

EMCシミュレーションにLTspiceを使って最良の結果を得る方法 - パート2：信号の完全性を改善

Richard Anslow、システム・アプリケーション・マネージャ
Sylvain Le Bras、フィールド・アプリケーション・エンジニア

LTspiceを使用したEMCと信号の完全性の確認

これは、LTspice®を使用したEMCと信号完全性のシミュレーション・モデルを示す3つの記事で構成されるシリーズのパート2です。「EMCシミュレーションにLTspiceを使って最良の結果を得る方法 - パート1」では、電源コンポーネント、伝導エミッション、およびEMI耐性を確認するシミュレーション・ツールとしてLTspiceを解説しました。

パート2では、LTspiceとCベースのプログラムを組み合わせ、設計者が有線ネットワークによる信号の完全性を理解し、改善するのを助ける方法について述べます。これらのツールは、設計者が行う実験室でのテストの繰り返しや、高価なハードウェアの再設計を回避する助けとなります。フィールドバス通信（RS-485、RS-232）、高速バックプレーン（LVDS）、ユビキタスUSB標準、およびデータ・ラインを介した電源供給（PoDL）を可能にする

新しいシングル・ペア・イーサネット（SPE）への適合性を実現するために、各種のシミュレーション・モデルが提供されています。

設計者が信号の完全性に注意すべき理由

信号の完全性は、単にプロトタイプでリンクが機能することを確認すれば確保できるというものではありません。以下のような理由から、リンクが正常に機能するように見えても、信号品質についてはより詳細なチェックを行うのが得策です。

- ▶ 信頼性：距離、スループット、および環境によってはバイナリ・エラー・レートが増大して、そのシステムのエラー・コレクション・メカニズムによって受け入れ難い値に達してしまう可能性があります。

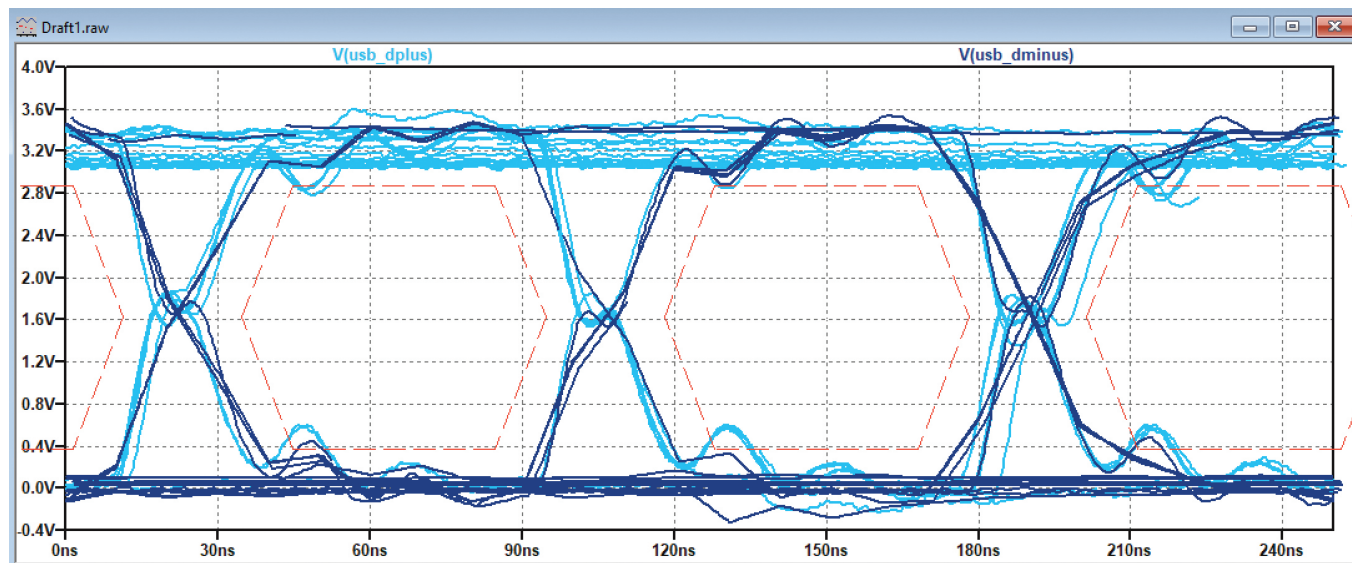


図1 LTspiceで行った分析の例

- ▶ IEEE 標準への適合：IEEE の推奨事項を満たしていれば、ネットワークのメンバー間でのシームレスな連携動作が確保されます。
- ▶ EMC 指令への適合：スルー・レートとオーバーシュートが大きい場合は高調波成分も大きくなるので、所定の要求を満たせないことが少なくありません。

LTspice を使って信号の完全性に関する問題を解決

本稿は、以下のような重要な疑問に答える助けとなります。

- ▶ このシステムは妥当な信号の安定性を確保できているのか。
- ▶ 最初に信号のどの側面を改善すべきなのか。信号のスルー・レート、リングング、ジッタ、マッチングのどれを改善すべきなのか。
- ▶ システムは正常に動作するが、時間、温度、およびコンポーネントの許容誤差範囲内でも一定の信頼性で機能するのか。

本稿では、以下のための方法を解説しています。

- ▶ LTspice 内で信号の全性を分析する
- ▶ 代表的テスト・ベクトルを作成、インポート、使用し、同様の分析を行って図1に示すような出力を得る
- ▶ 複数のパラメータをランダムに選んでシステムの統計的検証を行う

アイ・ダイアグラム・シミュレーションの実施

アイ・ダイアグラムは、トランスミッタ側またはレシーバ側の信号が基準に適合しているかどうかを評価する、便利な方法を提供します。アイ・ダイアグラムは信号を時間基準で表したものです。

この表示法は残光表示を使用して大量のシンボルを分析し、信号レベル、ジッタ、および立上がり時間が適切であることを確認します。

LTspiceはアイ・ダイアグラム分析に必要な機能をいくつか備えていますが、フル機能の分析を行うには追加的なステップが必要です。

テスト・ベクトルを使ってPWLファイルを作成

LTspiceは、シミュレーションでテスト・ベクトルを実行するための効率的な方法を備えています。不適合という結果に至る状況を正しく把握するには、大量のデータを使用することが不可欠です。

いくつかの不適合現象は、以下のような極めて特別な状況下で生じます。

- ▶ 連続ビットの長いシーケンス
- ▶ ゼロ以外のバランス・シーケンス
- ▶ 近傍にある他の伝送チャンネルからのクロストーク

これは例えば、ランダム関数を使ってデータが作成されている場合、ハイ・レベルが11個連続するという特別な状態を発生させるには、数千個のシンボルを使用しなければならない可能性があることを意味します。

LTspiceが想定しているPWL データ・フォーマットを図2に示します。

```
0.000000E+000→ 0.00CRLF
1.320000E-005→ 0.00CRLF
1.334583E-005→ 0.00CRLF
1.337083E-005→ 1.20CRLF
1.346667E-005→ 1.20CRLF
1.361250E-005→ 1.20CRLF
1.363750E-005→ 0.00CRLF
```

図2 PWLテスト・ベクトルのファイル・フォーマット

```
// initialize
fprintf (pFile, "%dn\t0\n", 0, initialVoltage);
fprintf (pFile, "%dn\t0\n", DELAY, initialVoltage);

// loop for every sample
for (sample = 0 ; sample < NUMBER_OF_SAMPLES ; sample++)
{
    // Compute a sample
    logicLevel = rand() % 2;

    // print sample
    if (logicLevel == 0)
    {
        fprintf (pFile, "%fn\t%5.2f\n", (sample*symbolTime)+DELAY+transitionTime, voltageLow);
        fprintf (pFile, "%fn\t%5.2f\n", ((sample+1)*symbolTime)+DELAY-transitionTime, voltageLow);
        printf ("%fn\t%5.2f\n", (sample*symbolTime)+DELAY, voltageLow);
    }
    else
    {
        fprintf (pFile, "%fn\t%5.2f\n", (sample*symbolTime)+DELAY+transitionTime, voltageHigh);
        fprintf (pFile, "%fn\t%5.2f\n", ((sample+1)*symbolTime)+DELAY-transitionTime, voltageHigh);
        printf ("%fn\t%5.2f\n", (sample*symbolTime)+DELAY, voltageHigh);
    }
}
```

図3 テスト・ベクトルの作成に使用したコードの抜粋

ここで、

- ▶ 最初の値は時間
- ▶ 2番目の値は出力（電圧、電流、温度など）
- ▶ ->はタブ文字（ASCIIコード#09）
- ▶ CRはキャリッジ・リターン文字（ASCIIコード#13）
- ▶ LFはラインフィード文字（ASCIIコード#10）

テスト・ベクトルを生成するオプションは数多くあります。このシミュレーションではCプログラミング言語を選択しました。

数行のコードで、そのままシミュレーションに入力できる大量のテスト・ベクトルを簡単に生成することができます。図3はテスト・ベクトルを生成するコードの抜粋で、図4はこのCプログラムで生成された波形です。

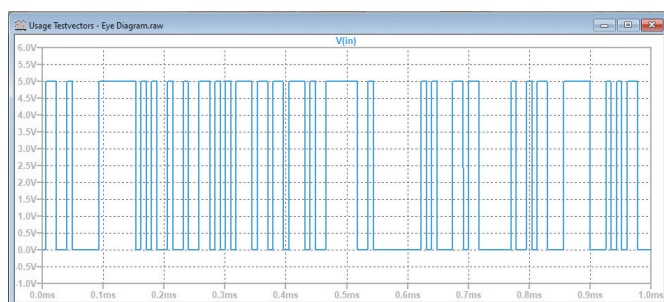


図4 Cプログラムを使って生成したテスト・ベクトル

ビット・スタッフィングなどの有線通信で使われる概念は、図5に示すような数行のコードで実装できます。

```
// Max consecutive identical symbol (for droop)
#define MAX_CONSECUTIVE_0 7
#define MAX_CONSECUTIVE_1 10

[...]
```

```
if (consecutiveHigh > MAX_CONSECUTIVE_1)
{
    printf("Hit Max 1 limit - Bit stuffing a Zero\n");
    logicLevel = 0;
    consecutiveHigh = 0;
}
if (consecutiveLow > MAX_CONSECUTIVE_0)
{
    printf("Hit Max 0 limit - Bit stuffing a One\n");
    logicLevel = 1;
    consecutiveLow = 0;
}
```

図5 コードは本稿のリンクに示されています^{P1}。

キャプチャしたテスト・ベクトルを使ってPWLファイルを作成

実際のデータを使い、そのデータをLTspiceにインポートすることも可能です。通常の実験装置を使ってデータを取り込めば、それも数ステップでインポートできます。

例えば、図6はオシロスコープで行ったUSB 1.0通信のキャプチャです。



図6 実験装置を使ってキャプチャしたUSB信号

図7はオシロスコープによって保存したデータセットの代表的な例です（正確なフォーマットは異なることがありますが、データをLTspiceへインポートするための一連の規則はそのまま適用できます）。

		A	B	C	2
1	1	in s	C1 in V	C2 in V	
3	2	-7.70E-07	3.08E+00	-2.59E-02	
	3	-7.70E-07	3.08E+00	-3.56E-02	
	4	-7.69E-07	3.07E+00	-2.59E-02	
	5	-7.69E-07	3.06E+00	-1.61E-02	

図7 キャプチャしたUSB信号の未加工データ

このデータセットをLTspiceのシミュレーションに使用するには、いくつか変更を加える必要があります。

1. ヘッダ行は削除します。
2. LTspiceに提供するPWLファイルのデータ列は1つだけとします。
3. 時間値は必ず正でなければなりません。

以上の条件を満たしていれば、そのデータはLTspiceで使用できます。

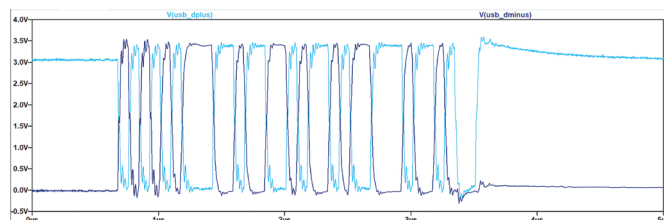


図8 LTspiceにインポートしたUSB信号

テスト・ベクトルPWLファイルの使用

図9に示すように、作成したPWLファイルを使用するため、設計に電源とファイル・パスを追加することができます。

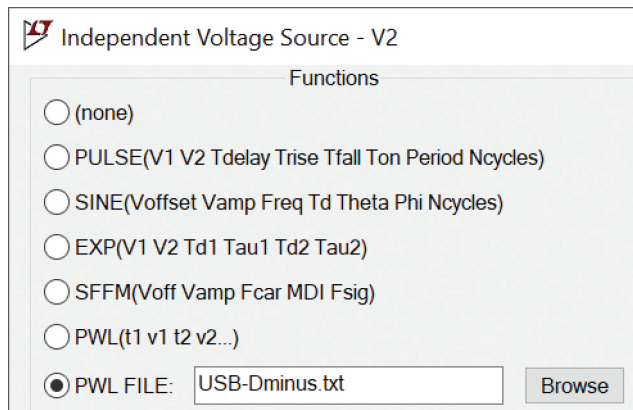


図9 電圧ソース用のPWLオプション

絶対ファイル・パスと相対ファイル・パスの両方を使用できますが、相対パスを使用すればシミュレーションの移植が可能になり、他の設計者とシミュレーションを共有することができるので、相対パスの使用を推奨します。

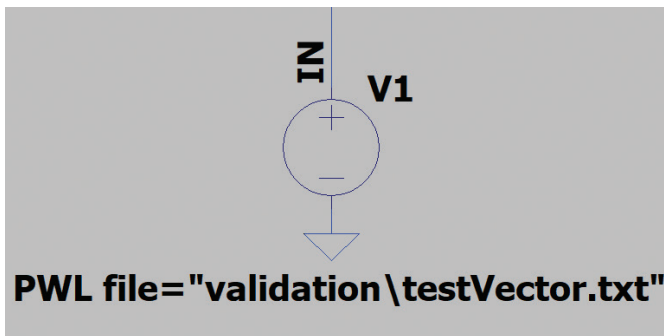


図10 相対パスの例

アイ・ダイアグラム機能のイネーブルとチューニング

LTspiceの隠れた能力を最大限に引き出してうまく利用するには、まずシミュレーションを行う必要があります。

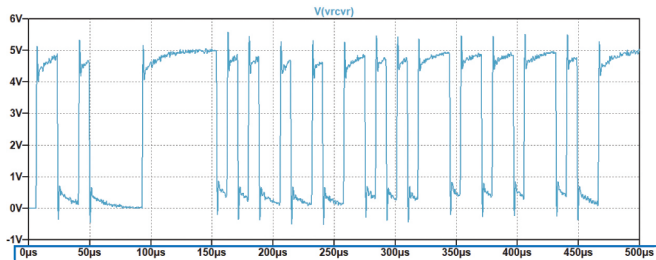


図11 アイの水平軸特性をイネーブルするにはこのエリアを右クリック

シミュレーションが終了して信号が表示されたら、水平（時間）軸を右クリックします。

ダイアログがポップアップ表示され、図12に示すようなアイ・ダイアグラム・ボタンが表示されます。

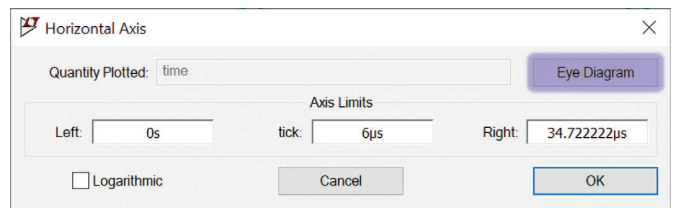


図12 アイ・ダイアグラム・オプションの場所

このポップアップ・ウィンドウを使用すれば、分かりやすい名前のパラメータを使い、アイ・ダイアグラムの表示をイネーブルして調整を加えることができます。

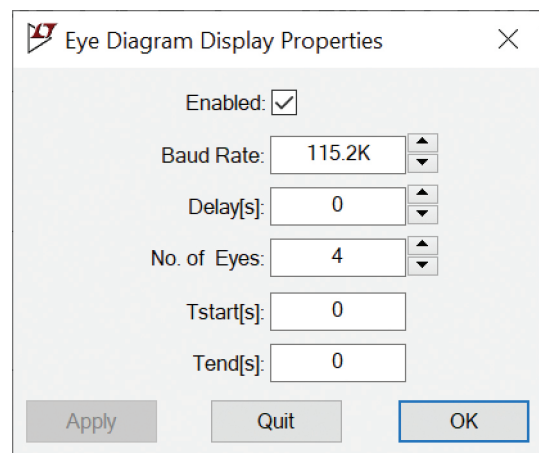


図13 アイ・ダイアグラムのセットアップ

検証時の表示は図14のようになります。

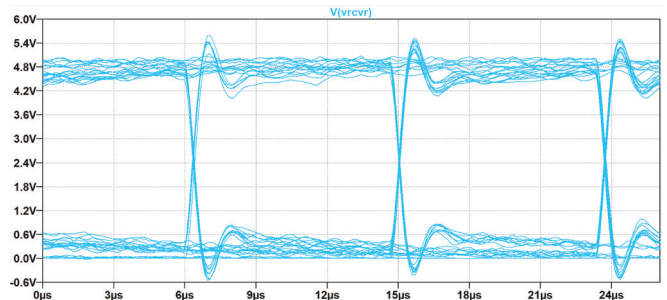


図14 アイ・ダイアグラム表示

アイ・ダイアグラム・マスクの計算と表示

信号の完全性の評価を容易にするために、アイ・ダイアグラムにはアイ・ダイアグラム・マスクを関連付けることができます。アイ・ダイアグラム・マスクはLTspiceの標準機能ではありませんが、(パート1のEMC上限ラインのように) 実装は可能です。

アイ・ダイアグラムの仕様は標準なので、図15に示すようにマスクのほとんどは、少数の変数の組み合わせで評価することができます。

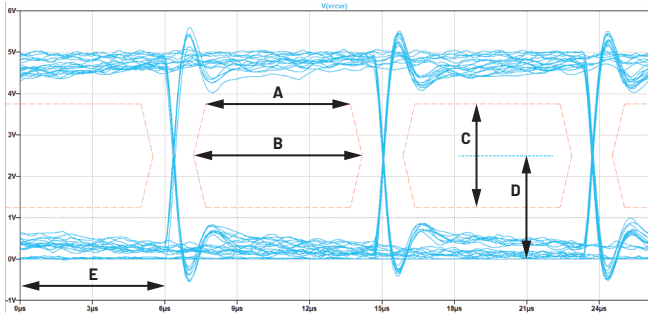


図15 アイ・ダイアグラムとアイ・ダイアグラム・マスク

アイ・ダイアグラム・マスクに示したA～Eの文字の定義を以下に示します。

- ▶ A: アイのフラット幅。これは、マスク定義によってはゼロにすることができます。
- ▶ B: アイの幅。これは、最大ジッタ、スルー・レート、ボー・レートなど、いくつかのパラメータを評価する際の鍵となります。
- ▶ C: アイの高さ。この値は通常、このパラメータの評価をケーブルの近端で行うか遠端で行うかによって異なります。
- ▶ D: アイの中心。これはアイの中心電圧を定義します。
- ▶ E: 遅延。これは、プロット・ウィンドウ内で信号を移動するためにLTspice内でのみ使われます。

アイ・ダイアグラム・マスクの作成

このシリーズ記事のパート1では、FFTスペクトラム内にEMC上限ラインを表示するための作画要素の使用方を解説しました。本稿では、同じツールを使用してアイ・ダイアグラム・マスクを描く方法を示します。

アイ・ダイアグラムの作成は、EMC上限ラインを作成して追加する場合より複雑です。アイ・ダイアグラムの場合は、JavaScript^{P2}が組み込まれたウェブページを使用してアイ・ダイアグラムの定義を作成します。作成した定義は、LTspice信号表示のプロット設定ファイル (*.plt) に貼り付けることができます。このJavaScriptプログラムは、図16に示すように、エンジニアがその設計を完了させるために使用できます。

一般的な有線インターフェース標準用のアイ・ダイアグラム定義は、既にプリセットとして用意されています。それぞれのラジオ・ボタンをクリックすることによって、フィールドに標準値が自動入力されます。

- ☐ RS232-9600
- ☐ RS232-115200
- ☐ RS485-9600-Differential Voltage
- ☐ RS485-115200-Differential Voltage
- ☐ USB 1 Low Speed
- ☒ USB 1 Full Speed
- ☐ USB 2 Full Speed - Differential
- ☐ SPE 10M
- ☐ SPE 100M
- ☐ LVDS 250 LV
- ☐ LVDS 250 HV

図16 アイ・ダイアグラム・ジェネレータのプリセット

精密調整やアイ定義も、所定のフィールドを使って行うことができます。

Baudrate	Eye Width [ns]
1500000	600
Eyes before	Eye Flat Width [ns]
1	500
Eyes after	Offset [ns]
3	0
Eye Height (mV)	Eye Center (mV)
3000	1650

図17 アイ・マスク入力フィールド

更新ボタンをクリックすると、対応するプロット設定コマンドが作成されます。これらのラインは、このシリーズ記事のパート1に説明する方法に従い、そのままプロット設定ファイルに追加できます。

```
Line: "V" 13 1 (-9.666666666666667e-7,1.65) (-9.166666666666667e-7,3.15)
Line: "V" 13 1 (-9.166666666666667e-7,3.15) (-4.166666666666667e-7,3.15)
Line: "V" 13 1 (-4.166666666666667e-7,3.15) (-3.666666666666667e-7,1.65)
Line: "V" 13 1 (-3.666666666666667e-7,1.65) (-4.166666666666667e-7,0.14999999999999999)
Line: "V" 13 1 (-4.166666666666667e-7,0.14999999999999999) (-9.166666666666667e-7,0.14999999999999999)
Line: "V" 13 1 (-9.166666666666667e-7,0.14999999999999999) (-9.666666666666667e-7,1.65)
Line: "V" 13 1 (-3e-7,1.65) (-2.5e-7,3.15)
Line: "V" 13 1 (-2.5e-7,3.15) (2.5e-7,3.15)
Line: "V" 13 1 (2.5e-7,3.15) (3e-7,1.65)
Line: "V" 13 1 (3e-7,1.65) (2.5e-7,0.14999999999999999)
Line: "V" 13 1 (2.5e-7,0.14999999999999999) (-2.5e-7,0.14999999999999999)
Line: "V" 13 1 (-2.5e-7,0.14999999999999999) (-3e-7,1.65)
Line: "V" 13 1 (3.666666666666667e-7,1.65) (4.166666666666667e-7,3.15)
Line: "V" 13 1 (4.166666666666667e-7,3.15) (9.166666666666667e-7,3.15)
Line: "V" 13 1 (9.166666666666667e-7,3.15) (9.666666666666667e-7,1.65)
Line: "V" 13 1 (9.666666666666667e-7,1.65) (9.166666666666667e-7,0.14999999999999999)
Line: "V" 13 1 (9.166666666666667e-7,0.14999999999999999) (4.166666666666667e-7,0.14999999999999999)
Line: "V" 13 1 (4.166666666666667e-7,0.14999999999999999) (3.666666666666667e-7,1.65)
Line: "V" 13 1 (0.0000010833333333333333,1.65) (0.0000010833333333333333,3.15)
Line: "V" 13 1 (0.0000010833333333333333,3.15) (0.0000015833333333333333,3.15)
Line: "V" 13 1 (0.0000015833333333333333,3.15) (0.0000016333333333333333,1.65)
Line: "V" 13 1 (0.0000016333333333333333,1.65) (0.0000015833333333333333,0.14999999999999999)
Line: "V" 13 1 (0.0000015833333333333333,0.14999999999999999) (0.0000010833333333333333,0.14999999999999999)
```

図18 ウェブページによって作成されたプロット設定

図19に示すような良好な表示を得るには、通常、表示するアイの個数とLTspiceの遅延設定を調整する必要があります。

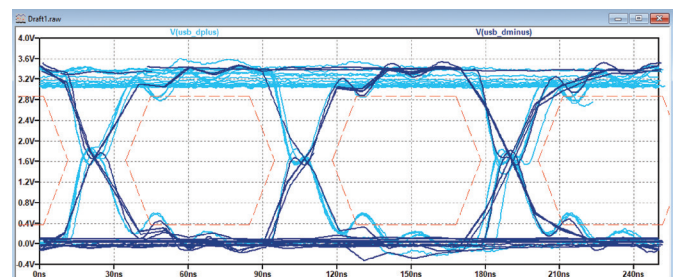


図19 ウェブページによって作成されたプロット設定を波形に適用した例

許容誤差範囲全体を通じた適合性評価

設計に使用するコンポーネントには広い許容誤差がありますが、これらの許容誤差が問題とならないかどうかは、計算によってチェックすることができます。しかし、数百個のコンポーネントを含む設計の場合は、手計算やスプレッドシートによる計算などの方法では時間がかかり、重要なパラメータを取り込めない場合もあります。デバイスによっては許容誤差範囲が小さいものを使用することも可能ですが、使用コンポーネントのすべてに許容誤差の小さいものを使用すると、価格や入手性の面で問題が生じます。また、経年変化の影響や温度依存性を考慮する必要もあります。

許容誤差範囲内での設計の妥当性を確認するために、SPICEとその拡張であるLTspiceは非常に優れた機能をいくつか備えています。

以下のセクションでは、モンテカルロ分布とガウス分布、およびLTspice内におけるワーストケース（最も厳しい条件）を使用した許容誤差分析手法について説明します。

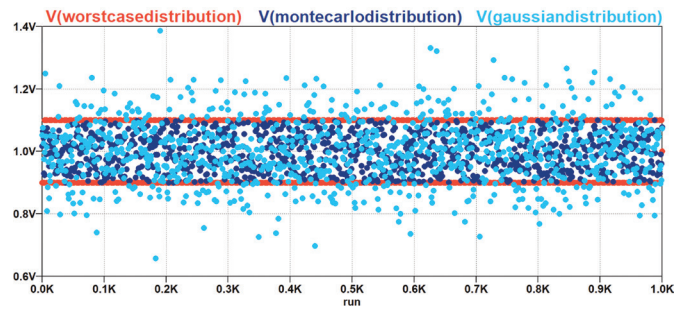


図 20 3つの主な方法におけるランダム値の分布

ガウス、ワーストケース、およびモンテカルロ分析用の共通テスト回路

実際の使用例における妥当性と徹底度を比較するために、ここではGraber⁴の研究に基づき、以下に示す例を取り上げました。このセットアップは、SPE 10Base-T1L 標準（10SPE）の物理層またはMDI用のシミュレーション回路を示しています。

図 21 に示すこのシミュレーション回路には、アナログ・デバイセズのADIN1110またはADIN1100 10BASE-T1L イーサネットPHY/MAC-PHY用に、100Ω ± 10%の終端抵抗が含まれています。

信号結合容量、電力結合インダクタ、共通モード・チョーク、その他のEMC保護コンポーネントがモデル化されています。一部のコンポーネントについては、推奨コンポーネント値と許容誤差範囲が追加されています。

リターン・ロス・プロットに使用する式は次の通りです。

$$(100+1/(V1))/(100-1/(V1))$$

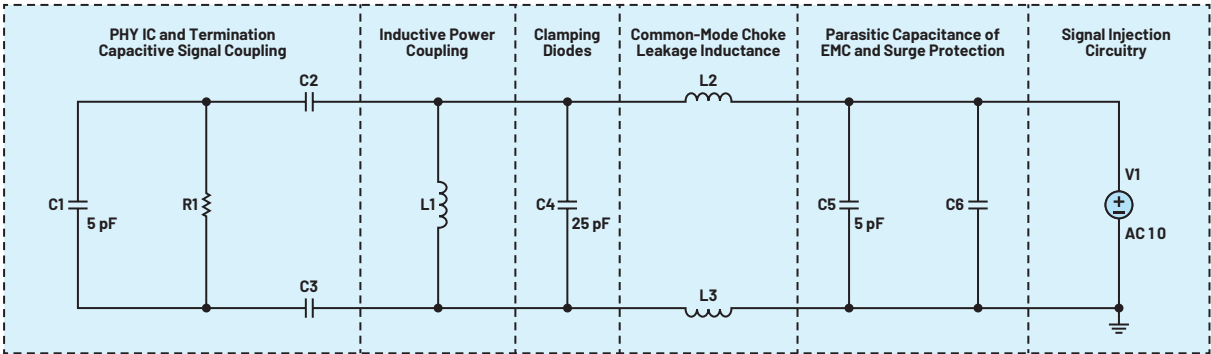


図 21 ガウス、ワーストケース、およびモンテカルロ分析用の共通テスト回路 - 参考資料に基づく⁴

モンテカルロ・シミュレーション

モンテカルロ・シミュレーションでは、使用シミュレーション回路における各指定コンポーネントの許容誤差範囲内のランダム値を使用します。コンポーネントの許容誤差範囲内にある値が回路シミュレーションで発生する確率は、すべて同じです。

LTspiceには、シンプルな構文の便利なモンテカルロ関数が組み込まれています。

例えば、許容誤差 10% の 100Ω 抵抗を作成するには、次の構文を使う必要があります。

```
{mc(100R, TolA)}  
.param TolA = 0.10
```

表 1 モンテカルロ法使用時のコンポーネント値と許容誤差の定義

記号	範囲	コンポーネント値の構文 (モンテカルロ)
R1	90Ω~110Ω	{mc(100, TolA)}
C2, C3	200nF~600nF	{mc(400 nF, TolB)}
L1	500μH~1500μH	{mc(1000 μH, TolC)}
L2, L3	0nH~500nH	{mc(250 nH, TolD)}
C6	0pF~200pF	{mc(100 pF, TolE)}

リターン・ロスのシミュレーションには図 21 に示す回路⁴を使用できます。リターン・ロスは、発生する可能性のあるすべての信号反射を表す指標です。

リターン・ロスは、ケーブル・リンク沿いのあらゆる位置におけるインピーダンス・ミスマッチによって生じます。リターン・ロスはデシベルで表され、10BASE-T1L で使われる高データ・レートの通信や長いケーブル（1700m）を使用する通信で特に問題となります。

プロットにMDIリターン・ロスの上限ライン（図 23 に示す赤いライン）を追加するには、Plot SettingsメニューのSave Plot Settingsをクリックします。

標準的なテキスト・エディタを使ってPLTファイルを開きます。Excelファイルに示す要領で、ライン定義構文をコピー・アンド・ペーストします (図22)。

LTSpice MDI Return Loss Mask				
Start Freq	End Freq	RL Start	RL Stop	Line def for LTSPICE plot settings file
100000	200000	-14.582	-20	Line: "dB" 4 0 (100000,0.18660574063097) (200000,0.1)
200000	1000000	-20	-20	Line: "dB" 4 0 (200000,0.1) (1000000,0.1)
1000000	10000000	-20	-3.3	Line: "dB" 4 0 (1000000,0.1) (10000000,0.683911647281429)

図22 LTSpiceプロット設定ファイルのライン定義

実際のシミュレーションでも同じ図を得るには、波形を右クリックしてDon't Plot Phaseボタンをクリックします。

モンテカルロ・シミュレーションに関するまとめ

モンテカルロ・シミュレーションは、電子設計の適合性をその全許容誤差範囲にわたって評価する方法の1つで、シミュレーションの実行回数を妥当な数に抑えながら、ほとんどの設計者のニーズを満たすことができます。

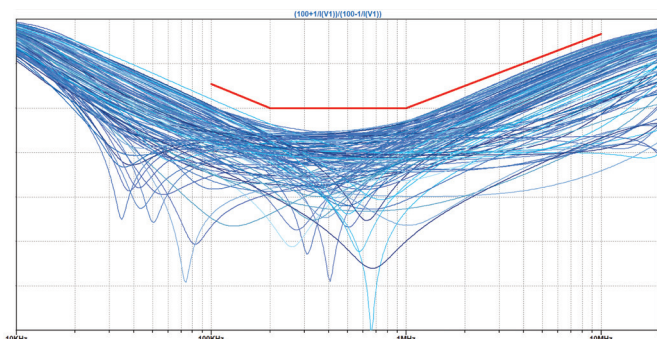


図23 SPE 終端の差動リターン・ロス：モンテカルロ分布パラメータを使い128回実行

ワーストケース・シミュレーション

ワーストケース・シミュレーション機能は、LTSpiceの組込み機能ではありません。しかし、Joseph SpencerとGabino Alonso²の研究に詳細が示されているように、ワーストケースをシミュレートするための機能を実装することができます。

最も厳しい条件のシナリオに従ってシミュレーションを行うには、func binary(run,index)関数とfunc wc(nom,tol,index)関数の両方が必要で、これらの関数をLTSpiceの回路図シートにSPICE命令として置く必要があります。

```
.func binary(run,index) floor(run/(2**(index))-2*floor(run/(2**(index+1))))
.func wc(nom,tol,index) if(run==numruns,nom,if(binary(run,index),nom*(1+tol),nom*(1-tol)))
```

これらの関数を使用するには、以下のことを行う必要があります。

- ▶ 最大値、最小値、および公称値をカバーするために必要なシミュレーションの実行回数を、指定されたワーストケース分析を行うコンポーネントごとにnumrunsを使って宣言します。実行回数は $2^N + 1$ で決定されます。ここで、Nはコンポーネントの許容誤差にリンクされた異なるインデックスの数です。図21の場合はこの値は129回なので（実行番号「0」も回数に含まれます）、構文は次のようになります。

```
.param numruns = 129
```

- ▶ コンポーネントの正則値ではなく、次の式を宣言する必要があります。

{wc(100R, 0.1, 0)}

ここで、

- ▶ 100Rは公称値。
- ▶ 0.1は許容誤差（ここでは+または-10%）。
- ▶ 0は変化するパラメータのインデックスで、変化する次のコンポーネントはインデックス1を取ります。

図21のシミュレーション回路は、静的なコンポーネント値ではなく、下の表に示す式を使って作動させます。

表2 ワーストケース法使用時のコンポーネント値と許容誤差の定義

記号	範囲	コンポーネント値の構文 (ワーストケース)
R1	90Ω~110Ω	{wc(100, TolA, 0)}
C2	200nF~600nF	{wc(400 nF, TolB, 1)}
C3	200nF~600nF	{wc(400 nF, TolB, 2)}
L1	500μH~1500μH	{wc(1000 μH, TolC, 3)}
L2	0nH~500nH	{wc(250 nH, TolD, 4)}
L2	0nH~500nH	{wc(250 nH, TolD, 5)}
C6	0pF~200pF	{wc(100 pF, TolE, 6)}

結果は図24の波形プロットに表示されます。

MDIリターン・ロス・マスクの上限ラインは、既に述べたように、プロット設定ファイルを編集することによって追加されます。

ワーストケース分析に関するまとめ

Steve Knudtsenは、システム設計にワーストケース分析を使用することの利点と制約について簡潔にまとめています³。

ワーストケース分析はよく使われるアプローチで、コンポーネント・パラメータはその最大許容限界に調整されます。

ワーストケース・アプローチの制約には、実際の結果と一般的に見られる結果が一致しないという点が含まれます。最も厳しい条件下での性能を示すシステムを観察するには、極めて多数のシステムを組み合わせなければならない場合があります。

システムが最も厳しい条件に合わせて設計されている場合は、コンポーネントの選択が高価になる可能性があります。

しかし、モンテカルロ・シミュレーションまたはガウス・シミュレーションと組み合わせるワーストケース法を使用すれば、そのシステムに関する貴重な情報を得ることができます。

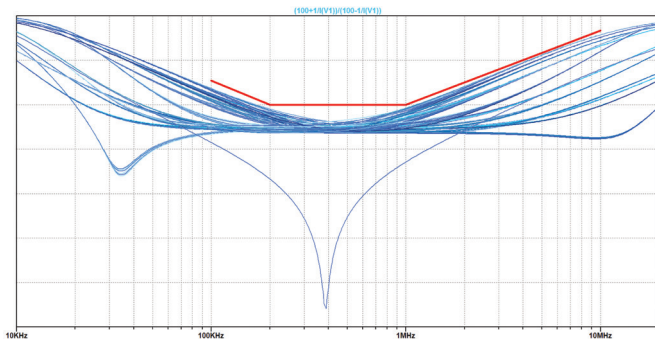


図24 SPE終端の差動リターン・ロス：
ワースケース分布パラメータを使い128回実行

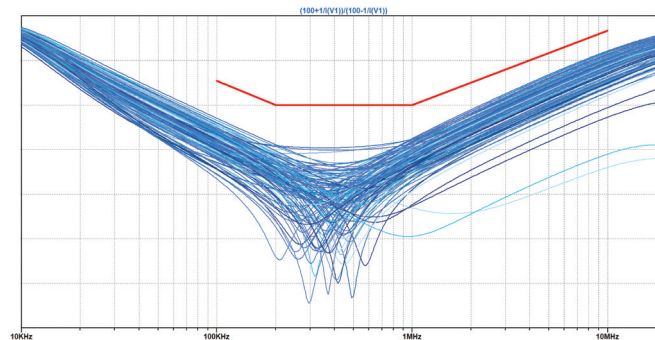


図26 SPE終端の差動リターン・ロス：
ガウス分布パラメータを使い128回実行

ワースケース分析は、シミュレーションに非常に時間がかかる場合や、公称値での動作が既に検証済みの場合の全体的な動作検証に適しています。

ガウス・シミュレーション

LTspiceにはガウス関数が組み込まれていて、中央値は発生確率が高くなっています。構文はシンプルです。

$$\{\text{nominal_value} * (1 + \text{gauss}(\text{tolerance} / \sigma))\}$$

下の表3に示す式を使い、ガウス分布の標準偏差パラメータ σ に従った調整を行うことができます。

表3 ガウス分布法使用時のコンポーネント値と許容誤差の定義

使用する式	許容誤差範囲内のサンプルの比率
$\{\text{nom} * (1 + \text{gauss}(\text{tol}/1))\}$	68.2% (1σ)
$\{\text{nom} * (1 + \text{gauss}(\text{tol}/2))\}$	95.4% (2σ)
$\{\text{nom} * (1 + \text{gauss}(\text{tol}/3))\}$	99.7% (3σ)
$\{\text{nom} * (1 + \text{gauss}(\text{tol}/4))\}$	99.99% (4σ)

もしくは、より視覚的な表現を使用します。

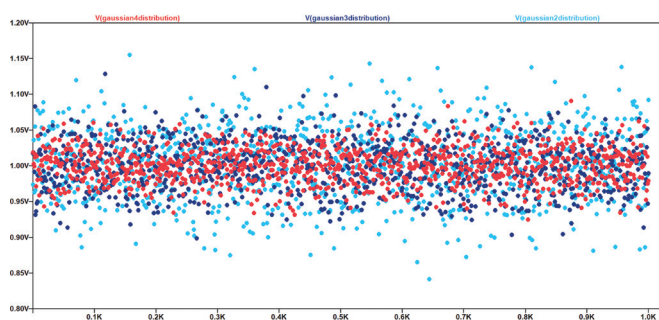


図25 サンプルのガウス分布と σ

例えば、許容誤差10%、許容誤差内の値の発生確率を 4σ として100 Ω 抵抗を作成するには、次の構文を使う必要があります。

$$\{100R * (1 + \text{gauss}(\text{TolA}/4))\}$$

$$\text{.param TolA} = 0.10$$

図26は、 4σ とガウス分布パラメータを使用して、図19に示すガウス・シミュレーションを128回実行した結果です。

ガウス分布法に関するまとめ

ガウス分布は、電子設計における変動のシミュレーションに最も多く使われる方法です。

公称値付近のパラメータのガウス分布は、許容誤差の影響を検討する上では依然として最も自然な方法です。

しかし、残念ながらこれには代償が伴います。徹底的な確認を行うには、シミュレーションを非常に多くの回数だけ繰り返す必要があります。

この分布は、許容範囲外にある値も拾い上げる一方、コンポーネントのメーカーが行うソーティングとビンニングの操作は省略します。

10BASE-T1Lリンクの分析への適用

10BASE-T1Lイーサネット標準を使用して、いくつかのフィールド・バスを置き換えることができます。従来型フィールドバスと10BASE-T1Lには、共に同じケーブルを使用することができます。このケーブルは、全二重通信と最終的なパワー・デバイス (PD) への電源供給に使われる、シンプルなバランス銅線ペアです。ただし、10BASE-T1L標準に適合するように、物理層通信トランシーバー (PHY) とパッシブ・コンポーネントは交換する必要があります。

LTspice内で行う10BASE-T1L信号の完全性の検証のほとんどは、同様の形状の信号を使って行うことができます。

表4 伝送する信号の振幅に応じたシングルペア・イーサネットの伝送距離

送信信号振幅	予想伝送距離
2.4V _{pp}	1000m～1700m
1.0V _{pp}	200m

使用するエンコーディングはPAM3 (Pulse Amplitude Modulation 3 levels) ⁶です。エンドポイントに必要とされる到達距離と能力に応じて、送信信号の振幅は1Vまたは2.4Vに調整できます。

ケーブル側の信号立ち上がり時間は-1から+1への遷移時に53.33nsで、立下がり時間も同じです。

スルー・レートは一定と見なせるので、0～1、1～0、-1～0、および0～-1の公称遷移時間は26.66nsです。

このようなテスト・ベクトルの作成には、図27に示すコードを使用します^{P3}。

```

// loop for every sample
for (sample = DELAY ; sample < (NUMBER_OF_SAMPLES+DELAY) ; sample++)
{
    // Compute a sample
    previousLogicLevel = logicLevel;
    logicLevel = (rand() % 3) ;
    transition = abs((previousLogicLevel - logicLevel));

    switch (transition)
    {
        case 2 : // +1 to -1 transition or +1 to -1 transition
            fprintf (pfile, "%E\t%5.2f\n", (sample*M_G_PERIOD), busvoltage[previousLogicLevel]);
            fprintf (pfile, "%E\t%5.2f\n", (sample*M_G_PERIOD)+(M_G_RISE), busvoltage[logicLevel]);

            break;

        case 1 : // transition to nearby state ( -1 <-> 0 <-> 1 )
            fprintf (pfile, "%E\t%5.2f\n", (sample*M_G_PERIOD)+((1.0*M_G_RISE)/4.0),
                    busvoltage[previousLogicLevel]);
            fprintf (pfile, "%E\t%5.2f\n", (sample*M_G_PERIOD)+((3.0*M_G_RISE)/4.0),
                    busvoltage[logicLevel]);

            break;

        case 0 : // no change in value
            fprintf (pfile, "%E\t%5.2f\n", (sample*M_G_PERIOD), busvoltage[logicLevel]);
            break;

        default :
            break;
    }
}
}

```

図 27 PAM3 テスト・ベクトルの作成に使用したコードの抜粋

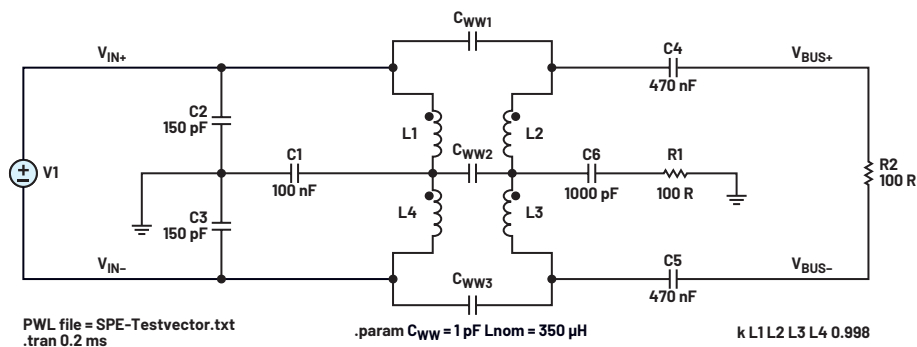


図 28 PAM3 PWL テスト・ベクトルを使用するトランス・ベースの終端

これは、5000個のPAM3シンボルからなるテスト・ベクトルをPWLフォーマットで出力します。

このテスト・ベクトルをここで使用する回路図に入力することによって、最小カップリング、巻線間容量を始めとする様々なパラメータを検証することができます。

図28、図29、図30は、10BASE-T1Lリンクのトランス・ベースの終端、PWLソース電圧ファイルの出力、およびPWL電圧源とケーブル側差動電圧のアイ・ダイアグラム表示です。これは、10BASE-T1L標準への適合性テストに使用できます。

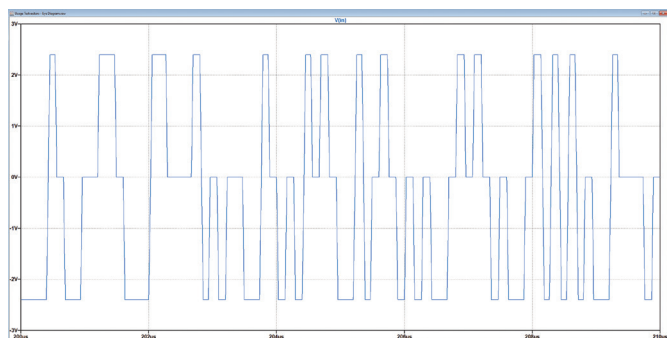


図29 PWL電圧源の出力

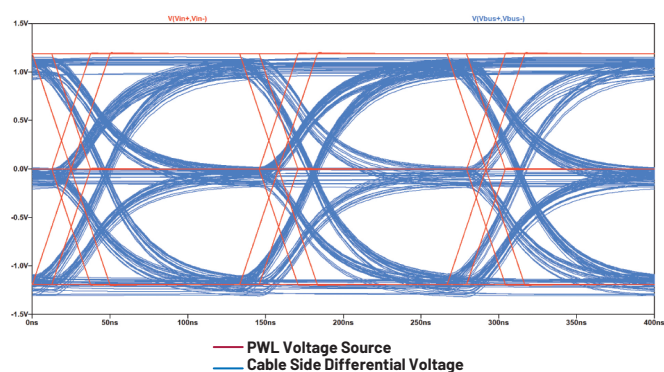


図30 PWL電圧源とケーブル側差動電圧のアイ・ダイアグラム表示

まとめ

LTspiceは無料で使用できる強力なシミュレーション・ツールで、CおよびJavaScriptコードを使用し、波形ジェネレータと組み合わせることで使用することができます。これは有線通信における信号の完全性を確認する強力なツールで、製品開発時の実験に要する時間を短縮し、製品設計の道筋をつけて、より迅速な製品市場投入を実現します。アナログ・デバイセズとWürth Elektronikは有線リンクを設計するエンジニアのためにこのツールを提供し、10BASE-T1L SPEなどの新しい標準の理解を支援します。

参考資料

- 1 “Signal Integrity Basics.” Signal Integrity White Paper. Anritsu, 2009.
- 2 Joseph Spencer and Gabino Alfonso. [LTspice : シミュレーションの実行回数を最小限に抑え、最も厳しい条件で回路を解析する方法] Analog Devices, Inc.
- 3 Steve Knudtsen. [LTspiceを使用し、複雑な回路を対象とする統計的な公差解析用のモデルを構築する] Analog Devices, Inc., August 2021.
- 4 Steffen Graber. “10 Mb/s Single Twisted Pair Ethernet.” IEEE, May 2017.
- 5 IEEE802.3-2022 clause 146 + annex 146
- 6 IEEE802.3-2022 annex 146A

プログラム :

- P1 限られた数の0と1の連続で単純なランダム・ビットストリームを生成するCプログラム
- P2 LTspiceプロット・ファイル用のアイ・ダイアグラム・マスク・コードを生成するHTML + Javascript
- P3 ランダム化されたPAM3ビットストリームを生成するCプログラム

著者について

Richard Anslowは、アナログ・デバイセズのオートメーション・ビジネス・ユニットに属するソフトウェア・システム設計エンジニアリング部門に勤務するシニア・マネージャです。状態基準保全、モータ・コントロール、産業用通信設計が専門です。アイルランドのリメリック大学で工学士号と工学修士号を取得しました。最近、パデュー大学でAIおよびMLの大学院課程を修了しています。

Sylvain Le Brasは、Würth Elektronikのフィールド・アプリケーション・エンジニアで、電源および電磁両立性が専門です。Würth Elektronik入社前はABBの研究開発部門と複数の技術移転研究所で様々な役職を経験しました。フランスのナント大学ポリテクニック・スクールで工学修士号を取得しています。

EngineerZone®

オンライン・サポート・コミュニティ

アナログ・デバイセズのオンライン・サポート・コミュニティに参加すれば、各種の分野を専門とする技術者との連携を図ることができます。難易度の高い設計上の問題について問い合わせを行ったり、FAQを参照したり、ディスカッションに参加したりすることが可能です。



Visit ez.analog.com

* 英語版技術記事は[こちら](#)よりご覧いただけます。