

IBIS モデルの開発方法【Part 2】 独自のIBISモデルの作成方法

著者：Rolynd Aquino、プロダクト・アプリケーション・エンジニア
 Francis Ian Calubag、システム・アプリケーション・エンジニア
 Janchris Espinoza、プロダクト・アプリケーション・エンジニア

概要

本稿では、LTspice®を利用して、独自のIBIS (Input/Output Buffer Information Specification) モデルを作成する方法を説明します。具体的には、モデルを作成するにあたっての事前準備からモデルの検証作業までの手順を、図やグラフを交えて詳細に解説します。特に、LTspiceを使用して、IBISモデルに必要なI-V (電流 - 電圧) データ、V-t (電圧 - 時間) データ、上昇率/下降率 (ramp) のデータ、C_compのデータを正確に取得するための手順を詳しく説明します。また、定性的/定量的なFOM (Figure of Merit : 性能指数) を使用して、IBISモデルの性能を検証する方法も示します。IBISモデルの作成対象としては、ADxxxxという架空の3ステート・バッファを例にとることにします。このバッファは、CMOSの入力インターフェースと3ステートの出力インターフェースを備えています。本稿で示す例は、読者の皆さんがIBISモデルの作成を開始する際の出発点として活用できます。

はじめに

どのようなシステムを構築する場合でも、シミュレーションは重要な役割を担います。シミュレーションを実施することにより、問題を事前に予測し、時間とコストのかかる改修を回避することが可能になります。開発プロセスにおいては、常に最初から正しく設計を行うことが目標になります。例として、高速デジタル・インターフェースのシミュレーションを実行するケースを考えます。その場合、設計が適切でなければ、プリント基板のシンプルなパターンが原因で、シグナル・インテグリティ (信号品質) に影響が及ぶ可能性があります。デジタル・インターフェースのシグナル・インテグリティについてシミュレーションを行う場合には、IBISモデルが使用されます。

この連載記事のPart 1で述べたとおり、IBISモデルでは、デバイスのデジタル・インターフェースの電気的特性を、表形式のI-VデータとV-tデータによって表します。このビヘイビア・モデル

については、構文エラーが含まれないようにしつつ、できるだけ正確に作成することが重要です。そうすれば、作成したモデルを使用する際に問題が生じることを避けられます。また、シミュレーションを実施する際には、デジタル・インターフェースを備える各種の部品/デバイスのIBISモデルも用意する必要があります。多くの場合、それらはメーカーのウェブサイトからダウンロードできますが、必ず提供されているとは限りません。実際、IBISモデルの利用にあたっては、モデルを入手できないということが問題になりがちです。設計に使用した部品/デバイスのIBISモデルが入手できない場合には、製品の開発が遅れる可能性があります。

IBISモデルについては、そのデバイスのメーカーから入手するのが最も良い方法だと言えます。ただ、ユーザが自分でIBISモデルを作成することも可能です。本稿では、SPICEモデルを基にして、最も基本的なIBISモデルを作成する方法を紹介します。その作成過程では、シミュレータとしてLTspiceを使用することにします。以下のセクションにおいて、LTspiceによるシミュレーションの設定は、「[IBIS Modeling Cookbook for IBIS Version 4.0](#)」を基にして行うことにします。また、定性的/定量的なFOMを使用して、IBISモデルを検証する方法についても詳しく説明します。

「最も基本的なIBISモデル」とは？

LTspiceを使って基本的なIBISモデルを作成するお客様を支援するためには、「基本的」という語の意味を定義する必要があります。基本的なIBISモデルの振る舞いは、主に、モデリングの対象となるデジタル・バッファの種類によって決まります。モデリングを行う際には、IBISのこれまでのバージョンを見返し、バッファのモデリングに必要な最小の要件と、モデリングの対象として取り上げられているデジタル・インターフェースの種類を確認する必要があります。シングルエンドのCMOSバッファは、IBISでモデル化できる最もシンプルなデジタルI/Oの例だと言えます。そこで、本稿では、同バッファをモデリングの対象として取り上げることにします。



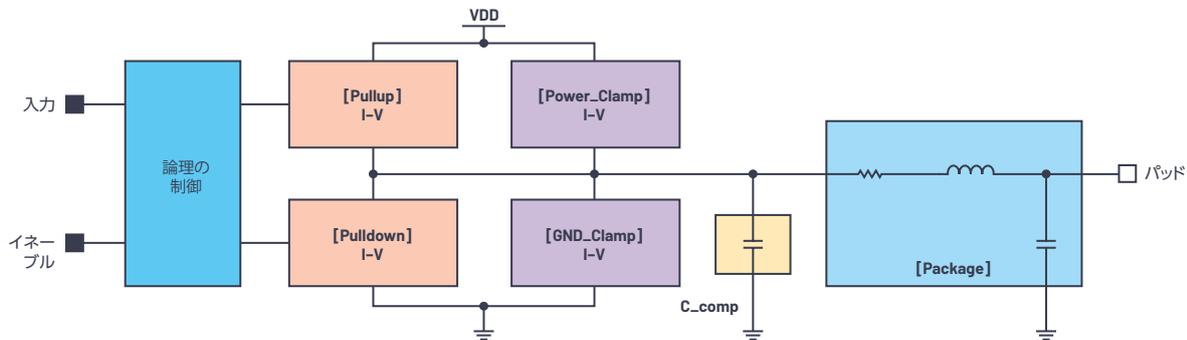


図1. CMOSの3ステート・バッファのIBISモデル

表1. Model_type別の構成要素

Model_type	[Package]	C_comp	[GND_Clamp]	[Power_Clamp]	[Pulldown]	[Pullup]	V-tデータ	[Ramp]
入力	✓	✓	✓	✓	—	—	—	—
3ステート	✓	✓	✓	✓	✓	✓	✓	✓
I/O	✓	✓	✓	✓	✓	✓	✓	✓

図1に示したのは、CMOSの3ステート・バッファに対応するIBISモデルの構造です。Part 1で説明したように、IBISモデルの構成要素や使用するキーワードは、モデルの種類によって異なります。表1は、各種モデル (Model_type) の構成要素と使用するキーワードについてまとめたものです。

ユース・ケース

本稿では、ADxxxxという架空のデバイスのLTspiceモデルを使用してIBISモデルを作成します。ここで、ADxxxxはイネーブル・ピンを備える単一入力、単一出力のデジタル・バッファであると仮定します。したがって、そのIBISモデルには、2つの入力 (DIN1とEN) と1つの3ステート出力 (DOUT1) が存在することになります。

一般に、IBISモデルの作成は、以下に示す5つのステップで行われます。

【ステップ1】モデルの作成に向けた事前準備を行う

【ステップ2】SPICEモデルを基にC_compのデータ、V-Iデータ、V-tデータを取得するために、LTspiceのシミュレーションを実行する

【ステップ3】IBISの仕様に準拠するようにファイルのフォーマットを行う

【ステップ4】IBISに対応するパーサ (Parser) によってテストを実施し、ファイルをチェックする

【ステップ5】同じ負荷条件の下で、IBISモデルとSPICEモデルを使用したシミュレーションをそれぞれ実施し、結果を比較する

IBISモデルには、標準値、最小値、最大値のデータが必要です。それらの値は、動作電源電圧範囲、温度、製造プロセスのコーナー条件によって決まります。本稿では、簡潔さを優先し、標準の条件だけを示すことにします。

ibischkシリーズは、作成したモデルがIBISの仕様に準拠していることを確認するための有効なツール (Golden Parser) です。その実行可能ファイルは、IBIS Open Forumのウェブサイト (<https://ibis.org/>) から無償ダウンロードすることが可能です。本稿では、ibischkを搭載するIBISモデルの編集用ソフト (サードパーティ製) を使って作成作業を行うことにします。

モデルの作成に向けた事前準備

上で示したステップのとおり、シミュレーションを開始する際には、事前の準備が必要です。具体的には、デバイスのデータシートをダウンロードし、SPICEモデルとLTspiceのファイルをインストールしなければなりません。また、最初の確認作業として、デバイスのデジタル・インターフェースの種類 (入力、オープンドレイン、3ステートなど) と数を明確にします。

また、デバイスのデータシートを参照し、動作電源電圧、動作温度、ICのパッケージの種類、ピン・リスト、デジタル出力のタイミング仕様に対応する負荷の条件 (R_{Load} と C_{Load})、デジタル入力に対するロー・レベルの入力電圧 (V_{inl}) とハイ・レベルの入力電圧 (V_{inh}) を確認します。図2にADxxxxのSPICEモデルのシンボル、表2にその仕様を示しました。

IBISモデルのファイル(拡張子は.ibs。以下、IBISファイル)には、キーワードを使用してデバイスのデジタル・インターフェースに関するあらゆる情報を記述します。IBISモデルで使われるキーワードは角括弧で囲まれており、識別子として機能します。キーワードの詳細については、Part 1をご覧ください。

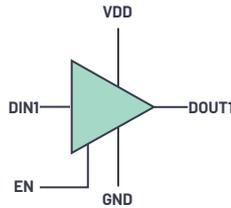


図2. ADxxxxのSPICEモデルのシンボル。
3ステートのデジタル・バッファです。

表2. ADxxxxのパラメータ。
データシートに記載されている情報に相当します。

パラメータ	値
VDD	1.8V (標準値)
動作温度	25°C
V_{inl}	$0.3 \times VDD$
V_{inh}	$0.7 \times VDD$
ICのパッケージ	6ピンのSOT-23
C_{Load}	15pF

ICのパッケージについては、キーワード[Package]を使用して記述します。その記述内容には、寄生要素であるRLC(抵抗/インダクタ/容量)成分の値が含まれます。それらの値は、ダイ・パッドとICのパッド/ピンの間の結合状態を表します。具体的な値はメーカーから入手することになります。他のIBISファイルに、モデリングの対象とするデバイスと全く同じパッケージで、同じメーカーから提供されているデバイスがあれば、キーワード[Package]を使って書かれているデータを流用することもできます。表3に、パッケージとして6ピンのSOT-23を採用したデバイスの寄生要素の値を示しました。

表3. 6ピンのSOT-23パッケージの寄生要素

[Package]			
変数	標準値	最小値	最大値
R_pkg	1.595E-01	NA	NA
L_pkg	4.455E-09	NA	NA
C_pkg	0.370E-12	NA	NA

表4に示したのは、ADxxxxのピン・リストです。IBISモデルでは、キーワード[Pin]を使用して、各ピンとそれに対応するモデル名を宣言します。この情報は、一般に3列のフォーマットで記述されます。1列目がピン番号、2列目がピンの説明、3列目がモデル名です。製品によっては、複数のピンに同じ名前が付与されていることがあります(VCCやGNDなど)。それらのピンについては、モデルにおいてグループ化し、まとめて記述することが可能です。本稿の例では、内部のトランジスタ・レベルの回路に関する情報が何もない状態でSPICEモデルが与えられているケースを想定しています。そのため、各デジタル・インターフェースに対し、個別にモデルを用意するのが適切な方法となります。このIBISモデルでは、電源ピンとグラウンド・ピンに対して[Power]と[GND]というモデル名を使用しています。デジタル以外のインターフェース・ピンと、「接続してはならない」ピンには「no connect」を意味する[NC]という名前が付与されます。なお、モデル名では、大文字と小文字を区別することに注意してください。モデル名は、後ほどモデルの作成工程で使用するので、正しく記述しなければなりません。

表5はADxxxxの真理値表です。この情報は、LTspiceによるシミュレーションを実施する際に必要になります。どのようにすれば、DOUT1ピンが論理レベルの1、論理レベルの0、高インピーダンス(High-Z)になるのかを理解することが重要です。

表4. ADxxxxのピン・リスト

[Pin]	Signal_name	Model_name
1	VDD	Power
2	DIN1	cmos_di1
3	EN	cmos_en
4	DOUT1	cmos_out1
5	GND	GND
6	NC	NC

表5. ADxxxxの真理値表

EN	DIN1	DOUT1
0	0	High-Z
0	1	High-Z
1	0	0
1	1	1

LTspiceの設定とシミュレーション

先述したとおり、IBISモデルでは、主にI-VデータとV-tデータによってデジタル・バッファの動作を表現します。表1に概要を示したように、デジタル・インターフェースの種類によってIBISモデルの作成に必要なI-V/V-tデータは異なります。表6は、必要なデータについてももう少し詳しく説明したものです。データの中には、「必須」のもの、「推奨」のものがある点に注意してください。「推奨」のデータについては、存在しなくても、パーサ(ibischk)によるテストでエラーは生じないということを意味します。ただ、それらのデータもシミュレーションに対して一定の影響を及ぼします。例えば、Power_Clamp、GND_Clampのデータは「必須」ではなく「推奨」ですが、信号の反射に関する解析に役立ちます。

表6. 入力/3ステートに必要なI-V/V-tデータ

	IBISのキーワード	入力	3ステート	
V-Iデータ	C_comp	必須	必須	
	[Power_Clamp]	推奨	推奨	
	[GND_Clamp]	推奨	推奨	
	[Pullup]	—	必須	
	[Pulldown]	—	必須	
V-tデータ	[Rising Waveform]	VDD基準の負荷	—	推奨
		GND基準の負荷	—	推奨
	[Falling Waveform]	VDD基準の負荷	—	推奨
		GND基準の負荷	—	推奨
	[Ramp]	—	必須	

キーワード [Power_Clamp] と [GND_Clamp]

キーワード [GND_Clamp] と [Power_Clamp] の下には、デジタル・バッファが備えるESD (Electro-Static Discharge) 保護用デバイスの動作を表形式のI-Vデータによって記述します。[Power_Clamp]は、VDDを基準とするESD保護用デバイスの全体的な動作を表し、[GND_Clamp]はGNDを基準とするESD保護用デバイスの全体的な動作を表します (図3)。

LTspiceでは、.DCというSPICEのコマンド/ディレクティブを使用することにより、I-Vデータを取得することができます。DOU1がグラウンドにクランプする動作については、図4の設定 (回路構成) によってシミュレーションを行い、結果を取得します。図4の回路では、適切な電源電圧を印加し、デバイスを高インピーダンスの状態に設定しています (表5を参照)。それにより、ESD保護用デバイスはコア回路から確実に分離されます。VSWEEPは、グラウンドを基準とする掃引電圧です。この電圧源により、グラウンド・クランプに関するESD保護用デバイスの挙動をシミュレーションすることができます。

IBISの仕様に基づき、I-Vデータは電源電圧を超える範囲 (できれば-VDD~2×VDD) で掃引を行って取得する必要があります。この例で言えば-1.8V~3.6Vの範囲です。ただ、この範囲でシミュレーションを実行すると、VDDを超える掃引電圧によって電源クランプに対応するESD保護用デバイスがオンになります。これを避けるために、VSWEEPは-1.8V~1.8Vの範囲で掃引するようにし、3.6Vのデータ・ポイントは外挿法によって追加することにします。この方法は、すべてのI-Vデータに適用できます。

どのI-Vデータにも、最大で100個までのデータ・ポイントしか含められないことに注意してください。100個を超えると、ibischkによるテストでエラーが発生します。データ・ポイントの数が99以下になるように、.DCコマンドのインクリメント設定を行ってください。なお、99以下にするのは、外挿によって2×VDDのデータ・ポイントを1つ追加する余地を残すためです。

DC掃引を行うと、シミュレーション上は非常に大きな逆電流が発生することがあります。これに対処するために、最初の掃引はダイオードのバリア電位 (-0.7V) 付近からVDD (1.8V) までに設定します。その後、-VDD~2×VDDのI-Vデータになるようにデータを外挿します。あるいは、小さな抵抗RserをVSWEEPに直列に追加し、過度の電流を制限するという方法も適用できます。

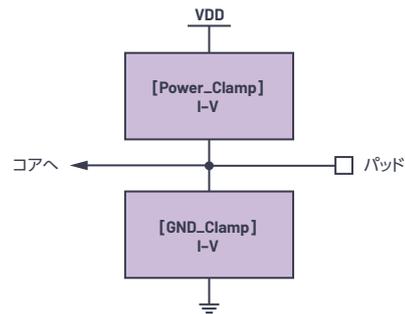


図3. キーワード [Power_Clamp]、[GND_Clamp] の構造を表す概念図

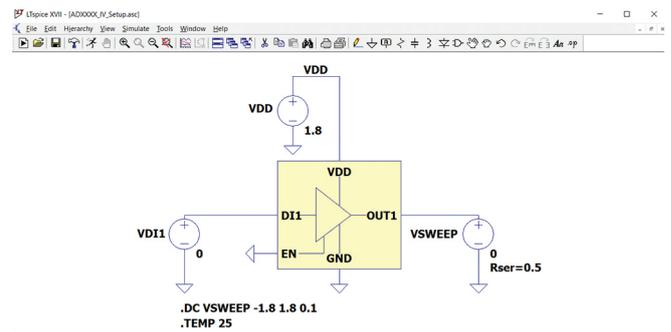


図4. グラウンド・クランプ用の設定。ADxxxxのDOU1を対象としています。

LTspiceの画面上で「Run」ボタンをクリックすると、シミュレーションが実行されます。測定の対象はDOUT1なので、対象となるノードはIx(U1:DOUT1)です。厳密に言えばI(VSWEEP)を対象にするべきなのですが、IBISモデルにはIx(U1:DOUT1)における電流の極性が必要です。このような設定にすることにより、モデルに適した形にI(VSWEEP)のデータを整えるための作業を減らすことができます。シミュレーション結果としては図5のグラフが得られるはずですが、シミュレーションが完了したら、「Results」ウィンドウをクリックし、「File」→「Export data as text」を選択してデータを保存します。データの保存先となるディレクトリを選択し、測定対象のノードをクリックして「OK」をクリックしてください（図6）。

[Power_Clamp]に関するデータの取得方法もグラウンド・クランプの場合と同様です。グラウンド・クランプの場合との違いは、掃引電圧VSWEEPの基準をVDDにすることです。必要な設定とシミュレーション結果を図7に示しました。

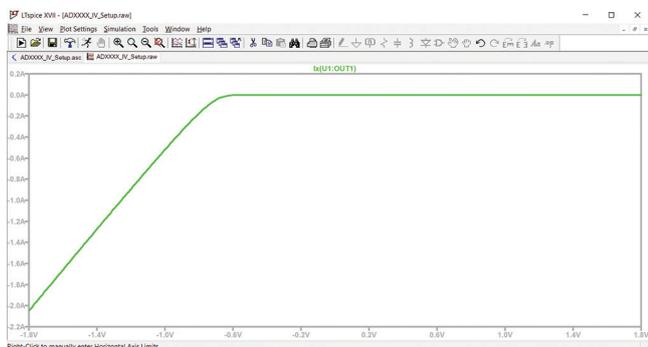


図5. グラウンド・クランプのシミュレーションの結果

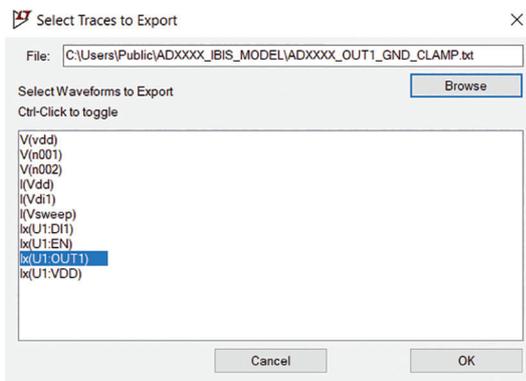


図6. シミュレーション結果のデータをテキストとしてエクスポート

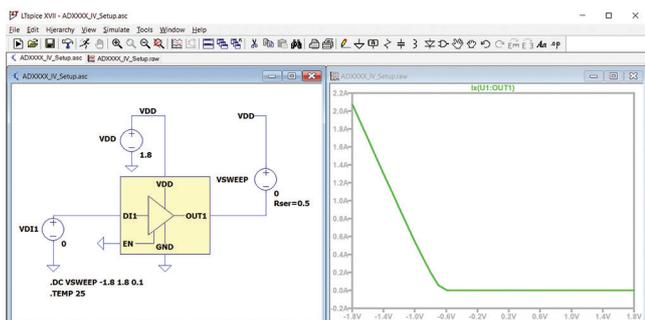


図7. パワー・クランプ用の設定とシミュレーション結果。ADxxxxのDOUT1を対象としています。

キーワード [Pulldown] と [Pullup]

図8の概念図は、I-Vデータに関連するキーワードについて示したものです。キーワード[Pulldown]と[Pullup]は、バッファ内のプルアップ素子/プルダウン素子の動作を表します。それらの動作は、グラフで表すとMOSFETのI-V特性に似ています。両キーワードに関連するデータを取得する際には、デバイスの真理値表を参考にし、出力ピンから所望の信号が得られるようにする方法を把握することが重要です。[Pulldown]と[Pullup]のデータを取得するための設定は、[GND_Clamp]と[Power_Clamp]の場合とほぼ同様です。DOUT1ピンをHigh-Zモードに設定するのではなく、イネーブルにするという点が異なります。

プルダウンのデータを取得するには、DOUT1ピンの出力を論理レベルの0、つまりは0Vに設定する必要があります。そこで、図9に示すように電圧源を配置します。この回路では、ハイの電圧に相当する1.8VをENピンに印加することによってDOUT1ピンをイネーブルにし、論理レベルの0 (0V) をDIN1ピンに印加することによって、DOUT1ピンから論理レベルの0が出力されるようになっています。この設定は、表5に示した真理値表に基づいています。

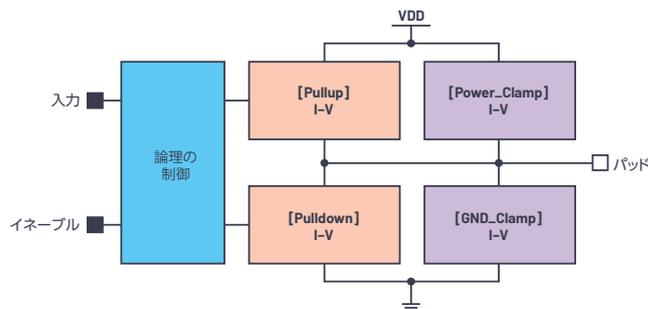


図8. I-Vデータに関連するキーワード

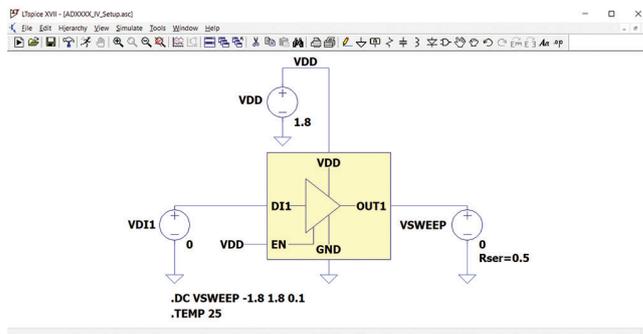


図9. プルダウン用の設定。ADxxxxのDOUT1を対象としています。

シミュレーション結果は図10のようになります。これを拡大すると、図11に示すように、MOSFETのI-V特性に似た結果が得られていることがわかります。

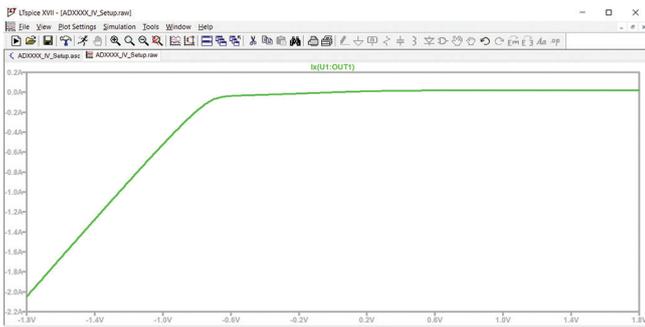


図10. プルダウンのシミュレーションの結果

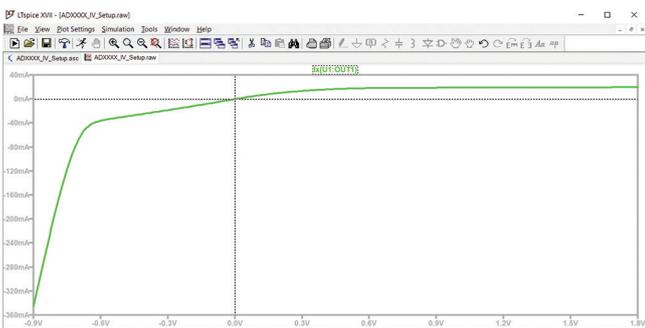


図11. 図10を拡大表示した結果

プルダウンのシミュレーションによって得られた電流のデータ（保存したデータ）は、グラウンド・クランプの電流とプルダウンの電流を合算したものになることに注意しなければなりません。これについては、図12をご覧ください。これに容易に理解できるでしょう。プルダウンのデータとして適切なものを得るには、データ・ポイントごとに、プルダウンの値からグラウンド・クランプの値を差し引く必要があります。この処理を簡単に行えるようにするには、[GND_Clamp]と[Pulldown]のDC解析において、電圧のインクリメント、開始電圧、終了電圧を同じ値に設定することが重要です。

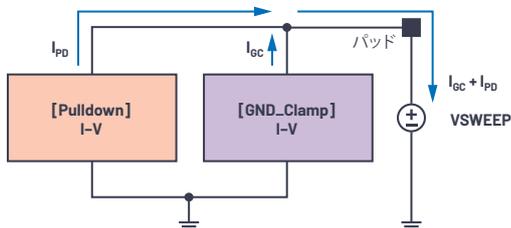


図12. プルダウンのデータに含まれる実際の電流

プルアップのデータを取得するための設定を図13に示しました。適切な電圧源を配置し、DOUT1を論理レベルの1（1.8V）に設定します。それにより、プルアップ素子がアクティブ（オン）になります。続いて、VDDを基準としてVSWEEEPを-1.8V～1.8Vの範囲で掃引します。VSWEEEPをこのように接続すれば、IBISの仕様に適合するようデータの編集を行う必要はありません。

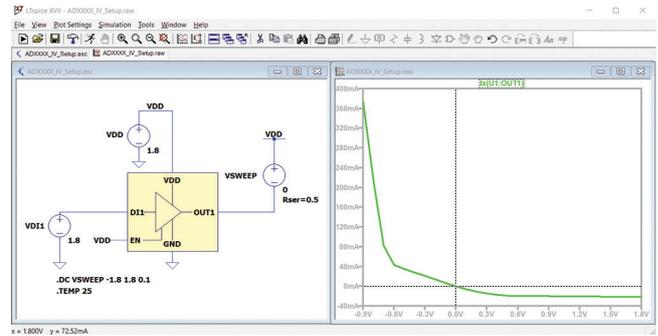


図13. プルアップの設定とシミュレーション結果。ADxxxxのDOUT1を対象としています。

プルダウンの場合と同様に、プルアップのシミュレーションによって得られた電流値（保存したデータ）は、電源クランプの電流とプルアップの電流を合算した値になります（図14）。そこで、データ・ポイントごとに、プルアップの電流値から電源クランプの電流値を差し引く必要があります。DC掃引を行う際、両者のパラメータを同じ値に設定しておけば、この処理を簡単に行うことができます。一般的な注意事項として、I-Vデータを取得するためのすべてのシミュレーションでは、DC掃引の設定として同じ値を使用すべきだということがわかります。

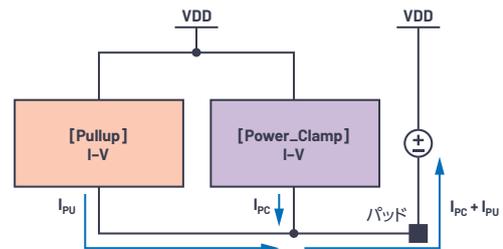


図14. プルアップのデータに含まれる実際の電流

キーワード [C_comp]

キーワード[C_comp]は、バッファの容量の最小値、標準値、最大値の記述に使用します。これはトランジスタとダイの容量を表しており、パッケージの容量とは異なります。[C_comp]のデータの取得方法は2つあります。1つは式(1)を使って概算値を計算する方法です。もう1つは、入力ピン/出力ピンにAC電圧を供給し、式(2)を使って計算を行う方法です。

$$C_{comp} = C_{IN} - C_{Package} \quad (1)$$

$$C_{comp} = - \frac{(Im_{Iac})}{2 \times \pi \times f \times V_{AC}} \quad (2)$$

各変数の意味は、以下のとおりです。

- ▶ Im_{Iac} : 電流の測定値における虚数部の値
- ▶ f : AC電源の周波数
- ▶ V_{AC} : AC電源の振幅

LTspiceによるC_compのデータの取得

図15に示すように、バッファの容量値は、入力ピンにAC電圧を供給して周波数掃引を行うことによって取得できます。AC電圧を供給することから、測定する電流には実数部と虚数部が存在することになります。AC電圧の供給時にバッファに流れ込む電流の値を測定するには、電流の極性を反転させる必要があります。出力バッファの容量値については、AC電源を出力ピンに接続するよう図15の回路を変更するだけで測定が可能です。

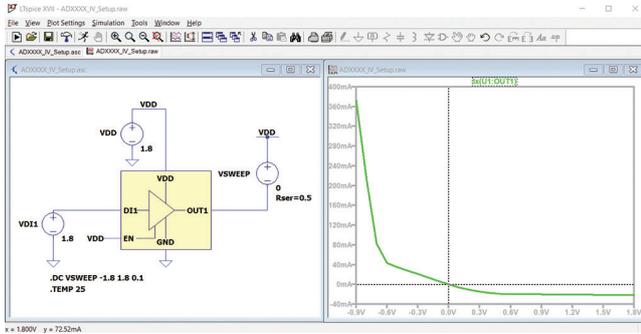


図15. ADxxxxのC_compの値を取得するための設定

供給するAC電圧の振幅値は任意ですが、通常は1Vに設定します。SPICEディレクティブの記述に従い、周波数掃引と測定が行われます。.ACコマンドを使用して波形をプロットする場合、デフォルトの設定ではBodeモード（ボード線図）で表示が行われます。その場合、単位はdBになります。バッファの容量の式にそのまま代入できるように電流の数値を表示したい場合には、Cartesianモード（直交座標）に設定を変更する必要があります。バッファの容量値の波形を表示するには、[Waveform] ウィンドウを右クリックして「Add Trace」を選択し、測定するピンを選びます（図16）。波形のプロット用のウィンドウには2本の線が表示されます。実線は電流の測定値の実数値、点線は虚数値を表します。

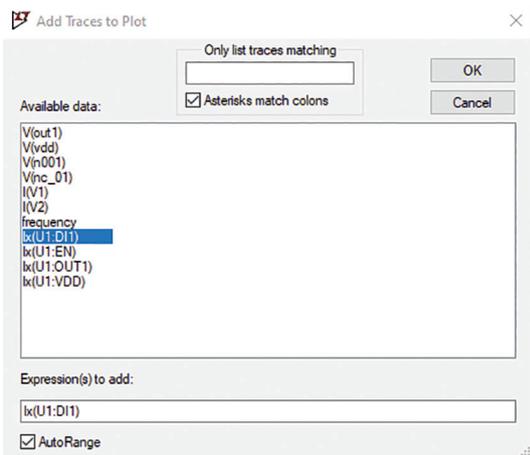


図16. [Add Traces] によるピンの選択

プロットの設定を「Bode」から「Cartesian」に変更するには、波形ウィンドウの左側のY軸を右クリックします。すると、「Left Vertical Axis-Magnitude」ダイアログ・ボックスが表示されます。このダイアログ・ボックスにおいて、プロットの「Representation」を「Bode」から「Cartesian」に変更してください（図17）。

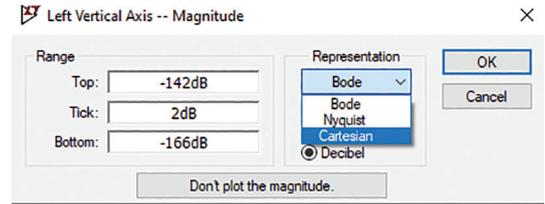


図17. [Bode] から [Cartesian] への設定変更

C_compの設定に使用するLTspiceのディレクティブ

LTspiceのディレクティブを使用し、回路の動作モードの設定、変数値の測定、C_compを計算するためのパラメータの処理を実施します。バッファのC_compの値を測定するためには、以下に示すLTspiceのディレクティブを使用します。

- ▶ .AC Lin 10 1k 10k : 動作モードとして、1kHz から 10kHz のAC線形周波数掃引を設定します。
- ▶ .Options meascplxfmt : .meas コマンドのデフォルトの複素数出力形式を、Bode、Nyquist、Cartesianのいずれかに設定します。
- ▶ .Options measdgt : .meas 文の有効桁数を設定します。
- ▶ .meas 文 : 回路内の特定のパラメータの値を算出するために使用するディレクティブです。

これらのディレクティブは、表示したいパラメータに応じて変更が可能です。LTspiceで利用できるディレクティブの詳細については、LTspiceのヘルプを参照してください。なお、.meas文の実行結果は、「[Tools] → [SPICE Error Log]」で確認できます。図18に示すように、SPICE Error Logには、実行結果がCartesian形式で表示されます。X座標は電流とバッファの容量の実数部、Y座標は虚数部を表します。上述したとおり、バッファの容量値の計算に必要なのは電流の虚数部です。実際のC_compは、図18の赤枠で囲まれた部分の値になります。

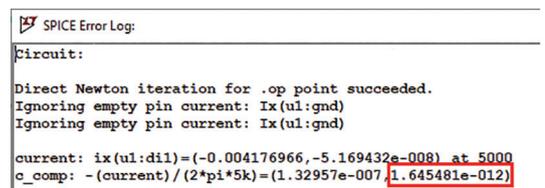


図18. .meas文の実行結果を表すSPICE Error Log

キーワード [Rising Waveform] と [Falling Waveform]

続いて、[Rising Waveform] と [Falling Waveform] の両キーワードについて説明します。

立上がり波形／立下がり波形とは？

キーワード [Rising Waveform] と [Falling Waveform] は、出力バッファのスイッチング動作のモデリングに使用します。そのモデルには、計4つのV-tデータを含めることが推奨されます。グラウンドを基準とする負荷を使用した場合の立上がり波形／立下がり波形と、VDDを基準とする負荷を使用した場合の立上がり波形／立下がり波形に対応するV-tデータです。

立上がり／立下がり時のV-tデータの取得

LTspiceでADxxxxのDOOUT1の立上がり波形／立下がり波形を取得するには、どのようにすればよいでしょうか。その方法は、立上がりエッジ／立下がりエッジに対応する入力スティミュラスを、PWL (Piecewise Linear) 信号またはパルス電圧源の形で入力ピンに印加するというものになります (図19)。モデルの作成にあたっては、最も高速な出力遷移を取得するために、シミュレーションで使用する入力スティミュラスについては遷移が高速になるよう設定しなければなりません。そのような信号を入力し、.TRANコマンドを使用して回路のトランジェント解析を実行しながら、出力ピンの電圧を測定します。ここでは、負荷として50Ωの抵抗を使用し、3ステート出力バッファの4つのV-tデータを取得します。それらの値は、バッファの設計と出力を遷移させるための駆動能力によって異なる可能性があります。V-tデータを取得する際に負荷の値を50Ωに設定するのは、プリント基板におけるパターンのインピーダンスの標準値が50Ωだからです。50Ωの負荷は、グラウンドを基準 (負荷をグラウンドに接続) にするか、またはVDDを基準 (負荷をVDDに接続) にして、バッファの出力ピンに接続します。

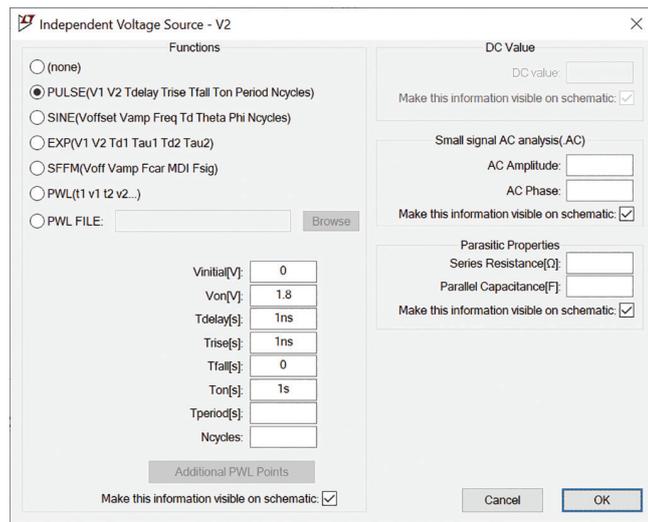


図19. 立上がりエッジに対応する入力スティミュラスの設定例。パルス電圧源を使用しています。

グラウンド基準の50Ω負荷を使用した場合の立下がり波形

グラウンドを基準とする場合の立下がり波形のデータを得るには、図20に示すような設定を行います。すなわち、50Ωの負荷をグラウンドに接続し、立下がりエッジに対応するスティミュラスを入力します。それにより、図21のような波形 (V-tデータ) が得られます。ご覧のように、出力は約16ナノ秒～20ナノ秒でセトリングしています。トランジェント解析の解析時間は、この立下がり波形のセトリングを十分に捉えられるように設定しなければなりません。

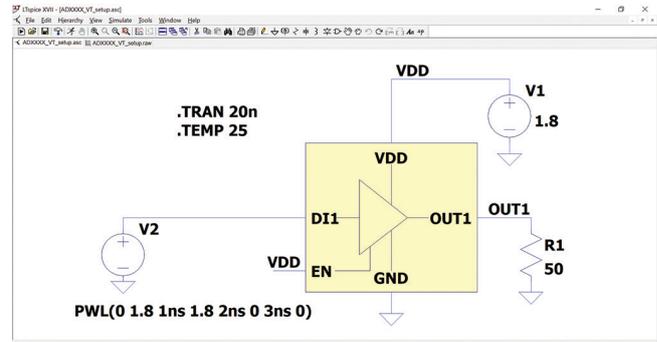


図20. グラウンド基準の50Ω負荷を使用した場合の立下がり波形を取得するための設定

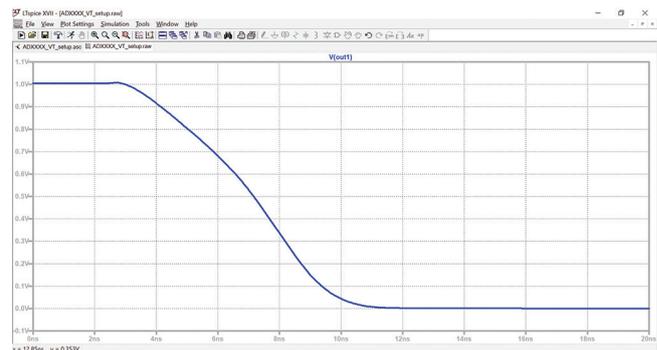


図21. 図20の回路のシミュレーション結果

VDD基準の50Ω負荷を使用した場合の 立下がり波形

続いて、VDDを基準とする50Ωの負荷を使用した場合の立下がり波形のデータを取得する方法を説明します。図22に示したのが、そのための設定とシミュレーション結果です。ご覧のように、立下がりの遷移全体を捉えるためには、トランジェント解析の解析時間として50ナノ秒以上が必要になります。

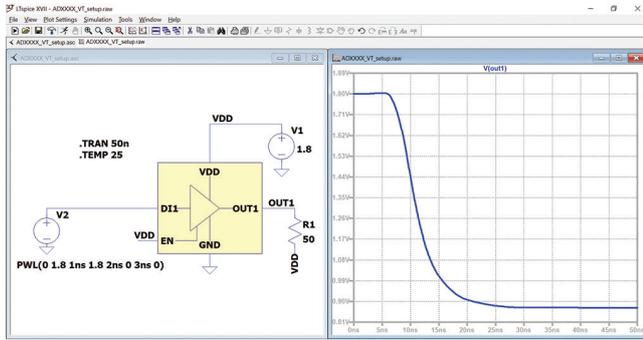


図22. VDD基準の50Ω負荷を使用した場合の
立下がり波形を取得するための設定。
シミュレーション結果も示しています。

グラウンド基準の50Ω負荷を使用した場合の 立上がり波形

立上がり波形については、PWL信号による入力スティミュラスを使用することにしました。ここでは、図23に示したように、グラウンドを基準とする50Ωの負荷を出力ピンに接続します。このような設定で、立上がり波形のV-tデータを取得します。

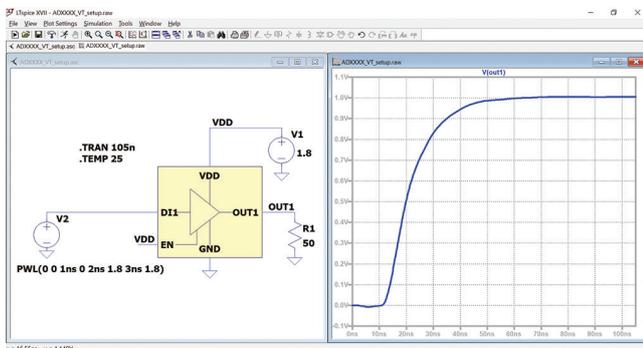


図23. グラウンド基準の50Ω負荷を使用した場合の
立上がり波形を取得するための設定。
シミュレーション結果も示しています。

VDD基準の50Ω負荷を使用した場合の 立上がり波形

VDDを基準とする50Ωの負荷を使用し、先ほどと同じ入力スティミュラスを使って立上がり波形のデータを取得します。そのための設定とシミュレーション結果を図24に示しました。

V-tデータが適切であることを確認するための1つの方法は、論理レベルのローとハイに対応する電圧を確認することです。VDDを基準とする場合の立上がり波形と立下がり波形では、ハイの部分の電圧がVDDとほぼ一致していなければなりません。また、ローの部分の電圧も同等の値であるはずで

す。ラウンドを基準とする場合の立上がり波形と立下がり波形では、ローの部分の電圧が約0Vになっていなければなりません。そして、ハイの部分の電圧も同等の値であるはずで

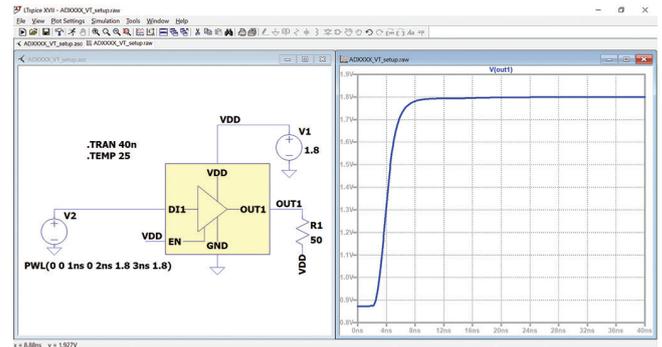


図24. VDD基準の50Ω負荷を使用した場合の
立上がり波形を取得するための設定。
シミュレーション結果も示しています。

波形データのエキスポート

上述した手順により、4つの条件に対応するV-tデータを得ることができました。得られた波形データは、次の手順によってエキスポート/保存します。まず、プロットを右クリックします。表示される「File」メニューから「Export data as text」を選択します（図25）。続いて、エキスポートの対象とする波形と、エキスポート先のディレクトリを選択します（図26）。

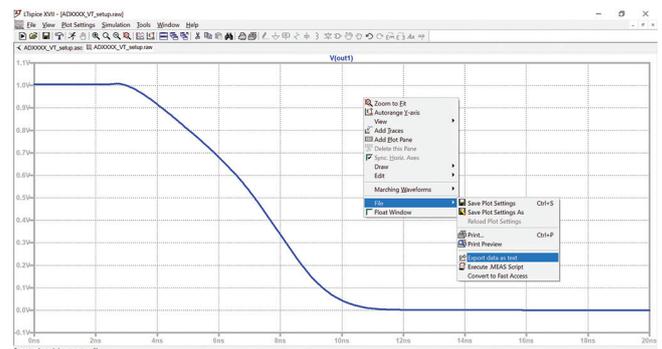


図25. データのエキスポート方法の選択

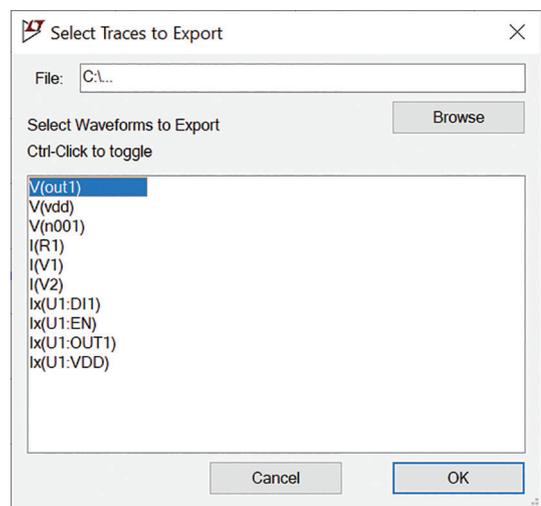


図26. 波形と保存先ディレクトリの選択

上昇率／下降率の値の取得

IBISモデルには、立上がり／立下がりのV-tデータの上昇率／下降率 (dV/dt) の値も記述できます。それにはキーワード [Ramp] を使用します。上昇率／下降率は、立上がり／立下がり波形において、遷移エッジの20%～80%の範囲で計算します。LTspiceでは、.MEAS/.PARAMディレクティブを使用することで、これらのパラメータの値を計算することができます。V-tデータを取得するための設定にディレクティブを追加すれば、V-tデータと上昇率／下降率を同時に取得することが可能です。

図27に、立上がり波形の上昇率を計算するための設定を示しました。立下がり波形の下降率を計算する場合には、VLOとVHIの時間の値を入れ替えます。出力が立下がる際には、バッファの論理レベルがハイから始まりローへと遷移するからです。

上昇率／下降率の計算用のLTspiceディレクティブ

上昇率／下降率の計算には、以下に示す3種のSPICEディレクティブを使用します。

- ▶ .TRAN：立上がり／立下がり波形のV-tデータを取得します。
- ▶ .OPTIONS：SPICE Error Logに表示される出力をCartesianモードに設定し、所望の有効桁数に丸めます。
- ▶ .MEAS：上昇率／下降率を計算します。

図27で使用している各ラベル（測定結果の表示名）の意味は以下のとおりです（併せて、図28もご覧ください）。

- ▶ VLO：論理レベルのローの電圧。
- ▶ VHI：論理レベルのハイの電圧。
- ▶ Diff：遷移における20%時点の電圧。この値をVLOに加算するかVHIから減算することにより、遷移における20%の時点と80%の時点の値が得られます。
- ▶ VX、VY：立上がり／立下がりの遷移エッジにおける20%時点と80%時点の電圧。
- ▶ dVとdT:IBISモデルのキーワード [Ramp] 用に計算された値。

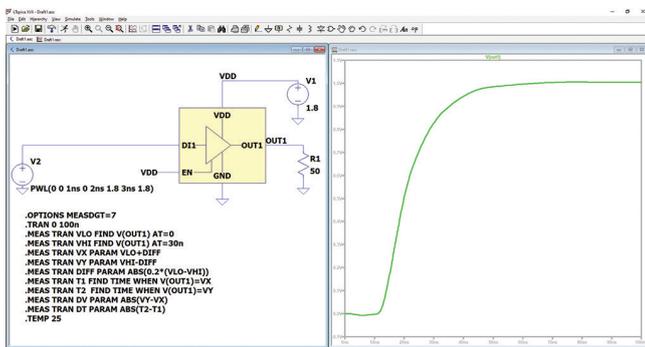


図27. 上昇率／下降率の計算用のディレクティブ。V-tデータの取得用の設定に追加します。

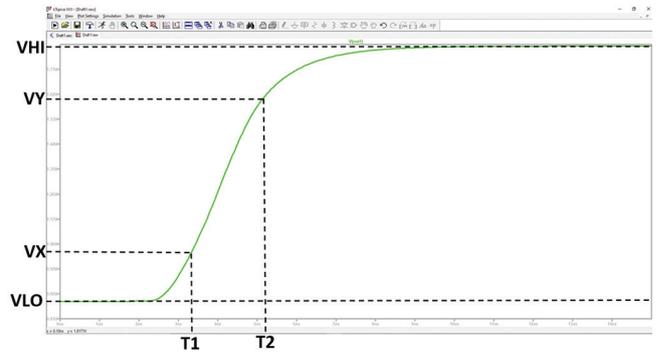


図28. 立上がり波形と各ラベルの関係

```

SPICE Error Log:

Circuit:

Direct Newton iteration for .op point succeeded.
Ignoring empty pin current: Ix(ul:gnd)
Ignoring empty pin current: Ix(ul:gnd)

vlo: v(out1)=0.8722517 at 0
vhi: v(out1)=1.798981 at 2.4e-008
vx: vlo+diff=1.057598
vy: vhi-diff=1.613635
diff: abs(0.2*(vlo-vhi))=0.1853458
t1: time=3.350427e-009 at 3.350427e-009
t2: time=5.192001e-009 at 5.192001e-009
dv: abs(vy-vx)=0.5560373
dt: abs(t2-t1)=1.841574e-009
    
```

図29. 上昇率／下降率の計算に対応するSPICE Error Log

IBISモデルの構築

必要なI-Vデータ、V-tデータをすべて取得したら、IBISファイル(.ibs)にまとめます。以下に示したのは、IBISファイルで使用するテンプレートの例です。このテンプレートは、IBISモデルを作成する際のリファレンスとして使用できます。

```

*****
ADI IBIS Modeling
*****

[[IBIS Ver]      3.2
[File Name]     adxxxx.ibs
[File Rev]      1.0
[Date]          April 15, 2021
[Source]        Analog Devices, Inc.
[Disclaimer]

[Copyright]

This model is an exclusive property of Analog Devices Inc
and is not to be resold or redistributed without the written
permission of Analog Devices.

Copyright(c) 2021 Analog Devices, All Rights Reserved
*****
    
```

IBISファイル(.ibs)は、キーワード[IBIS Ver]で始まります。その下には、ファイル名と改訂番号が続きます。3ステートの出力バッファをモデリングする場合、IBISのバージョン3.2以降に対応している必要があります。この例では、キーワード[IBIS Ver]によって「3.2」というバージョンを指定しています。IBISファイルのファイル名とキーワード[File Name]で指定したファイル名は同一でなければなりません。このルールに反すると、パーサによるテストでエラーが発生します。また、ファイル名には大文字を使用してはなりません。すべて小文字のファイル名でなければ、パーサによるテストでエラーになります。以下では、その他の重要なキーワードについて説明します。

IBISファイルの次のパートには、キーワード[Component]、[Manufacturer]、[Package]、[Pin]を記述します。ADxxxxには2つの入力バッファ(DIN1とEN)と1つの出力バッファ(DOUT1)が存在します。つまり、IBISファイルには計3つのバッファ・モデルについて記述することになります。キーワード[Package]は、デバイスのパッケージ・モデルに相当します。この部分には、パッケージの寄生要素(RLC)の値を記述します。デバイスが備えるすべてのバッファのモデル名は、キーワード[Pin]の下に宣言します。同キーワードは、変数名の定義に使用するものだと考えればよいでしょう。

```

|
[Component] adXXXX
[Manufacturer] Analog Devices, Inc.
[Package]
| variable          typ          min          max
R_pkg              3.59E-02      NA           NA
L_pkg              2.52E-09      NA           NA
C_pkg              1.38E-12      NA           NA
|
=====
|
[Pin]  signal_name  model_name
1      VDD1         Power
2      DIN1         cmos_dil
3      EN           cmos_en
4      DOUT1        cmos_out1
5      GND          GND
6      NC           NC
|
=====

```

各モデルの詳細は、キーワード[Model]の下に記述します。この部分に、取得したI-VデータとV-tデータを挿入することで、デバイスのデジタル・バッファのモデリングを行います。バッファ・モデルに記述する内容は、Model_typeに指定されたバッファの種類ごとに異なります。下に示したコードのcmos_dilというモデルは入力バッファを表します。このバッファ・モデルには、C_comp、[Power_Clamp]、[GND_Clamp]のデータだけが含まれることとなります。なお、入力バッファのモデルには、VinlとVinhの値も含まれます。これらの値は、デバイスのデータシートに記載されています。DIN1とENは、どちらも入力バッファなので、両者のバッファ・モデルは同じ構造になります。

```

*****
|
|                                     Model:cmos_dil
|                                     *****
|
[Model]          cmos_dil
Model_type      Input
Vinl=0.36
Vinh=1.44
|
|   variable          typ          min          max
C_comp          1.645481e-012      NA           NA
|
[Temperature Range]  25              NA           NA
[Voltage Range]      1.8              NA           NA
[Power Clamp Reference] 1.8          NA           NA
[GND Clamp Reference] 0              NA           NA
|
|
[Power_clamp]
| Voltage          I(typ)          I(min)          I(max)
-1.800000         1.898645E+00      NA           NA
-1.700000         1.714524E+00      NA           NA
-1.600000         1.511816E+00      NA           NA
|...
1.700000          7.601960E-12      NA           NA
1.800000          2.800000E-12      NA           NA
3.600000          -4.400000E-12      NA           NA
|
[GND_clamp]
| Voltage          I(typ)          I(min)          I(max)
-1.800000         -1.895275E+00      NA           NA
-1.700000         -1.710353E+00      NA           NA
-1.600000         -1.507593E+00      NA           NA
|...
1.700000          1.455209E-11      NA           NA
1.800000          1.620648E-11      NA           NA
3.600000          2.140000E-11      NA           NA
|
|                                     End of Model:cmos_dil
|                                     *****
|

```

一方、3ステートの出力バッファのバッファ・モデルには、入力バッファのモデルと同じキーワードに加えて、更に多くのI-VデータとV-tデータが記述されます。次に示すコードのcmos_out1というバッファ・モデルには、出力の容量性負荷を表すCrefと、リファレンス電圧のレベルを表すVmeasというサブパラメータが記述されています。通常、VmeasにはVDDの1/2の値が指定されます。

```

*****
|
|                                     Model:cmos_out1
|                                     *****
|
[Model]          cmos_out1
Model_type      3-state
Cref=15pF
Vmeas=0.9
|
|   variable          typ          min          max
C_comp          4.143501E-11      NA           NA
|
[Temperature Range]  25              NA           NA
[Voltage Range]      1.8              NA           NA
[POWER Clamp Reference] 1.8          NA           NA
[GND Clamp Reference] 0              NA           NA
[Pullup Reference]  1.8              NA           NA
[Pulldown Reference] 0              NA           NA
|
[POWER_clamp]
| Voltage          I(typ)          I(min)          I(max)
-1.800000E+00     2.074265E+00      NA           NA
-1.700000E+00     1.887999E+00      NA           NA
-1.600000E+00     1.685262E+00      NA           NA
|...
1.700000E+00      -6.471900E-11      NA           NA
1.800000E+00      -1.606903E-10      NA           NA
3.600000E+00      -8.012131E-10      NA           NA
|
[GND_clamp]
| Voltage          I(typ)          I(min)          I(max)
-1.800000E+00     -2.047257E+00      NA           NA
-1.700000E+00     -1.861165E+00      NA           NA
-1.600000E+00     -1.658421E+00      NA           NA
|...
1.700000E+00      1.221660E-10      NA           NA
1.800000E+00      1.638958E-10      NA           NA
3.600000E+00      5.271379E-10      NA           NA
|

```

また、3ステート・バッファのモデルには、C_comp、[Power_Clamp]、[GND_Clamp]のデータを記述します。それに加えて、以下に示すように、[Pullup]と[Pulldown]のI-Vデータも記述されます。

```
[Pullup]
| Voltage      I(typ)      I(min)    I(max)
-1.800000E+00  2.075567E+00  NA        NA
-1.700000E+00  1.889618E+00  NA        NA
-1.600000E+00  1.686874E+00  NA        NA
|...
1.700000E+00   -2.166668E-02  NA        NA
1.800000E+00   -2.181376E-02  NA        NA
3.600000E+00   -2.453158E-02  NA        NA
|
[Pulldown]
| Voltage      I(typ)      I(min)    I(max)
-1.800000E+00  -2.048355E+00  NA        NA
-1.700000E+00  -1.862534E+00  NA        NA
-1.600000E+00  -1.659785E+00  NA        NA
|...
1.700000E+00   1.934561E-02  NA        NA
1.800000E+00   1.942086E-02  NA        NA
3.600000E+00   2.086263E-02  NA        NA
|
[Rising Waveform]
R_fixture = 50
V_fixture = 1.8
|
| time          V(typ)      V(min)    V(max)
0.000000E+00   8.722465E-01  NA        NA
4.000854E-10   8.722824E-01  NA        NA
8.001709E-10   8.723076E-01  NA        NA
|...
3.880829E-08   1.799903E+00  NA        NA
3.920837E-08   1.799910E+00  NA        NA
4.000000E-08   1.799923E+00  NA        NA
|
[Falling Waveform]
R_fixture = 50
V_fixture = 1.8
|
| time          V(typ)      V(min)    V(max)
0.000000E+00   1.800005E+00  NA        NA
5.001068E-10   1.799995E+00  NA        NA
1.000214E-09   1.799995E+00  NA        NA
|...
4.851036E-08   8.723745E-01  NA        NA
4.901047E-08   8.723730E-01  NA        NA
5.000000E-08   8.723702E-01  NA        NA
|
[Rising Waveform]
R_fixture = 50
V_fixture = 0
|
| time          V(typ)      V(min)    V(max)
0.000000E+00   -5.744911E-06  NA        NA
1.050224E-09   7.964322E-06  NA        NA
2.100449E-09   4.059370E-05  NA        NA
|...
1.018718E-07   1.004326E+00  NA        NA
1.029220E-07   1.004331E+00  NA        NA
1.050000E-07   1.004340E+00  NA        NA
|
```

IBISモデルの末尾には、必ずキーワード[End]を記述します。

```
[Falling Waveform]
R_fixture = 50
V_fixture = 0
|
| time          V(typ)      V(min)    V(max)
0.000000E+00   1.004369E+00  NA        NA
2.000427E-10   1.004357E+00  NA        NA
4.000854E-10   1.004356E+00  NA        NA
|...
1.940414E-08   1.912548E-04  NA        NA
1.960419E-08   1.816631E-04  NA        NA
2.000000E-08   1.631222E-04  NA        NA
|
[Ramp]
| variable      typ          min          max
dv/dt_r      6.026073E-01/1.327850E-08  NA          NA
dv/dt_f      5.565810E-01/5.189650E-09  NA          NA
R_load = 50
|
| *****
|                               End of [Model]: cmos_out1
| *****
[End]
```

IBISモデルの検証

Part 1で説明したように、IBISモデルの検証は、パーサによるテストと関連の確認の2つのプロセスで構成されます。パーサによるテストでは、作成したIBISファイルがIBISの仕様に準拠しているか否かを検証します。関連の確認では、モデルがリファレンス(元になったSPICEモデル)に近い形で動作するか否かを検証します。

パーサによるテスト

作成したIBISファイルについては、関連の確認のプロセスに進む前に、パーサによるテストを実施する必要があります。ibischkは、IBISファイルのチェックに使用されるGolden Parserです。同パーサを使うことで、作成したファイルがIBISの仕様に従っているかどうかを確認できます。本稿執筆の時点で、ibischkの最新バージョンは7です。詳細については、ibis.orgを確認してください。

パーサによるテストには、ibischkを搭載するIBISモデルの編集用ソフトを使用するとよいでしょう。例えば、Cadence Model IntegrityやHyperlynx Visual IBIS Editorといったツールを使用するという事です。これらのツールを使えば、構文を簡単にチェックすることができます。この種のツールを所有していない場合には、実行可能コードをibis.orgから入手すればよいでしょう(無償)。各種OSに対応するコードが用意されているので、使用環境についての心配は無用であるはずで。

関連の確認

関連の確認というのは、作成したIBISモデルがリファレンス(ここではSPICEモデル)と同じように動作するか否かを確認する作業のことで。IBISモデルについては、表7に示すような品質レベルが定義されています。この品質レベルは、リファレンスに対するIBISモデルの精度を表します。この精度は、実施したテストに応じて確認されることとなります。本稿の例の場合、リファレンスはADxxxxのSPICEモデルなので、検証が完了すれば、作成したIBISモデルの品質レベルは2aになります。品質レベルが2aであるということは、作成したモデルがIBISの仕様に準拠しており、すべてのパラメータがデータシートに記載された値と比較して適切で、関連もとれているということを意味します。

表7. IBISモデルの品質レベル

品質レベル	説明
レベル0	Golden Parser (ibischk) によるテストに合格
レベル1	チェックリスト・ドキュメントの定義に完全に従っている
レベル2a	シミュレーションによって関連を確認済み
レベル2b	実測によって関連を確認済み
レベル3	上記のすべてを満たす

IBISモデルとリファレンスであるSPICEモデルの相関は、一般的なステップに従って確認することができます。そのフローチャートを図30に示しました。

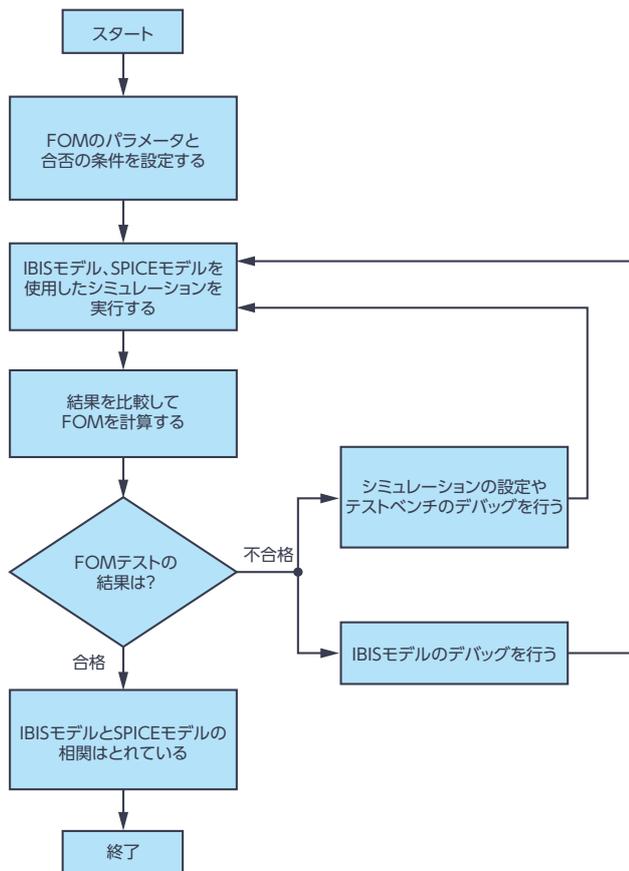


図30. IBISモデルとSPICEモデルの相関を
確認するためのフローチャート

FOMの設定

相関がとれていることの根拠になるのは、次のようなことです。すなわち、同じ入力スティミュラスと同じ負荷の条件下で、IBISモデル、SPICEモデルによって表現した各デジタル・インターフェースが同等に動作することが確認されたということです。これは、両者の出力が理論的に完全に一致することを意味します。一般に、IBISモデルの出力がリファレンスであるSPICEモデルの出力とどれだけ近いかを示す方法は2つあります。1つは定性的なFOMテスト、もう1つは定量的なFOMテストです。それらのテストによって、IBISモデルとSPICEモデルの相関がとれているか否かを判定します。

定性的なFOMテストの1つの方法は、目視で確認するというものです。2つの出力を目で見て、相関がとれているか否かを判断します。IBISモデル、SPICEモデルを使用して行ったシミュレーション結果を重ね合わせて、2つのグラフに相関があるかどうかを確認するというものです。この方法は、定量的なFOMテストに進む前の事前テストとして利用できます。比較的低い周波数/ビット・レートでインターフェースが動作する場合には、このテストでも十分です。

IBIS IO Buffer Accuracy Handbookには、別の定性的なFOMテストとして、グラフの包絡線を指標にする方法が示されています。その方法では、プロセス/電圧/温度が極端な値に振れた場合の最小/最大の曲線を、相関の有無を表す境界線として使用します。IBISモデルを使用したシミュレーション結果をグラフ化し、そのグラフ上のすべての点が最小/最大曲線の間にあれば合格と見なすというものです。ただ、本稿の例は標準的な条件にしか対応していないので、この方法を適用することはできません。

一方、定量的なFOMテストでは、算術演算を使用してIBISモデルとSPICEモデルの相関の有無を判定します。IBIS IO Buffer Accuracy Handbookには、曲線の重ね合わせに基づく方法が示されています。その方法では、IBISモデル、SPICEモデルを使用したシミュレーション結果のデータ・ポイントを使用します。まず、それぞれのデータ・ポイントの間で、X軸/Y軸方向の差を求めます。続いて、それらの絶対値を合計し、対象となる軸の範囲とポイント数の積で割るという計算を行います。以下のような式に基づいた計算を実施することにより、相関の判定を行うということです。

$$FOM_{COM} = 100\% \times \left[1 - \frac{\sum_{i=1}^N |x_i(\text{reference}) - x_i(\text{IBIS})|}{\Delta x \times N} \right] \quad (3)$$

この方法は、本稿のユース・ケースに対する相関の確認手段として適しています。但し、他にも考察すべき事柄が存在します。上式の方法では、IBISモデル、SPICEモデルを使用したシミュレーションの結果を同じXY座標上に表示する必要があります。そのためには、数値アルゴリズムと補間処理を適用しなければなりません。本稿では、定量的なFOMテストを簡単に実施したいという方に向けて、曲線面積を指標とする方法を紹介します。この方法では、曲線とX軸を境界線とする領域を使用します。SPICEモデルを使用した場合の結果と、IBISモデルによって得られた曲線下面積の計算値を比較します。この方法は、以下の式で表すことができます。

$$FOM_{CAM} = 100\% \times \left[1 - \frac{|A_{IBIS} - A_{SPICE}|}{A_{SPICE}} \right] \quad (4)$$

なお、この方法を適用する際には、作成したIBISモデルが定性的なFOMテストに合格することを事前に確認しておかなければなりません。それにより、IBISモデルとSPICEモデルから得られる曲線が同一の位相で重なって表示されることが保証されます。曲線下面積の計算には、台形則や中点則といった数値的手法を使用することができます。ただ、IBISモデル、SPICEモデルから得られた両方の結果に対しては、同じ手法を適用しなければなりません。また、この方法を使用する場合、面積の近似値ができるだけ正確に得られるように、できるだけ多くのポイントを使用する必要があります。

ADxxxxのIBISモデルの検証

作成したIBISモデルを検証するための最初のステップは、パーサによるテストです。本稿の例では、HyperLynx Visual IBIS Editorを使用してadxxxx.ibsというIBISファイルを作成しました。図31に示したのは、このファイルをパーサによってテストした結果です。パーサによるテストでは、エラーをなくすることが目標になります。テストを実行した結果、エラーや警告が表示された場合にはファイルを修正します。パーサによるテストを実施することで、作成したIBISモデルが各種シミュレーション・ツールに対する互換性を備えていることが保証されます。

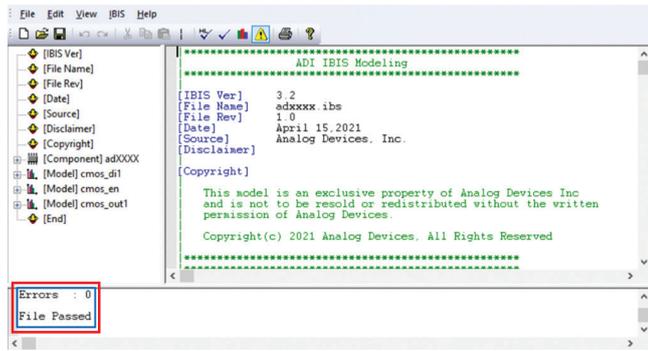


図31. ADxxxxのIBISファイルをパーサでテストした結果

次のステップでは、FOMパラメータの設定を行います。本稿では、定性的なFOMと曲線面積指標を相関の判定基準として使用する方法だけを示すことにします。ここで使用するのは、同じ入力刺激、同じ負荷の条件下で、IBISモデル、SPICEモデルを使用したシミュレーションの結果として得られるトランジェント応答曲線です。曲線面積指標の計算に基づくFOMが95%以上である場合に、両者は相関がとれている状態であると判定することにします。以下では、DOUT1、DIN1、ENの相関を確認する方法を示します。

DOUT1の相関

図32に示したのは、DOUT1の相関を確認するためのテストベンチです。これにより、LTspiceモデルのシミュレーションを実施します。回路にドライバをイネーブルにするための適切な電圧源を配置し、DOUT1を駆動するためのパルス信号源をDIN1ピンに接続しています。LTspiceにおいてDOUT1のモデルを完成させるために、他のコンポーネントも追加しています。図中のC_compはダイの容量です。また、負荷容量C_loadを配置しています。更に、パッケージの寄生要素であるR_pkg、L_pkg、C_pkgを追加しています。

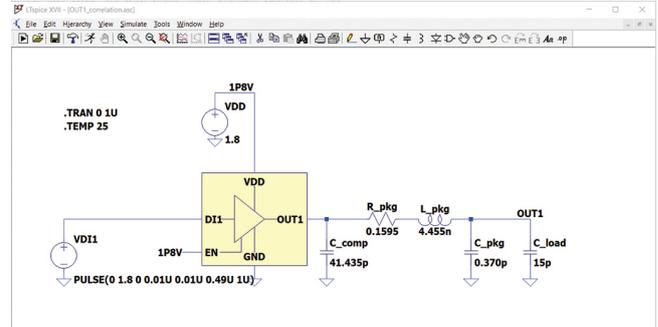


図32. DOUT1の相関を確認するためのLTspiceのテストベンチ。LTspiceモデルのシミュレーションに使用します。

DOUT1の相関を確認するには、IBISモデルのテストベンチも必要です。こちらは、Keysight Advanced Design System (ADS)上で図33のように構成しました。LTspiceモデル用のテストベンチと同じ入力刺激、C_load、電圧源、トランジェント解析を使用しています。ただ、C_compとパッケージの寄生要素は、3状態・バッファのモデルに既に含まれています。したがって、ADS上の回路図に対してそれらを付加する必要はありません。

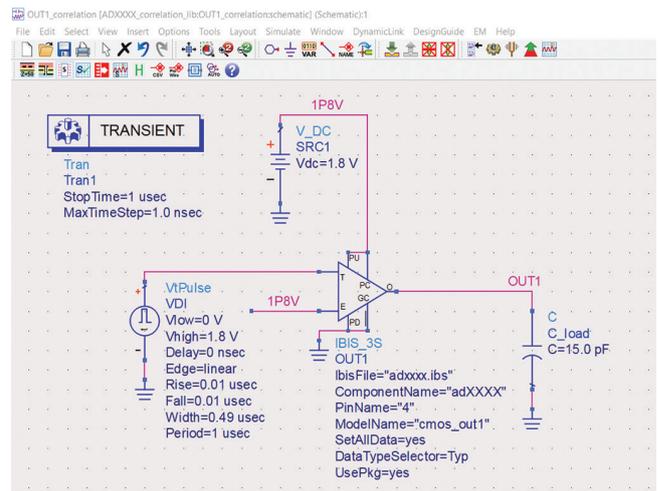


図33. DOUT1の相関を確認するためのADSのテストベンチ。IBISモデルのシミュレーションに使用します。

トランジェント応答の曲線は、C_loadのノードで取得します。LTspiceによるシミュレーション結果とADSによるシミュレーション結果を取得し、両者を重ね合わせて定性的なFOMを確認しました。図34に示すように、DOU1の両応答はよく一致しています。両者の違いは、曲線面積指標を使うことで定量化できます。この例では、1マイクロ秒のトランジェント時間に対する曲線下面積を計算しました。すると、曲線面積指標の計算値は99.79%となりました。これは、95%以上で合格という条件を満たしています。つまり、IBISモデルとSPICEモデルにおいて、DOU1は相関のとれている状態にあります。

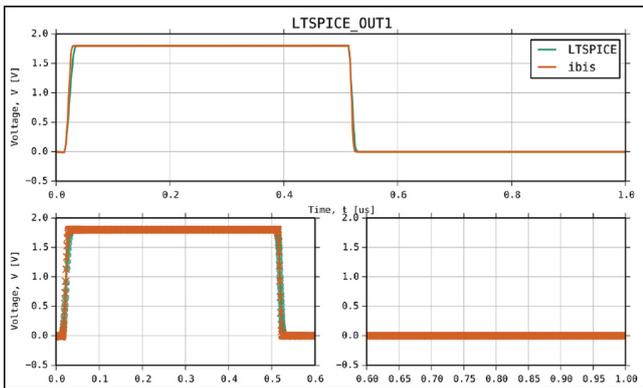


図34. LTspiceモデルとIBISモデルから得たDOU1の応答

DIN1とEN

続いて、入力バッファの検証を行います。DOU1の場合と同様に、LTspice上、ADS上でトランジェント応答曲線を取得し、定性的なFOMと曲線面積指標によって相関の有無を判定します。図35に、LTspice用のテストベンチを示しました。この方法は、DIN1ピンとENピンの両方に適用できます。DOU1の場合と同様に、DIN1ピンのすぐ近くにC_compを配置します。また、パッケージの寄生要素（RLC）も付加しています。更に、50Ωの抵抗R_seriesと入力スティミュラス用のパルス電圧源も追加します。この状態で、DI1_probeをプローブ・ポイントとして応答を取得します。

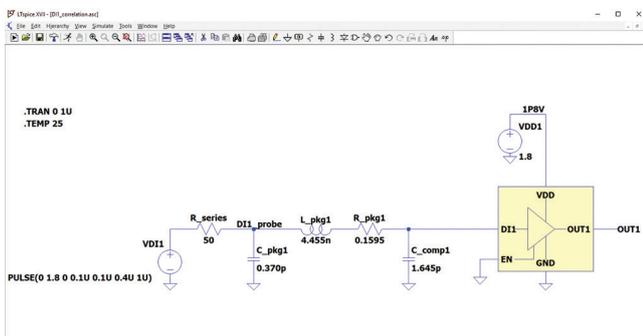


図35. DIN1 (EN) の相関を確認するためのLTspiceのテストベンチ

図36に示したのは、入力バッファの検証に使用するADS用のテストベンチです。50Ωの抵抗R_seriesを入力に近いに配置し、LTspiceの場合と同じ入力スティミュラスを適用します。ただ、C_compと寄生要素（RLC）はIBISモデルに既に含まれているので、回路図上には追加しません。この状態で、DI1_probeをプローブ・ポイントとしてトランジェント応答を取得します。

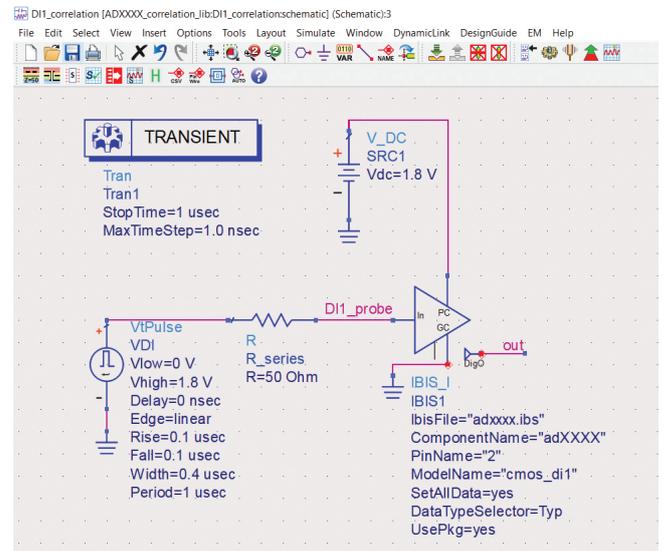


図36. DIN1 (EN) の相関を確認するためのADSのテストベンチ

まず、LTspiceとADSで取得した各トランジェント応答曲線を重ね合わせて、定性的なFOMテストを実施します。図37に示したとおり、両曲線は完全に一致しているように見えます。また、DIN1の曲線面積指標を計算すると、その値は100%となります。つまり、95%以上で合格という条件を満たしています。同様の手法により、ENピンの相関が得られていることも確認できました。

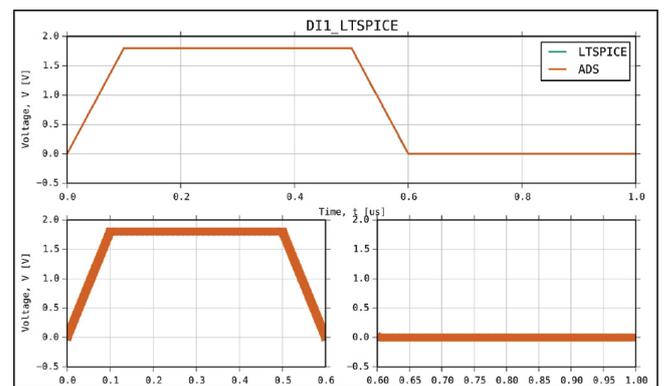


図37. LTspiceモデルとIBISモデルから得たDIN1の応答

まとめ

本稿では、LTspiceを使用してデータを取得し、IBISモデルを作成する方法を紹介しました。また、IBISモデルと、リファレンスであるSPICEモデルの相関を、定性的なFOMと定量的なFOM（曲線面積指標）によって確認する方法も示しました。このような確認を行うことにより、作成したIBISモデルがSPICEモデルと同じように動作するという確信を得ることができます。デジタルI/Oとしては、本稿で取り上げたのとは異なる種類のものも存在します。ただ、本稿で示したC_comp、I-Vデータ、V-tデータの取得手順は、他のI/Oのモデルを作成する場合にも応用することが可能です。

LTspiceを無償ダウンロード／インストールすれば、独自のIBISモデルの作成を直ちに開始することができます。

参考資料

Mercedes Casamayor [AN-715 Application Note: A First Approach to IBIS Models: What They Are and How They Are Generated (初めてのIBISモデル：モデルの概要と開発方法)] Analog Devices、2004年

IBIS [I/O Buffer Accuracy Handbook (I/Oバッファの精度に関するハンドブック)] IBIS Open Forum、2000年4月

Roy Leventhal, Lynne Green [Semiconductor Modeling: For Simulating Signal, Power, and Electromagnetic Integrity (半導体のモデリング：信号、電力、電磁界インテグリティのシミュレーション)] Springer、2006年

Michael Mirmak, John Angulo, Ian Dodd, Lynne Green, Syed Huq, Arpad Muranyi, Bob Ross [IBIS Modeling Cookbook for IBIS Version 4.0 (IBISモデリング・クックブック IBIS Version 4.0対応版)] The IBIS Open Forum、2005年9月



著者について

Rolynd Troy Aquino (rolynd.aquino@analog.com) は、アナログ・デバイスズのプロダクト・アプリケーション・エンジニアです。新技術統合チームに所属しています。主に、アナログ・デバイスズの製品を対象としたIBIS/IBIS-AMI/LTspiceモデルの作成とシミュレーションを担当。2014年にインターンとして勤務した後、2016年に入社しました。2015年にマプア大学（マニラ）で電子工学の学士号を取得しています。



著者について

Francis Ian Calubag (francislan.calubag@analog.com) は、アナログ・デバイスズのシステム・アプリケーション・エンジニアです。2019年、システム・アプリケーション・チームにインターンとして参加。2020年に入社し、新技術統合チームに加わりました。主に、アナログ・デバイスズの製品を対象としたIBIS/LTspiceモデルの作成とシミュレーションを担当しています。2020年にリセウム・オブ・ザ・フィリピン大学カビテ校で電子工学の学士号を取得しました。



著者について

Janchris Espinoza (janchris.espinoza@analog.com) は、アナログ・デバイスズのプロダクト・アプリケーション・エンジニアです。新技術統合チームに所属しています。主に、アナログ・デバイスズの製品を対象としたIBISモデルの作成とシミュレーションを担当。2019年にAnalog Garageチームでインターンとして勤務し、2020年に入社しました。2020年にデラサール大学で電子工学の学士号を取得しています。