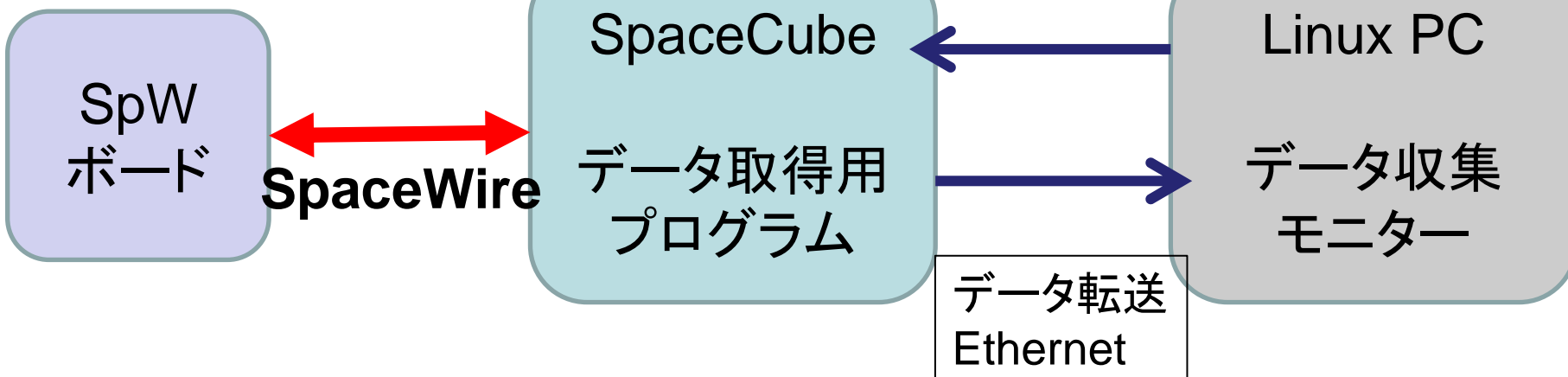
Space Cube

- シマフジ電機製のSpaceWireポートを実装した5cm角の小型コンピュータ。
- リアルタイムOS T-Kernel
- 地上試験および開発用。



SpWボード上のメモリに
アクセスしデータを読む

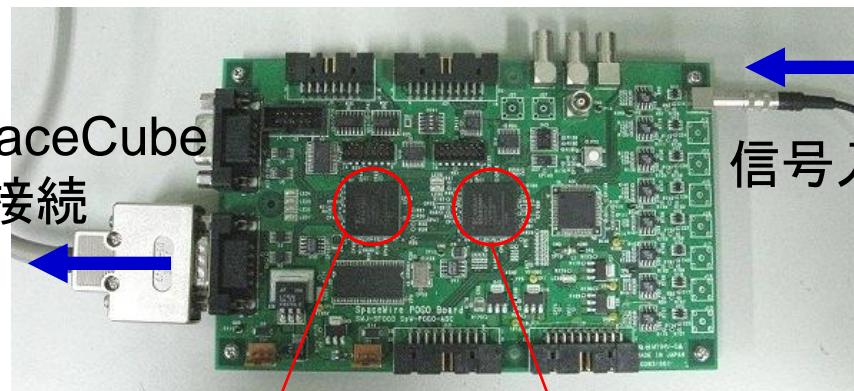
SpaceCube制御



2.実験セットアップ

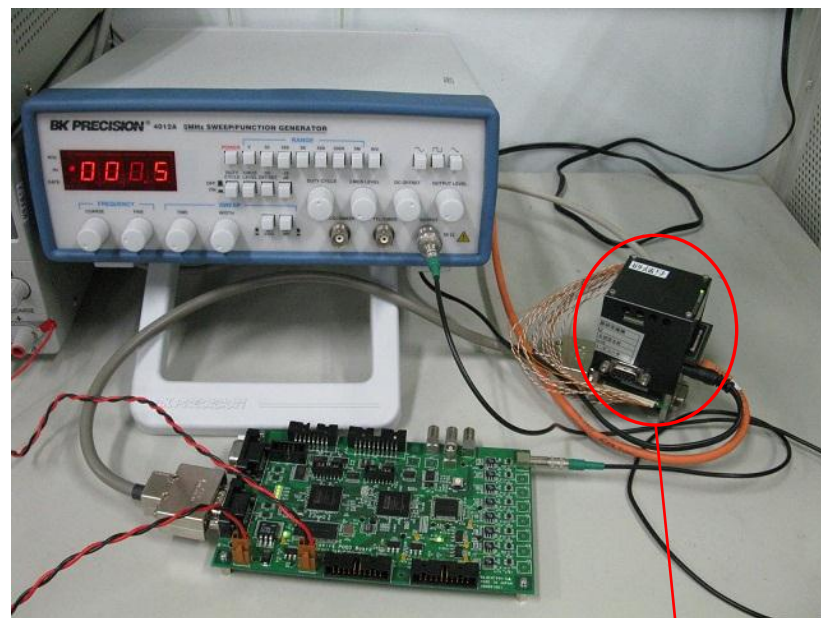
- VHDLでユーザーモジュールを記述し、SpWボード上のFPGAに書き込む。
- SpaceCube上でプログラム(C++)を実行する。

SpW Sampling ADCボード



SpW FPGA

User FPGA



Space Cube

3.圧縮アルゴリズム

- 実際の衛星では16bit ADCを使用予定。
- 今回使用したADCは12bit。これを1サンプルにつき4bitにするアルゴリズムをFPGAに実装した。

- (1) 200HzでTime Ordered Dataを取得し、ある個数ごとに
(今回は128回)平均値を求める。
- (2) (データ) - (平均値)を1サンプルごとに求める。
- (3) $\sigma=50$ のホワイトノイズを想定し、 $\sigma/2=25$ として、
データをbinningし、0~15までの数字(4bit) を割り当てる。
- (4) 平均値、 σ の値、0~15に変換されたTODを読み出す。

圧縮率

- 現在は12bit ADCのデータを128個ごとに圧縮する。
1サイクルで、4bitのデータ128点、平均値、 σ の値を読む。

$$\frac{\text{圧縮後のデータサイズ}}{\text{圧縮前のデータサイズ}} = \frac{4\text{bit} \times 128 + 16\text{bit} \times 2}{12\text{bit} \times 128} = 35.4\%$$

- 将来、16bit ADCを使用した場合でも同様に圧縮できる。

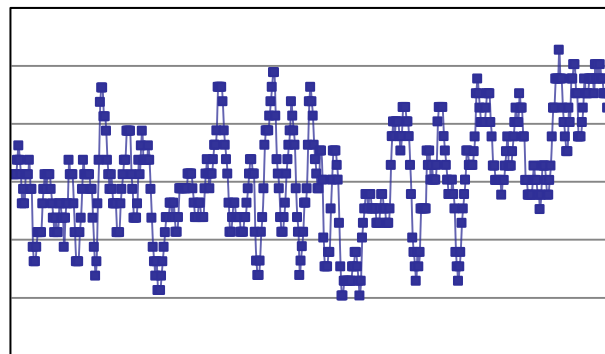
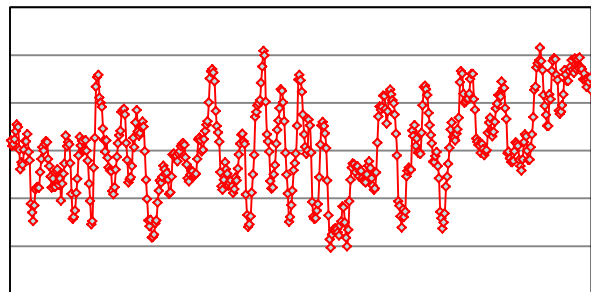
$$\frac{4\text{bit} \times 128 + 16\text{bit} \times 2}{16\text{bit} \times 128} = 26.6\%$$

アルゴリズムの検証

- アルゴリズムが動作していることを確かめるため、圧縮前後のTODを比較し、情報が保たれているかを確認する。

ADCを通さずに既知の信号をFPGA内から直接入力し、圧縮前後を比較し、再現性を確認する。

FPGA上の
メモリーに
TODを用意



TODを
200Hzで
入力

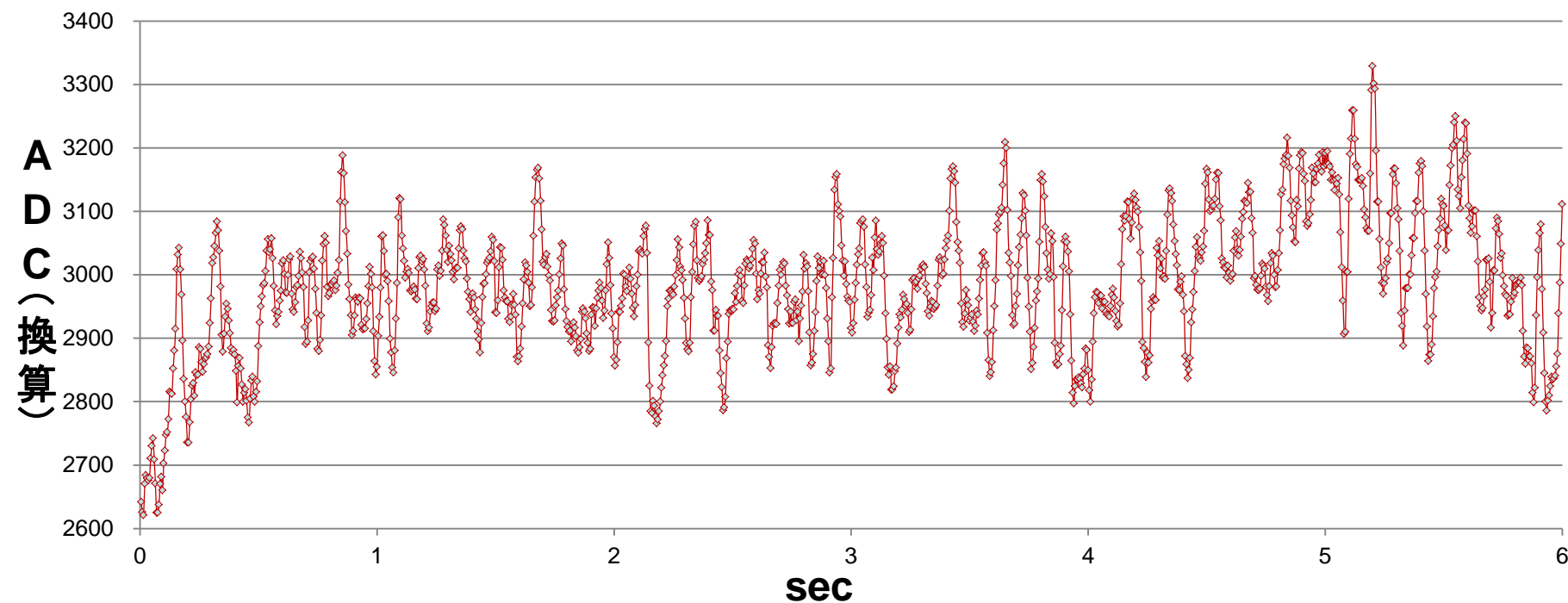
FPGA上で
圧縮アルゴリズムを
実行しTODを
0~15にbinning

SpaceCubeで
データを
読み出す

アルゴリズムの検証

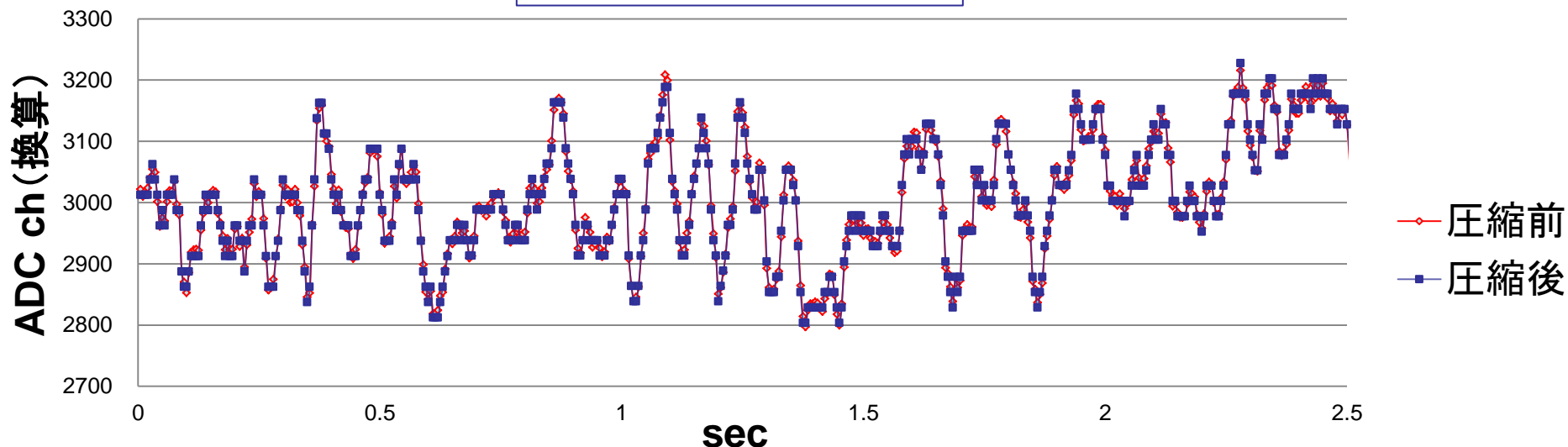
- 80GHzのCMBについてシミュレーションし、200Hzでサンプリングしたノイズがない場合のTODを使用した。
- データは12bitに収めるため、2700~+3300の程度の値を取るようにして、入力した。

CMB TOD

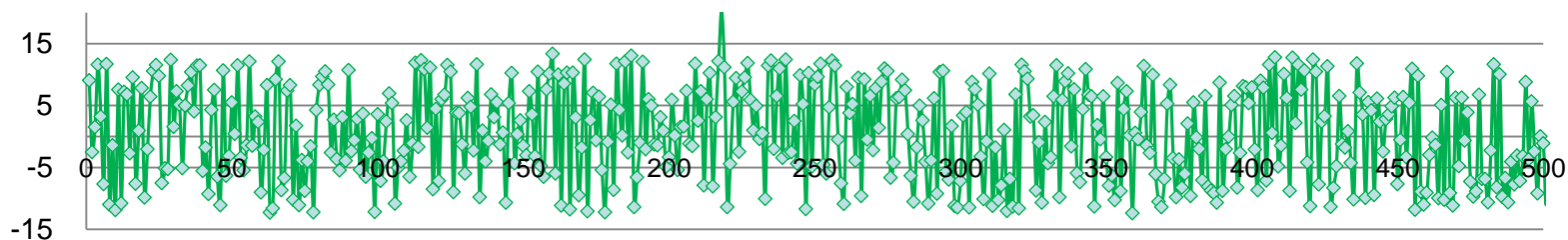


アルゴリズムの検証

圧縮前後のデータの比較



圧縮前後のデータの差



- 圧縮後のデータから圧縮前の概形が再現できている。
- 前後のデータの差はランダムで「 $\sigma/2=25$ 」以内に収まっている

量子化ノイズ

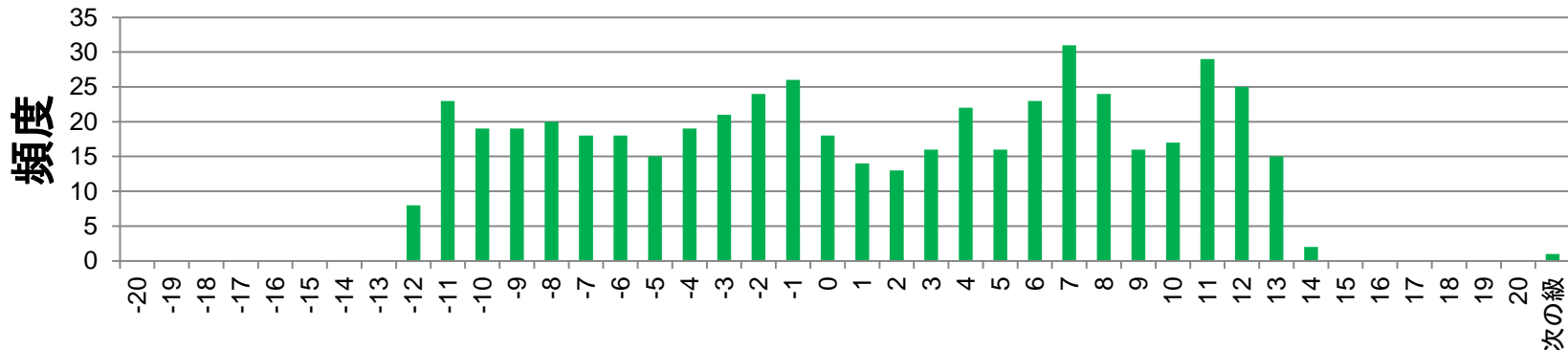
- $-25/2 \sim +25/2$ の区間で均一に分布する。

$$\sigma_q^2 = \int_{-a/2}^{a/2} x^2 \frac{1}{a} dx = \frac{a^2}{12} \quad a = 25 \quad \sigma_q = \frac{a}{\sqrt{12}} = \frac{25}{\sqrt{12}}$$

- 圧縮後のデータには量子化ノイズが含まれるが、ホワイトノイズの寄与に比べると小さく、圧縮前のデータが再現できている。

$$\sqrt{50^2 + (25/\sqrt{12})^2} = 50.5$$

圧縮前後の差についてのヒストグラム



4.まとめ

目的：データ圧縮のアルゴリズムをFPGA上に実装し、
SpaceWireを使ったDAQシステムを構築する。

進展

- 12bitのADCの場合は圧縮率35.4%を達成するアルゴリズムが実装できた。
- 圧縮前後でのデータの再現性が確認できた。もともとあるホワイトノイズ($\sigma=50$)にこの量子化ノイズが加わり全体で
 $\sqrt{50^2 + (25/\sqrt{12})^2} = 50.5$ のノイズとなるが寄与は小さく問題はない。

今後の課題

- 多チャンネル化をおこなう。(LiteBIRDでは約2000ch必要)
- 宇宙線陽子による突発的な信号と点源によるスパイクが含まれる場合の対処を実装する。

圧縮率

- 理論的に達成可能な圧縮率はシャノンエントロピー h を用いて表される。

$$h = \log_2 \left(\frac{\sqrt{2\pi e} \sigma_e}{\Delta} \right)$$

デジタル化の分解能 : $\Delta = 25$

ホワイトノイズ : $\sigma_e = 50$

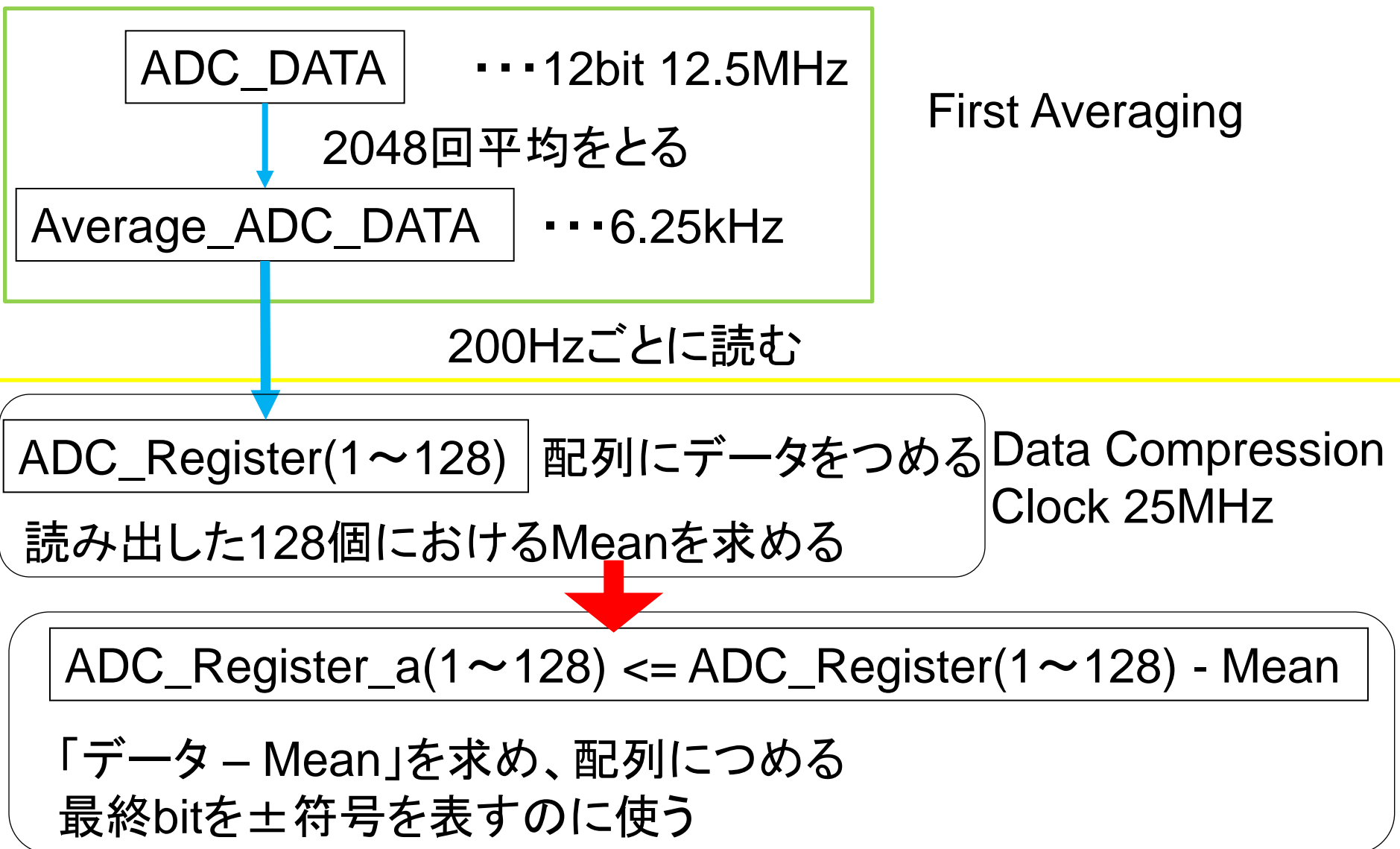
可能な圧縮率 = $\frac{h}{1 \text{ サンプルの圧縮前のデータサイズ}}$
--

12bit : 25.4%

16bit : 19.0%

- この圧縮率は圧縮アルゴリズムとして、Huffman 符号化や算術符号化を用いれば、近い値を達成することができる。

実装した圧縮アルゴリズム(詳細)

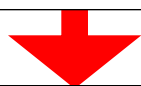


実装した圧縮アルゴリズム(詳細)

ADC_Register_a(1~128)について

2乗の和をとって $\sum \frac{(x-\mu)^2}{N}$ を計算し

さらに平方根を計算し $\sigma = \sqrt{\sum \frac{(x-\mu)^2}{N}}$ を求める



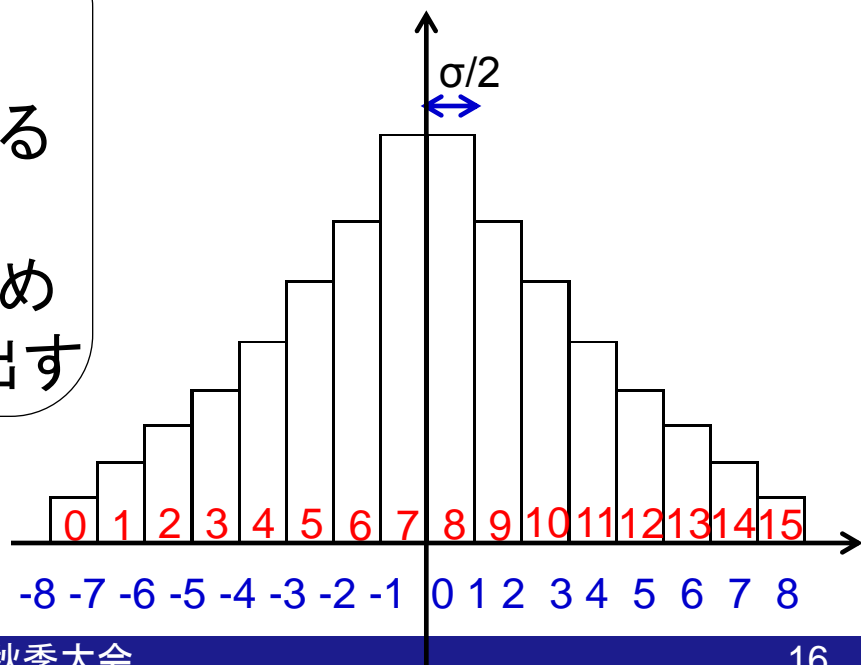
ADC_Register_a(1~128)について

Meanを中心として

$\sigma/2$ ごとにbinningし、0~15を割り当てる

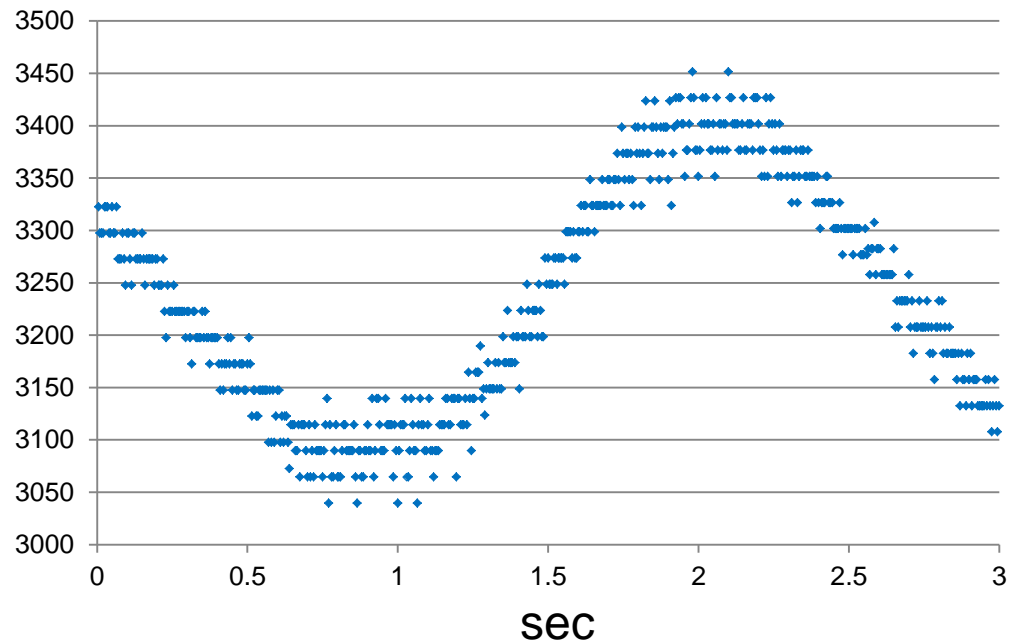
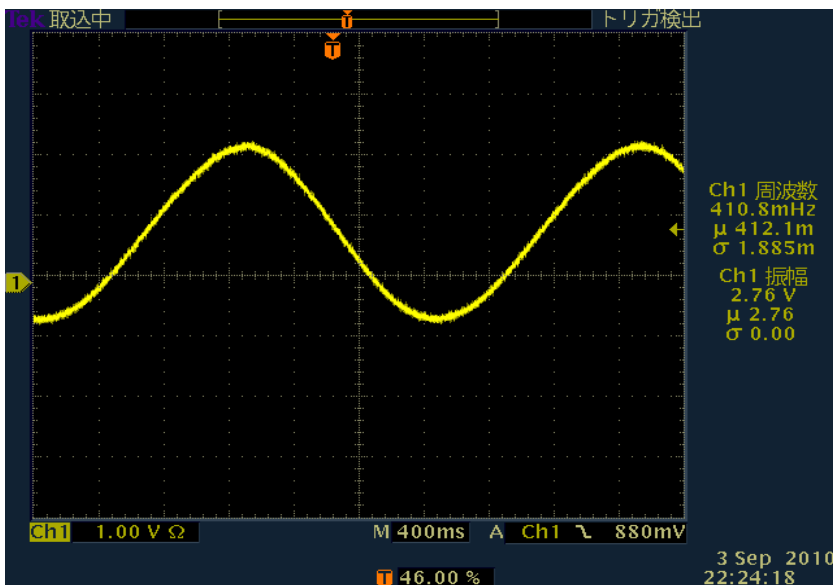
結果をADC_Register_f(1~128)につめ
Meanと σ とともにSpaceCubeで読み出す

200Hzでサンプリング
データ圧縮可能



アルゴリズムの検証(2)

- ファンクションジェネレータから信号を入力し、ADCからのデータを圧縮し、入力前の信号の比較する。
- ファンクションジェネレーターより0.4HzのSin波を入力。



- 圧縮後のデータから圧縮前の波形が再現できている。

アルゴリズムについて(補足)

- ◆ 今回のアルゴリズムではTOD128個ごとにbinningをおこなったが、2のn乗でさえあればよい。(割り算がしやすいため)

個数	圧縮率(12bit)	圧縮率(16bit)
256	34.4%	25.8%
128	35.4%	26.6%
64	37.7%	28.1%
32	41.7%	31.2%

- ◆ このアルゴリズムのメリットは、最初から粗い精度の4bit ADCで測定するのと違い、CMBのTODは変化が少ないため、平均の差分をbinningすることで精度が確保できる点。