

進捗報告

柳瀬 健太郎

報告

- Link sharingを使ったthreshold scanの結果(通常の見出しとの比較)

まとめ(前回)

- PreprodQ18を使ってData Mergingを適切に行うための試験として、デシリアライザーのlaneのDelay値、Cmlbias、の2つのパラメータを決定した。
- Eyediagramでは各Delay値でデータフォーマット上のidle block(data block以外のblock)を数え上げ、Link qualityを導出した。
→ **lane0 : 17 ns lane1 : 5 ns lane2 : 6 ns lane3 : 11.5 ns に決定**
- Cmlbias scanではCmlbias0,1を2次元的に変化させ、各条件でEyediagramを行い、Link quality=1の範囲が最も多かった条件を導いた。
→ **Cmlbias0=800 Cmlbias1=400 に決定**

Eye diagram参考 :

https://indico.cern.ch/event/1223749/contributions/5668600/attachments/2750773/4787859/YARR_Mironova_1311_AUW.pdf

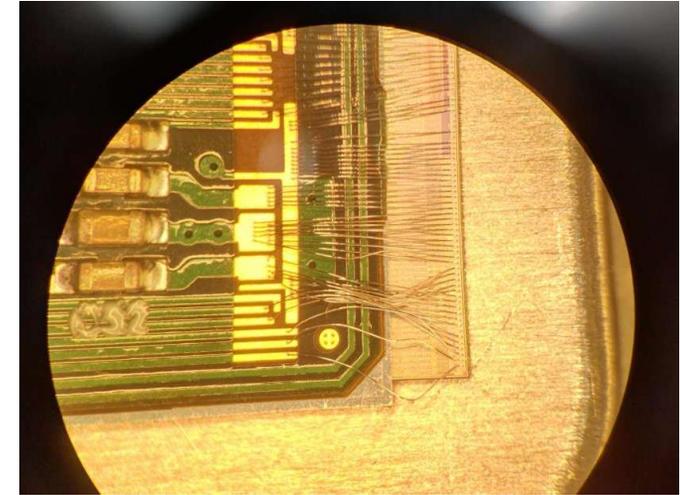
PreprodQ18状況

- ARTB後の生存確認で、5.88 A(CC)で~1.8 V程度しかかからず、全Chipとコミュニケーションが取れない
- Chip3のワイヤーが何かしらに触れてしまった痕跡 (TB中は読み出し快調だったので片付け中だと思われる)



- 電圧は5.88 A(CC)で変わらず~1.8 V程度しかかからない..
- でもChip3以外とコミュニケーションが取れた

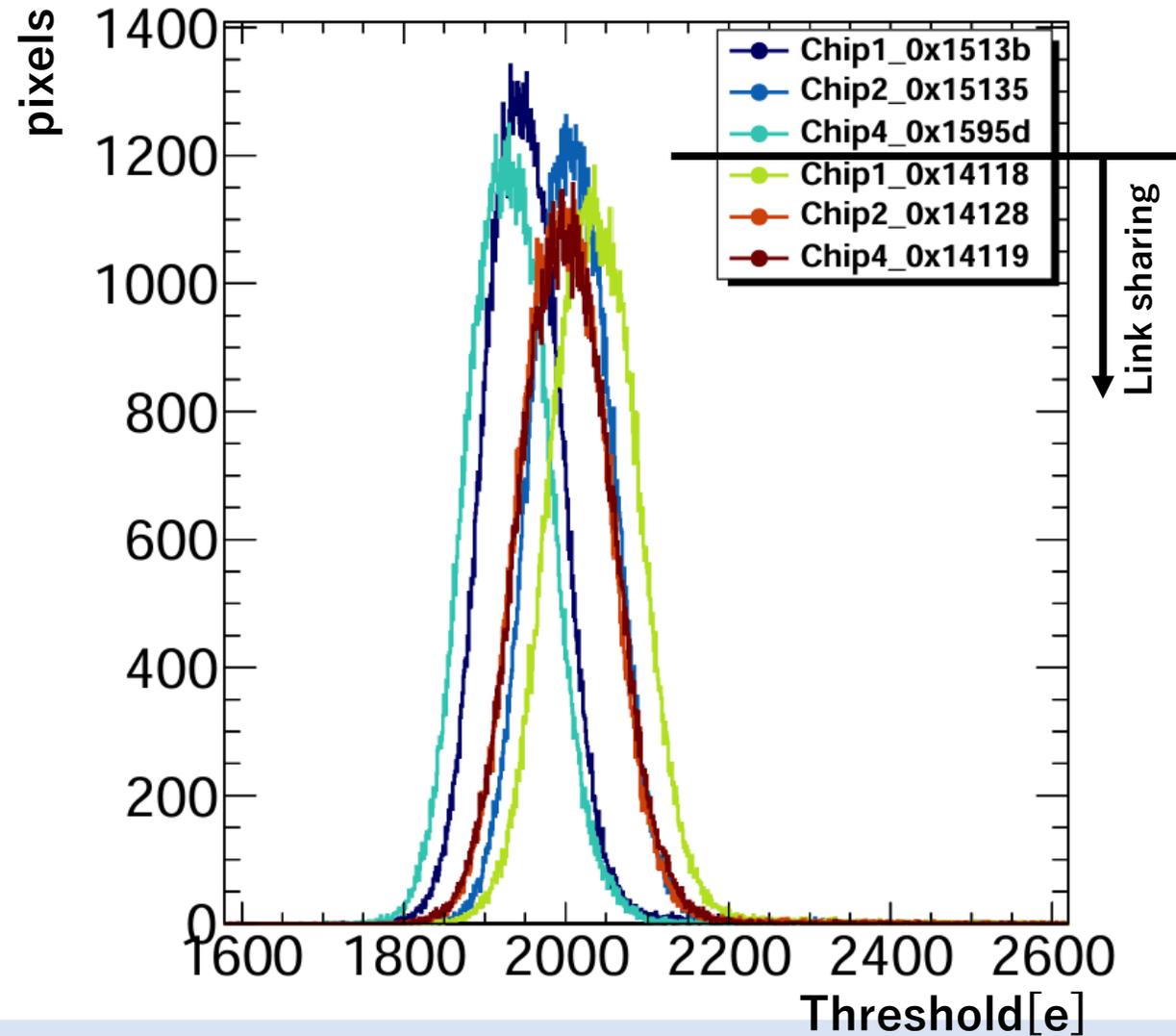
- 3Chipでもリンクシェアリング自体は可能なのでリンクシェアリングを用いた読み出しと通常の読み出しでthreshold scan結果を比較してみる



Chip3ワイヤー打ち直し前

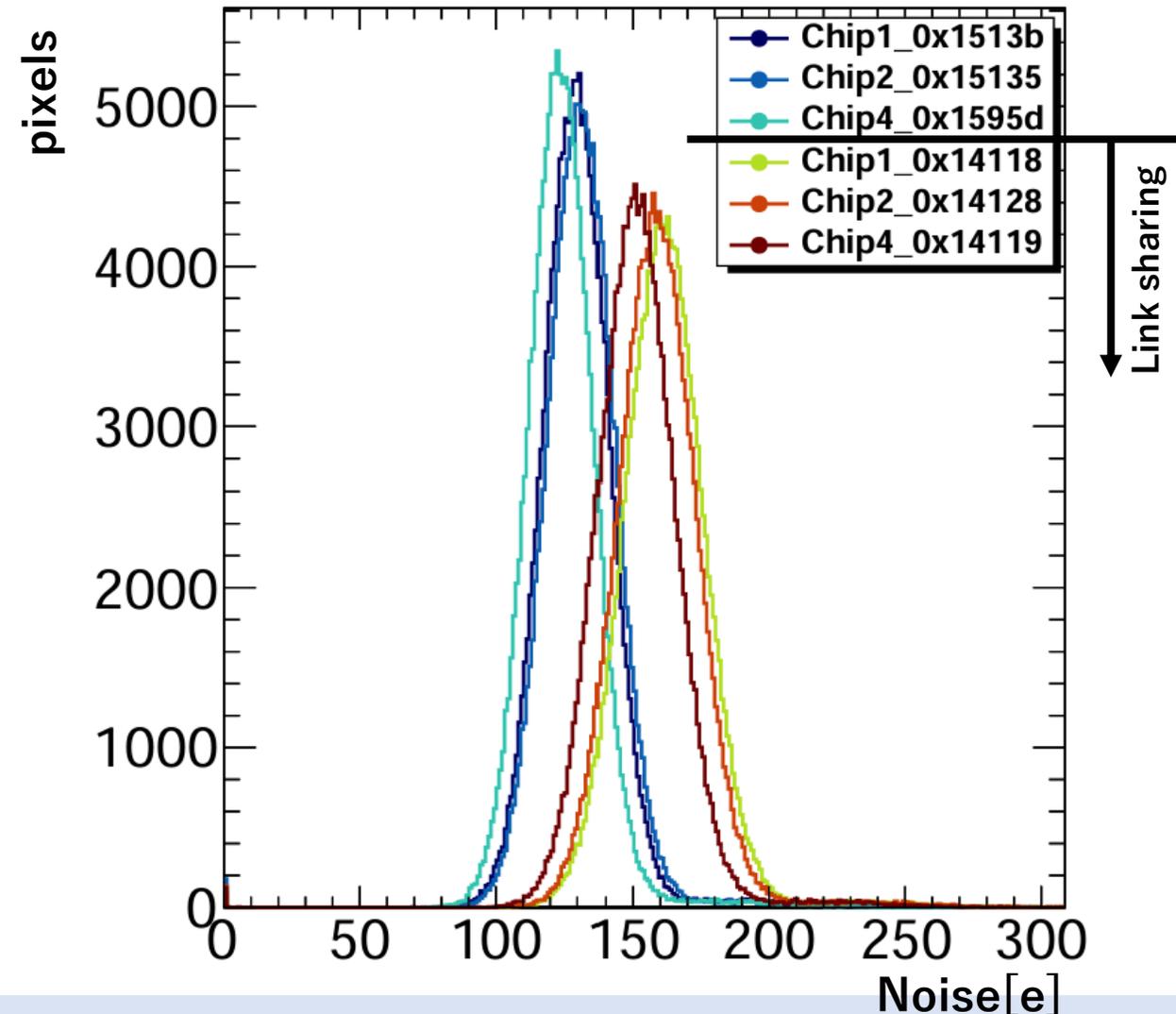
Threshold scan (Threshold)

- いずれも～2000 eにTuningはできている
- リンクシェアリングを用いた結果は通常の結果と比べ**Threshold sigma**が大きい



Threshold scan (Noise)

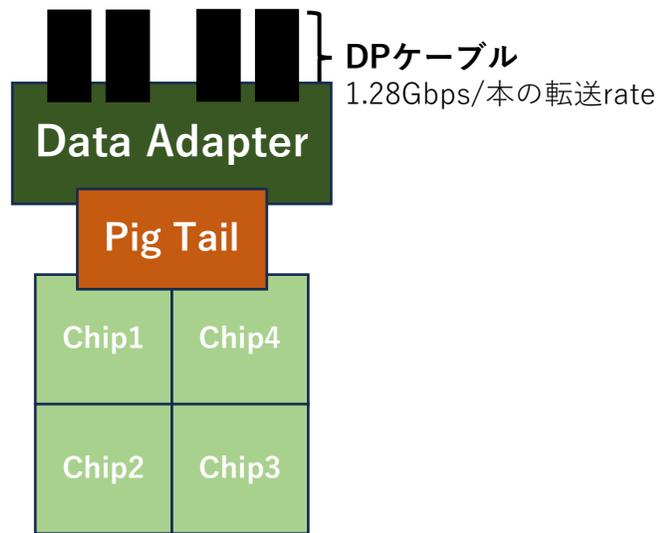
- リンクシェアリングを用いた結果は通常の結果と比べ**約40 e Noise Meanが高い**
- 同日に繰り返し測定を行ったが同様の結果だったので、測定誤差ではなさそう
- Link sharingそのものが測定結果に寄与するとは考えづらい
- Link sharingを用いて読み出しするのにあたり多くのパラメータを変更したので、それが影響している？
(TO DO：各パラメータについて確認する)



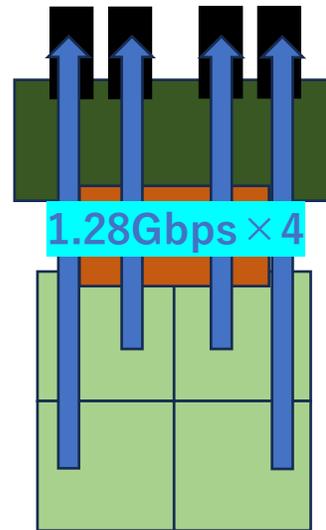
Back up

Data Merging

- Event数の多いITkでは荷電粒子の散乱を防ぐために検出器の物質量を最小化したい
- **Link Sharing**によって、データの読み出しLink(ケーブル)を減らしてデータの読み出しを行うことが可能
- Linkの数と転送rateはトレードオフであり、層ごとにLink Sharingを使い分ける必要がある

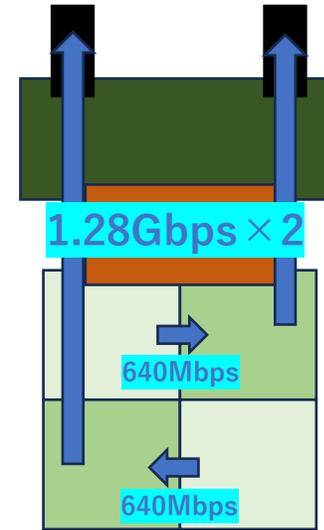


読み出したデータはpigtailを通して Adapter cardに送られ、DP(Display Port)ケーブルで解析用PCへと送る



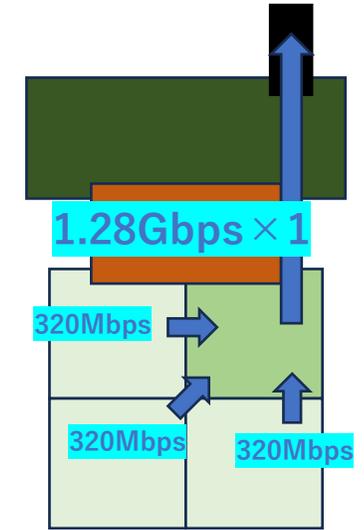
No Link Sharing

ケーブルとChipを1対1
転送rateは5.12Gbps
物質量は最も多い
ITk1層目で用いられる



50% Sharing

ケーブルとChipを1対2
転送rateは2.56Gbps
ITk2,3層目で用いられる



25% Sharing

ケーブルとChipを1対4
転送rateは1.28Gbps
物質量は最も少ない
ITk4層目以降で用いられる

データエンコーディング

- モジュールのデータは、Chip上で**シリアライザー(スクランブラー)**により順番処理された後でフレキを通して**データストリーム**として外部へ読み出され、FPGA上の**デシリアライザー(デスクランブラー)**によりChip毎のデータとして読み出す。

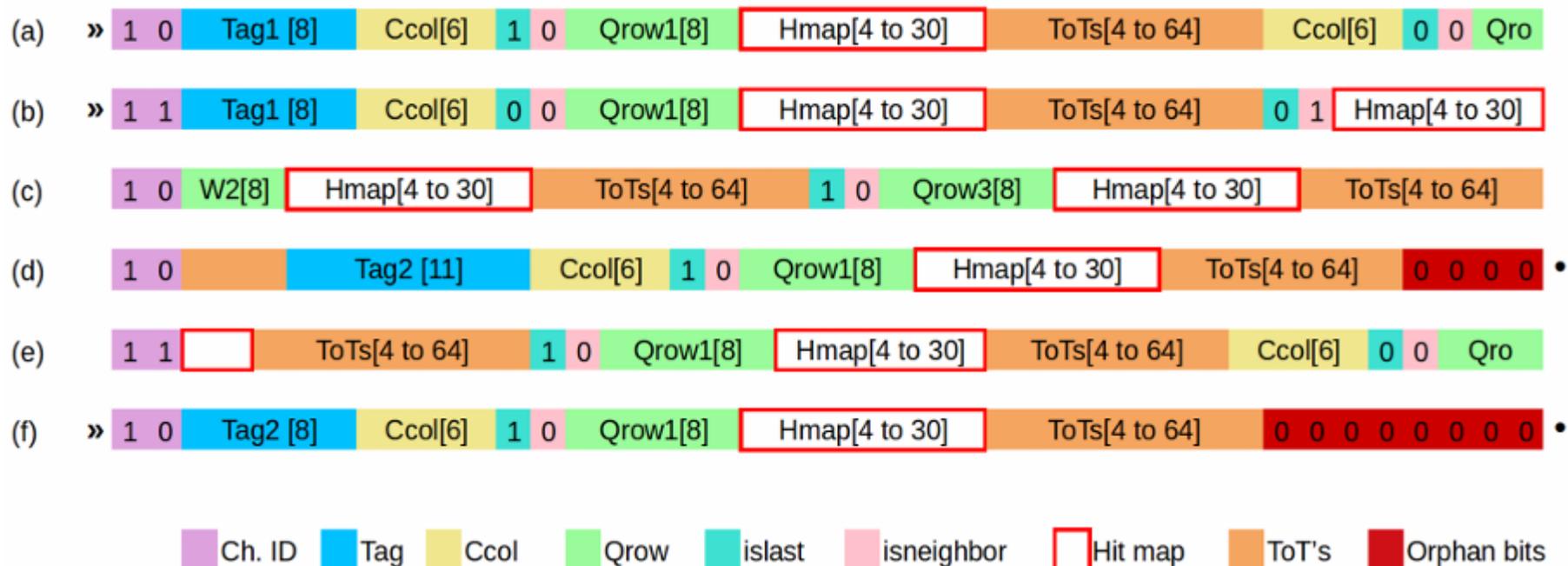


Fig:順番処理後に出力されるデータストリーム(2Chipマルチ)の例

データエンコーディング

- データストリームはたくさんの**データブロック**から成る。
- データフォーマット(データが出力される形式)は、**N個のデータブロックと1個のサービスブロック**が定期的に出力される。

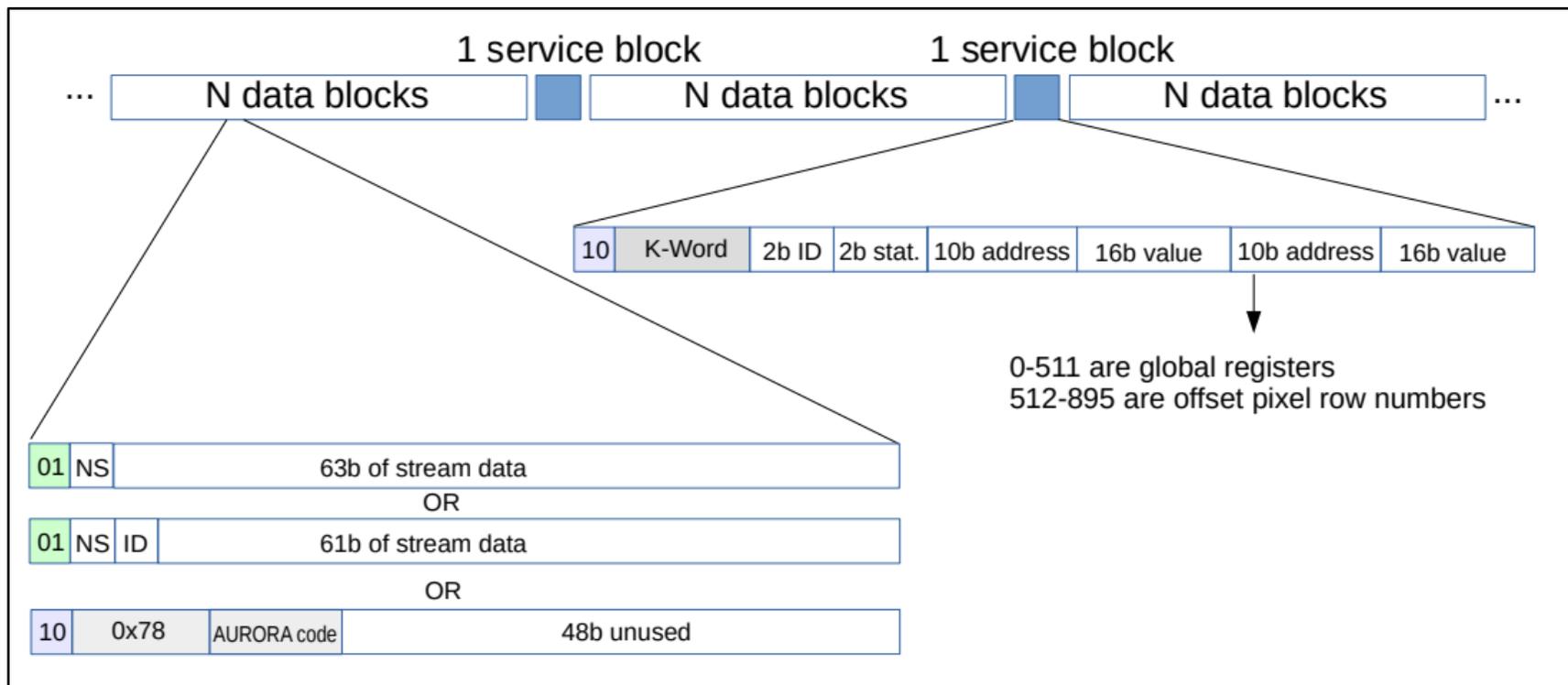


Fig:データブロックとサービスブロックから成るデータフォーマットの例

Eye Diagram scan

- **Eye Diagram**

…デシリアライザー上の4つのChipに対応するLaneのdelay parameterを0~31[ns]の範囲で変えながらデータフォーマットを10e5 blockで分割したときのidle blockを数え上げるscan

※10e5 blockで分割したとき、20420 or 20421 blockはidleであることが既知(定期的にservice blockが存在するため)

- Eye Diagramでは4Chip分の情報が含まれるデータストリームを適切にChip毎のデータとして収集するため、**デシリアライザーの4つのLinkに与える適切なDelay値を探る**

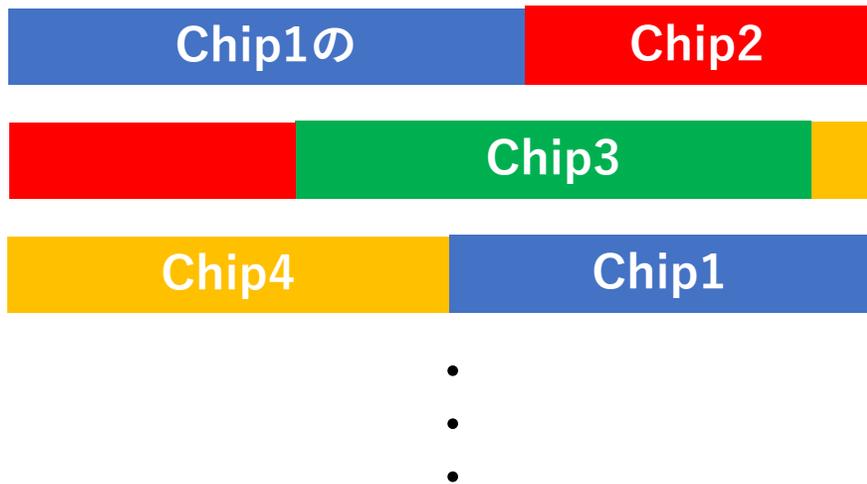


Fig:4Chip分のデータストリーム

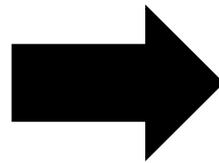
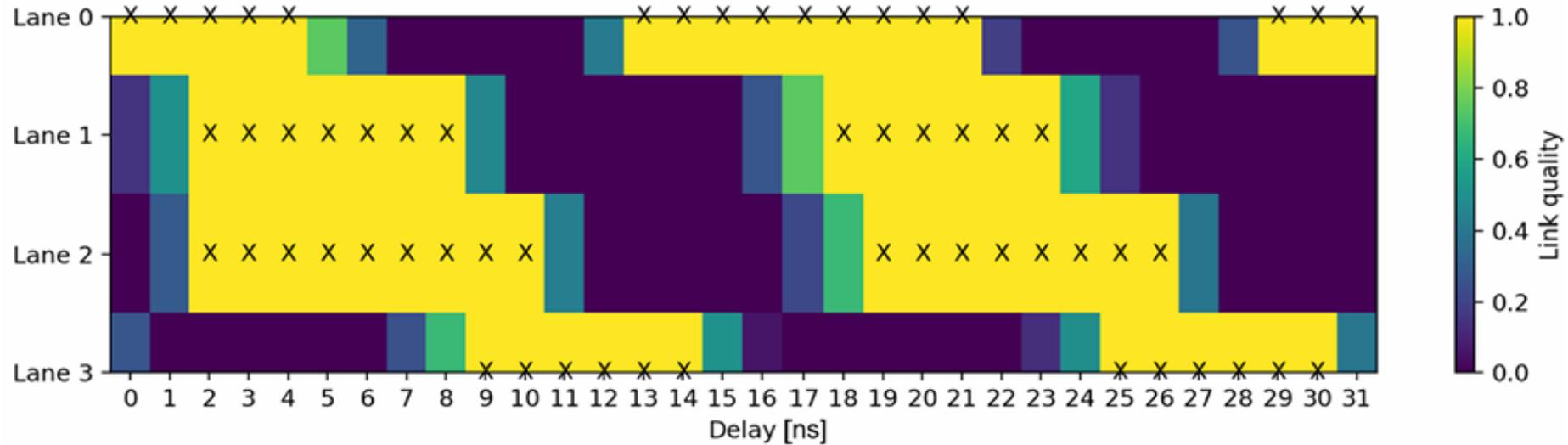


Fig:適切にデシリアライズされた後

Eye Diagram scan

【結果】



X軸 : Delay [ns]

Y軸 : lane(上からChip3,2,1,4に対応)

Z軸 : **Link quality**

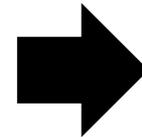
If(non idle block = 20420 or 20421)

Z = 1

else

$$Z = \log\left(\frac{1}{\frac{\text{abs}[\text{non idle} - 20421]}{20421}}\right) / 13$$

- Link quality=1の領域の中央値が適切なdelay値とする



lane0 : 17 ns

lane1 : 5 ns

lane2 : 6 ns

lane3 : 11.5 ns

Cmlドライバー

- Cmlドライバー

…主に減衰のしやすい高周波信号の減衰を補正する役割を担う

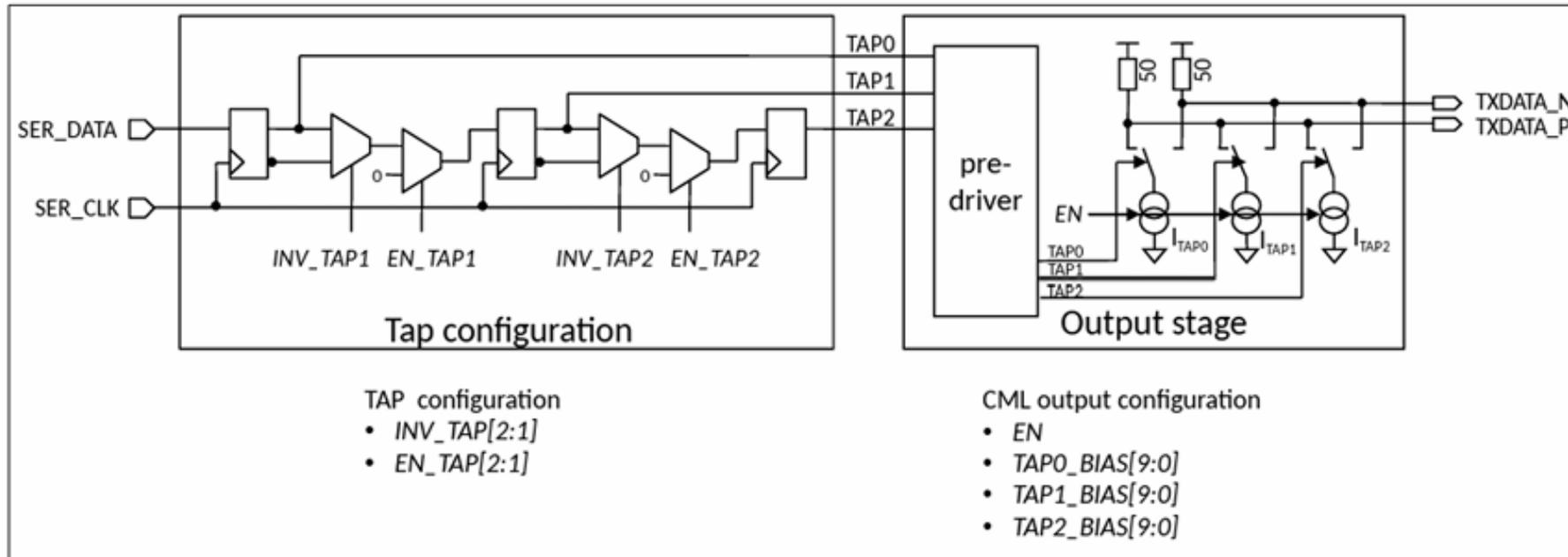


Fig:Cmlドライバー回路図

- 4つのINV_TAP[1,2]、EN_TAP[1,2]でTAP[0,1,2]_BIASのON/OFFを決める。
- TAP[0,1,2]_BIASの大きさはCmlbias[0,1,2]で調整可能

Cmlドライバー

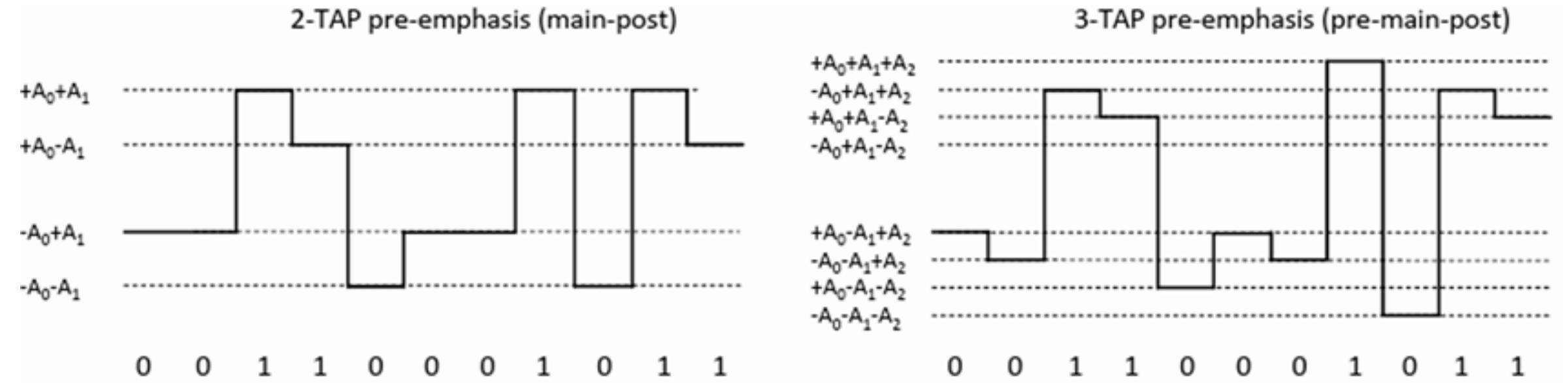


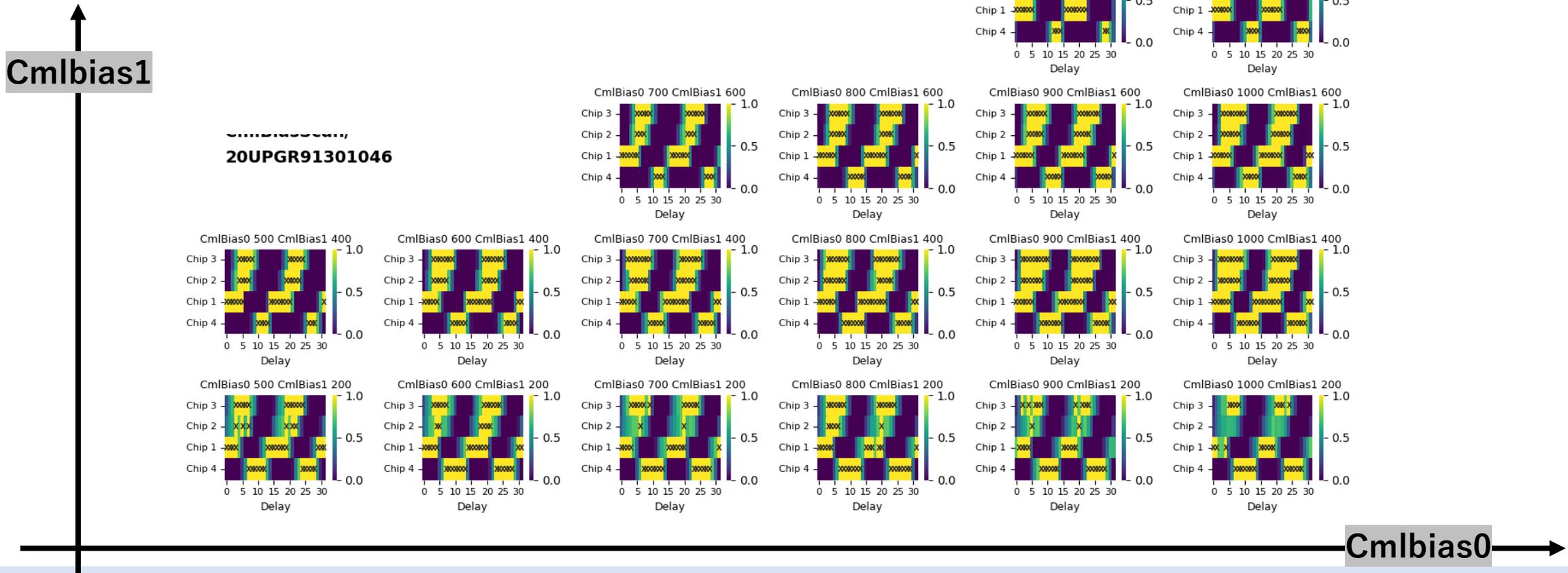
Fig:Cmlドライバーにより出力された信号波形の例

- 3つの振幅 A_0 、 A_1 、 A_2 はTAP[0,1,2]_BIASに対応(つまりCmlbias[0,1,2]にも対応)
- Mainは信号の立ち上がりと立ち下がり、Preは立ち上がり前のアンダーシュート、Postは立ち上がり後のオーバーシュートを制御

Cmlbias scan

- Cmlbias Scan

…データ転送に関する電圧のパラメータであるCmlBias0とCmlBias1を2次元的に変化させてそれぞれのパラメータでEye Diagramを行う



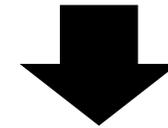
Cmlbias scan

【結果】

各Cmlbias条件で行ったEye Diagramの全Delay値のうち、Link quality=1だった割合[%]を示す

Cmlbias1	Cmlbias0										
	800	600	400	200	0	500	600	700	800	900	1000
800										27.3%	32.8%
600								31.3%	34.4%	38.3%	41.4%
400			32.0%	37.5%	39.8%	46.9%	46.9%	46.9%			
200			30.5%	29.7%	28.1%	25.8%	23.4%	22.7%			
0			0.0%	0.0%	0.0%	0.0%	0.0%	0.0%			

Cmlbias0=800~1000
Cmlbias1=400
が最も割合高い



Cmlbias0=800
Cmlbias1=400
(Defaultの設定)
に決定