

第13回日本加速器学会年会

**高精度時刻同期技術をベース  
としたトリガ・タグ情報配信  
システムの開発**

高輝度光科学研究センター  
増田 剛正



---

# Outline

---

- ❖ 開発の目的
- ❖ 検証システム(Version1)の構築
  - ❖ 評価試験
- ❖ システムの改良(Version2)
  - ❖ 評価試験
- ❖ まとめ
- ❖ 今後の予定



---

# 開発の目的

---

- ❖ 汎用ネットワークを介した高精度時刻同期技術を用いて、トリガ・タグ情報の配信システムを構築する



---

# 開発の目的

---

- ❖ 汎用ネットワークを介した高精度時刻同期技術を用いて、トリガ・タグ情報の配信システムを構築する
- ❖ 従来のアナログ型の信号配信
  - ❖ 信号の分周や遅延のための各種回路が必要
  - ❖ タイミングの調整に手間が掛かる
  - ❖ ショット番号などの付加情報は別途用意
  - ❖ 計算機との親和性が低い



# 開発の目的

- ❖ 汎用ネットワークを介した高精度時刻同期技術を用いて、トリガ・タグ情報の配信システムを構築する
- ❖ 従来のアナログ型の信号配信→デジタル型のタイミング情報の配信へ
  - ❖ 扱いやすい、拡張が容易
  - ❖ タイミングの調整がソフトウェアにより簡便に行える
  - ❖ 入射番号などのタグ情報を容易に付加出来る



# 開発の目的

- ❖ 汎用ネットワークを介した高精度時刻同期技術を用いて、トリガ・タグ情報の配信システムを構築する
- ❖ 検証システムの開発
  - ❖ SPring-8蓄積リングの特定バンチに同期した周回周波数信号（～209kHz）の生成と、ショット番号などのタグ情報の配信
  - ❖ 構築した新システムによって生成される周回周波数信号のジッタ等を計測



---

# 開発の目的

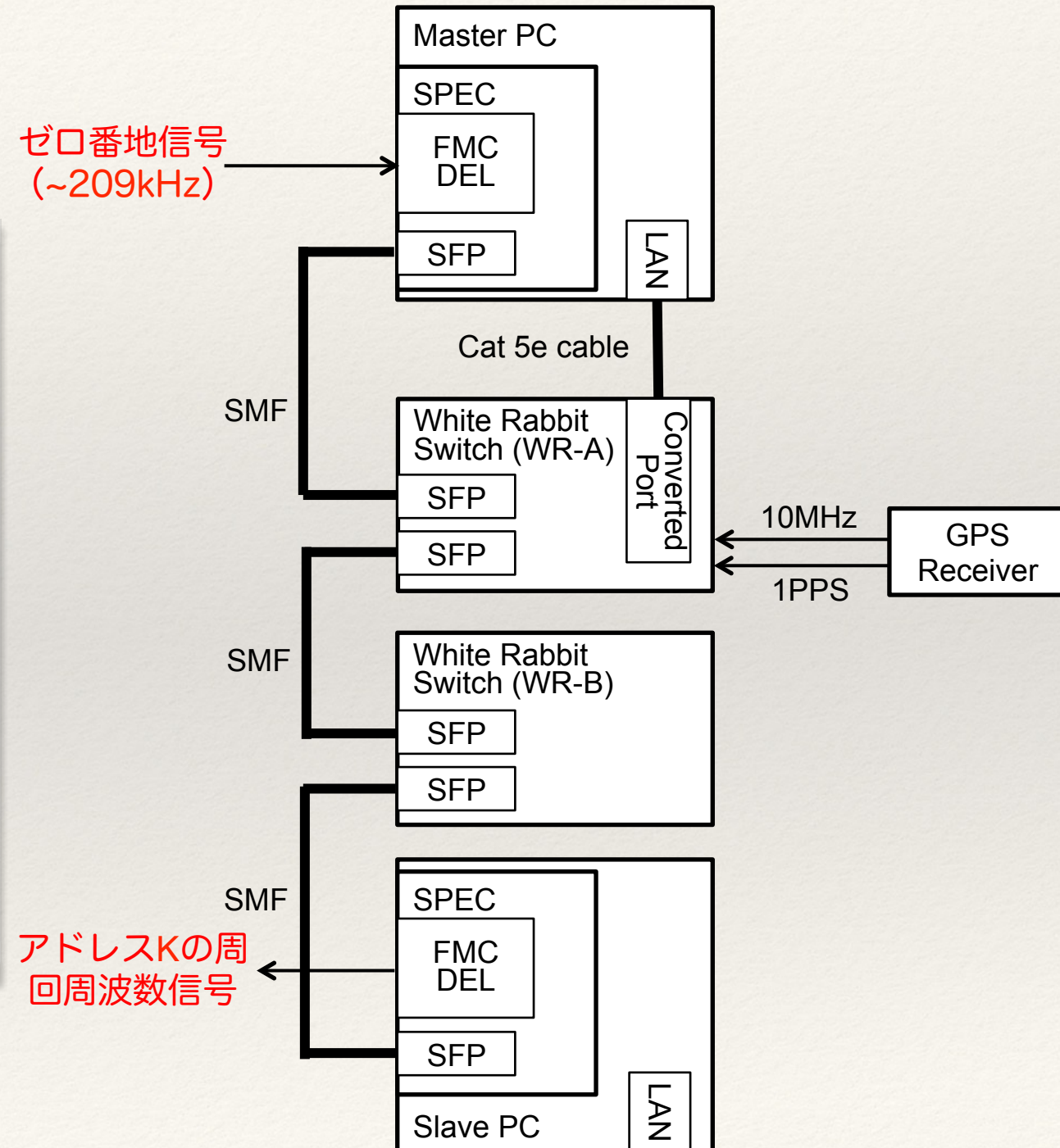
---

- ❖ 汎用ネットワークを介した高精度時刻同期技術を用いて、トリガ・タグ情報の配信システムを構築する
- ❖ 高精度時刻同期技術としてWhite Rabbitを採用
  - ❖ CERNを中心に研究が進められている汎用ネットワークを用いたサブナノ秒以下の高精度時刻同期+リアルタイム通信技術（IEEE1588の改良版）



# 検証システム (Version 1) の構築

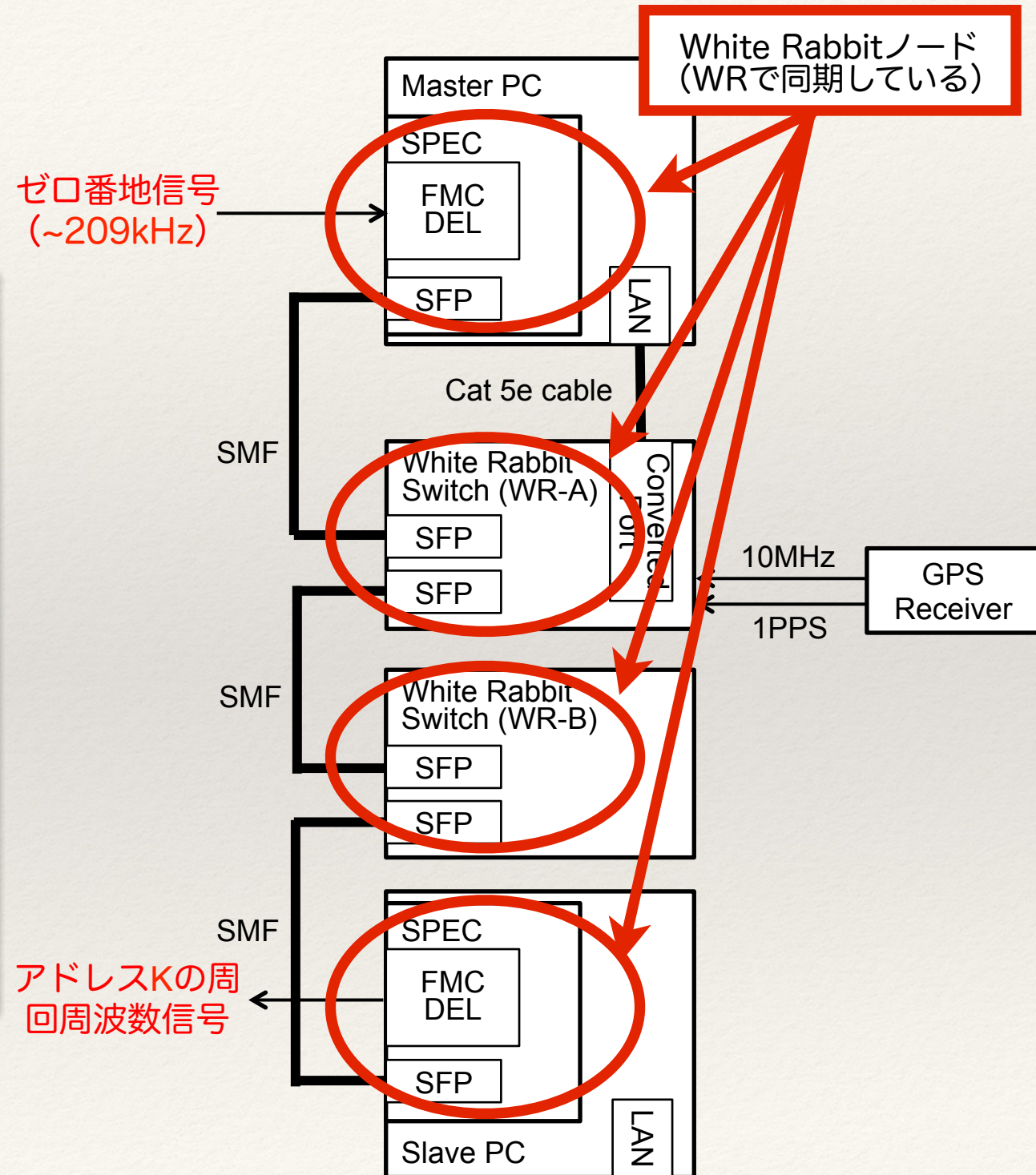
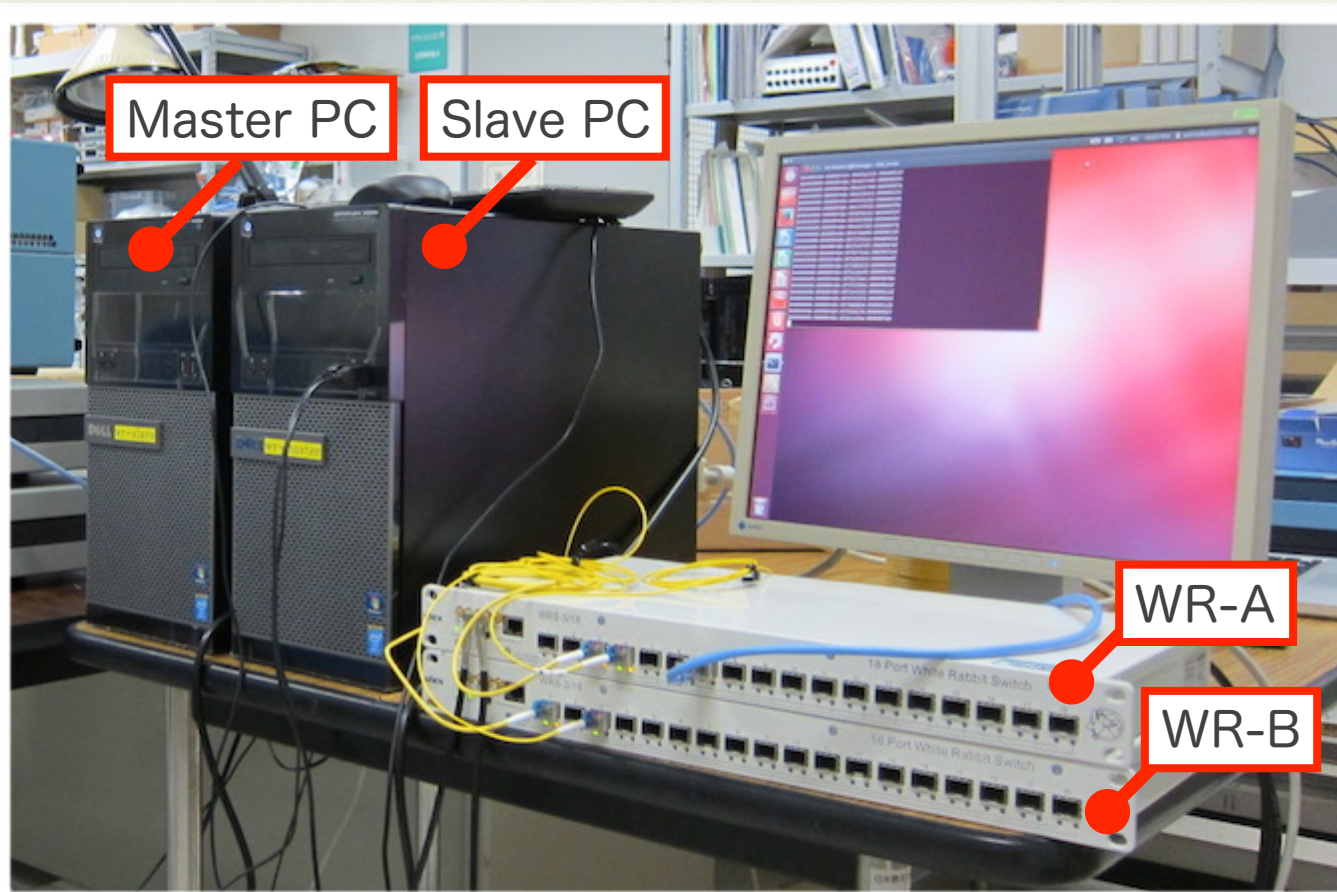
- ❖ 右図のような最小構成の検証システムを構築





# 検証システム (Version 1) の構築

- ❖ 右図のような最小構成の検証システムを構築





# 検証システム (Version 1) の構築

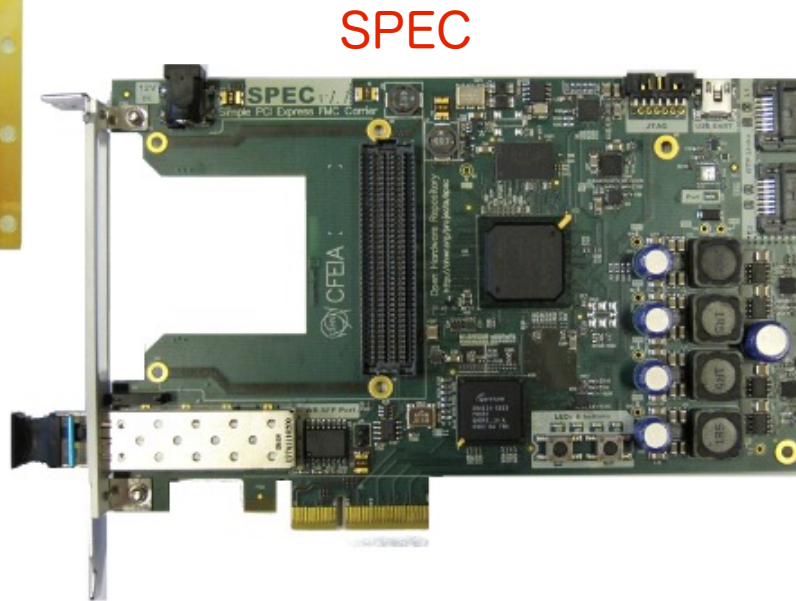
- ❖ 右図のような最小構成の検証システムを構築

## PCI Express FPGAボード (SPEC) + Fine Delay FMC

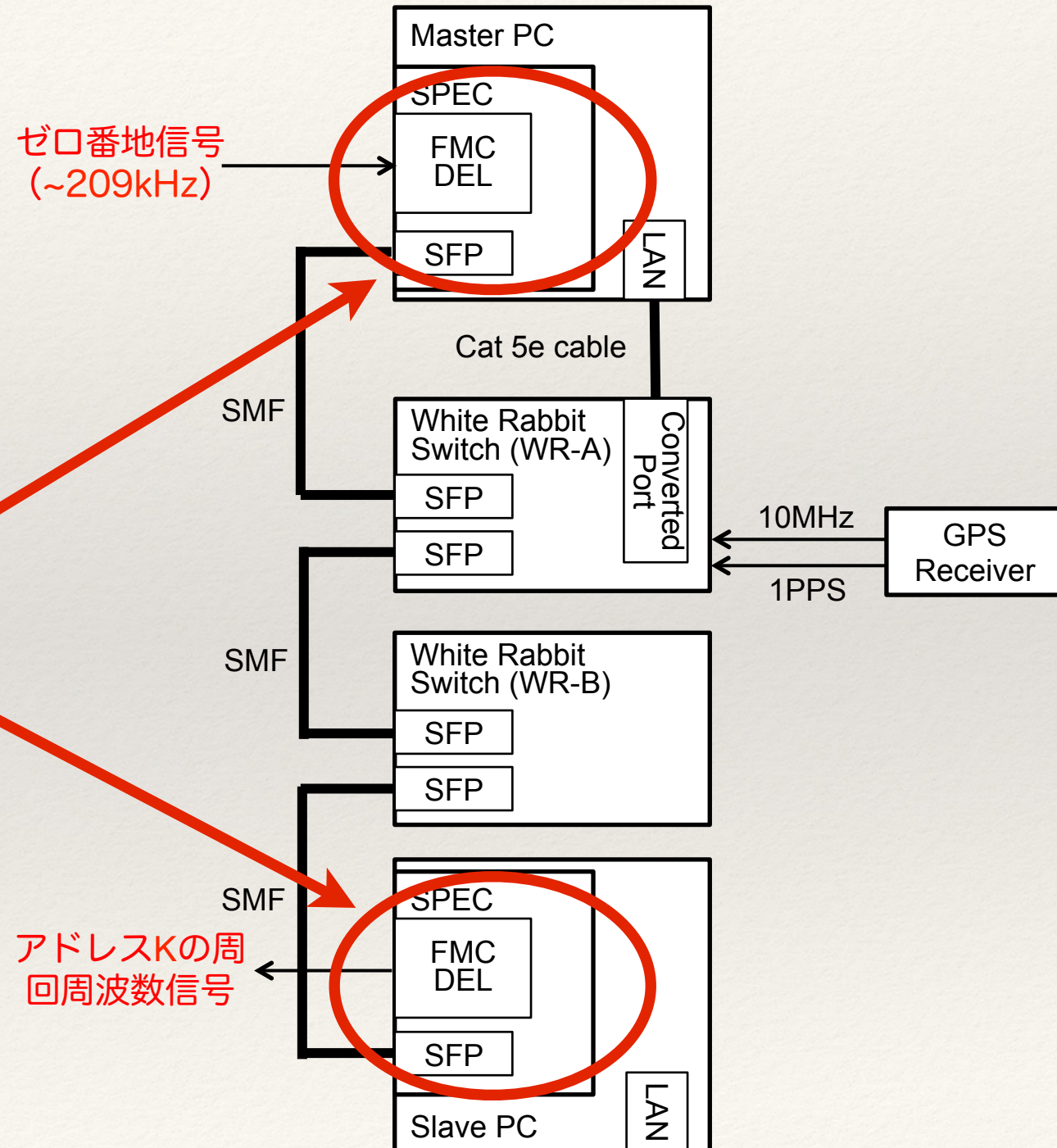
共にCERNを中心に進められているOpen Hardware Repository (OHR)で開発、自由に情報を入手可能。



Fine Delay FMC



SPEC





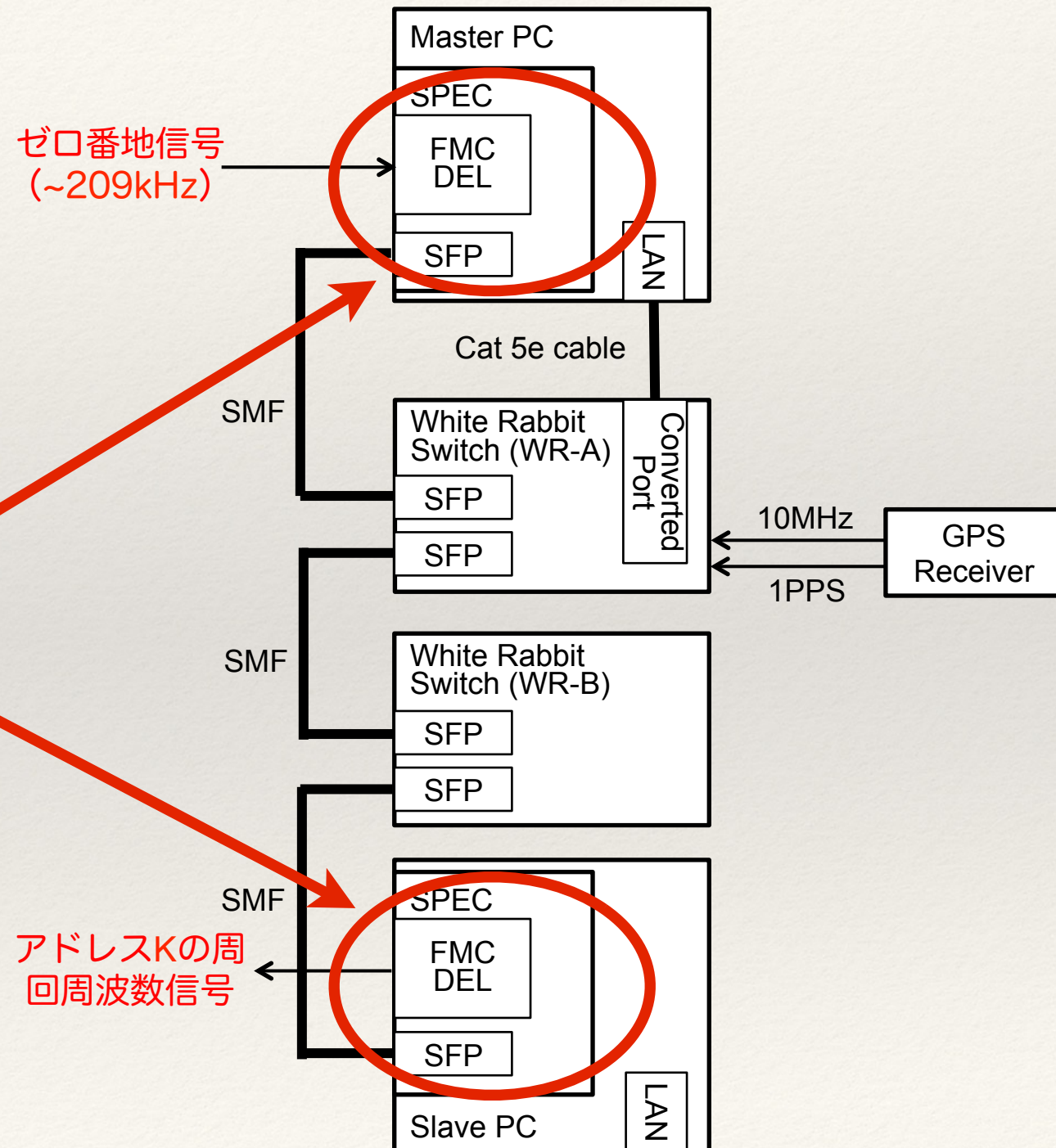
# 検証システム (Version 1) の構築

- ❖ 右図のような最小構成の検証システムを構築

## PCI Express FPGAボード (SPEC) + Fine Delay FMC

共にCERNを中心に進められているOpen Hardware Repository (OHR)で開発、自由に情報を入手可能。

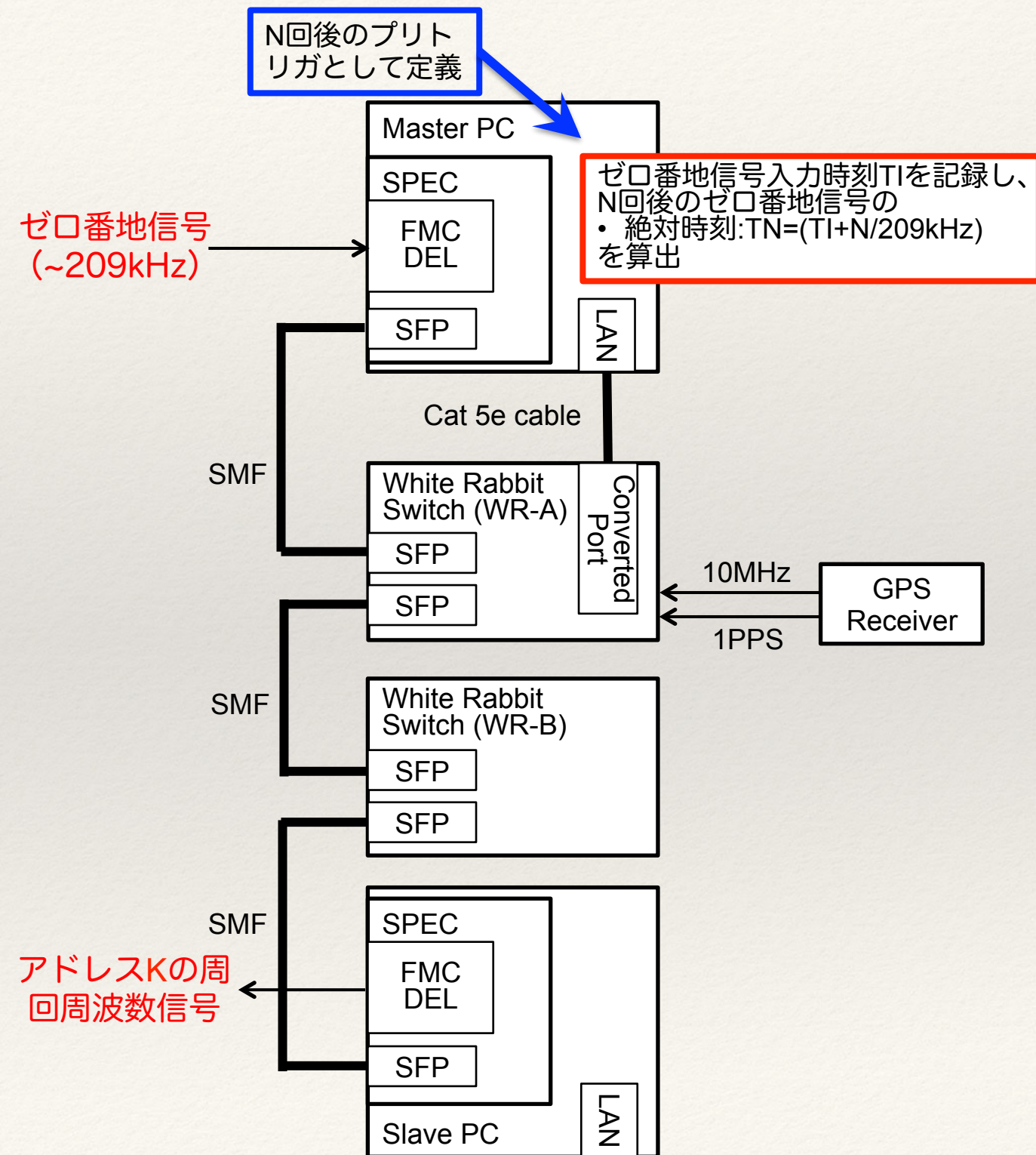
- トリガ入力チャンネル：1
  - 信号レベル：TTL/LVTTL
  - バンド幅：1MHz
- 出力チャンネル：4
  - 信号レベル：TTL互換
  - 立ち上がり/立ち下がり時間：2.5ns
- 3つの動作モード
  - Pulse Delayモード
    - 入力トリガから指定された時間後に一つまたは複数のパルスが発生。  
10ps分解能で600ns~12sまで設定可能。
  - Pulse Generatorモード
    - 与えられた絶対時刻に一つまたは複数のパルス（1~65536個）を発生。分解能10ps。
  - Time-to-Digital Converterモード
    - 全入力トリガにタイムスタンプをつけてユーザアプリケーションに分配する





# 検証システム (Version 1) の構築

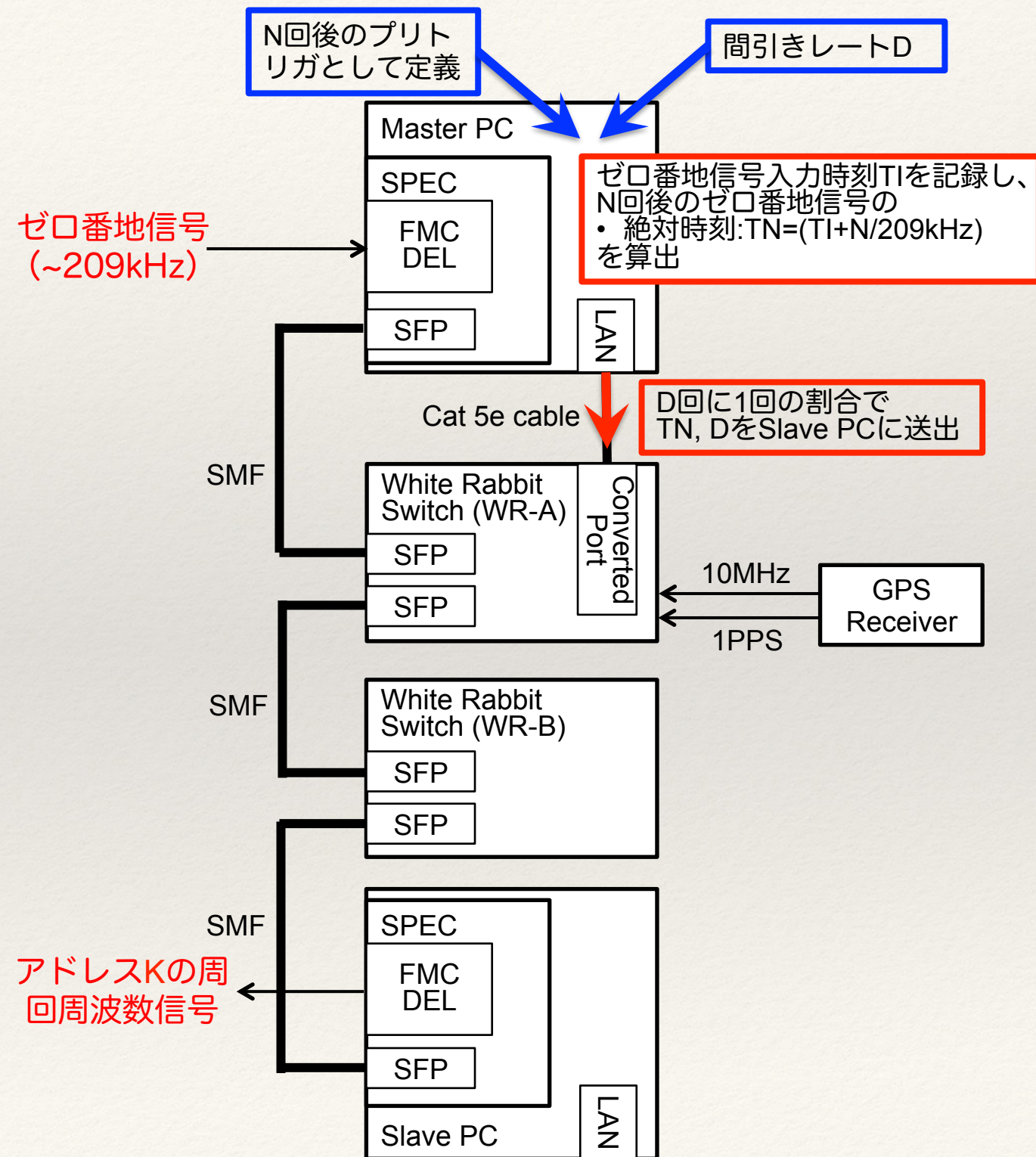
- ❖ 右図のような最小構成の検証システムを構築
  - ❖ チューナブルパラメータ
    - ❖ N (N周後にトリガ出力)





# 検証システム (Version 1) の構築

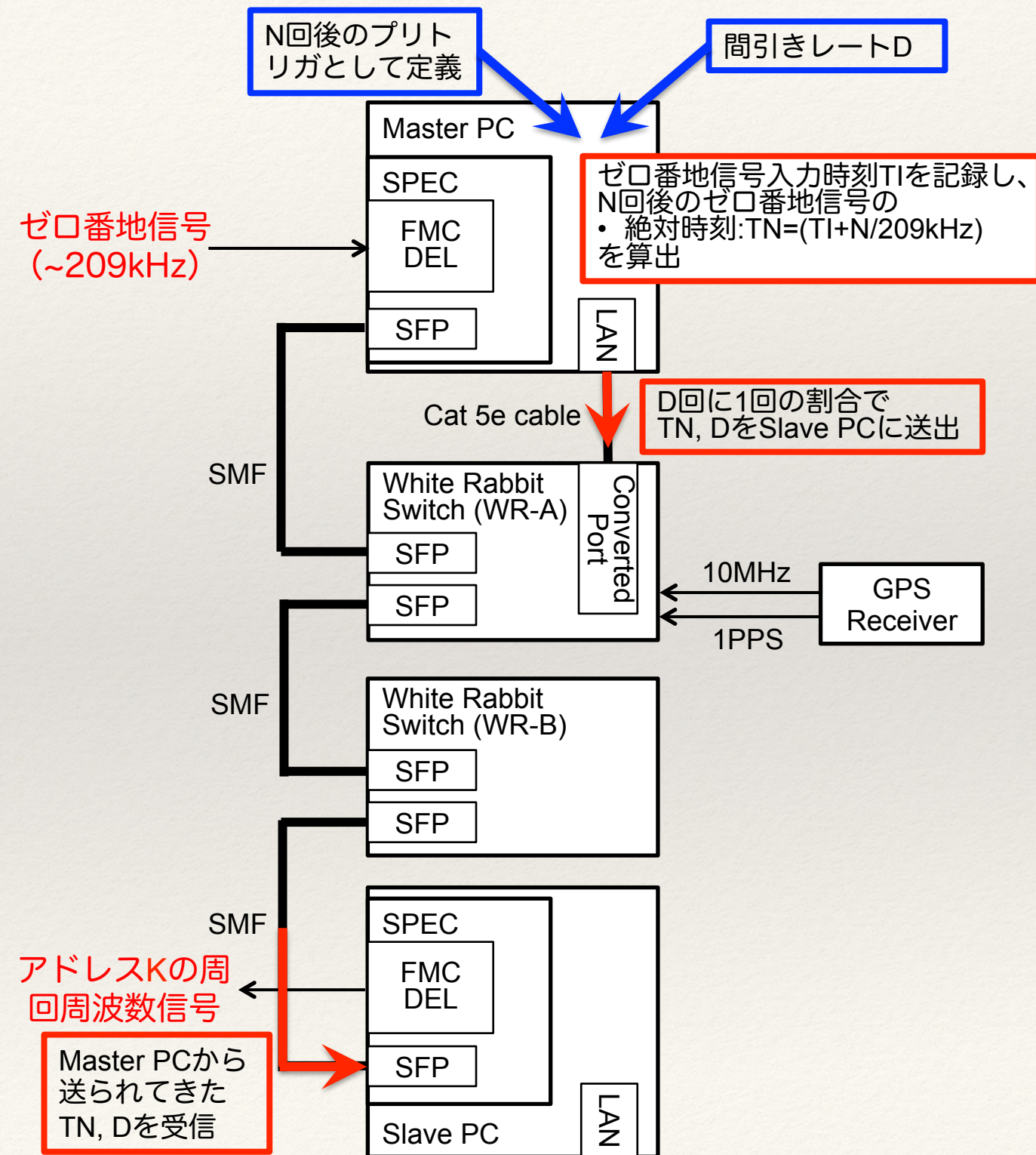
- ❖ 右図のような最小構成の検証システムを構築
- ❖ チューナブルパラメータ
  - ❖ N (N周後にトリガ出力)
  - ❖ D (間引レート)





# 検証システム (Version 1) の構築

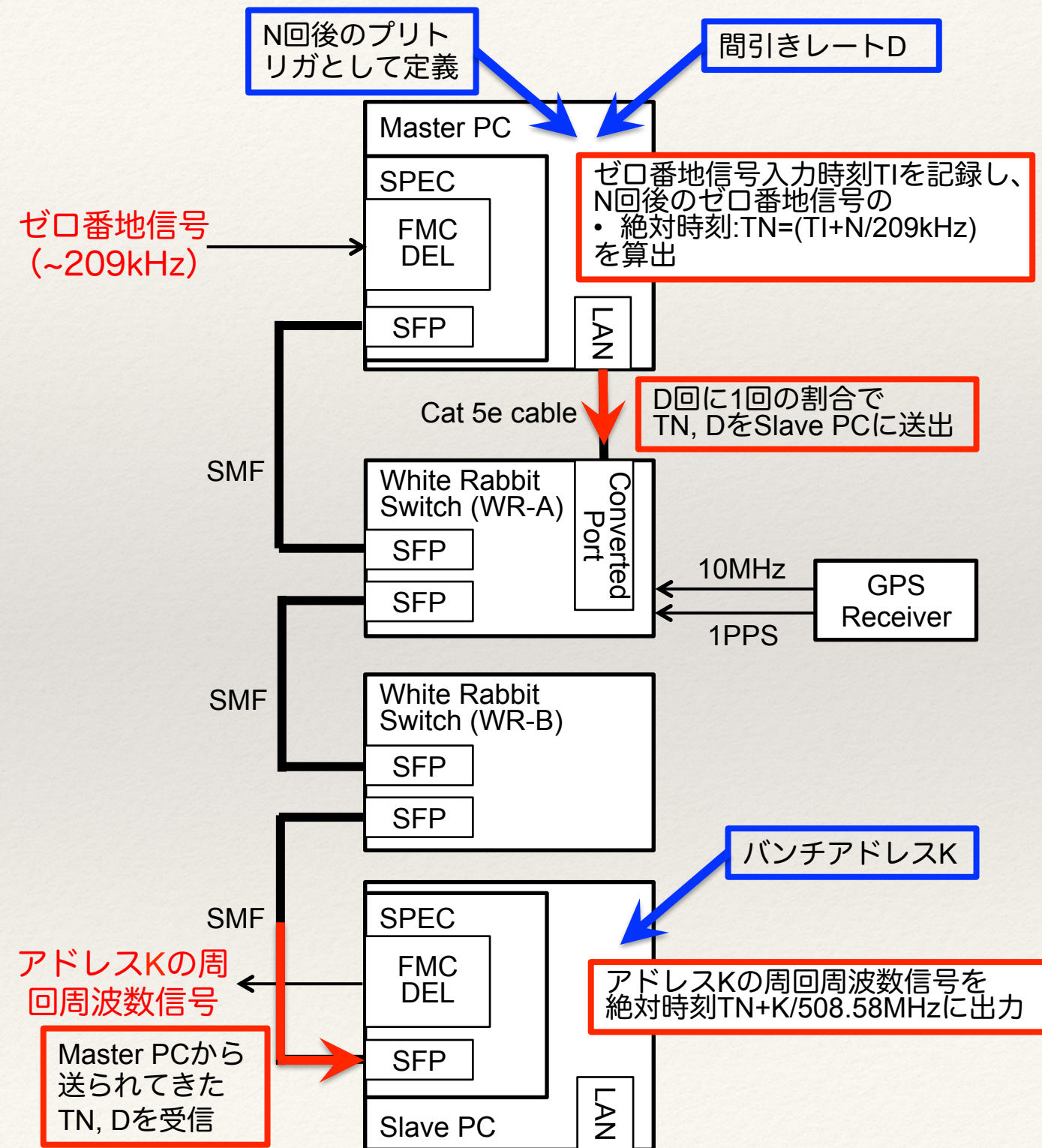
- ❖ 右図のような最小構成の検証システムを構築
- ❖ チューナブルパラメータ
  - ❖ N (N周後にトリガ出力)
  - ❖ D (間引レート)





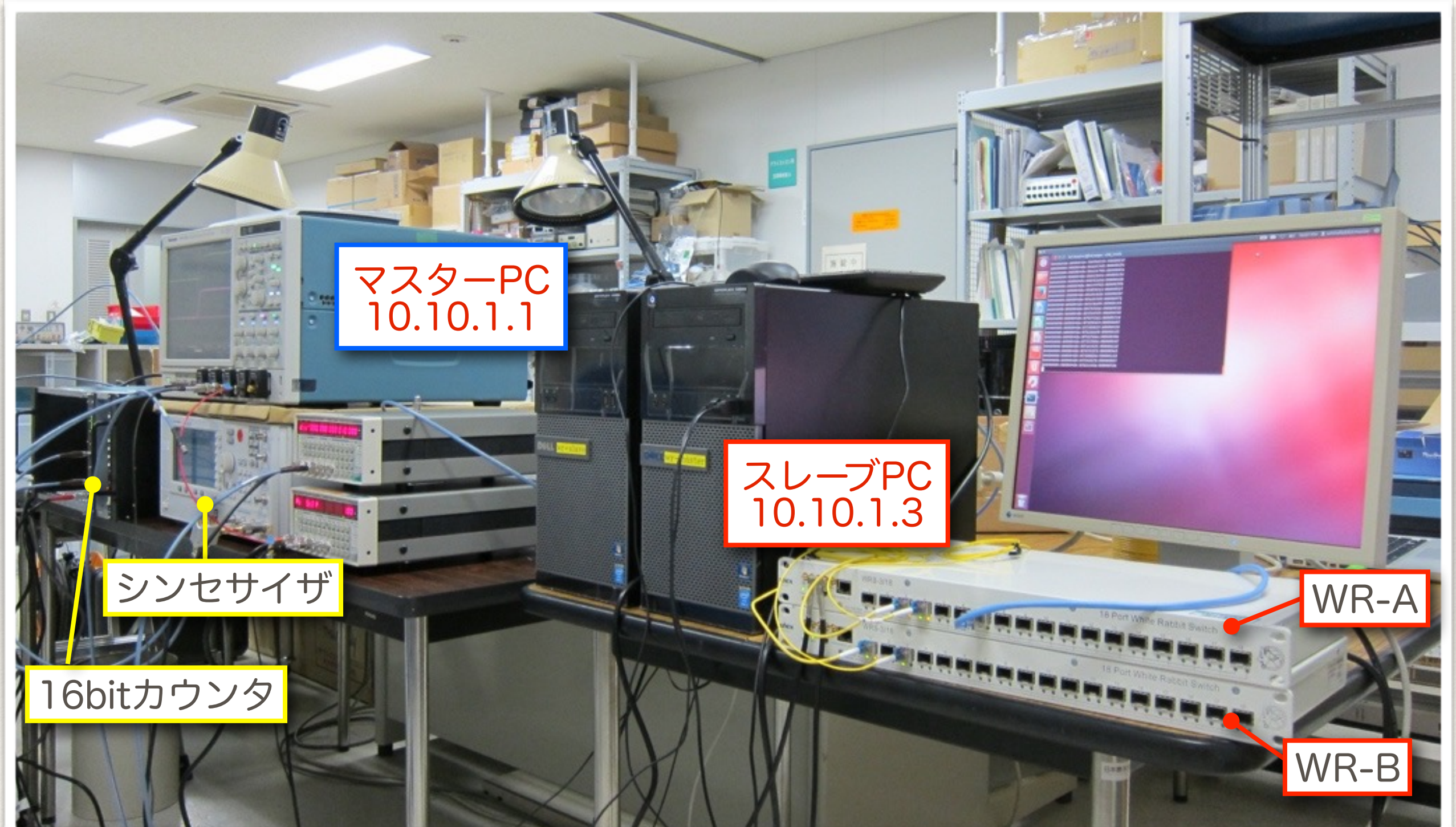
# 検証システム (Version 1) の構築

- ❖ 右図のような最小構成の検証システムを構築
  - ❖ チューナブルパラメータ
    - ❖ N (N周後にトリガ出力)
    - ❖ D (間引レート)
    - ❖ K (ターゲットバンチアドレス)
  - ❖ 出力周波数(1.95ps単位)
  - ❖ パルス幅 (160ns単位)





# 評価試験



マスターPC  
10.10.1.1

スレーブPC  
10.10.1.3

シンセサイザ

16bitカウンタ

WR-A

WR-B



# 評価試験

マスターPC上で以下のコマンドを実行する

```
sudo ./s8_init_master -n N -d D
```

N: N周回後にトリガ出力, D:間引きレート

```
sudo ./s8_master_on -s 10.10.1.3
```

-sで指定されたアドレスにトリガー情報を送る

マスターPC  
10.10.1.1

スレーブPC  
10.10.1.3

WR-A

WR-B





# 評価試験

マスターPC上で以下のコマンドを実行する

```
sudo ./s8_init_master -n N -d D
```

N: N周回後にトリガ出力, D:間引きレート

```
sudo ./s8_master_on -s 10.10.1.3
```

-sで指定されたアドレスにトリガー情報を送る

マスターPC  
10.10.1.1

スレーブPC  
10.10.1.3

スレーブPC上で以下のコマンドを実行する

```
sudo ./s8_init_slave -k K -f F -w pulse-width
```

K: ターゲットバンチアドレス, F: fine delay( $F \cdot 1.95\text{ps}$ )

```
sudo ./s8_slave_on -c 1
```

Fine delay FMCのCH1より出力する

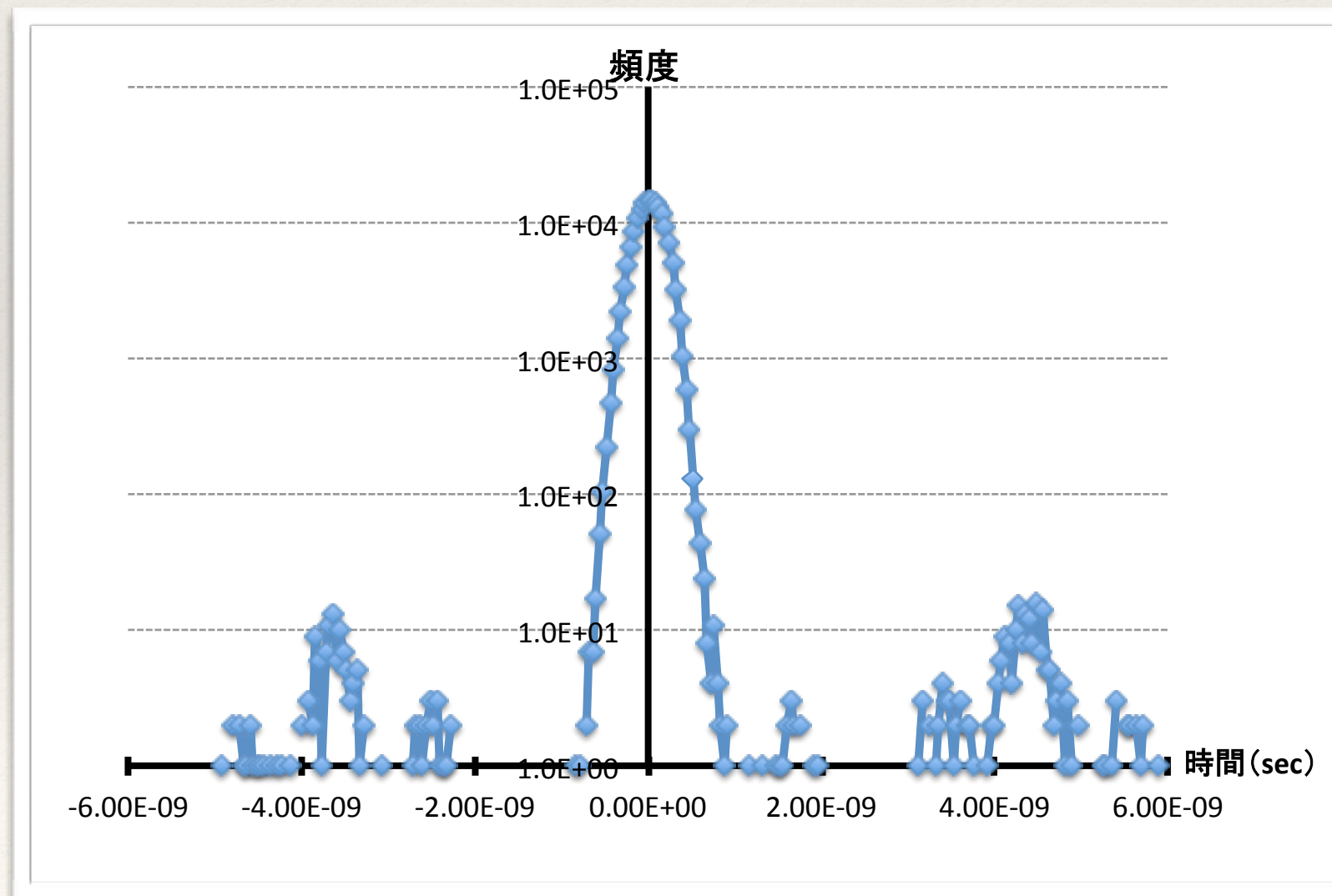
WR-A

WR-B



# 出力信号のジッタ計測

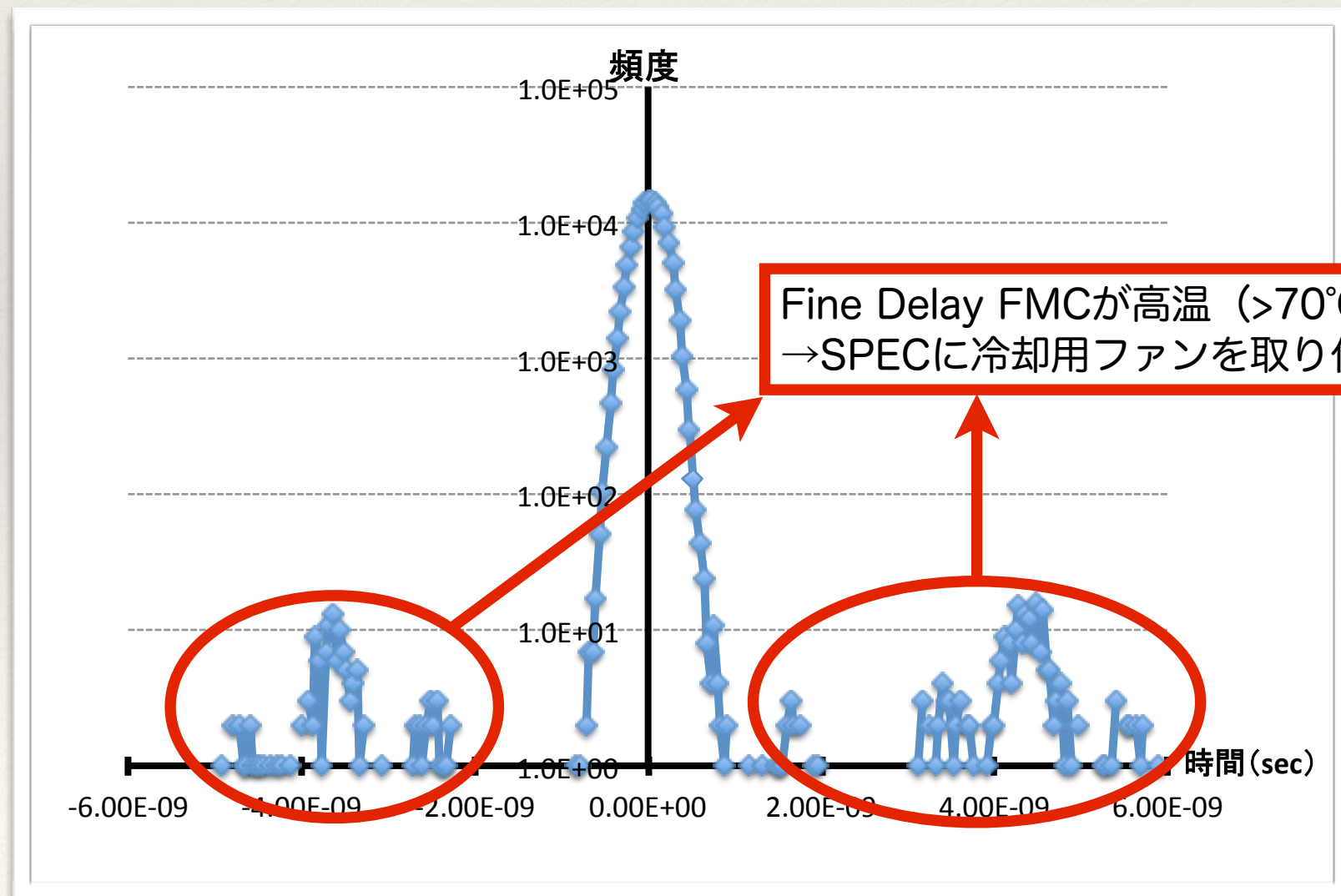
- ❖  $D=1000, N=900$ でジッタ計測
  - ❖ スレーブPCからの209kHz出力信号のジッタが最小となるように出力周波数を微調整





# 出力信号のジッタ計測

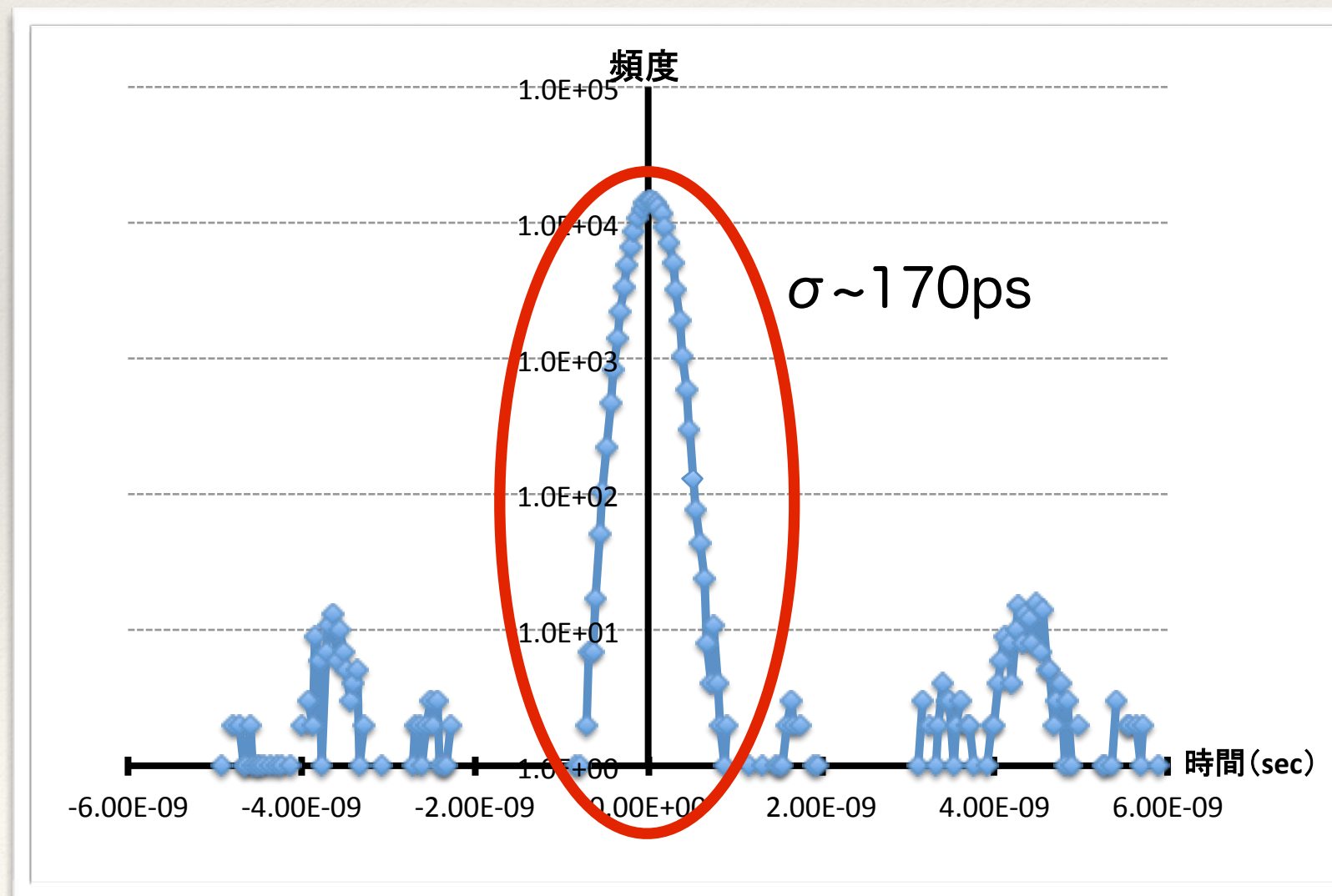
- ❖  $D=1000, N=900$ でジッタ計測
  - ❖ スレーブPCからの209kHz出力信号のジッタが最小となるように出力周波数を微調整





# 出力信号のジッタ計測

- ❖  $D=1000, N=900$ でジッタ計測
  - ❖ スレーブPCからの209kHz出力信号のジッタが最小となるように出力周波数を微調整





# ジッタの間引きレートD依存性

## ❖ 間引きレートDを変えてジッタを計測

- ❖ Version1では $D > N$ という条件が付いているため、あまりDを小さくできない

	標準偏差 (全体)	標準偏差 (中央部)	サンプル数
D=1000, N=900	254ps	171ps	163,233
D=600, N=500	221ps	139ps	68,321
D=300, N=200	201ps	131ps	71,249



# WR Switch間SMFの長さ依存性

- ❖ WR Switch間のSMFの長さを1 mから100mに伸ばして計測
  - ❖ 期待通り、SMFの長さには影響を受けないことが分かった。

条件	平均 (ns)	最小値 (ns)	最大値 (ns)	標準偏差 (ns)	サンプル数
D=600,N=500, K=0, F=0, 出力周波数微調 (-23.4ps)					
SMF 1m	4.75	-2.34	10.23	0.45	118.8k
SMF 100m	4.68	-2.11	10.86	0.47	600.8k



---

# システムの改良 (Version 2)

---

❖ Version1の以下の課題の解消を目指した

1.スレーブPCからの出力が時々抜ける

2.D (間引きレート)  $> N$  (N周回後にトリガ出力) という  
制約がある



# システムの改良 (Version 2)

❖ Version1の以下の課題の解消を目指した

## 1. スレーブPCからの出力が時々抜ける

- ❖ マスターPCからの出力時刻 $T_N$ と間引きレート $D$ の送出手間がソフトウェアによって行われていることに起因
- ❖ FPGAロジックによって送出手間を行うようVersion2で実装

2.  $D$  (間引きレート)  $> N$  ( $N$ 周回後にトリガ出力) という制約がある



# システムの改良 (Version 2)

❖ Version1の以下の課題の解消を目指した

## 1.スレーブPCからの出力が時々抜ける

- ❖ マスターPCからの出力時刻TNと間引きレートDの送出手間によって行われていることに起因
- ❖ FPGAロジックによって送出を行うようVersion2で実装

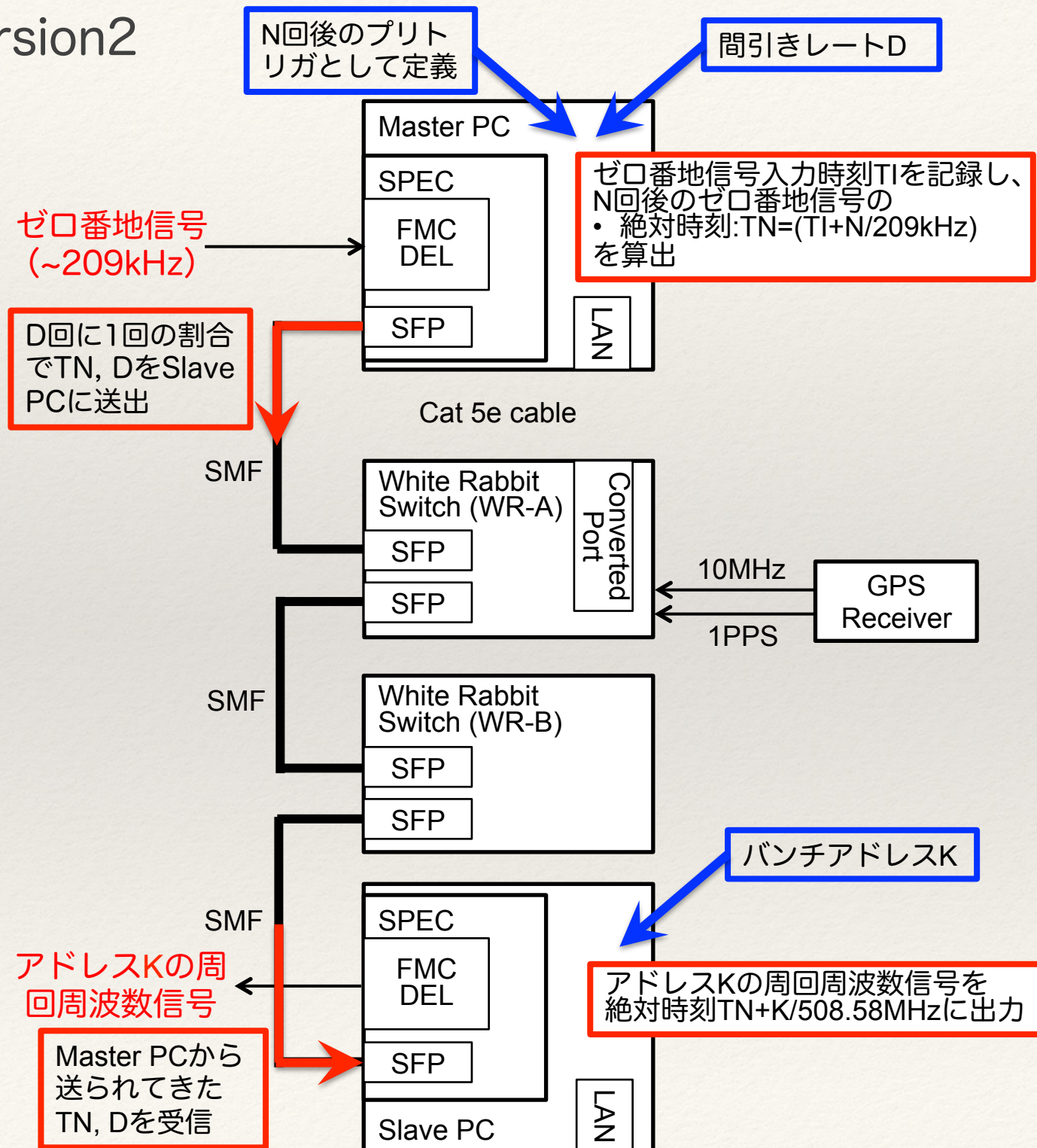
## 2.D (間引きレート) > N (N周回後にトリガ出力) という制約がある

- ❖  $D < N$ の場合、スレーブ側が最初のパルス出力を行う前に次のデータが送られてきてしまうため実装が難しかった→Version2で実装



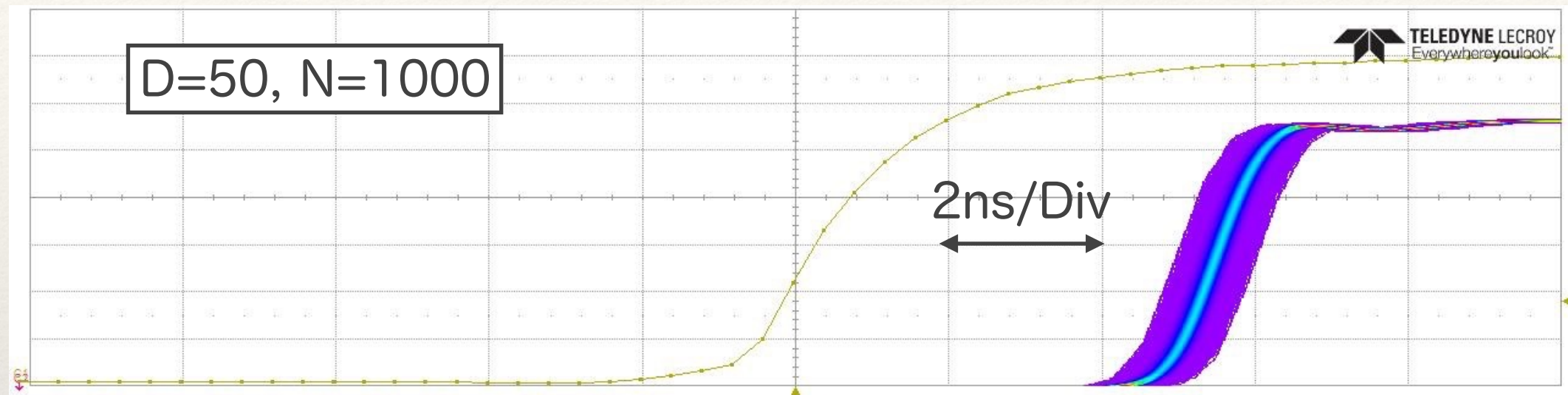
# システムの改良 (Version 2)

Version2





# Version2のジッタ計測



条件	平均 (ps)	最小値 (ps)	最大値 (ps)	標準偏差 (ps)	サンプル 数
スレーブ側設定項目 K=0, F=0, 出力周 波数微調(-31.2ps) は共通					
D=1000, N=1000	4.89	4.24	5.53	155.2	108.2k
D=50, N=1000	5.06	4.55	5.62	97.3	458.3k



---

# まとめ

---

- ❖ 高精度時刻同期技術 (White Rabbit) をベースとしたトリガ・タグ情報配信システム構築に向けて、最小構成の検証システムを開発した
  - ❖ SPring-8蓄積リングの特定バンチに同期した周回周波数信号 (~209kHz) の生成
- ❖ **Version1**では、現実的な間引きレートDの設定値 (600~1000) で出力ジッタは140~170ps(1 $\sigma$ )であった
- ❖ データの送出をFPGAで行う**Version2**では、出力ジッタ < 100ps(1 $\sigma$ )を実現できた



---

# まとめ

---

- ❖ トリガ信号の取り扱いが非常に容易
- ❖ ソフトウェアによって簡単にタイミング調整が出来る
- ❖ マスター/スレーブPC間をSMF 1本で接続するだけ
  - ❖ SMFの長さやスイッチの存在、温度等によるSMFの経路長変化によるタイミング調整は不要
- ❖ スレーブ側の増設は、ネットワークスイッチに接続するだけ
- ❖ タグ情報の付加もソフトウェアによって簡便に行える



---

# 今後の予定

---

- ❖ 放射光利用実験への試験的な適用
- ❖ Version2システムの現在の課題（チューナブルパラメータの変更が出来ない、システムが不安定）を解決する必要がある



---

# 今後の予定

---

- ❖ 機会が得られれば以下の高度化を行いたい
  - ❖ 低ジッタ化
  - ❖ 入力信号の周波数変化への自動追従
  - ❖ ブロードキャスト／マルチキャスト送信
    - ❖ マルチスレーブ構成への対応
- ❖ 任意の付加情報をソフトウェアで送出できるようにする



ご清聴、ありがとうございました。