

SPICE の伝送線路モデルを使って遅延信号を作ってみる

著者: 石井 聡

はじめに

とあるところで、複数の遅延した信号同士が足し算されたようすを示す必要性（どうやら「マルチパス」というものの例を示したかったらしい…）があったため、ちょっと ADIsimPE (ADI simulator Personal Edition; 新しいアナログ・デバイセズの SPICE ツール) を使って作ってみました。その結果をご紹介します。なおこのツールは SIMetrix ベースなので SIMetrix でも同じシミュレーションを実行できます。

伝送線路モデルで遅延信号を作る

「さて、どうやって遅延した信号を作るか?」ですが、SPICE では「伝送線路」(Transmission Line) というモデルで遅延した信号を作ることができます。

ADIsimPE ではロス有り、ロス無し(ロスレス。抵抗成分による減衰が無い) 伝送線路の解析は簡単にできます。ADIsimPE の回路入力画面の左側にあるリンク、Search をクリックして、以下の図 1 の Search ボックスに Transmission (もしくはその途中まで) と入力すると、伝送線路モデルを選ぶことができます。ロス有りの伝送線路 (Lossy Transmission Line) もうまく使えば大変うれしいことですが、とはいえ、今回のこの目的(信号遅延を発生させるだけ)であれば、このロスレスのモデルでも十分に対応ができます。

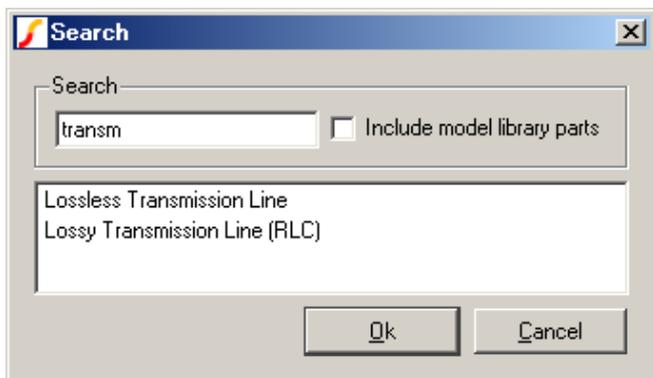


図 1. Search ボックスで見つけられる伝送線路モデル

一番簡単な回路でまず実験してみる

ちゃんと動くかどうかの確認を、図 2 のような一番簡単な回路にして、シミュレーションしてみました。この回路は基本的な動作を確認してみるもので、目的の遅延時間が得られるかどうかのポイントです(というよりそれだけが目的)。信号源は 5V デジタル信号 (CLK) + 50Ω の出力インピーダンス、負荷側は 120Ω を接続してあり、これで反射係数は $\Gamma = 0.41$ になります。

信号源のクロックは矩形波 5V とし、CLK 周波数は 2.5kHz (周期 400us)、伝送線路は 50Ω で遅延時間 50us です。

これでシミュレーションしてみた結果が図 3 です。緑が入力 (P1) で、赤が出力 (P2) です。緑のように伝送線路の入力側の波形が最初 2.5V まで立ち上って、50us の時間後に出力の赤が変化していることがわかります。

さらに反射係数が 0.41 なので、緑の入力側が $2.5 + (2.5 \times 0.41) = 3.525V$ にステップ状に (50us + 50us = 100us のタイミングで) 変化するようすが得られます。このように 50Ω 伝送路を SPICE で実現できることがわかりました。

なお ADIsimPE では、矩形波信号源となるものは Voltage Waveform Generator と、この Universal Source が利用できます。Voltage Waveform Generator モデルについては rise/fall 時間がデフォルトでは 500ns になっています。そのため、より高速な時間変動をシミュレーションする場合には、図 4 で示す Universal Source の「立ち上がり時間 Rise」と「立ち下り時間 Fall」のパラメータを変更してください。ここでは 10ps に設定してあります。

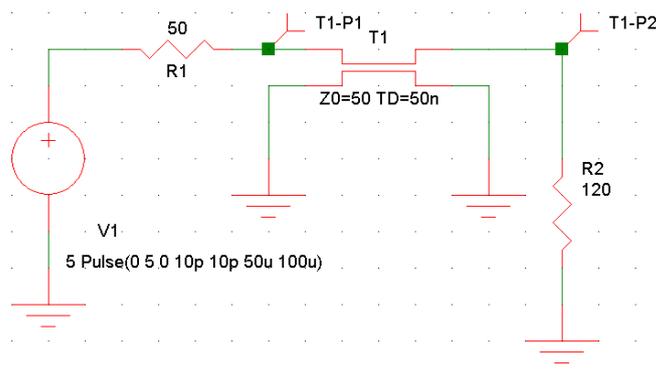


図 2. 一番簡単な回路で実験してみる

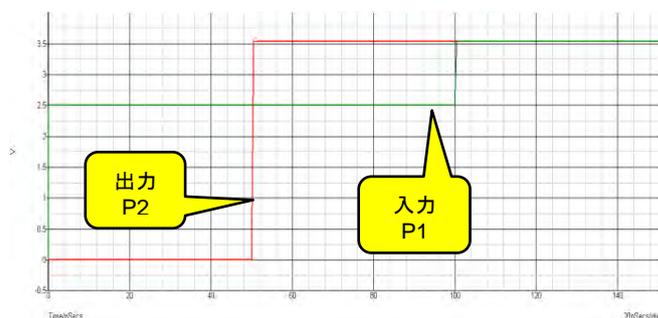


図 3. 一番簡単な回路でのシミュレーション結果

アナログ・デバイセズ株式会社は、提供する情報が正確で信頼できるものであることを期していますが、その情報の利用に関して、あるいは利用によって生じる第三者の特許やその他の権利の侵害に関して一切の責任を負いません。また、アナログ・デバイセズ社の特許または特許の権利の使用を明示的または暗示的に許諾するものでもありません。仕様は、予告なく変更される場合があります。本紙記載の商標および登録商標は、それぞれの所有者の財産です。
©2014 Analog Devices, Inc. All rights reserved.

Rev. 0

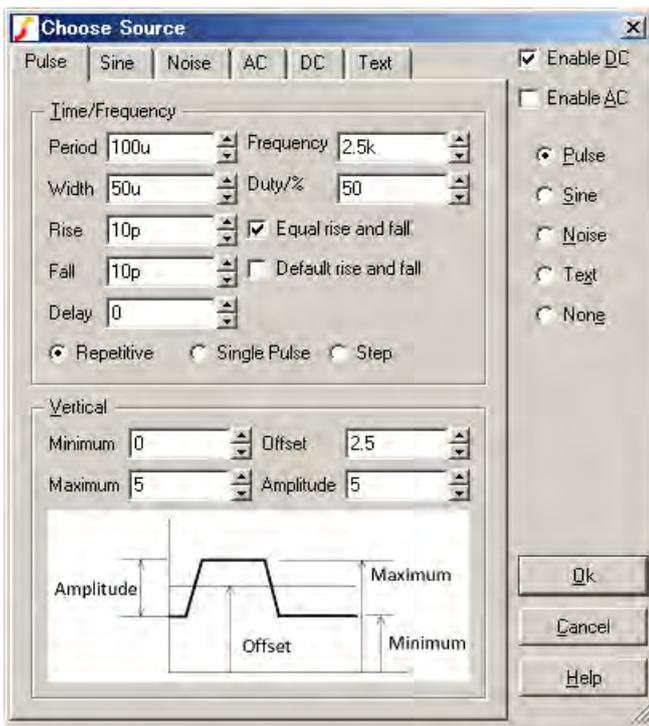


図 4. 高速でシミュレーションする際は CLOCK_VOLTAGE の立ち上がり時間と立ち下り時間を設定する

ネットリストで表すとどうなる？

ところでこれを SPICE ネットリストでは、以下のように表すことができます。

T1 2 0 3 0 Z0=50 TD=50us

T1 は伝送線路機能自体（頭文字の T）とそのモデル名、2 は入力ポート、3 は出力ポート（入出力は双方向なので入/出は意識することはない）、2 個の 0 はグラウンドのポートになります。Z0 は伝送線路の特性インピーダンス、TD は線路の遅延時間です。

[Tx] [InPort] [InPort] [InGnd] [OutPort] [OutGnd] [Z0] [PropagationTime] という文法です。ここでは、[InPort] = 2、[OutPort] = 3 と（仮に）してあります。これは大体 10km の同軸ケーブルに相当します（波長短縮率を 66%と考えると。真空での位相速度/光速なら 15km に相当）。

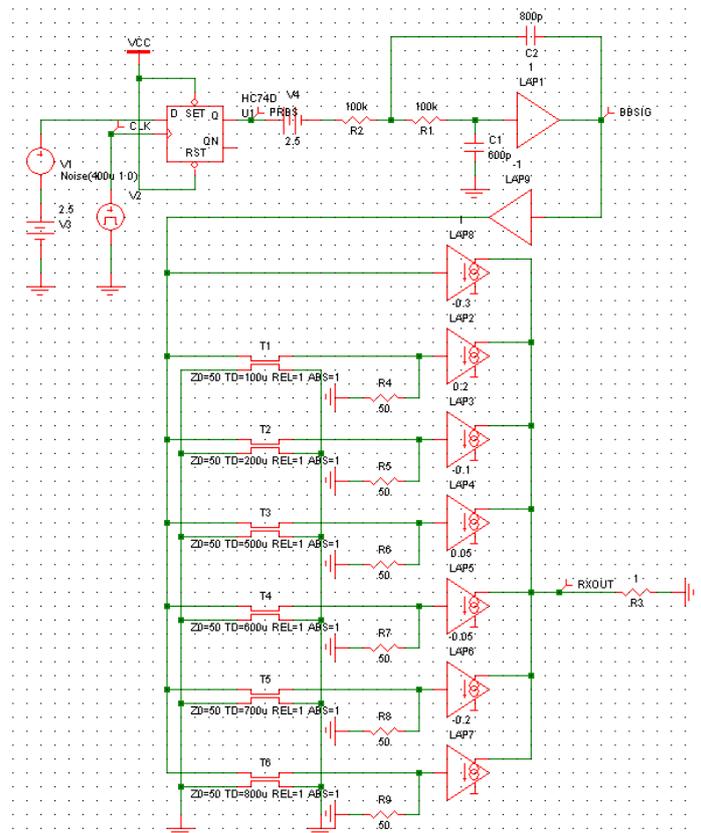


図 5. 7 path 遅延モデルの SPICE シミュレーション回路

実際の遅延モデルを作ってみる

この伝送路をシミュレーション上で作りたかったのは、複数の伝送遅延路を作りたかったからでありました。最終的なシミュレーション回路では 7 経路（7 path とする）の遅延波形を合成したいというところでした。またそれぞれの遅延系（6 経路）には加算係数をつけてあります。それぞれのパラメータを以下に示します。

- T1 TD=100us × -0.3
- T2 TD=200us × 0.2
- T3 TD=500us × -0.1
- T4 TD=600us × 0.05
- T5 TD=700us × -0.05
- T6 TD=800us × -0.2

Tx がそれぞれの経路の線路、TD は経路ごとの遅延時間を示しています。「×」以降が加算係数で、T1～T6 の経路（path）の遅延波形をこの係数で重みづけして足し合わせることにします。また遅延時間の無いぶんも×1 倍として一緒に合成します。この最終目的のシミュレーション回路図を図 5 に示します。

この回路では U1 にて V1 のランダム・ノイズをラッチすることで 2.5kbps のランダム・パターンを発生させ、LAP1 をバッファとしたサレンキー・フィルタでその信号を波形整形します。電圧源 V4 で信号を -2.5V オフセットさせて、LAP1 の出力が 0V を中心にスイングするようにしてあります。

LAP1 出力には伝送線路の信号源抵抗に相当する 50Ω を接続してありませんが、これはそれぞれの伝送線路の出力側でそれぞれ終端してあるので（反射が無いので）、省いてあります。

合成の部分は電流出力モデルを使用

合成の部分ですが、それぞれの 50Ω 伝送路は 50Ω で終端整合させないといけないので、これをそれぞれ Laplace Transfer Function の出力を電流出力にして、これらを直結と LAP2～LAP7 の計 7 経路を束ね合せて 1Ω の抵抗に流入させます。なお VCCS モデルで変換することもできます。

こうすれば（電流出力なので）シミュレーション上で簡単に信号合成ができるわけですね。

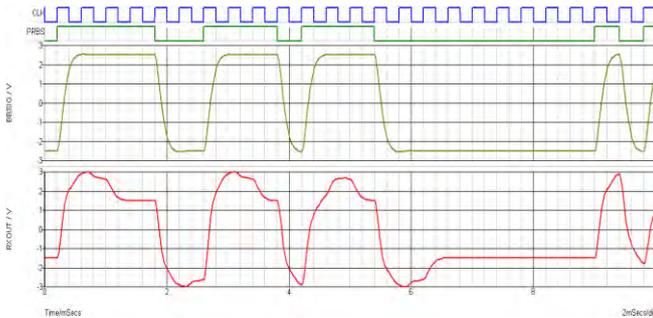


図 6. 遅延が合成された状態のシミュレーション結果

図 6 は「遅延なし ($\times 1$ 倍) + Σ (LAP1～LAP6)」として全体を合成してみた結果です。ビットレートは 2.5kbps です。なお繰り返しますが、 50Ω の抵抗で各伝送線を終端しているのので、反射なく（伝送遅延だけの）信号を得ることができています。

図中で一番上が 2.5kbps 相当のクロック波形で、真ん中のふたつが LAP1 の入力と出力である単一経路相当 (T_0 だけの伝送路) の波形、いちばん下が複数 (7 path) の伝送遅延路を通して得られた波形です。

アイパターン表示は別のツールを利用

最終的には 1 UI (Unit Interval; 1 ビット相当の時間) に相当するアイパターンを作りたいのですが、その図は ADIsimPE や SIMetrix Intro だけでは作ることはできません。SIMetrix 製品版では Command Shell ウィンドウに Command Line フィールドが用意されており、そこで波形データをファイルに吐き出すことができます。詳細はヘルプの Data Analysis Data Import and Export の章にある Exporting Data の項をご覧ください。

そこで NI Multisim を今回は使って

そこでこの図の作成は NI Multisim の力を借りることになりました。同じ回路を作って、そのシミュレーションで得られた波形データを Grapher から

ツール > Excel にエクスポート

という機能を使って、Excel ファイルとして落として、それを必要な部分だけをコピー&ペーストしてテキストファイルとし、MATLAB に読み込ませます。さらに SPICE シミュレーションではタイムポイントが一定ではないため、MATLAB の interp1() 関数で一定時間にリサンプルし、同じく eyediagram() で表示させてみました。

図 7 と図 8 はこれにより得た波形です。図 7 は LAP1 相当の出力として単一経路相当の（伝送遅延路を通さない）波形（図 6 の真ん中の 2 波形に相当）、図 8 は複数 (7 path) の伝送遅延路を通して得られた波形（図 6 の下に相当）です。

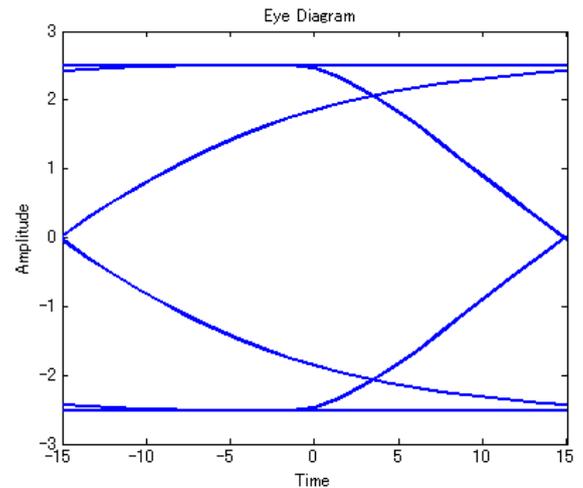


図 7. LAP1 の出力として単一経路相当の（伝送遅延路を通さない）波形（図 6 の真ん中の 2 波形に相当）

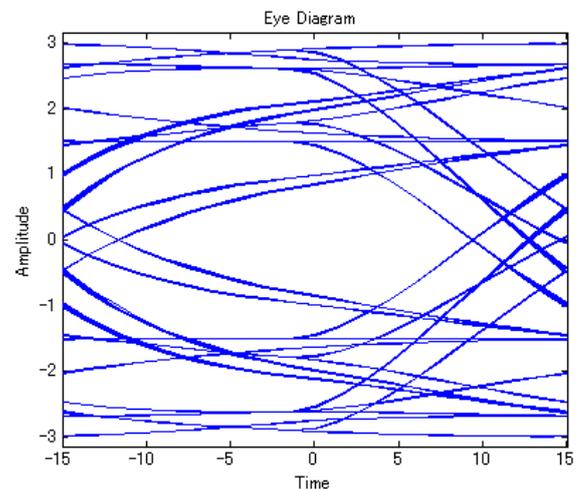


図 8. 複数 (7 path) の伝送遅延路を通して得られた波形（図 6 の下に相当）

伝送線の SPICE ネット表記についてなど

SPICE での伝送線路ブロックのネットリスト上の表記は先に示したとおりですが、このように表記もすることができます。

T1 2 0 3 0 Z0=50 F=1kHz NL=0.2m

ここで [Txx は伝送線路機能名] [InPort] [InGnd] [OutPort] [OutGnd] [Z0] [Frequency] [Normalized Electrical Length] という文法です。F [Frequency] は伝送線路に通すと仮定する周波数、NL [Normalized Electrical Length] はその周波数での「波長で規格化」した電気長（波長短縮率も考慮した長さ）です。NL はデフォルトでは 0.25、つまり 1/4 波長となります。結局は

$$TD = NL/F$$

という関係になっているだけで、TD と同じことを表しています。ADIsimPE では TD で指定します。なお NI Multisim では、TD を指定するものが LOSSLESS_LINE_TYPE1 に相当し、F と NL を指定するものが LOSSLESS_LINE_TYPE2 に相当します。

ロスレス伝送線路の基本モデルのご紹介

またおまけですが、SPICE でのロスレス伝送線路を表現している基本モデルを図 9 にご紹介しておきます。この基本モデルを見てみると、「グラウンド側」という明確な規定がされていません。なぜ SPICE ツールの伝送線路モデルが図 2 で示したような 4 端子の図形になっているのかが分かるかと思います。

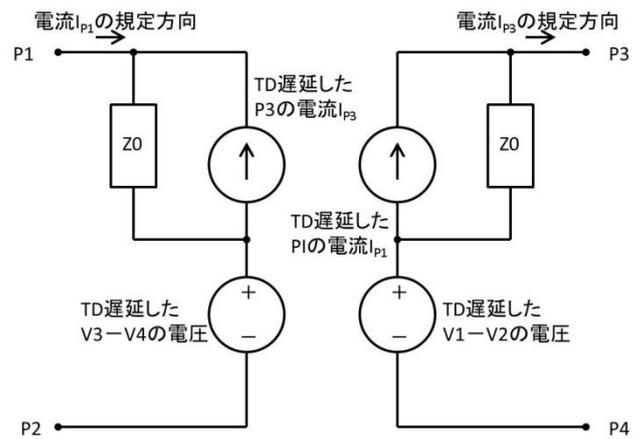


図 9. ロスレス伝送線路の基本モデル