

クロック分配システムMIKUMARIおよび 時刻同期プロトコルLACCP のマルチファミリへの展開と検証

濱田英太郎

高エネルギー加速器研究機構
素粒子原子核研究所
測定器開発センター(ITDC)

E-sysの開発する新しいフロントエンド回路の時刻同期プロトコル

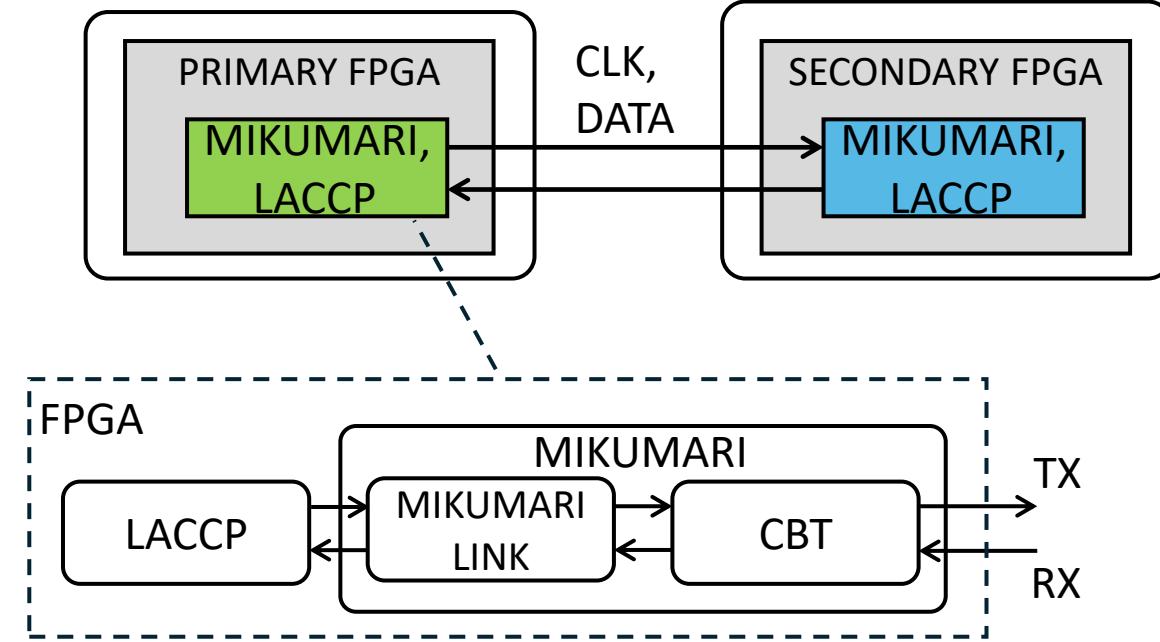
MIKUMARI (クロック分配システム)^[1]

- CBT (CDCM-based transceiver)
 - 1本のラインでクロックとデータを送信
 - FPGA標準I/Oでの実装が可能
- MIKUMARI Link (Link layer protocol)
 - MIKUMARI間通信のルールを定義

LACCP (時刻同期プロトコル)^[2]

- MIKUMARI Linkを利用し、時刻同期を可能にする
- 全てのフロントエンド回路をサブナノ秒の確度で同期するための軽量でシンプルなプロトコルとして開発

これまでXilinx Kinex-7 FPGA搭載のデバイスに対して300ps確度での同期を達成^[2]



※注: 確度はタイムスタンプのズレ具合 クロック信号のジッタではない

^[1] R. Honda, IEEE TNS, Vol. 70 (6), 1102 (2023)

^[2] R. Honda, IEEE TNS, Vol. 72 (3), 614 (2025)

MIKUMARIとLACCPをマルチファミリに展開

目標:

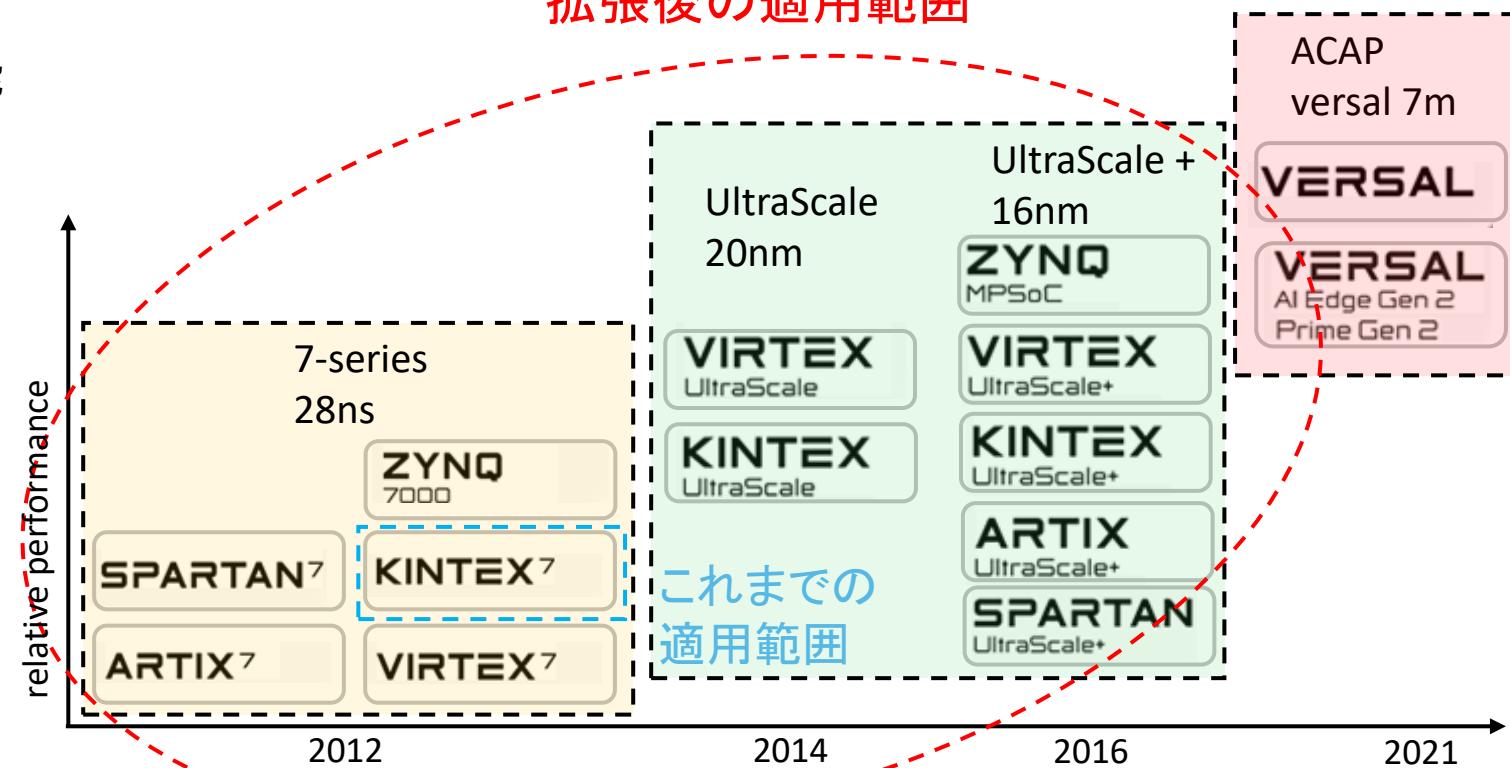
UltraScaleと7シリーズで使用可能にする
異なるデバイス間の通信を可能にする

- より高性能なFPGAであるUltraScaleが使用可能
- 最適なFPGAを選択可能になり、設計の拡張性や柔軟性を向上
- 本技術を様々なシステムに採用していきたい

ポイント:

- 専用ハードウェア(IDELAY、ISERDES等)の扱い
- FPGAの特性の違い

拡張後の適用範囲



MIKUMARIとLACCPの応用例

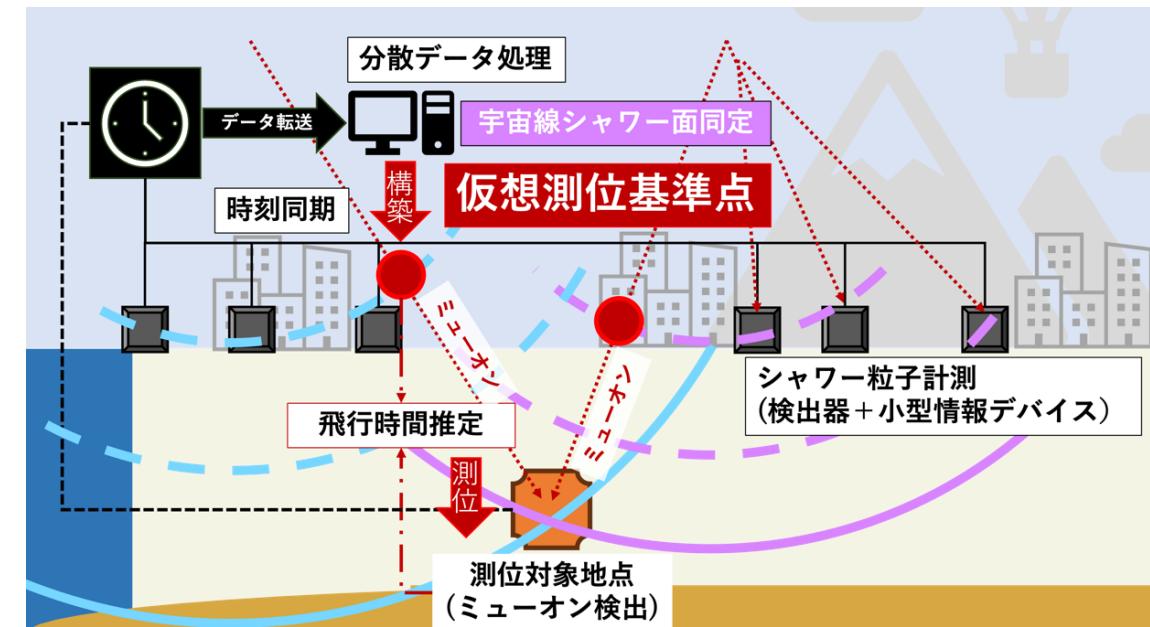
「JST K-program 仮想測位基準点を構築する即時分散データ処理技術」へ応用

衛星測位システム(GNSS)が使用できない環境下において
ミューオンを活用して時刻同期・測位を行う技術を開発

- 宇宙線シャワー到来を測定するための測定器群を
MIKUMARIとLACCPを用いて同期

MIKUMARI、LACCPへの要求

- 10km長のシングルモードファイバーによって通信
- 2つの回路間で300 psの正確さで時刻同期



LACCPによる時刻同期

PRIMARYのハートビートタイミングにSECONDARYを一致させる

クロック位相の差はfine offsetを求めてことで対応

- ハートビート: 16-bitカウンタが0のときに output

「時刻同期の方法」

- パルス信号の往復時間を測定し、SECONDARYの16-bitカウンタが持つ

オフセットを調べる

※8ns精度で調整

- fine offsetの計算

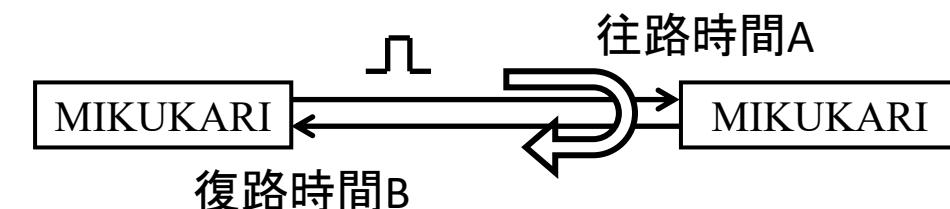
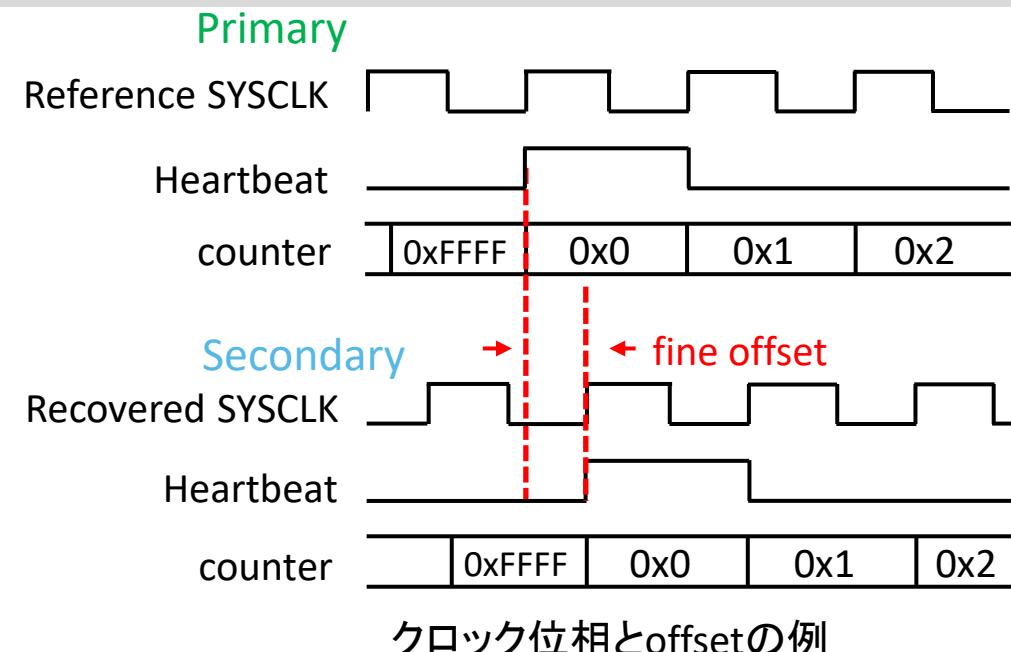
パルス信号の復路時間Bと往路時間Aの差より計算^[1]: $\text{fine offset} = (B - A) / 2$

- 同じデバイスで通信: $\text{fine offset} = (\text{可変遅延量} d(B) - \text{可変遅延量} d(A)) / 2$

- 違うデバイスで通信: $\text{fine offset} = (\text{可変遅延量} d(B) - \text{可変遅延量} d(A)) / 2$

$$+ (\text{初期遅延量} D(B) - \text{初期遅延量} D(A)) / 2$$

初期遅延量を調べ、正しいfine offsetを計算

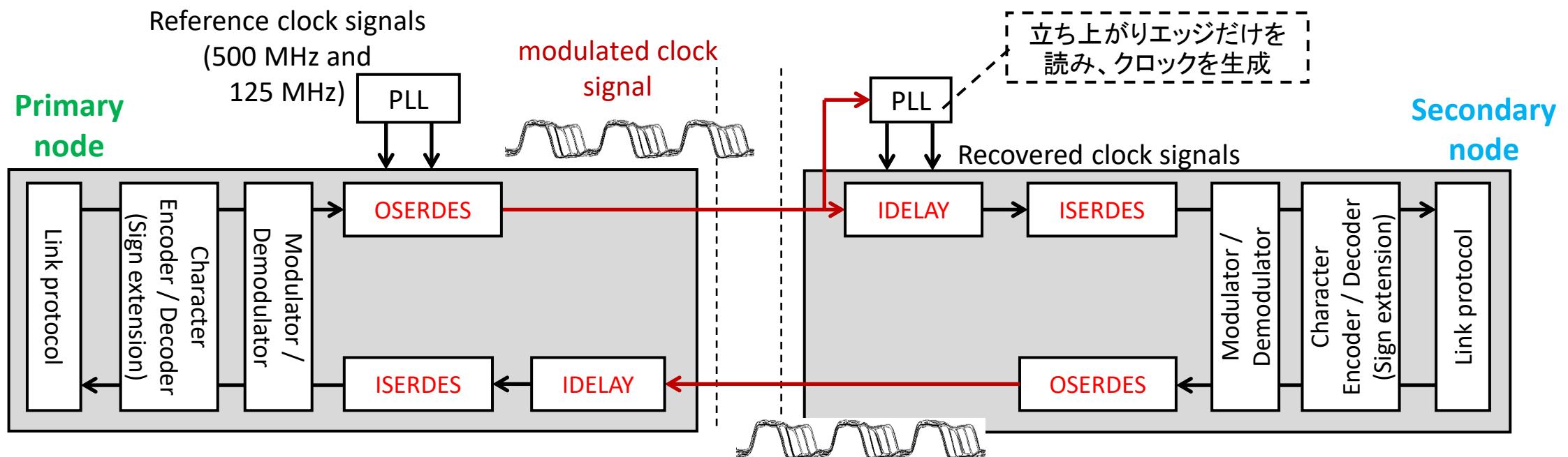


^[1] R. Honda, IEEE TNS, Vol. 72 (3), 614 (2025)

MIKUMARIによる通信

FPGA にあらかじめ組み込まれた専用ハードウェアで機能を実現

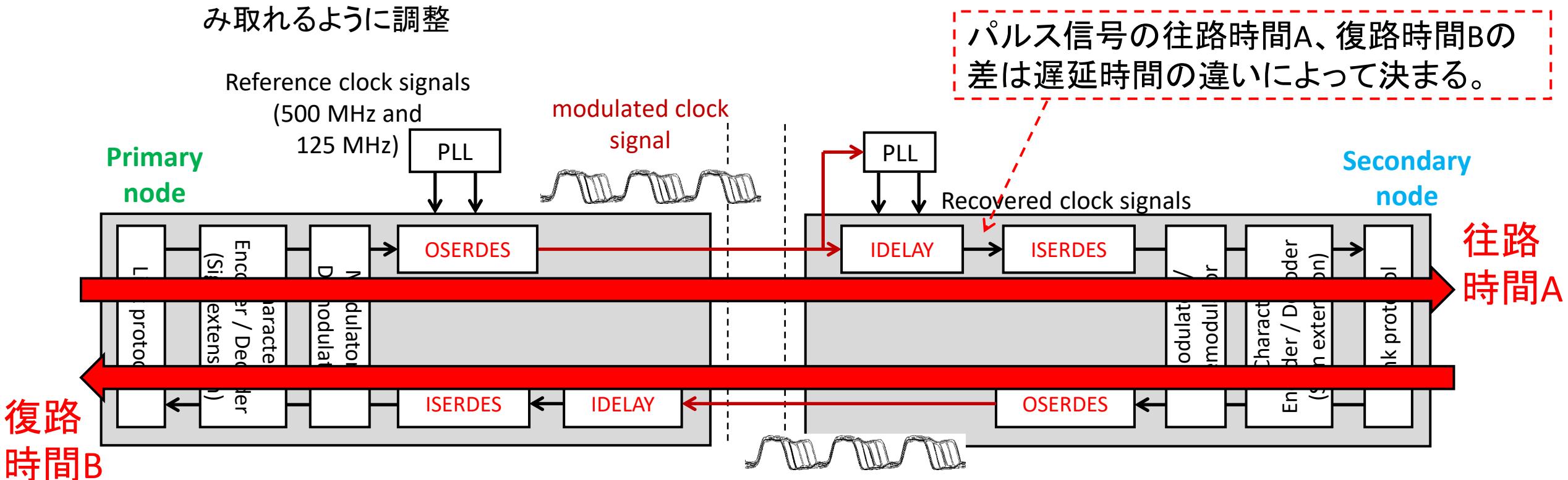
- ISERDES: シリアルパラレル変換を行う専用モジュール
bit単位で遅延を与え、正しいデータを読み出す
- IDELAY: 受信信号を細かく遅らせ、ISERDES が安定して bit を読み取れるように調整



MIKUMARIによる通信

FPGAにあらかじめ組み込まれた専用ハードウェアで機能を実現

- ISERDES: シリアルパラレル変換を行う専用モジュール
bit単位で遅延を与え、正しいデータを読み出す
- IDELAY: 受信信号を細かく遅らせ、ISERDESが安定してbitを読み取れるように調整



ISERDES による遅延

ISERDES

- ・ シリアルパラレル変換を行う専用モジュール
- ・ bit単位で遅延を与え、正しいデータを読み出す

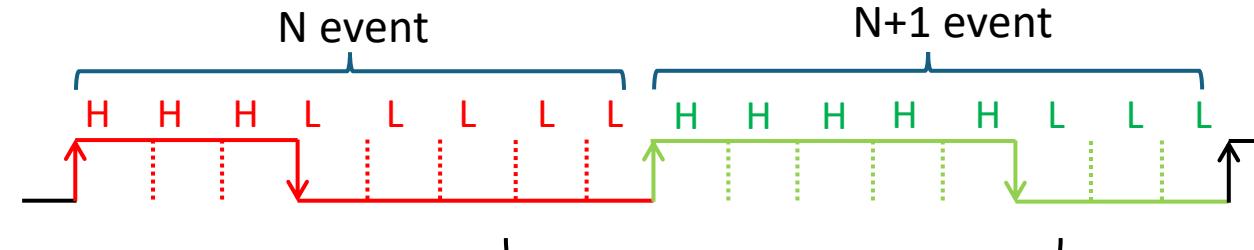
(例) SYSCLK



FAST CLK



ISERDES
入力



変換範囲がこの範囲
だと間違い

ISERDES
出力(8bit)



ISERDESによる遅延

ISERDES

- シリアルパラレル変換を行う専用モジュール
- bit単位で遅延を与え、正しいデータを読み出す

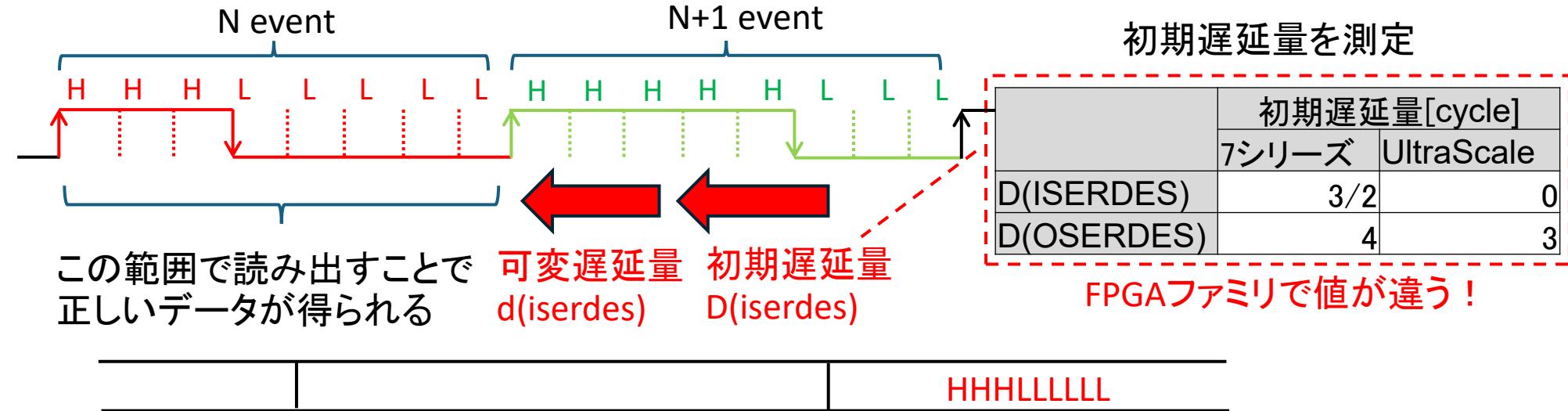
(例) SYSCLK



FAST CLK



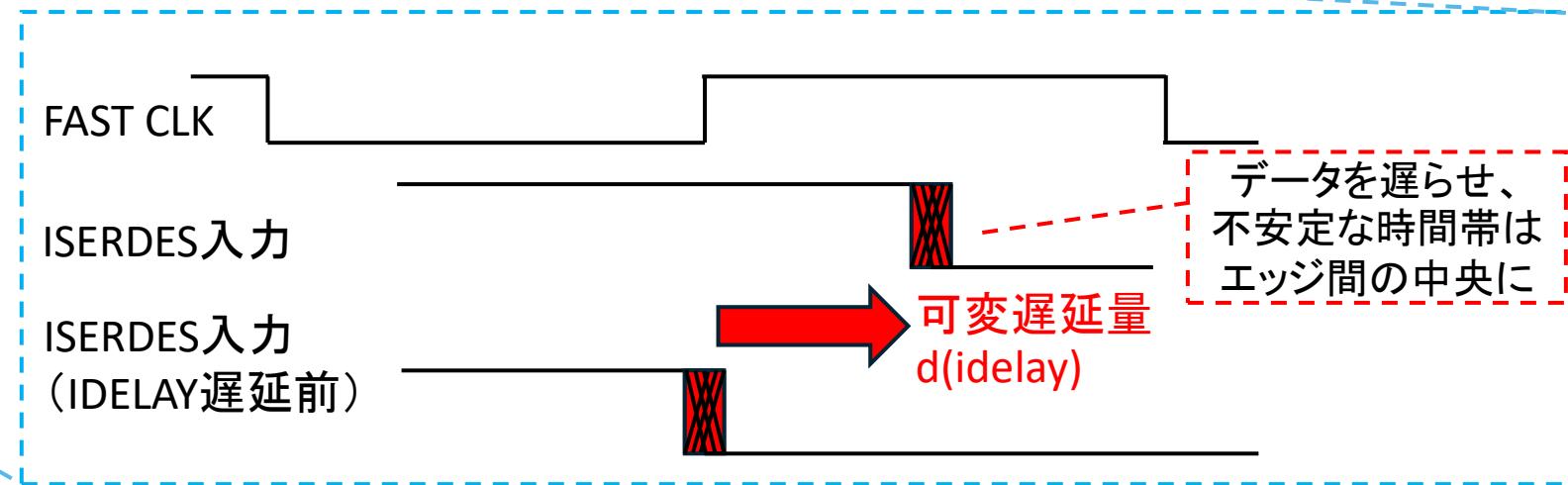
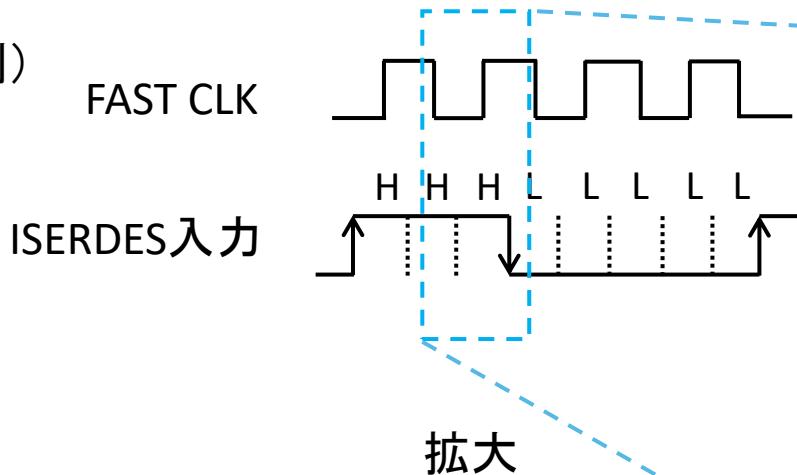
ISERDES
入力



IDELAYによる遅延



(例)



現在のMIKUMARIの設定

- 7シリーズ: 可変遅延量 $d(idelay) = 78\text{ps} \times n$ (n の最大は31)
- UltraScale: 可変遅延量 $d(idelay) = \text{約}110\text{ps} \times n$ (n の最大は31)

※UltraScaleはデバイスによって値が違う

※現在の設定では、UltraScaleの遅延量の調整精度は低く設定している

入出力の遅延

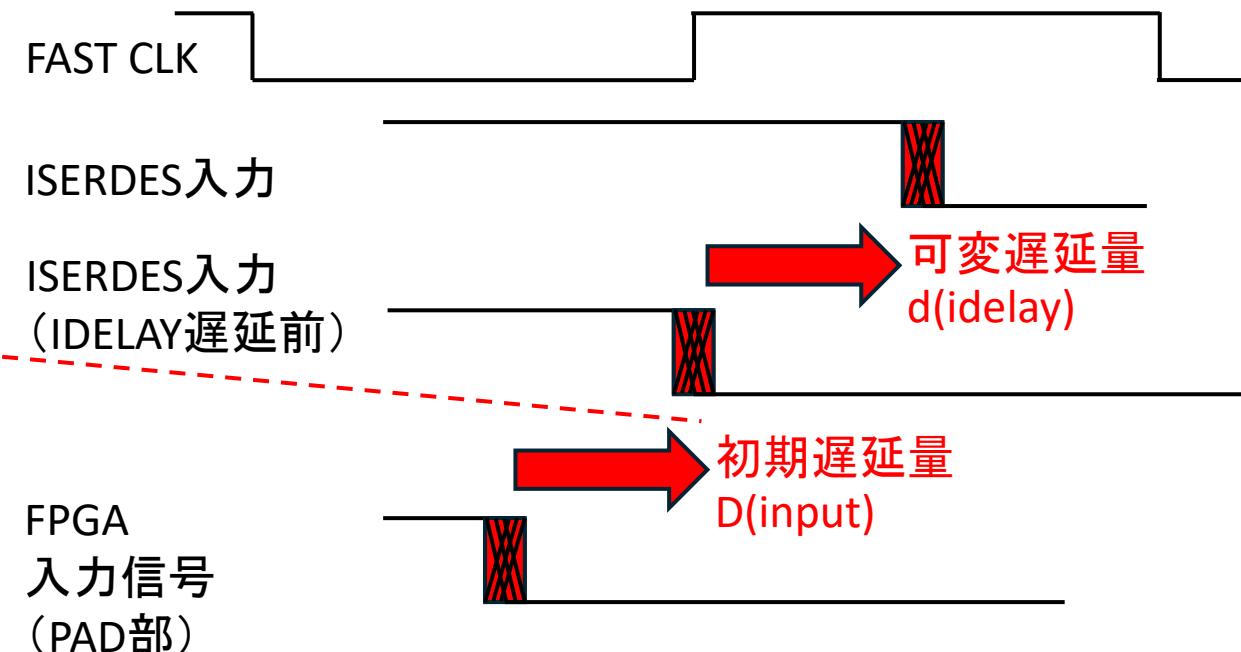
配線の伝搬遅延やバッファ遅延により、入力信号はISERDESに届くまで時間がかかる



FPGAに信号が入ってからISERDESに届くまで時間がかかる

開発ツールvivadoのタイミング解析により計算

| Location | Delay type | Incr(ns) | Path(ns) |
|--------------------------------|--|----------|----------|
| <hr/> | | | |
| (clock MIKUMARI_RXP rise edge) | | | |
| W1 | 0.000 | 0.000 | r |
| | 0.000 | 0.000 | r |
| <hr/> | | | |
| W1 | net (fo=0) | 0.000 | 0.000 |
| | IBUFDS (Prop_ibufds_I_0) | 0.435 | 0.435 r |
| IDELAY_X1Y90 | net (fo=2, routed) | 0.000 | 0.435 |
| | IDELAYE2 (Prop_idelaye2_IDATAIN_DATAOUT) | 0.244 | 0.679 r |
| ILOGIC_X1Y90 | net (fo=1, routed) | 0.000 | 0.679 |
| | ISERDESE2 | | r |



入出力の遅延 初期遅延量の大きさ

- 初期遅延量の影響は大きくないケースが多い(誤差に埋もれてしまう)
 - (例1) PRIMARY/SECONDARYがArtix-7、Artix US+ →入出力の初期遅延量の項 = 9ps
 - (例2) PRIMARY/SECONDARYがKintex-7、Artix US+ →入出力の初期遅延量の項 = 84ps
- PRIMARYとSECONDARYのFPGAが同じ場合、キャンセルされる
→ Fine offset 計算に入出力の初期遅延量を反映させるかどうかを、ファームウェア設定で切り替え可能とした

Fine offset = (復路時間B – 往路時間A) / 2

$$= (\text{可変遅延量} d(B) - \text{可変遅延量} d(A)) / 2 + (\text{初期遅延量} D(B) - \text{初期遅延量} D(A)) / 2$$

入出力の初期遅延量の項 = (入出力の初期遅延時間(Secondary) – 入出力の初期遅延時間(Primary)) / 2

| | Kintex-7 | Artix-7 | Artix US+ | Kintex US | Kintex US+ |
|-----------|----------|---------|-----------|-----------|------------|
| 初期遅延量[ps] | 460 | 309 | 292 | 124 | 293 |

入出力における初期遅延量

タイミング解析の計算結果

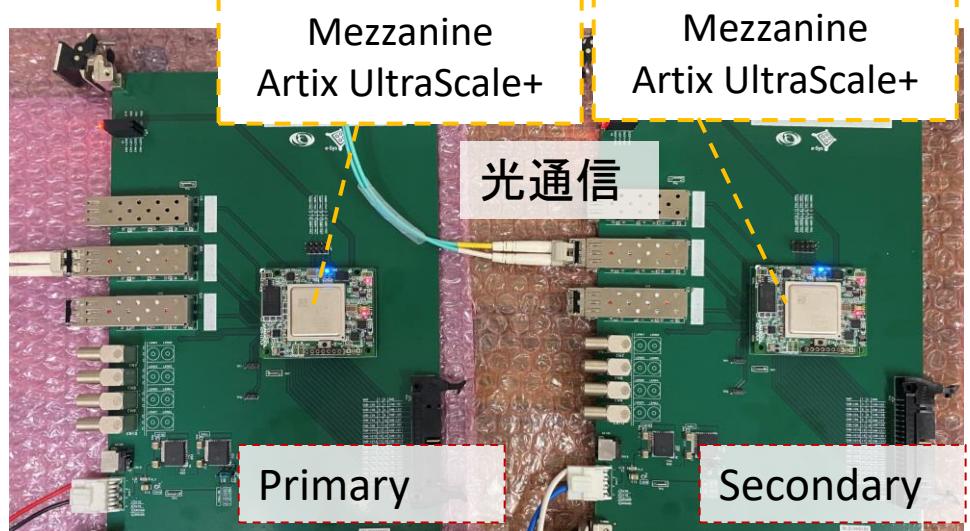
※IOピンによっても数~数10ps変わる

※Kintex UltrascaleとKintex Ultrascale+はタイミング解析を実施しただけで実機での試験は行っていない

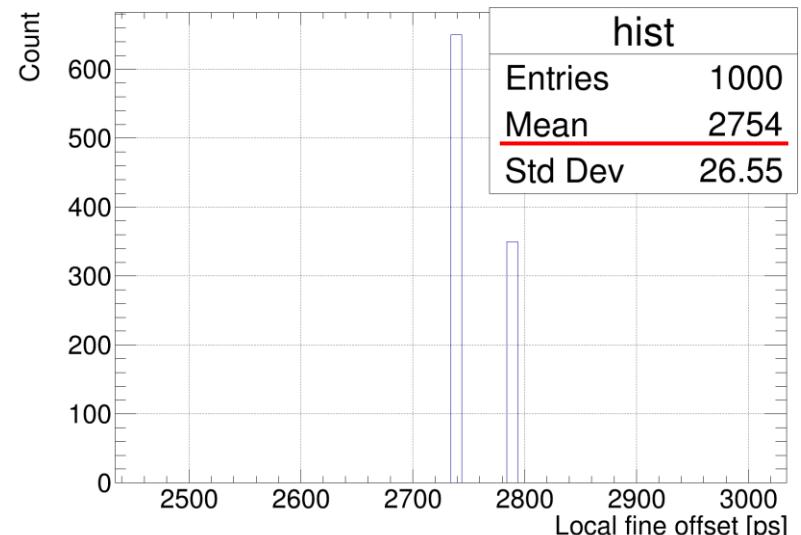
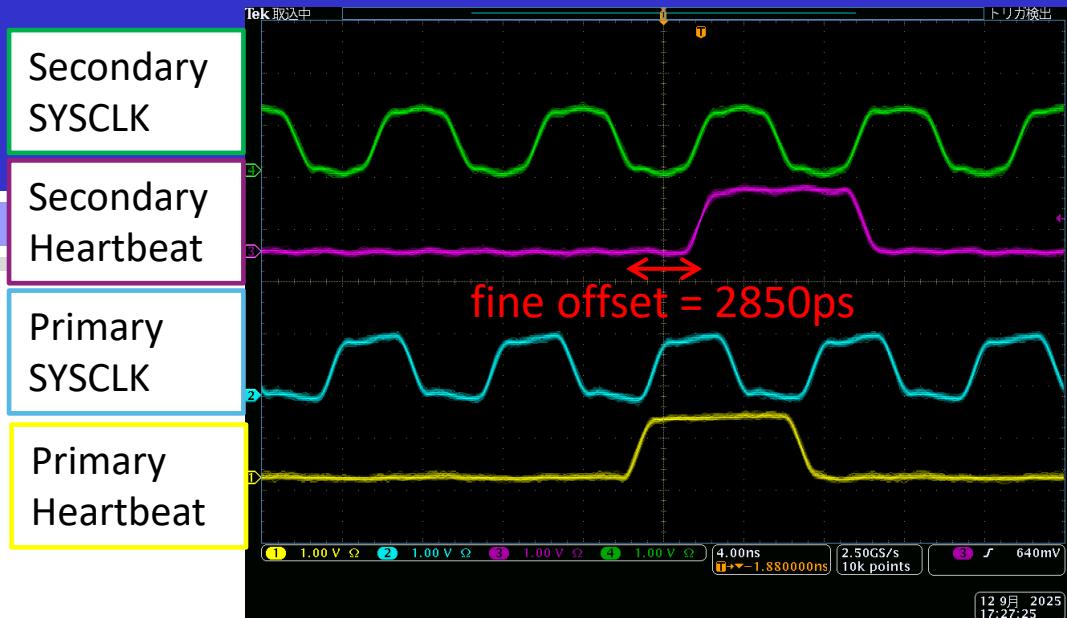
MIKUMARI、LACCPの測定

- Heartbeat(16bit-counterが0になるときの信号)を測定
- 実測値 - 計算値 = $2850\text{ps} - 2754\text{ps} = 96\text{ps}$

$$\text{fine offset} = (\text{可変遅延量d(B)} - \text{可変遅延量d(A)}) / 2 + (\text{初期遅延量D(B)} - \text{初期遅延量D(A)}) / 2$$



通信試験 セットアップ



fine offsetの計算結果
(1000回の計算を実施)

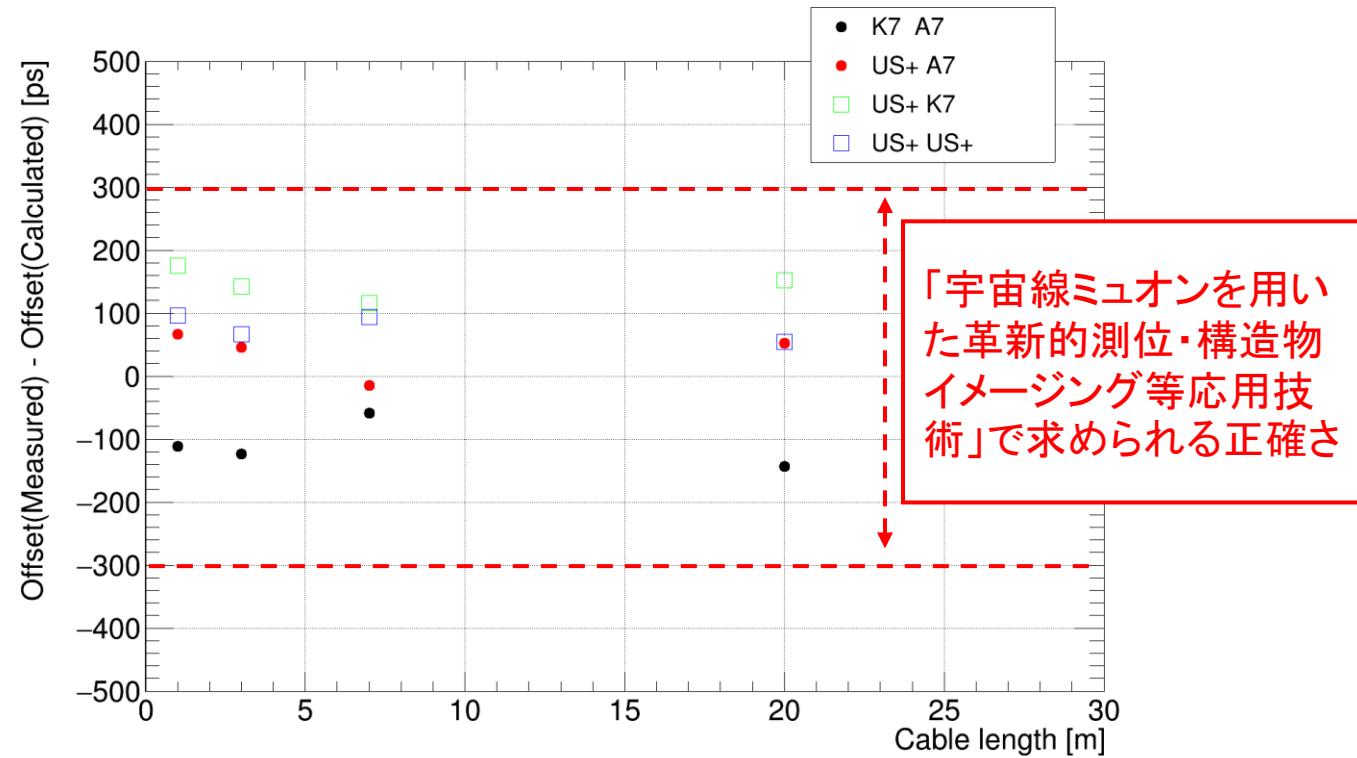
MIKUMARI、LACCPの測定

どのデバイスのパターン、ケーブルの長さでも $\pm 200\text{ps}$ 以内の正確さ

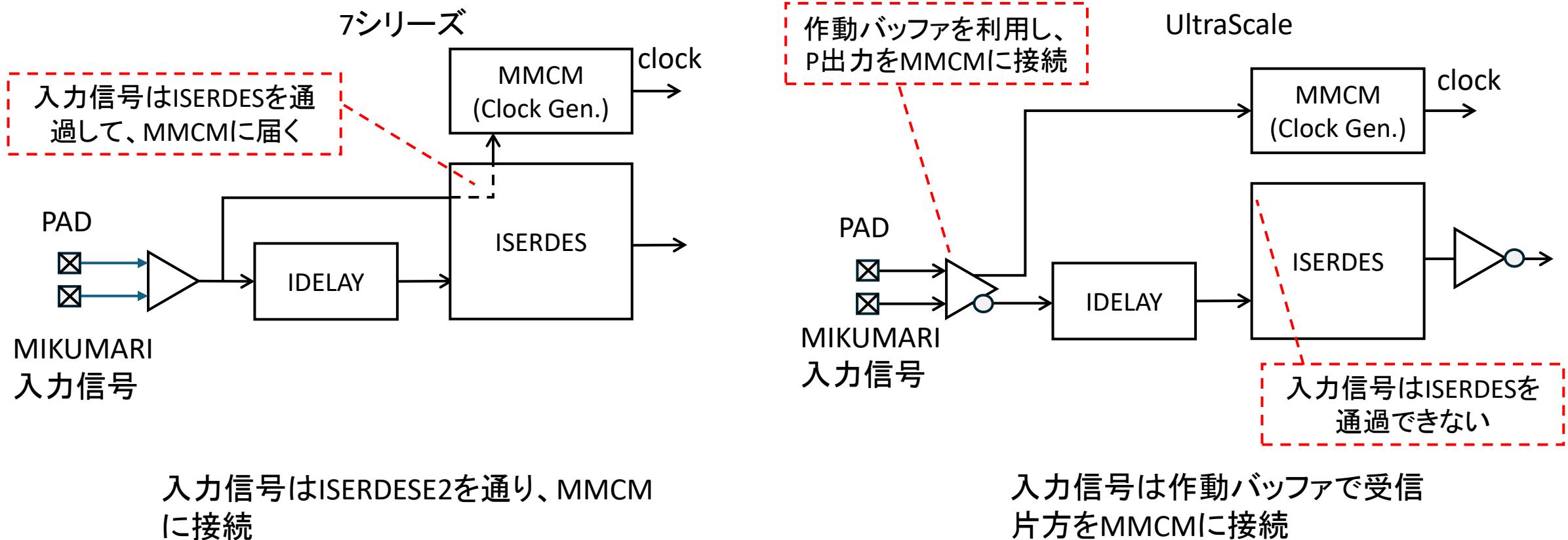
デバイスとケーブルの長さを変えて、
fine offsetの実測値-計算値を測定

(デバイスのパターン)

- Primary: Kintex-7, Secondary: Artix-7
- Primary: Artix UltraScale+, Secondary: Artix-7
- Primary: Artix UltraScale+, Secondary: Kintex-7
- Primary: Artix UltraScale+, Secondary: Artix UltraScale+



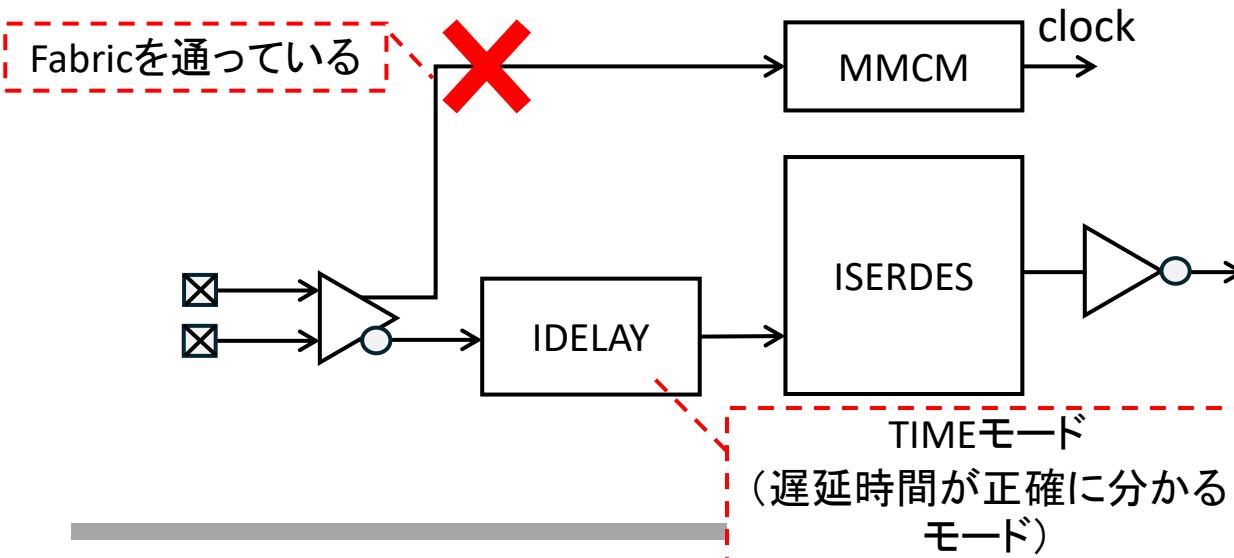
7シリーズとUltraScaleにおけるSecondary 入力部の違い



UltraScaleにおけるピンアサインの注意

Secondaryでは、BitSlice_0またはBitSlice_6のpinにMIKUMARIの入力信号を割り当ててはいけない！

- UltraScaleのI/OピンにはBITSLICE(BITSLICE_0 ~ BITSLICE_11)が割り当ててられている
- IDELAYE3をTIMEモードにした場合、BITSLICE_0またはBITSLICE_6はキャリブレーションを行う制御スライスになる
- Fabric(ユーザーが自由にロジックを組める部分)へ信号をつなげられない



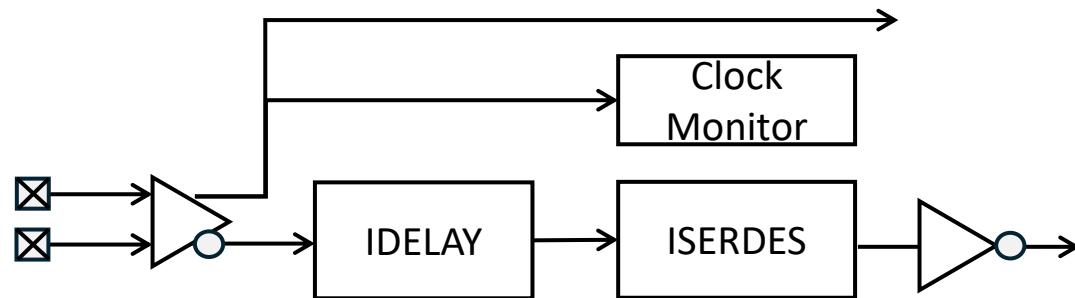
| | |
|-----|-----------------------------|
| B26 | IO_L24N_T3U_N11_66 |
| B25 | IO_L24P_T3U_N10_66 |
| C26 | IO_L23N_T3U_N9_66 |
| D26 | IO_L23P_T3U_N8_66 |
| C24 | IO_L22N_T3U_N7_DBC_AD0N_66 |
| D23 | IO_L22P_T3U_N6_DBC_AD0P_66 |
| D25 | IO_L21N_T3L_N5_AD8N_66 |
| D24 | IO_L21P_T3L_N4_AD8P_66 |
| E23 | IO_L20N_T3L_N3_AD1N_66 |
| F23 | IO_L20P_T3L_N2_AD1P_66 |
| E26 | IO_L19N_T3L_N1_DBC_AD9N_66 |
| E25 | IO_L19P_T3L_N0_DBC_AD9P_66 |
| H22 | IO_L18N_T2U_N11_AD2N_66 |
| H21 | IO_L18P_T2U_N10_AD2P_66 |
| G26 | IO_L18P_T2U_N10_DBC_AD9P_66 |

Artix UltraScale+の回路図(一部)

ピン名のNの後に数字がBitSliceの番号

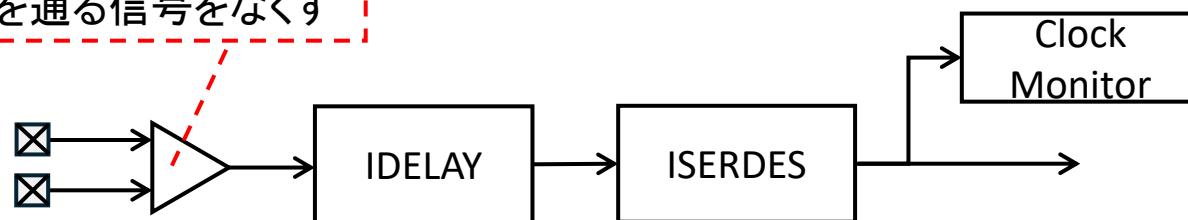
UltraScaleにおけるピンアサインの注意

Primaryでは、BitSlice_0とBitSlice_6のピンにMIKUMARIの入力を割り当てることは可能

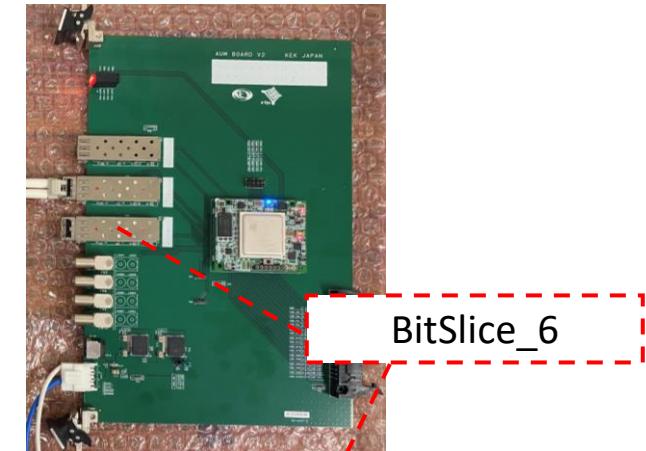
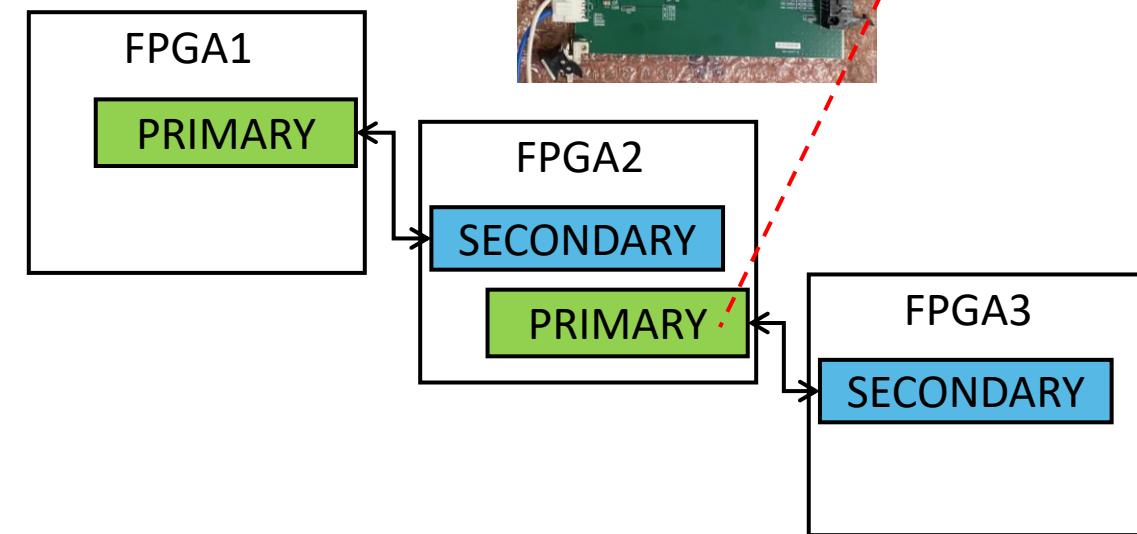


入力信号をBitSlice_0、BitSlice_6以外に割り当てた場合

Fabricを通る信号をなくす



入力信号をBitSlice_0、BitSlice_6に割り当てた場合
(PRIMARYではこの構成でも動作可能)

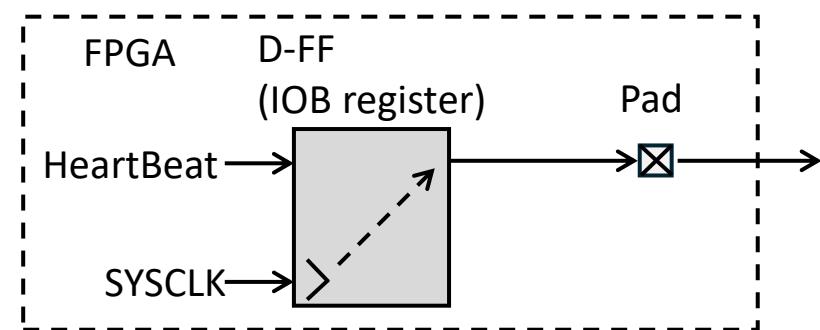


Fine Offsetの検証

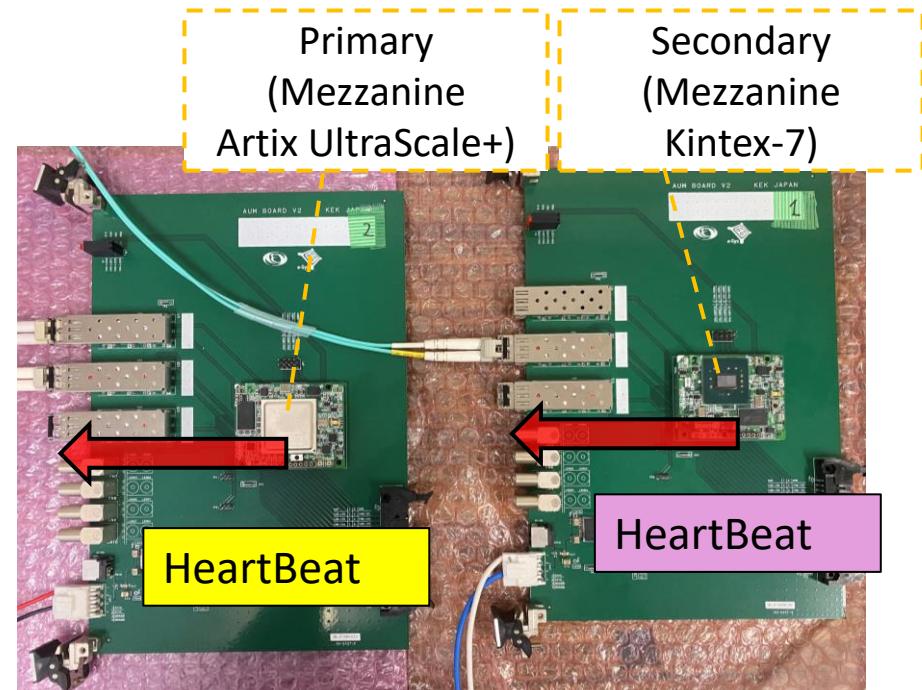
問題:

PrimaryとSecondaryでFPGAファミリが異なる場合、
Fine Offset(HeartBeat信号の時間差)を正確に測ることが難しい

原因:FPGA内部でHeartBeat信号の出力の伝搬時間が異なる
対処法:タイミング解析結果による補正



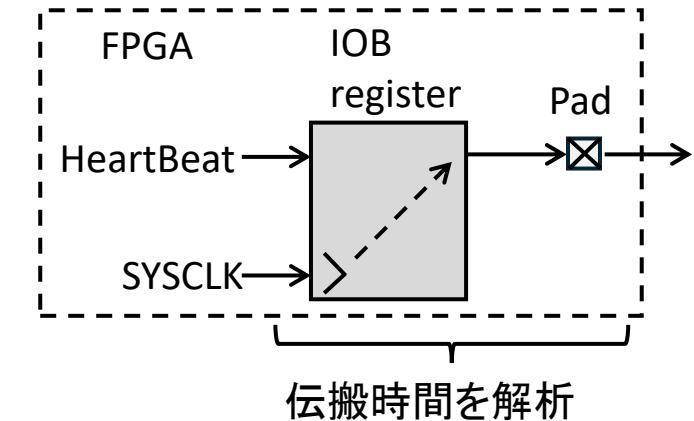
FPGA内部
HeartBeat信号が外部に出力される回路



Fine Offsetの検証

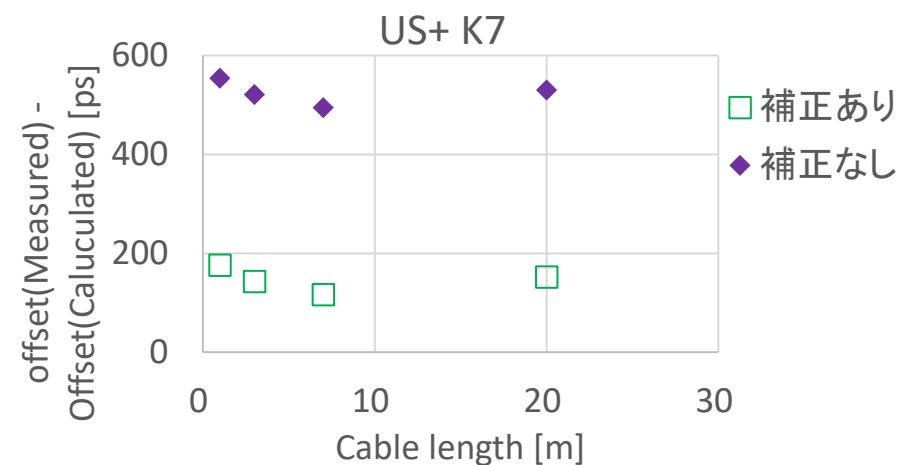
タイミング解析結果による補正

- vivadoでHeartBeat信号が出力されるまでの時間を解析
- オシロスコープで測定した実測値Offset(Measured)を補正

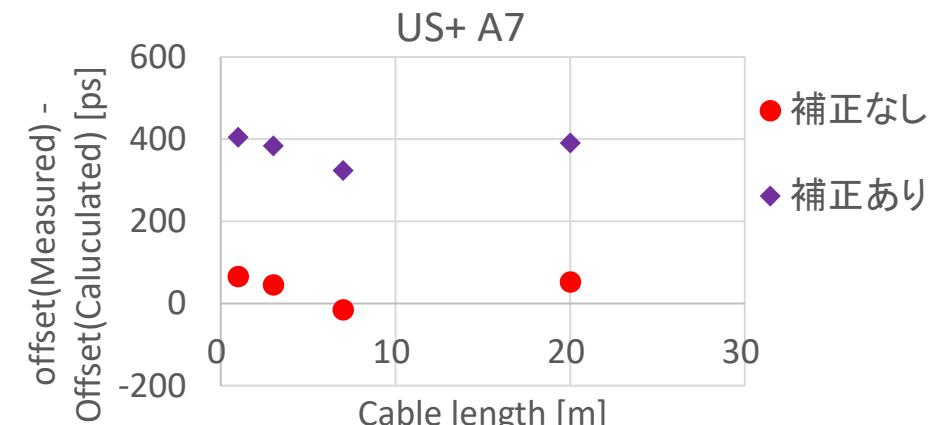


| | |
|-------|--------|
| Artix | 1230ps |
| US+ | 1608ps |
| K7 | 1608ps |
| A7 | 1569ps |

タイミング解析によって求めた遅延量[ps]



※補正することで1230ps-1608ps=-378psを加えている



※補正することで1230ps-1569p=-339psを加えている

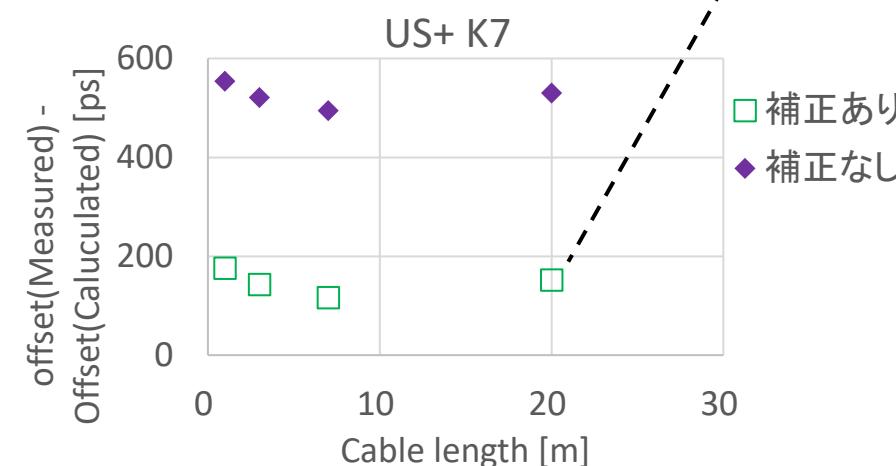
Fine Offsetの検証

タイミング解析結果による補正

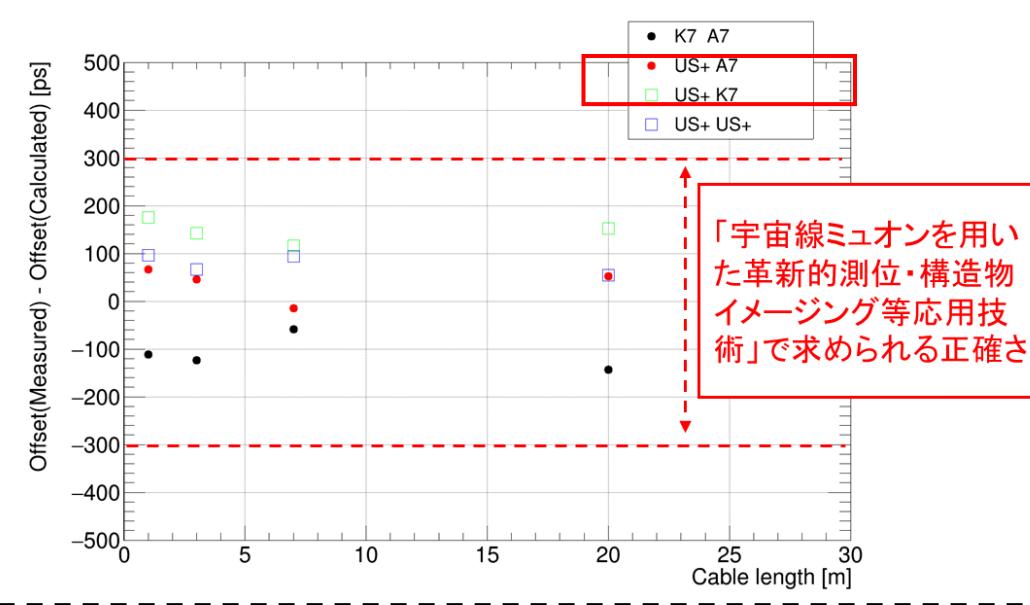
- vivadoでHeartBeat信号が出力されるま
- オシロスコープで測定した実測値Offset

| | |
|-------|--------|
| Artix | 1230ps |
| US+ | |
| K7 | 1608ps |
| A7 | 1569ps |

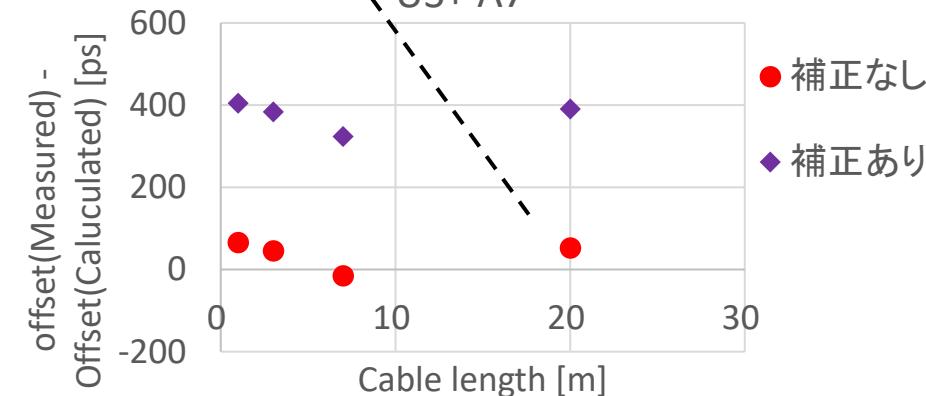
タイミング解析によつ
て求めた遅延量[ps]



※補正することで1230ps-1608ps=-378psを加えている



伝搬時間を解析

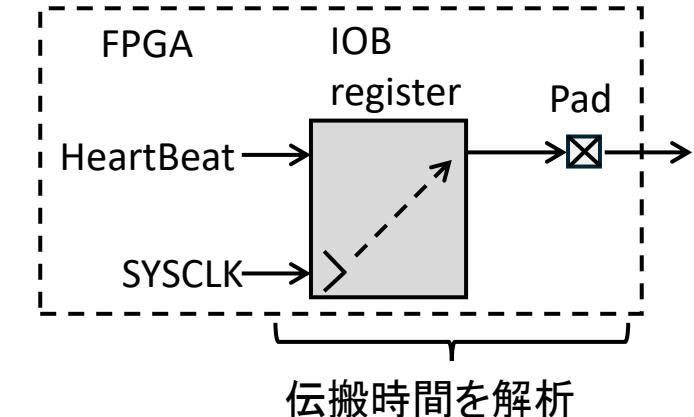


※補正することで1230ps-1569p=-339psを加えている

Fine Offsetの検証

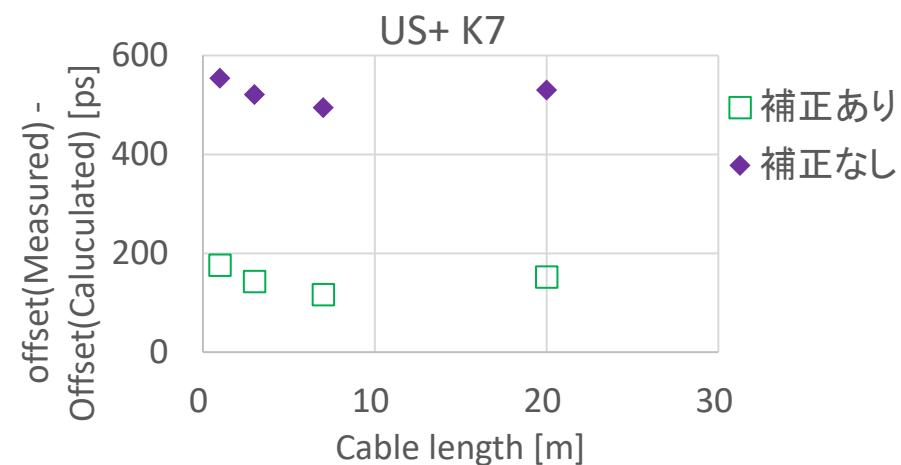
タイミング解析結果による補正

- vivadoでHeartBeat信号が出力されるまでの時間を解析
- オシロスコープで測定した実測値Offset(Measured)を補正

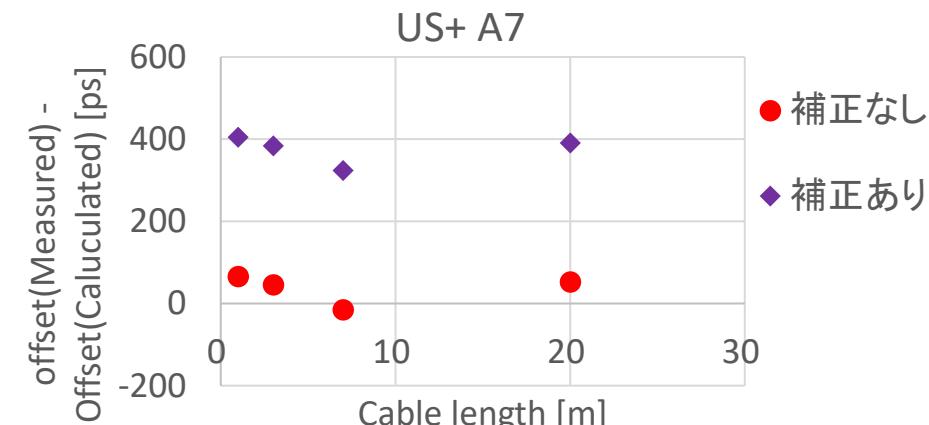


| | |
|-------|--------|
| Artix | 1230ps |
| US+ | 1608ps |
| K7 | 1608ps |
| A7 | 1569ps |

タイミング解析によって求めた遅延量[ps]



※補正することで1230ps-1608ps=-378psを加えている



※補正することで1230ps-1569p=-339psを加えている

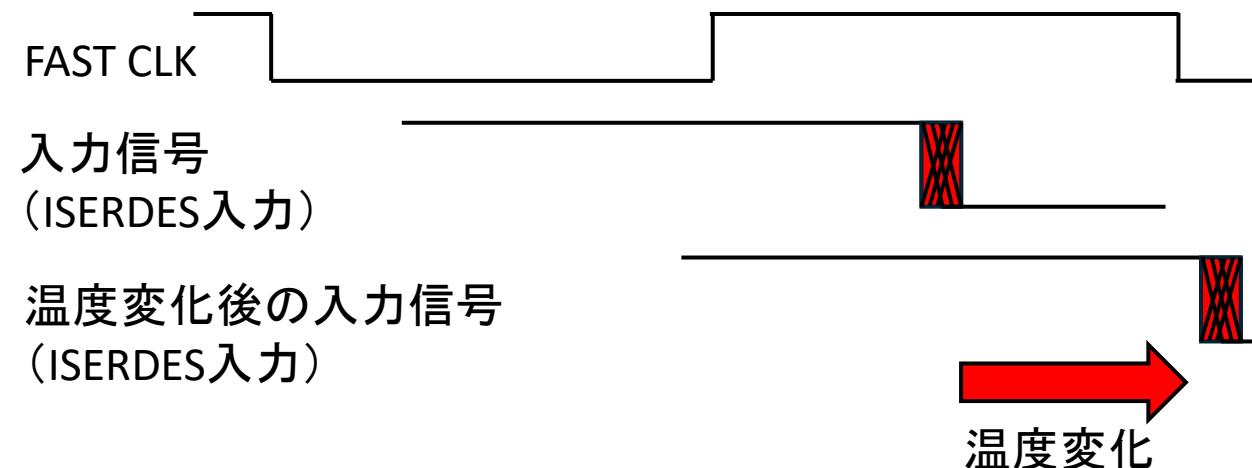
MIKUMARIとLACCPの応用

今後の課題: 長距離通信における光ファイバーの温度変化への対応

温度変化に伴い、光ファイバーの屈折率および長さが変化

→光の伝搬遅延が変動

→ LACCP時刻同期性能の劣化、通信負荷



まとめと今後の予定

- MIKUMARI(クロック分配システム)とLACCP(時刻同期プロトコル)をアップデート
 - 専用ハードウェア(I SERDES、I DELAY等)動作の調査を実施
 - 7シリーズとUltraScaleで使用可能にした
 - 異なるデバイス間でも時刻同期を可能にした
- 複数のデバイスでLACCPの時刻同期の試験を実施し、200ps以内の正確さであることを確認
- 「宇宙線ミュオンを用いた革新的測位・構造物イメージング等応用技術」に採用予定

「今後の予定」

- ファームウェアコードの確認後、リリース予定
- 長距離通信における光ファイバーの温度変化への対応

謝辞

本研究は、JST経済安全保障重要技術育成プログラム
【JPMJKP24J2】の支援を受けたものです。

