

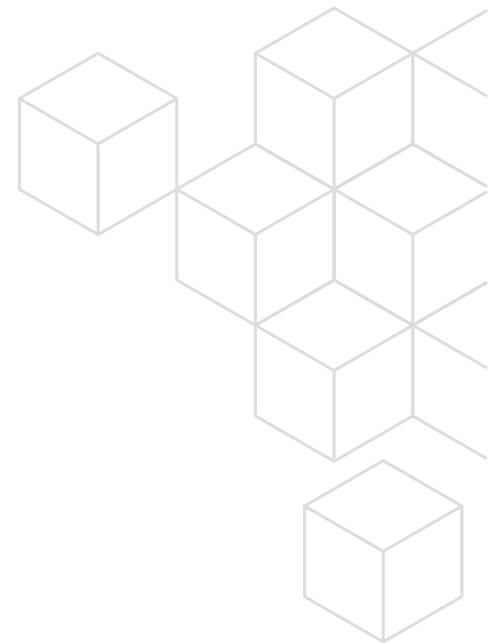
Backchannel Modeling and Simulation Using Recent Enhancements to the IBIS Standard

By Ken Willis, Product Engineering Architect; Ambrish Varma, Senior Principal Software Engineer; Dr. Kumar Keshavan, Senior Software Architect; Cadence

IBIS スタンドアートの近年の機能改善ではバックチャネル・トレーニングがサポートされ、IBIS-AMI モデルを使用し実世界の SerDes の動作をエミュレートできるようになりました。AMI モデラーは、バックチャネル・アルゴリズムを IBIS-AMI モデルに組み込むことが可能となり、実際の SerDes ハードウェア・デバイスと同様に、トランスミッタとレシーバーのイコライゼーションの設定の最適化を自動化することができます。これにより、システム設計者は、リンクに最適なイコライゼーションの設定を決定するための計算量の多いスイープをさけることで、大幅な時間短縮になり、同時に、モデル化しようとするハードウェアとの整合性が取れた、より現実的で高品質な結果を得ることができます。

目次

Backchannel Basics	2
Backchannel Enhancements to the IBIS Standard	3
How Backchannel Simulations Work.....	4
1. The transmitter initiates the process.....	4
2. The receiver evaluates the waveforms	5
3. The receiver requests TX FFE adjustments.....	6
4. The transmitter adjusts and re-transmits.....	7
5. The backchannel training completes.....	7
Backchannel Simulation Example	7
Simulation Testbench Setup.....	8
Sweep Results.....	10
FFE Adaptation	12
DFE Adaptation.....	14
Pre- and Post-Cursor Roles Regarding ISI.....	15
Conclusion	17
References.....	17



Backchannel Basics

シリアル・リンクでは「リンク・トレーニング」と呼ばれるバックチャネルの機能は、シリアル・リンクのトランスミッタ (TX) のイコライゼーションの設定を自動的に調整して、シグナル・インテグリティ (SI) とビット・エラー・レート (BER) を最適化する起動です。この機能は、InfiniBand、Fibre Channel、SAS、10GBASE-KR、PCI Express® 3.0 および 4.0 などの一般的なシリアル・リンクの標準規格に存在しています[1]、[2]。バックチャネル・トレーニングを使用することで、RX と TX のイコライゼーションの設定を組み合わせることで最適化することが可能です。これにより、TX または RX のみを個別に最適化するよりも良い BER マージンが生成されます。これは、より高いデータ・レートの場合、マージンが厳しくなることから重要です。さらに、実際のハードウェアの動作をシミュレーションでより再現することが可能となり、設計者はシミュレーション結果に基づいてより優れた設計判断が可能となります。

ハードウェアでは、TX と RX はトレーニング・モードに入り、フィードフォワード・イコライゼーション (feed-forward equalization、FFE) 設定の調整方法について、TX は RX からの提案を受け入れます。

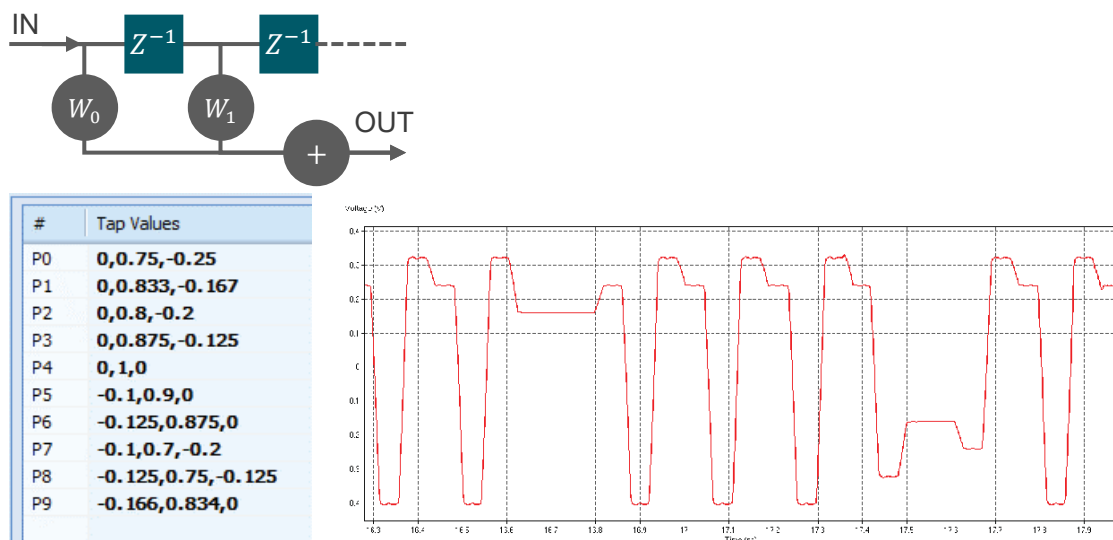


図 1: フィードフォワード・イコライゼーション

FFE は、送信信号の高周波を強調する（もしくは低周波を目立たせないようにする）ことに利用できる、複数の「タップ」ももしくはブースト・ドライバーで構成されます。これらのタップには、駆動力を制御するための係数があります。これらの係数には、順守する必要がある実質的な制限があります。

バックチャネル・トレーニング中、TX は RX にテスト・パターンを送信します。続いて、RX は送信された信号の品質を評価し、物理チャネル（即ち、バックチャネル）を介して TX と通信して、FFE の設定の調整方法を送信します。設定が調整されると、評価のために RX に新しいパターンが送信されます。このプロセスは、RX が信号品質に満足するか、トレーニング・プロセスの時間もしくはサイクルの制限に達するまで続きます。TX FFE の設定がロックされると、実際のデータの転送が開始されます。このプロセスを次の図に示します。

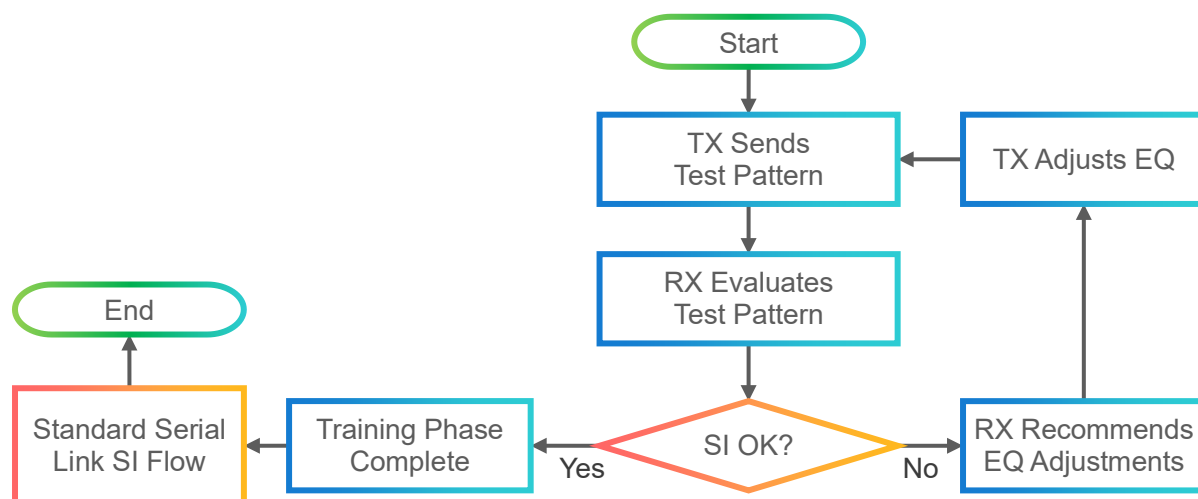


図 2: バックチャネル・プロセス

Backchannel Enhancements to the IBIS Standard

IBIS (I/O Buffer Information Specification) [3]は1993年に承認され、SIシミュレーションを実現するためにデバイス・モデルの標準となっています。2007年にAMI (Algorithmic Modeling Interface) セクションが仕様に追加され、最新のSerDes (シリアライザ/デシリアライザ) デバイスに見られるイコライゼーション機能に対応しました。これにより、業界に新しいレベルのシミュレーション機能がもたらされました。近年、バックチャネル・サポートが追加されたBuffer Issue Resolution Document (BIRD 147) [4]が、IBISの仕様の次の改訂版に入ることが承認されました。ケイデンスは社内外の顧客の要望に基づいてこの機能を早期にサポートするために組み込みました。

バックチャネル・サポートのIBISへの組み込みは簡単なファイル・インターフェイスとして定義されています。これは、関連するIBIS-AMIモデルのamiファイルに5つの新しいReserved_Parametersの追加により構成されています。バックチャネル・トレーニングをサポートするTXモデルとRXモデルの両方に、以下のReserved_Parametersが含まれています。

▶ **BCI_Protocol** : 使用されている標準プロトコルの名前

リンク・トレーニングを有効にするには、TXおよびRXの両方で同じプロトコルを指定する必要があります。モデルは複数のプロトコルのサポートが可能です。現在のバージョンの仕様では、プライベート・プロトコルのみを定義することができます。将来、モデルがサポートする可能性のあるバックチャネル・トレーニングのために、仕様にはIBISが承認したプロトコルを公開する規定があります。

▶ **BCI_ID** : TXとRX間で受け渡されるファイルに使用されるプレフィックス

BCI_IDと関連ファイルの規則はプロトコル依存で、BCI_Protocolに記述されます。この調査で使用したプライベート・プロトコルは、TXおよびRXがcdns_bciのBCI_IDで同じファイルを読み書きする必要があります。

▶ **BCI_State** : モデルの状態

パラメータの可能な値は、*off*, *training*, *converged*, *failed*, *error*です。TXとRXの両方で、BCI_Stateパラメータを更新することが可能です。EDAツールは、BCI_Stateが収束した場合、トレーニングを終了し、従来のチャネル・シミュレーションを継続することが可能です。

レシーバーAMI モデルは、バックチャネルをサポートするために以下の 2 つの追加パラメータが必要となります。

- ▶ `BCI_Message_Interval_UI`: TX AMI モデルと RX AMI モデル間の受け渡される波形で使用される推奨される `AMI_GetWave` ブロック・サイズ
- ▶ `BCI_Training_UI`: バックチャネル・トレーニングを評価するための UI の最大数

EDA ツールはこのパラメータを使用して、両方のモデルが相互に収束状態になっていない場合にリンク・トレーニングを終了させます。これは、トレーニング中の暴走ルーチンや無限ループを制限するために重要です。

バックチャネルに対応するために、TX と RX の両方の IBIS-AMI モデルでは関連する AMI の `Reserved_Parameters` パラメータが含まれているいることが必要です。

How Backchannel Simulations Work

IBIS に準拠したチャネル・シミュレータでバックチャネル・トレーニングを有効にするには、いくつかの簡単な前提条件を満たす必要があります。

- ▶ TX と RX の両方に IBIS-AMI モデルがアサインされている
- ▶ これらの IBIS-AMI モデルで、TX のモデルと RX のモデルの両方で、`BCI_State` に *Training* が設定されている
- ▶ TX のモデルと RX のモデルの両方に同じ `BCI_Protocol` の設定が与えられている

上記の条件が満たされると、バックチャネル・トレーニングが有効となり、TX FFE の調整と設定が可能となります。トレーニングが完了すると、FFE の設定がされた状態でチャネル・シミュレーションが動作し、データ転送がシミュレーションされます。

以下のセクションでは、バックチャネルが初期化されたときにシミュレーション時に発生する 5 段階のイベントのシーケンスについて説明します。ここでは、例題として、ケイデンスが開発した新しい IBIS ベースのバックチャネル機能を持った PCI Express 4.0 モデルを使用します。バックチャネル・トレーニング機能を持った IBIS-AMI モデルを有効にするには、以下の 5 つのステップを実行するためにコードの編集が必要となります。

1. The transmitter initiates the process

バックチャネルを有効にすると、TX はテスト・パターンとして定義されたチャネルを介して 1 と 0 の組み合わせのパターンをレシーバーに送信します。このテスト・パターンは `BCI_Protocol` により予め定義することや、シミュレーション・ツールにより設定された任意の PRBS のステイミュラス・パターンを使用して実行できます。この実行で使用されるモデルの場合、TX はシミュレーション・ツール (Cadence® Sigrity™ SystemSI™テクノロジー) で生成された PRBS パターンを利用し、リンク・トレーニングに他の特定のパターンを必要とはしません。

続いて、TX はその機能について RX に通信します。これは、ファイル・インターフェイスを介して実行されます。最初に TX は、両方の AMI モデル内で定義されている `BCI_ID` のプレフィックスを使用して、このファイルを書き出します。このファイルは RX モデルにより読み込まれ、TX タップの状態を確認します。どちらのモデルも同じ `BCI_Protocol` をサポートし、このプロトコルがファイルの読み書きと内

容のルールを決めるため、RX モデルはこのファイルを開きます。

以下は、ケイデンスの TX AMI モデルにより出力された初期 BCI ファイルの例です。

```
(amitx
  (tapincdec
    (-1 0)
    (0 0 )
    (1 0 )))
```

この簡単な 2 列の書式は、各行が <tap_id> <tap_state> の数字の組み合わせで設定されて、内容は以下を表しています。

タップ	数値	内容
tap id	-1	プリカーソル・タップ
tap id	0	メイン・タップ
tap id	1	ポストカーソル・タップ
tap state	0	調整のため、タップがオープンとなっている
tap state	1	タップがリミットの最大値に達している
tap state	-1	タップがリミットの最小値に達している

表 1: tap_id と tap_state

TX は、RX にタップ数がいくつあるか、タップの構成（即ち、存在するプリカーソルとポストカーソルのタップ数）、ストレングスの調整にこれらをオープンにするかなどを伝えるためのテキスト・ファイルを書き出します。上記の例では、3 つのすべてのタップは、どちらの方向でも調整できるようにオープンになっています。

RX モデルは TX モデルの制限を知る必要があるため、タップを調整するルールは TX AMI モデルに記述する必要があります。（この例では、TX および RX モデルのどちらもケイデンス AMI モデルなので、これは問題になりません。）

2. The receiver evaluates the waveforms

続いて、レシーバーは受信した波形の信号品質を評価します。これを実行するための事前に定義された方法がなく、RX AMI モデルは単にブラック・ボックスです。受信波形を受信し、受信波形に AMI イコライゼーション機能を適用し、それらのイコライズした波形を出力します。続いて、定義されている独自のアルゴリズムを用いて、結果の信号品質を評価します。

SerDes レシーバーが適用することができるアルゴリズムの 1 つの例は、図 3 のエラー・サンプリングに基づく方法です。レシーバーはアイの密度の電圧分布の中央値をモニターすることができ、サンプリング・ポイントで実際の波形電圧が効果する場所と中央値との差分を測定することが可能です。差分は「誤差」の値として考えることができます。正の値は中央値より上の値を意味し、負の値は中央値より下の値を意味して、どちらかの値を取ります。

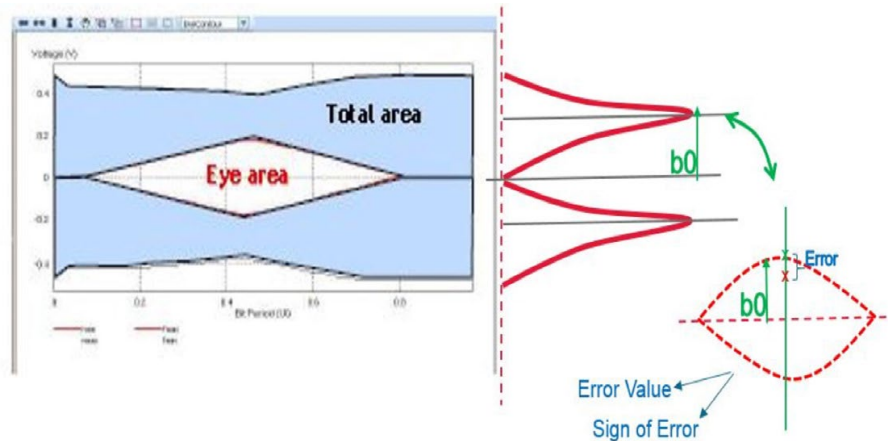


図 3: レシーバーの信号品質の評価

DEF のエラー・サンプリング・アルゴリズムは、RX のみでポストカーソル・タップを訂正しますが、サンプラーは、プリカーソル・タップから全体的な誤差への寄与について、以前のサンプルから十分な情報を得ています。続いて、この情報を用いて訂正すべき方向を判断し、この情報を TX に伝えます。この訂正は、安定させタップの値のディザリングを低減するために、一定の時間区間にわたり平均化される必要があります。

一般的なサンプラー・アルゴリズムは、以下の式に基づいています。

$$UP_DN_n = f_n(\epsilon_0, S_n)$$

ここで、

- UP_DN_n は n 番目のタップのアップもしくはダウンの補正です
- ϵ_0 は、目的の分布の中央値 (b_0 は図 3 を参照) に基づく現在の場所の誤差です
- S_n は n 番目のタップの状態 (例えば NRZ [Non Return to Zero] の場合は -1 や 1) です

この機能の簡単な組み込みは、 UP_DN の取り得る値として -1 、 0 、 1 があり、これは、TX に対して n 番目のタップの値の減少や維持、もしくは、増加させることを示します。

これは、RX で処理される波形の信号品質を決定する 1 つの方法に過ぎません。バックチャネル・トレーニングでは、各 RX は波形を解析し信号品質を改善するためにさらに TX タップを修正するかを判断する必要があります。

3. The receiver requests TX FFE adjustments

TX FFE の設定の現在の状態を評価し知ること、RX は次のトレーニングのトランザクションのための FFE の設定について TX に調整してほしい内容を書き出します。この例題で使用されているモデルでは、RX モデルは、もともと TX から取得した同じファイルの 2 列目の内容を変更し上書きすることで、これを実行します。

以下は、ケイデンスの RX AMI モデルにより出力された初期 BCI ファイルの例です。

```
(amirx
 (tapincdec
  (-1 1)
  (0 0 )
  (1 -1 )))
```

簡単な 2 列の書式は、各行が<tap_id> <tap _ equalization _ directive>の数字の組み合わせで設定されて、内容は以下を表しています。

タップ	数値	内容
tap id	-1	プリカーソル・タップ
tap id	0	メイン・タップ
tap id	1	ポストカーソル・タップ
tap equalization directive	0	調整をしない
tap equalization directive	1	イコライゼーションの増加
tap equalization directive	-1	イコライゼーションの減少

表 2: tap_id と tap_equalization_directive

RX は、TX に実行してほしい FFE の調整を示すように、元の BCI ファイルを上書きします。TX と RX は同じ BCI_Protocol プロトコルをサポートしているので、イコライゼーションに変更を要求したときに、RX は TX のタップの値がどの程度上下するかを理解しています。

4. The transmitter adjusts and re-transmits

TX は新しく書き込まれた BCI ファイルを読み込み、TX の FFE タップの設定を調整し、RX が評価するための新しいパターンを送信します。

5. The backchannel training completes

上述した全体的なプロセスは、以下の 2 つのうち 1 つが生じるまで複数回実行されます。

- ▶ RX がこれ以上調整の必要ないことを示す内容を BCI ファイルに書き出し（即ち、BCI ファイル内のすべての tap_equalization_directives が 0 として書き出され）、Reserved_Parameter BCI_State が *converged* に変更される
- ▶ BCI_Training_UI の上限に達する

いずれの場合も、上記の 2 つの条件のうち 1 つが満足され場合、バックチャネル・トレーニングが終了され、TX の FFE の設定がロック・ダウンされ、シミュレーション・ツールによってすべてのチャネル・シミュレーションが実行されます。

Backchannel Simulation Example

信号品質に対するバックチャネル・トレーニングの効果を実験するために、以下のテストベンチを構築しました（図 4）。

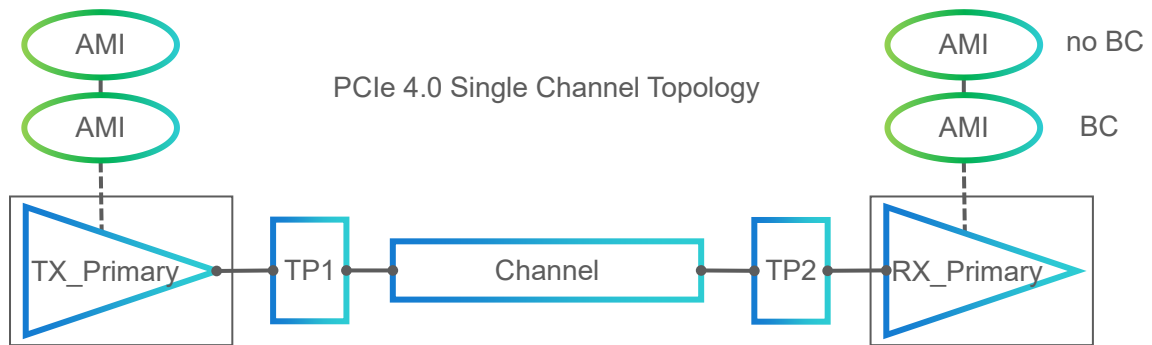


図 4: シミュレーション・テストベンチ

Simulation Testbench Setup

チャンネル・ブロックは、物理的な長さが 2 から 40 インチの範囲で 16Gbps の PCI Express 4.0 のデータ・レートのナイキスト周波数の 8GHz において -1.4dB から -21.7dB の挿入損失を表す散乱パラメータ (S パラメータ) 群として与えられています。これらの S パラメータの波形を以下に表します。

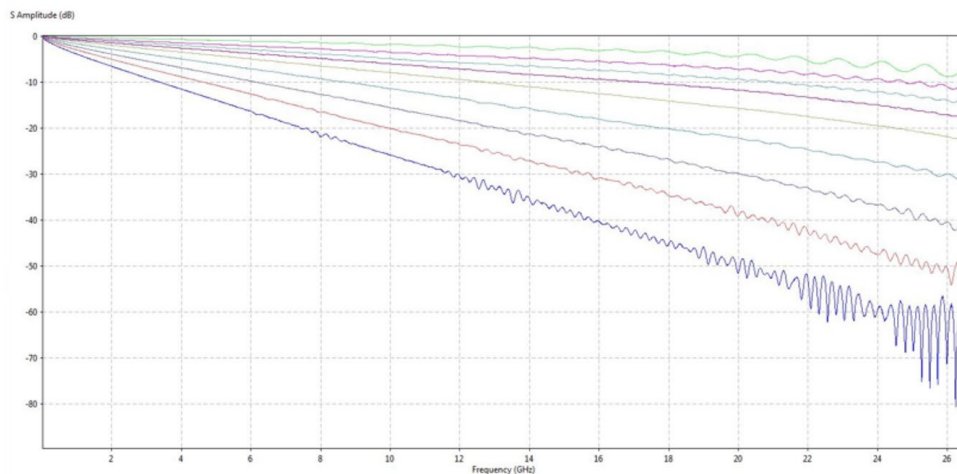


図 5: チャンネル S パラメータ

このテストベンチでは、IBIS-AMI モデルの 2 つのセットが含まれています。それらはそっくりで、これらには同じ仕様レベルの PCI Express 4.0 の機能が含まれています。違う点は、1 つのセットは IBIS ベースのバックチャネル機能が有効であるのに対して、もう 1 つのセットは有効ではないことです。

以下の波形は、トポロジー内のチャンネル・モデルのステップ応答を示しています。

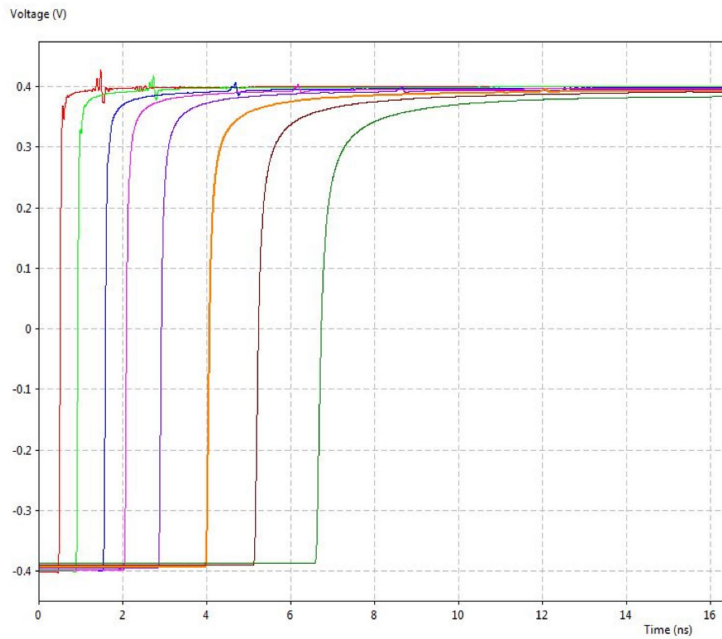


図 6: チャンネルのステップ応答

関連するインパルス応答を以下の図に示します。

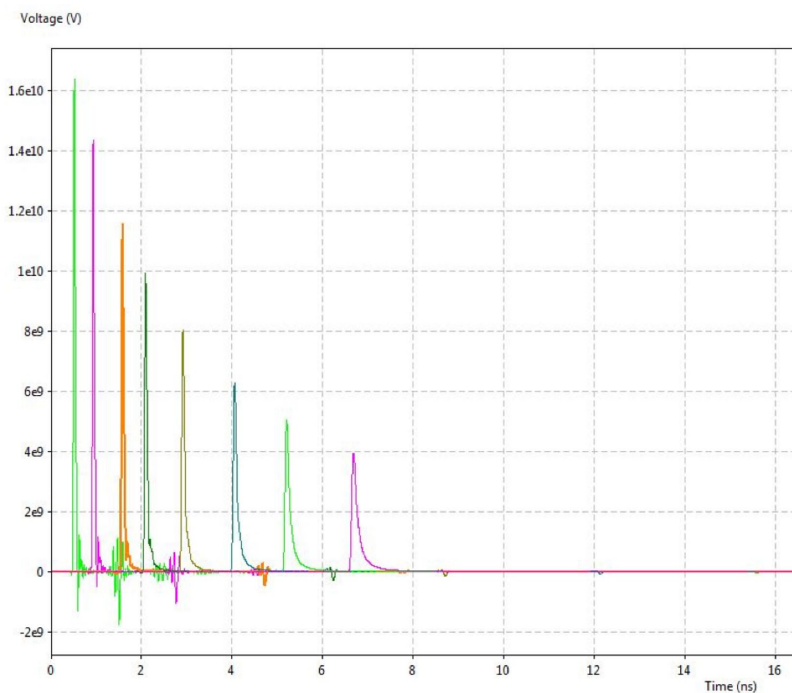



図 7: チャンネルのインパルス応答

上述のプロットから観測できるように、ショート・チャンネルでは顕著な反射のふるまいが観測され、一方、ロング・チャンネルはほとんど損失が支配的で、反射の振幅はレシーバーに到達するまでに大きく減衰されることが観測されます。

Sweep Results

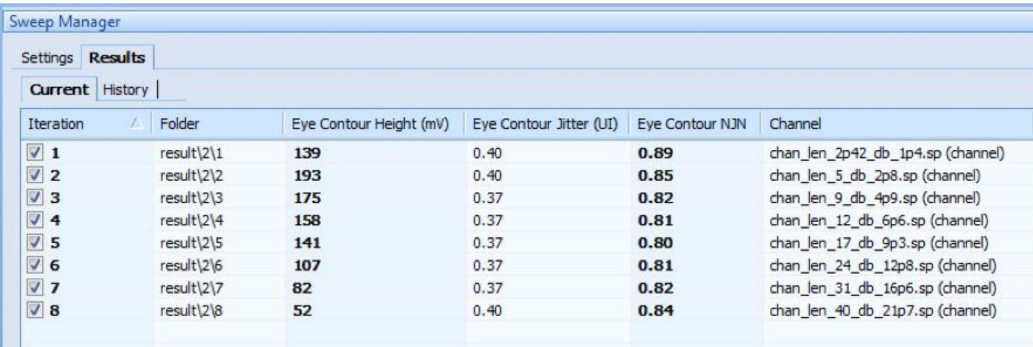
テストベンチで、2つのチャンネル・モデルのセットのスweepが設定されています。スweepの両方のセットは、レシーバーでアダプティブな2タップのDEF（decision feedback equalization、判定帰還型等価化）を持っています。スweepの最初のセットでは、TX FFEは、チャンネルのインパルス応答に基づいてタップ係数を自己最適化することが可能でした。これは、LMS（least mean square、最小二乗平均）アルゴリズムを用いて、与えられたチャンネルのアイのオープンに最適なタップ係数を計算します。これは、RXイコライゼーション機能の知識が無くても、TXの観点から厳密に実行されます。これにより、以下の結果が得られました。



Iteration	Folder	Eye Contour Height (mV)	Eye Contour Jitter (UI)	Eye Contour NJN	Channel
1	result\1\1	117	0.43	0.91	chan_len_2p42_db_1p4.sp (channel)
2	result\1\2	157	0.43	0.86	chan_len_5_db_2p8.sp (channel)
3	result\1\3	140	0.37	0.83	chan_len_9_db_4p9.sp (channel)
4	result\1\4	123	0.43	0.84	chan_len_12_db_6p6.sp (channel)
5	result\1\5	107	0.43	0.83	chan_len_17_db_9p3.sp (channel)
6	result\1\6	88	0.40	0.83	chan_len_24_db_12p8.sp (channel)
7	result\1\7	59	0.46	0.86	chan_len_31_db_16p6.sp (channel)
8	result\1\8	40	0.46	0.88	chan_len_40_db_21p7.sp (channel)

図 8: バックチャンネル無しのスweepの結果

スweepの2つ目のセットでは、バックチャンネルが有効です。レシーバーの知識なしにTXにその係数を自己最適化させるのではなく、RXは、TXによってチャンネルを介し送信されたトレーニング・パターンを評価し、TXにその等価のための調整を実施するように指示することが可能です。



Iteration	Folder	Eye Contour Height (mV)	Eye Contour Jitter (UI)	Eye Contour NJN	Channel
1	result\2\1	139	0.40	0.89	chan_len_2p42_db_1p4.sp (channel)
2	result\2\2	193	0.40	0.85	chan_len_5_db_2p8.sp (channel)
3	result\2\3	175	0.37	0.82	chan_len_9_db_4p9.sp (channel)
4	result\2\4	158	0.37	0.81	chan_len_12_db_6p6.sp (channel)
5	result\2\5	141	0.37	0.80	chan_len_17_db_9p3.sp (channel)
6	result\2\6	107	0.37	0.81	chan_len_24_db_12p8.sp (channel)
7	result\2\7	82	0.37	0.82	chan_len_31_db_16p6.sp (channel)
8	result\2\8	52	0.40	0.84	chan_len_40_db_21p7.sp (channel)

図 9: バックチャンネルを有効にしたときのスweepの結果

バックチャンネルを有効にしたときの結果は、アイの高さ、ジッタ、NJN（Normalized Jitter and Noise）などの測定全体を使用した場合より、例外なく優れていました。NJNは正規化されたジッタおよびノイズとして定義された信号品質の測定基準です。これは、1ユニットのインターバル（UI）のノイズとジッタの領域を合計のアイの領域で割ることにより計算されます（図 10）。

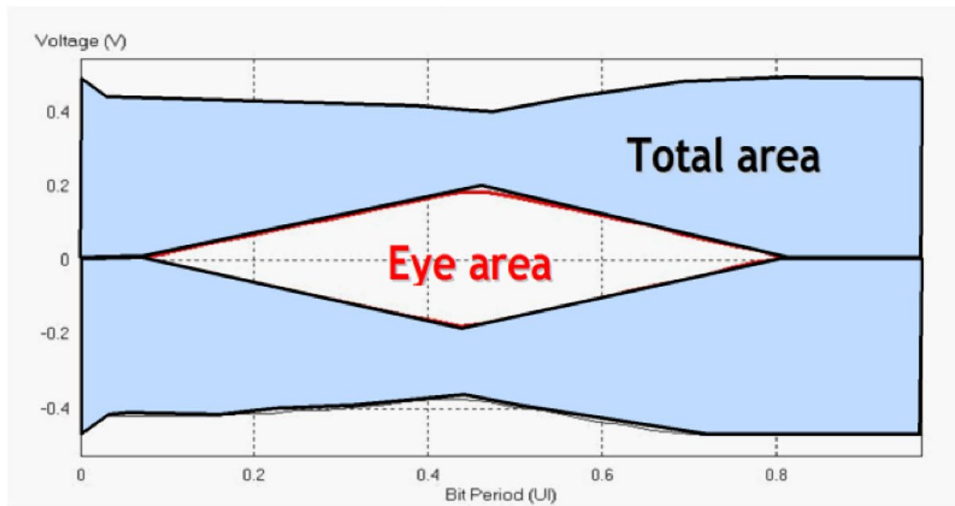


図 10: NJN

以下のプロット（図 11、図 12）は、チャンネルの挿入損失のスペクトラムにわたる 3 つの測定基準（アイの高さ、幅、NJN）の改善率を示しています。

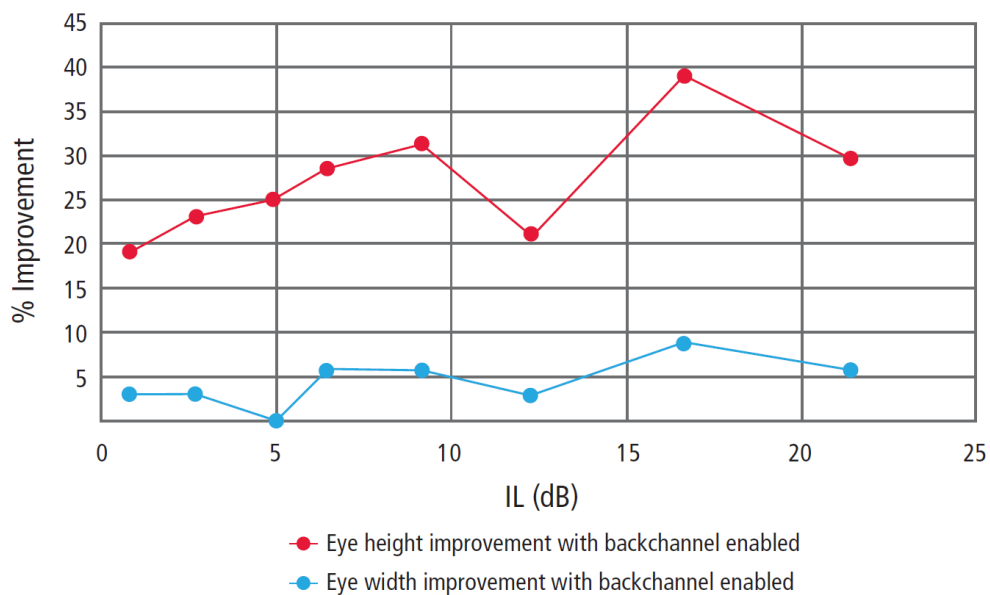


図 11: 挿入損失に対するアイの高さと幅の改善率

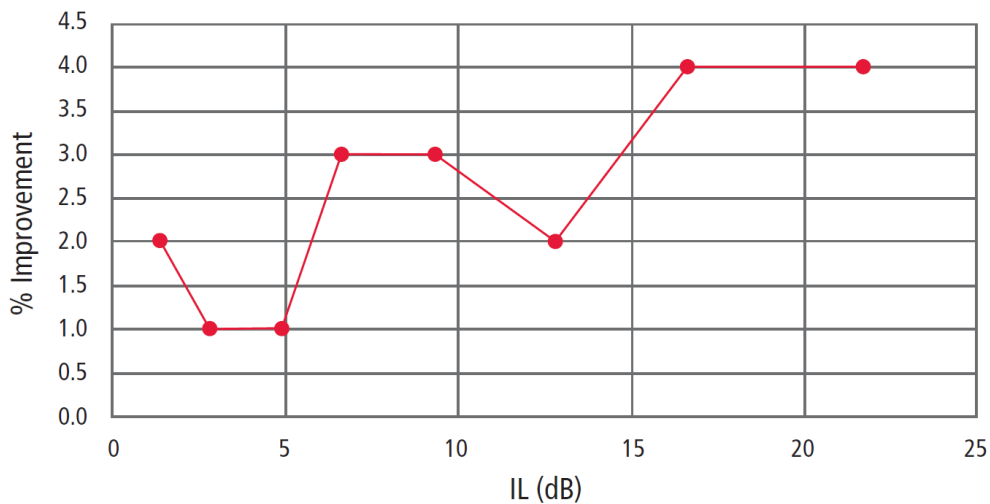


図 12: 挿入損失に対する NJN の改善率

各ケースで全ての測定基準が改善され、大幅な改善が確認できます。改善の重ねた結果を表示するために、各測定基準の改善率を各ケースで平均化しました。このプロットを以下に示します。

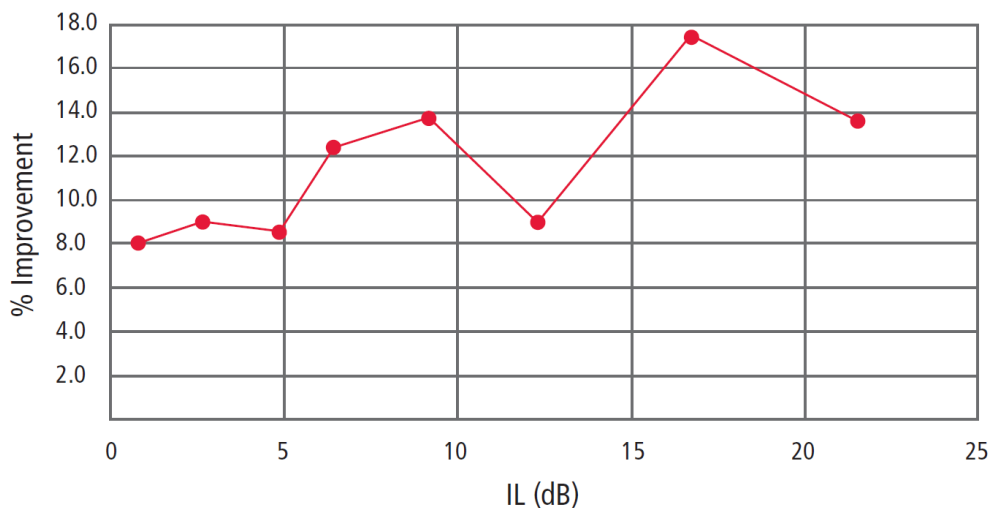


図 13: 挿入損失に対する複合した改善率

一般的な傾向としては、イコライゼーションが損失の補償の限界に達し始めるまで、チャンネルは損失が多く反射が少ないほど、バックチャンネルからの改善が増加します。これにより、イコライゼーションはチャンネルの損失を補償するように設計されており、通常、反射の抑制には効果が低いことを、直感的に理解できます。

FFE Adaptation

バックチャンネル・トレーニングの効果をさらに理解するために、これらのケースのうちの1つをさらに詳しく確認します。31 インチのチャンネルでは、アイの高さで 39%の改善、ジッタで 9%の改善、NJN で 4%の改善が確認できます。

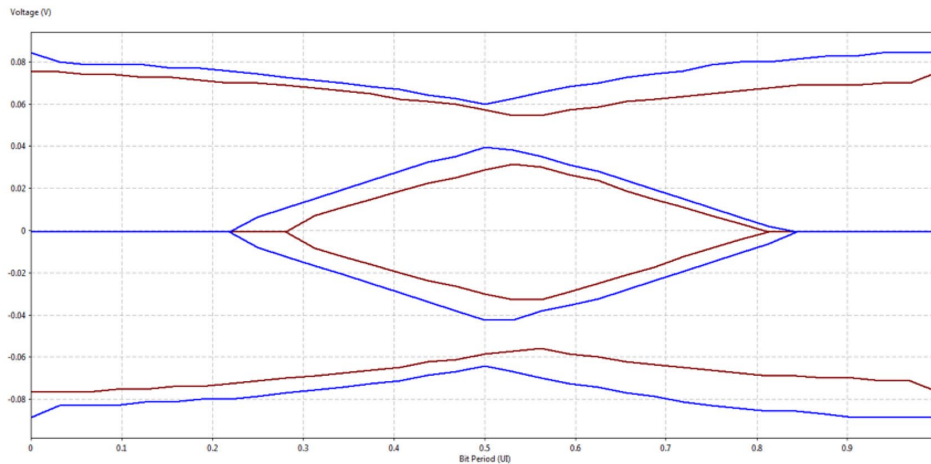


図 14: 31 インチのチャンネルの Eye Contour プロット
バックチャンネルあり (青)、バックチャンネルなし (赤)

自己最適化およびバックチャンネルのいずれのケースにおいて、このケースの FFE タップの係数は同じ値で開始されます。「non-backchannel」(非バックチャンネル) のケースでは、チャンネル・シミュレーションの期間中、これらの係数を維持していましたが、バックチャンネル・トレーニングを有効にすると、これらの係数は大きく変化しました。以下のプロットは、これら FFE タップの係数です。プリカーソルとポストカーソルの FFE タップの係数は実際には負の値ですが、プロットでの比較を容易にするために、ここでは絶対値でプロットされていることに注意してください。

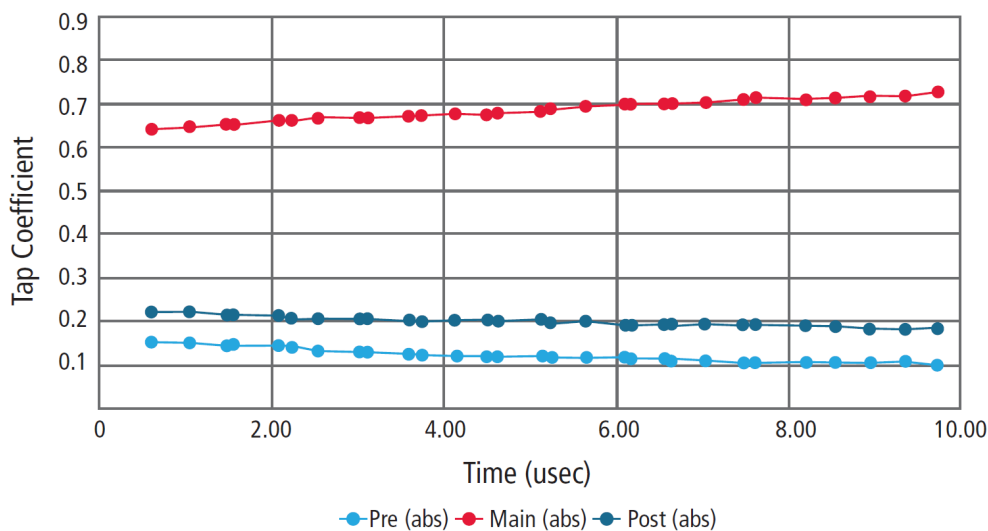


図 15: 適合された FFE タップの係数

プリカーソルとポストカーソルの FFE タップの係数の振幅はそれぞれ約 34%と 13%減少します。これは、バックチャンネル・トレーニング・プロセスの間中 FFE が基本的に「下降」し、RX により多くのイコライゼーションの作業とその DFE 機能を任せることを意味します。

DFE Adaptation

31 インチのチャンネルのバックチャンネルのオフとオンの両方のケースの DFE の係数をプロットしました。以下の図は、バックチャンネルを有効にしたときの DFE 適応のプロットです。

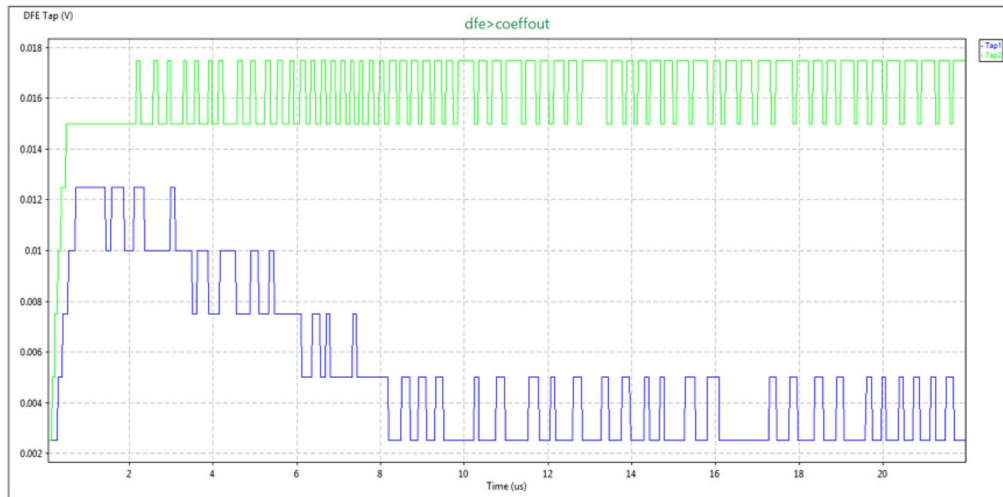


図 16: バックチャンネルを有効にしたときの DFE 適応
タップ1 (青) とタップ2 (緑)

次は、バックチャンネルを無効にしたときの結果です。

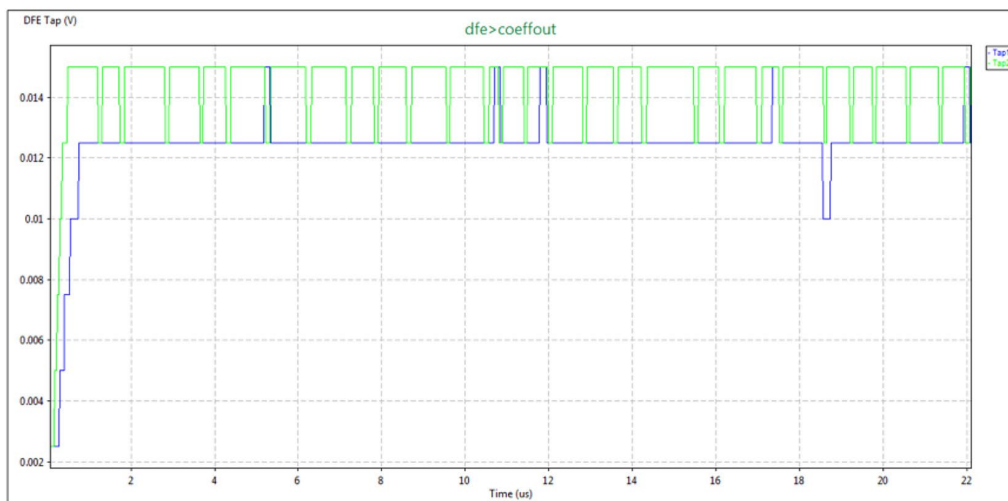


図 17: バックチャンネルを無効にしたときの DFE 適応
タップ1 (青) とタップ2 (緑)

図 16 のバックチャンネル・トレーニングを有効にした場合、最初、DFE タップの係数は低く始まり、高いレベルに上昇し、続いて、低いレベルに低下することが確認できます。これは関心深い動作です。最初、FFE のポストカーソルのタップは信号を実際に過剰に均等化し、1 番目の DFE タップはそれを補正しようと試みています。そのポストカーソルのイコライゼーションを下げるように FFE と通信が始まると、

この DFE タップは低いレベルで安定します。これは、図 17 とは対照的です。DFE タップは高くなり、その状態を続け、シミュレーションの期間中に FFE ポストカーソルの過剰なイコライゼーションとの調整を続けます。

両方のケースで報告された最終的な DFE 係数を以下に示します。

ケース	DFE タップ 1	DFE タップ 2
31" channel, BC off	0.0125	0.015
31" channel, BC on	0.0025	0.0175
% difference	-80.0	16.7

表 3: バックチャネルの有りと無しの場合の最終的な DFE 係数

この場合、FFE は徐々に下降しますが、1 番目の DFE タップは大幅に下降し、2 番目の DFE タップはほぼ 17%増加しています。

Pre- and Post-Cursor Roles Regarding ISI

FFE および DFE などのイコライゼーションの基本的な目的は、符号間干渉 (inter-symbol interference) つまり ISI をキャンセルすることの支援です。トランスミッタの FFE には、プリカーソルおよびポストカーソルの両方の ISI をキャンセルする機能がありますが、適応機能があり強力なレシーバーの DFE はポストカーソルの ISI に対応するだけに制限されています。これを念頭に置いて、主にプリカーソルの ISI の FFE に集中して、ポストカーソルの ISI のすべてを DFE に任せることで、何が起るかを考察することは興味深いことでした。これを有効にするには、FFE をプリカーソル・タップのみ可能になるように構成を変更し、バックチャネルを有効にして 31 インチのチャネルのシミュレーションを再実行することで、RX は TX にプリカーソルの設定を伝えるようになります。これにより、元のバックチャネルが有効な時のシミュレーションよりもさらに結果が改善されました。以下に結果を示します。

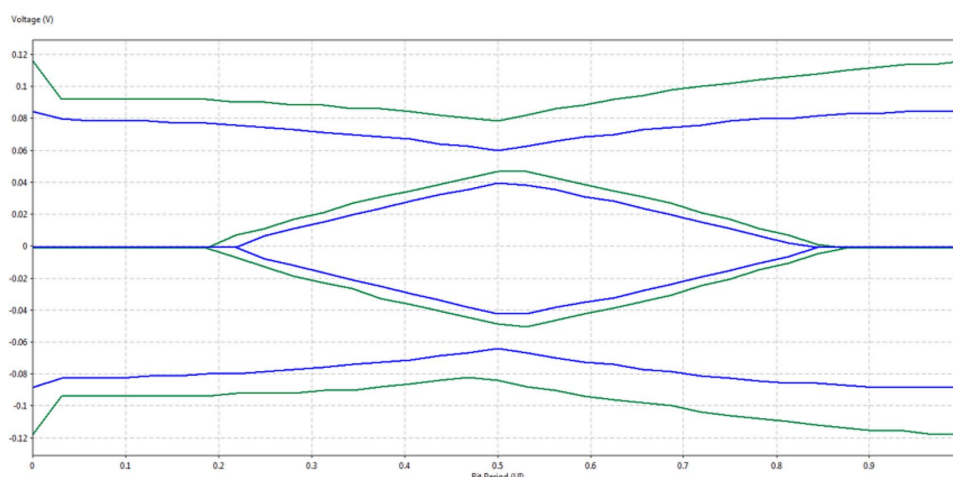


図 18: プリカーソル・タップに焦点をあてた FFE によるアイ開口部の改善

アイの高さは 82mV から 95mV に改善され、ジッタは 0.37UI から 0.31UI に改善され、NJN は 0.82 から 0.81 に改善されました。このシミュレーションからのプリカーソルの FFE タップ適応は以下のようになります。

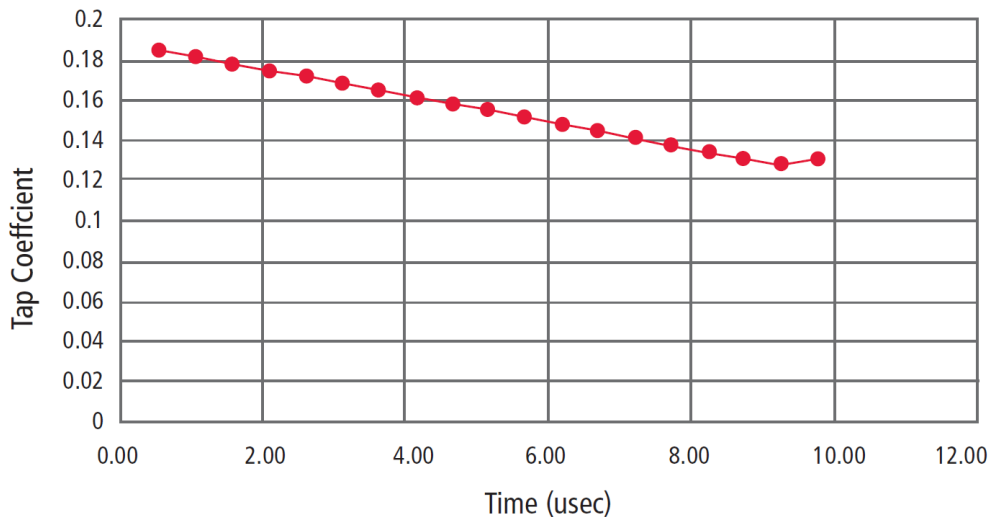


図 19: 「プリカーソルのみ」に FFE タップを適応

この場合、プリカーソル FFE タップは、-0.13 の値に調整され、これはプリカーソルとポストカーソルのタップの両方が有効な場合に生じる -0.094 の値とは対称的です。したがって FFE は前よりもさらにプリカーソルのイコライゼーションを実行します。図 20 に、この場合の DFE の適応の結果をプロットしています。

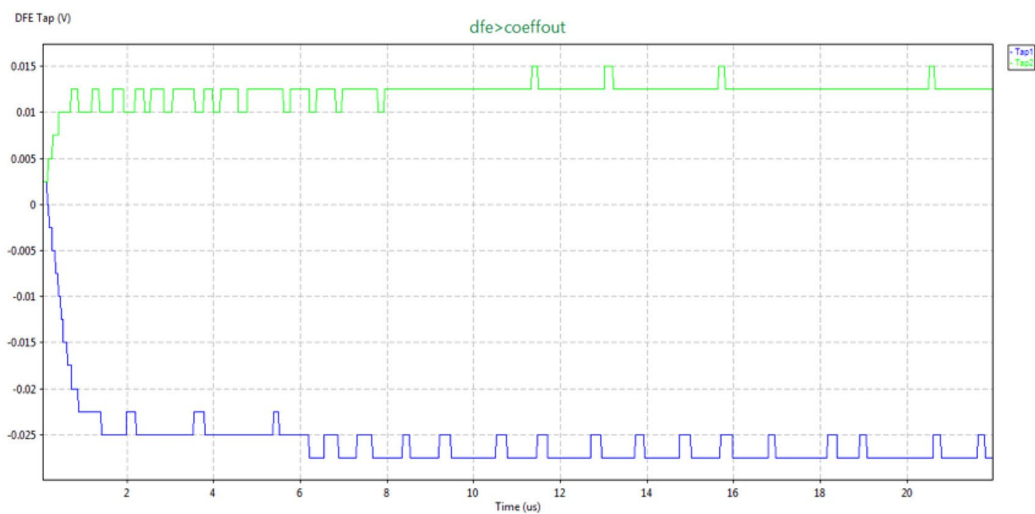


図 20: 「プリカーソルのみ」に DFE タップを適応

1 番目の DFE タップは 25mV から 30mV の値よりも高く調整されます。これは、FFE のポストカーソルのタップを有効にして使用されている場合よりもほぼ 1 桁高くなります。これは、DFE がポストカーソルのイコライゼーションが大幅に行われること示しています。続いて、2 番目の DFE タップは前のケースの約 16mV と比較して約 12.5mV と、前よりも低い値で安定し、ほぼ 28%減少しています。これは、イコライゼーションがより効果的に動作して、2 番目の DFE タップが本質的に実行する作業が少ないことを示しています。このケースでは、FFE をプリカーソル ISI に集中させ、ポストカーソルのイコライゼーションのすべてを DFE にまかせることで、全体的に最良な結果が得られました。

Conclusion

このホワイト・ペーパーでは、近年のハイスピード・シリアル・リンク・インターフェイスにおけるバックチャンネル・トレーニング機能の概要および、IBIS-AMI モデルを使用したチャンネル・シミュレーションにバックチャンネル・トレーニングを組み込むことを可能にする IBIS の仕様の最近の改善について論じました。バックチャンネル・トレーニングが組み込まれた複数のシミュレーション結果を紹介し、これらすべてにおいて、レシーバーとの通信が無い場合やトランスミッタがイコライゼーションの設定を自己最適化する場合よりも、大幅な改善を示しています。FFE (Feed-Forward Equalizer) と DFE (Decision Feedback Equalizer) によるプリカーソルおよびポストカーソルの役割について確認しました。IBIS の新機能を利用して IBIS-AMI モデルにバックチャンネル・トレーニングを組み込む方法についてのガイダンスも提供され、バックチャンネル機能を持つ SerDes デバイスの動作をより正確に表現することが可能となりました。

References

- 1 Mohammad S. Mobin et al., TX back channel adaptation algorithm and protocol emulation with application to PCIe, SAS, FC, and 10GBASE-KR, DesignCon 2012.
- 2 PCI Express Base Specification Revision 4.0 Version 1.0 September 27, 2017.
- 3 <http://ibis.org/ver6.1/>
- 4 <http://ibis.org/birds/bird147.6.docx>

