Validating and Using IBIS Files

National Semiconductor Corp.
Interface Products Group

Overview

The IBIS (Input/Output Buffer Information Specification) behavioral model is widely used for high-speed designs to evaluate Signal Integrity issues. With board designs getting faster and faster, designers are increasingly turning to simulation to evaluate designs before or instead of prototyping. This can reduce cycle time and headaches to be able to simulate critical nets beforehand. IBIS has the advantages over SPICE of file size, not revealing any intellectual property and multiple simulator platform support, so vendors have endorsed the IBIS specification and most vendors readily provide IBIS files. Yet understanding what an IBIS file is or how it works in different simulators can be challenging. This paper intends to show IBIS users how to verify an IBIS file, how IBIS files work in different simulators, and general tips and tricks on how to get up and running using IBIS.

Table of Contents

List of Figures

Viewing and Editing an IBIS File

Once you have an IBIS file, you need to verify the file in some manner before you should use it in a system level simulation. The first thing you need to do is to run the IBIS file through the “golden parser”. The IBIS Open Forum (an EIA group that has ownership of the IBIS specification) developed the “golden parser” as a way for users to check IBIS files for consistency. The “golden parser” is a software parser that checks the IBIS file for correct syntax and basic waveform properties. (Remember, the IBIS format is ASCII text) The “golden parser” can be downloaded for free from the IBIS Open Forum website. Multiple operating systems are supported. The Visual IBIS Editor, a free tool provided by Innoveda (now (CY2002) Mentor Graphics), allows the editing of an IBIS file, viewing the I/V and V/T waveforms, and includes the “golden parser”.

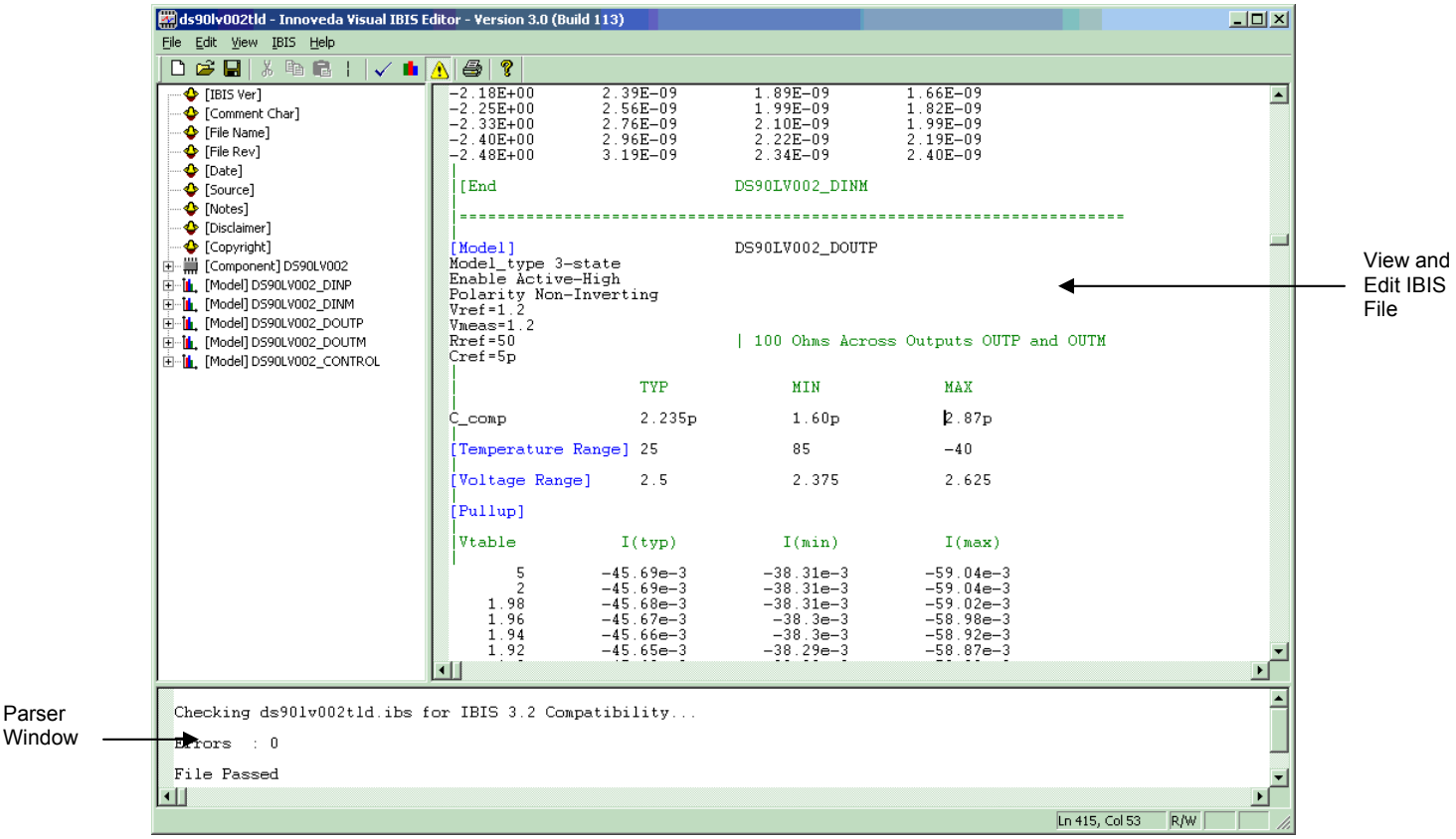

Figure 1. Visual IBIS Editor

An Overview of a Model Spec in IBIS

To better understand the IBIS standard, let's look at an input model. A typical input model contains a power clamp and a ground clamp (that represents the ESD structure). Two additional keywords are required for an input model: Vih and Vil, the logic thresholds of the input. These parameters are important because an IBIS simulator uses these threshold values to compute Signal Integrity issues such as overshoot/undershoot, noise margins, and so forth.

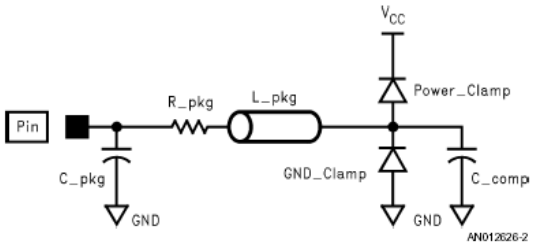

Figure 2. Input Model Structure in IBIS

The input model also includes the package parasitics (which are added separately and will be covered later) and the input die capacitance, C_comp. The C_comp parameter is connected to the input usually with reference to ground when the IBIS file is used in an IBIS simulator. The syntax for the input model is below:

```
Define Model as Input ──►  [Model]                LVDS_INPUT
                           Model_type Input
                           Vinh=.05               | Vth
                           Vinl=-.05              | Vtl   ◄── Input Logic Threshold Values
                           |
                           |                       TYP      MIN      MAX
                           |
Receiver Input Capacitance ►  C_comp               3p       3p       3p
                           |
                           [Temperature Range]     25       85       -40
                           |
                           [Voltage Range]         2.5      2.375    2.625
                           |
                           [GND Clamp]
                           |
                           |Vtable        I(typ)        I(min)        I(max)
                           |
                           -2.50E+00      -5.70E-01     -5.32E-01     -6.12E-01
Ground Clamp I/V Data ──►  …
                           …
                           2.45E+00       7.30E-07      7.43E-07      7.18E-07
                           |
                           [POWER Clamp]
                           |
                           |Vtable        I(typ)        I(min)        I(max)
                           |
                           0.00E+00       4.11E-10      5.49E-10      1.85E-10
Power Clamp I/V Data ──►   …
                           …
                           -2.48E+00      3.19E-09      2.34E-09      2.40E-09
                           |
                           | End                  LVDS_INPUT
```

Figure 3. Input Model Syntax in IBIS

The I/V data for the power and ground clamp curves can be generated from simulation or from lab measurement. The data is entered in table format and includes typical, minimum, and maximum data to represent process-voltage-temperature variations. The keywords such as voltage, temperature, and logic thresholds define the rest of the model.

Visual Check

The next step is to visually inspect the waveforms. By visually looking at the I/V and V/T waveforms, a lot of issues such as non-monotonic points and missing data can be found.

The Ground Clamp Curve

The ground clamp curve should have negative current and approach zero current as the voltage goes from –Vcc to Vcc. A ground clamp circuit goes active when the input goes below ground, and for a typical circuit this is usually around 0.7V below ground. A typical ground clamp curve for an LVDS input buffer is shown below.

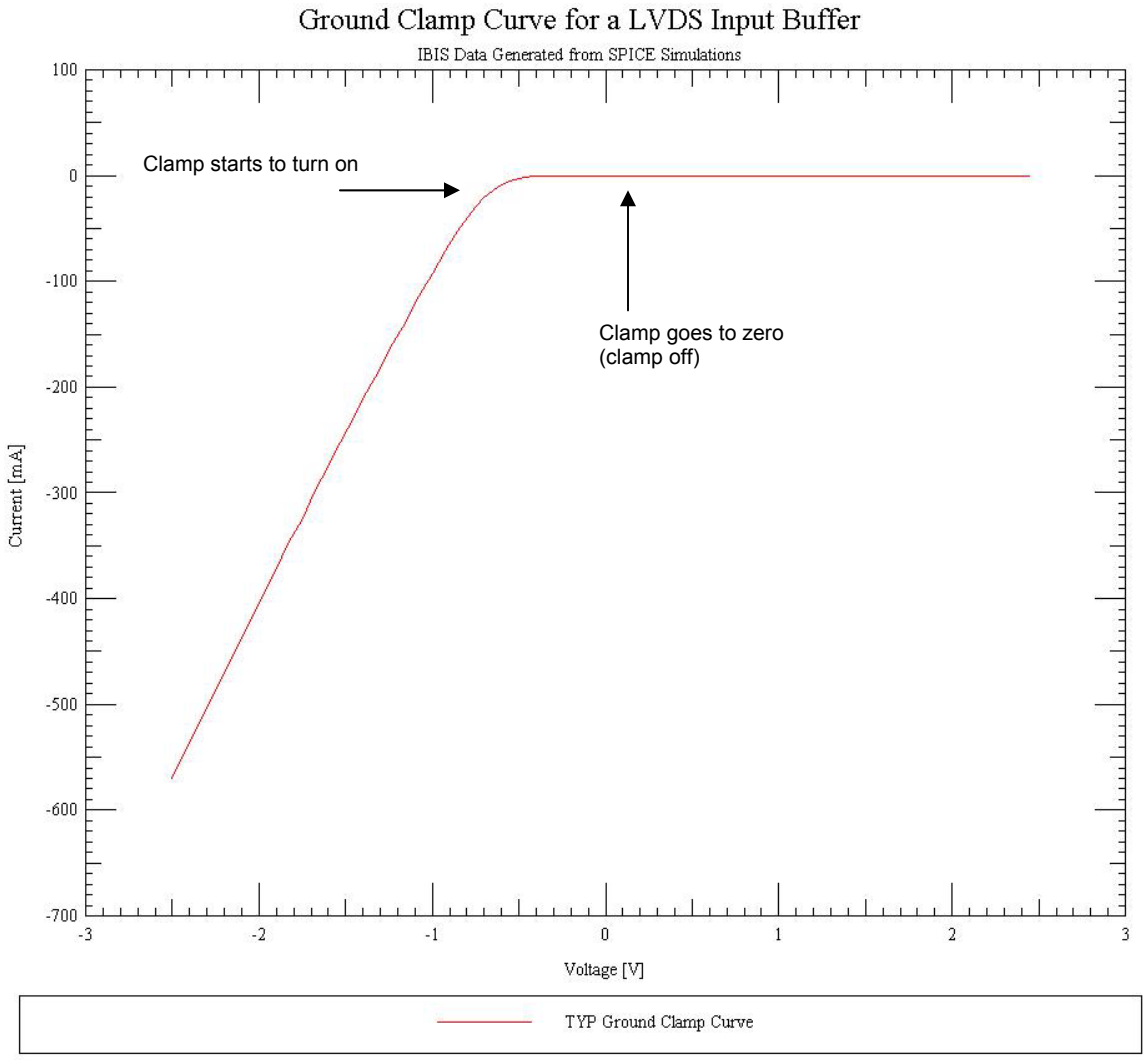

Figure 4. Ground Clamp Curve

Power Clamp Curve

A power clamp circuit goes active when the input goes above Vcc, and for a typical circuit this is usually around 0.7V above Vcc. However, in IBIS, the power clamp data in IBIS is entered in as Vtable=Vcc-Vin. The input voltage, Vin, is swept from Vcc to 2* Vcc. So instead of the data going from Vcc to 2*Vcc (referred to as ground relative), the data is shown as going from –Vcc to 0. (Referred to as Vcc relative) The power clamp curve should have positive current and approach zero current as the voltage goes from –Vcc to 0. A typical power clamp curve for an LVDS input buffer is shown below. (Vcc relative.)

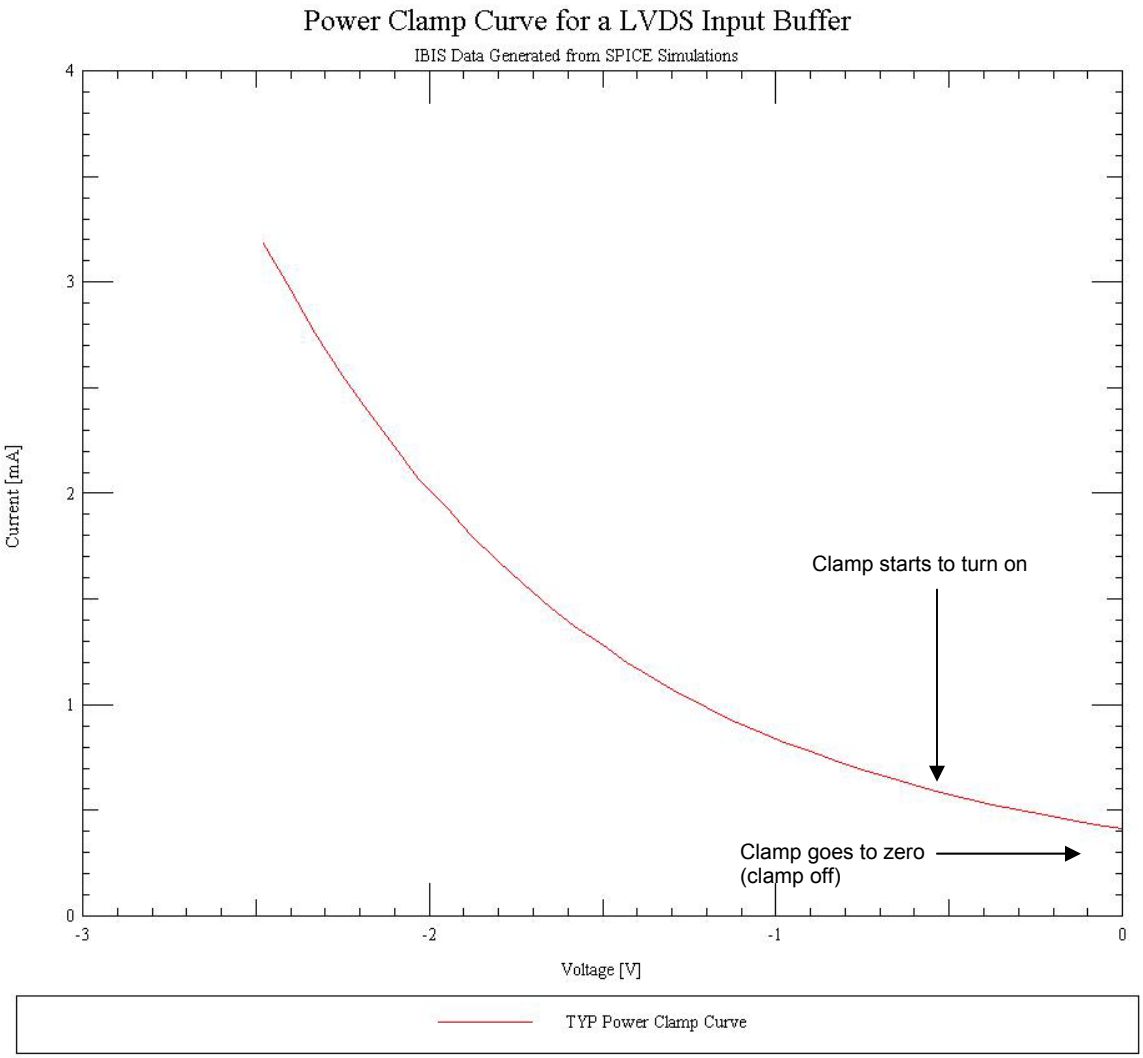

Figure 5. Power Clamp Curve

Pulldown Curve

The pulldown curve should have negative current until it crosses zero current, and then the current should be positive. The pulldown curve describes the strength of the output drive in a low state. The zero current crossing point can be an indicator of the steady state value of the output. For a typical LVDS device that switches from 1.0 V to 1.4 V, the pulldown should generally cross at 1.0 V. (This depends on the drive of the output, the internal circuitry of the device, and other factors). The power and ground clamp curves need to be subtracted from the pulldown curve if it is a 3-State output. If the power and ground clamp curves are not subtracted, then the effects of the ground clamp can clearly be seen on the negative current of the pulldown curve. (Since the power clamp curve is usually so small, the effects are negligible) A typical pulldown curve for an LVDS 3-State output buffer is shown below.

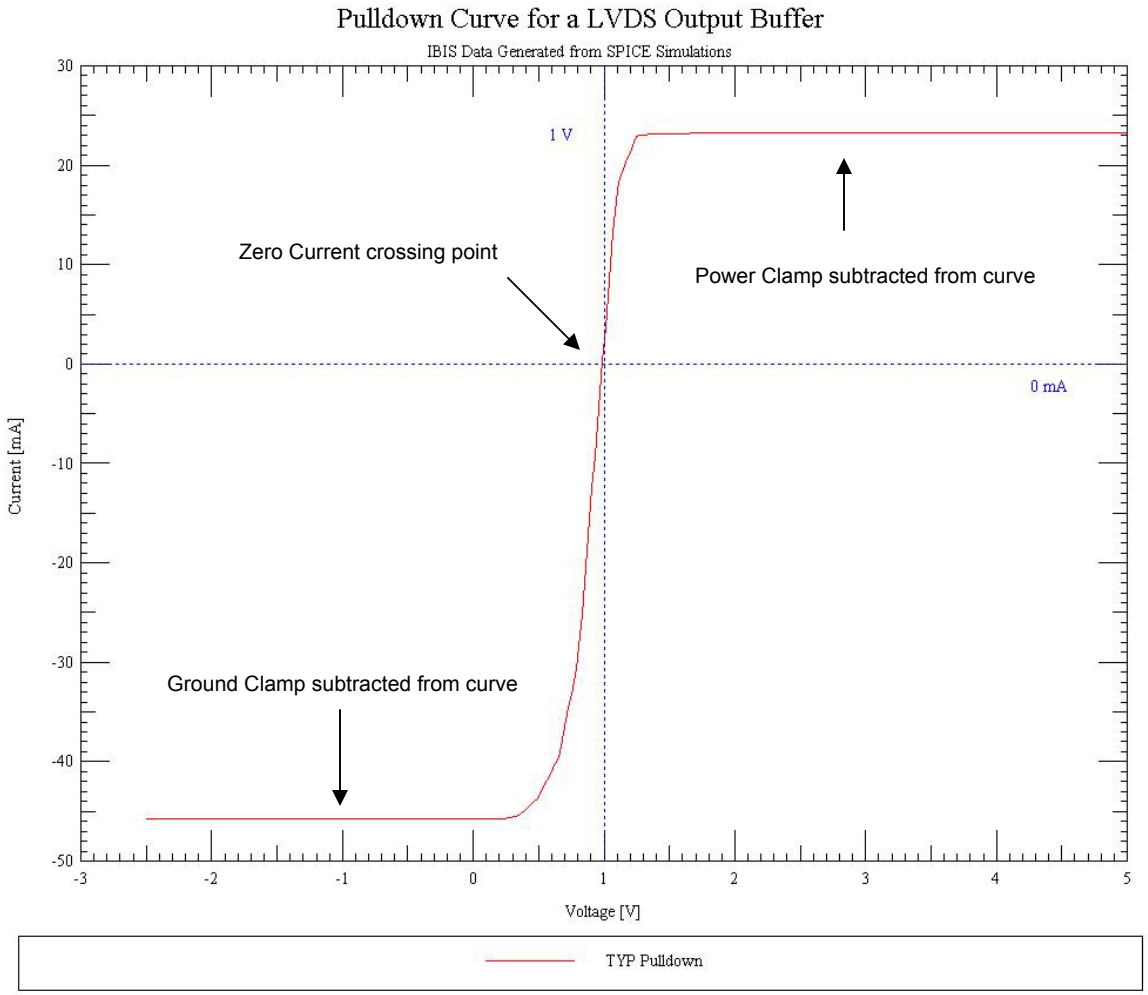

Figure 6. Pulldown Curve

Pullup Curve

The pullup curve should have positive current until it crosses zero current, and then the current should be negative. The pullup curve describes the strength of the output drive in a high state. The zero current crossing point can be an indicator of the steady state value of the output. For a typical LVDS device that switches from 1.0 V to 1.4 V, the pullup curve should generally cross at 1.4 V. (This depends on the drive of the output, the internal circuitry of the device, and other factors). Since the pullup curve is entered into the IBIS format Vcc relative (like the Power Clamp curve), the zero crossing point will be 1.0 V. The power and ground clamp curves need to be subtracted from the pullup curve if it is a 3-State output, just like the pulldown curve. A typical pullup curve for an LVDS 3-State output buffer is shown below. (Vcc relative.)

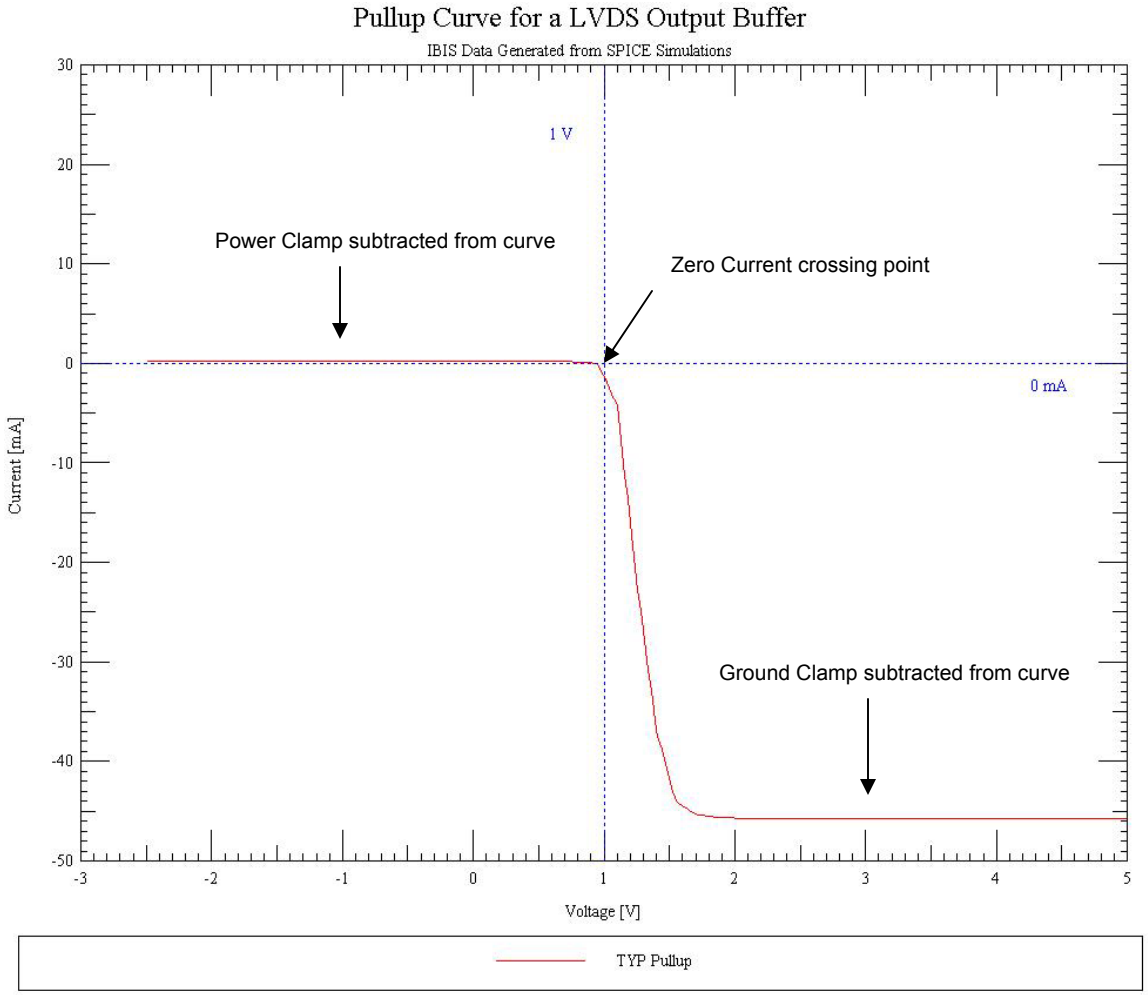

Figure 7. Pullup Curve

Rising and Falling Waveforms

The rising and falling waveform data is generated by applying a voltage ramp to the input, and measuring the time it takes the output to reach a steady state while it is connected through a load. For a LVDS device, this load is usually half the recommended differential load resistor (R_fixture) tied to a reference voltage (Vref) that is equal to the VOS of the device. This ensures that the single-ended LVDS output operates properly. It is important for the waveforms to reach a steady state value. The rising and falling slew rates of the device entered under the Ramp keyword are derived from these waveforms. The slew rates are calculated at the 20% and 80% points. A typical rising and falling waveform for an LVDS 3-State output buffer is shown below.

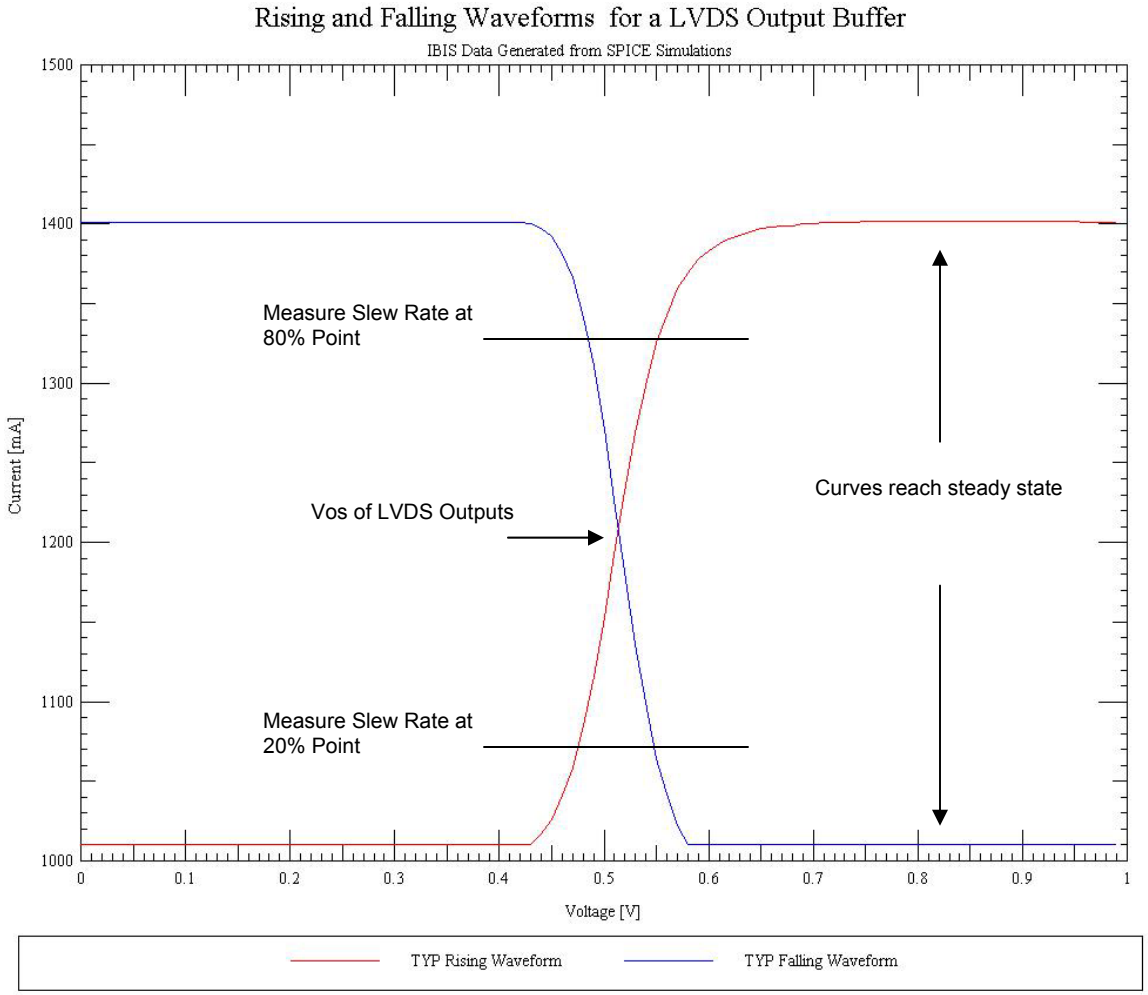

Figure 8. Rising and Falling Waveforms

Parsing an IBIS File

When you parse the IBIS file, if you receive no warnings or errors, you can move on to the next step of using your IBIS file. If you do get warnings or errors, then you will have to do some probing to see what's going on.

Syntax Warnings and Errors

An IBIS file contains I/V and V/T data of the input/output of a device. This data, when formatted into the IBIS format, is used along with keywords to create a behavioral model of the device. The "golden parser" checks to ensure that the keywords are used properly. An example of a syntax warning is the following:

WARNING - Model 'DS90LV002_DINP': Model_type 'Input' must have Vinh set

If the logic thresholds are not defined for an Input model, then the parser will give a warning. Syntax warnings and errors are the easiest to fix. One can refer to the IBIS standard for the model type to determine what keywords need to be present. If a typo is not causing the syntax issue, then usually it is a missing parameter, which can almost always be obtained from the Datasheet. The model supplier can also be contacted for missing information or other issues with the IBIS file.

Mismatch between DC and AC Endpoints

The "golden parser" checks to make sure that the DC and AC endpoints of a data set match. The parser uses load line analysis to calculate the driver's high and low DC output levels using the I/V data, and then checks if these levels match the AC output levels in the given V/T data. If there is a mismatch, the following warning will be generated:

WARNING - Model DS90LV002_DOUTM: The [Falling Waveform]
with [R_fixture]=50 Ohms and [V_fixture]=1.2V
has TYP column DC endpoints of 1.01V and 1.41v, but
an equivalent load applied to the model's I-V tables yields
different voltages (1.01V and 1.40V),
a difference of 0.11% and 0.58%, respectively.

The warning states that for a given load (R_fixture and V_fixture), the DC endpoints are 1.01V and 1.41V. These are the endpoints from the Falling Waveform V-T curve. The warning continues on to state that from an equivalent load applied to the I-V data, the DC endpoints are 1.01V and 1.40V. The parser also gives the "error" percentage between the two sets of endpoints. If the IBIS data is generated correctly, then there should be no warnings concerning mismatched data. The parser gives a warning for mismatches under 2%, and anything above 2% is an error. Ideally, there should be no mismatch in the endpoint data. However, a less than 2% error will not generally affect the overall accuracy of the IBIS file.

To better understand how a load line is created, we will use a typical LVDS device as an example. The pullup and pulldown curves in an output model define the output drive in the high and low state. A load line based upon the load R_fixture and V_fixture is constructed on each I/V curve to obtain the DC high and low endpoints. These endpoints should match the AC endpoints that were generated using the same test load in the V/T data.

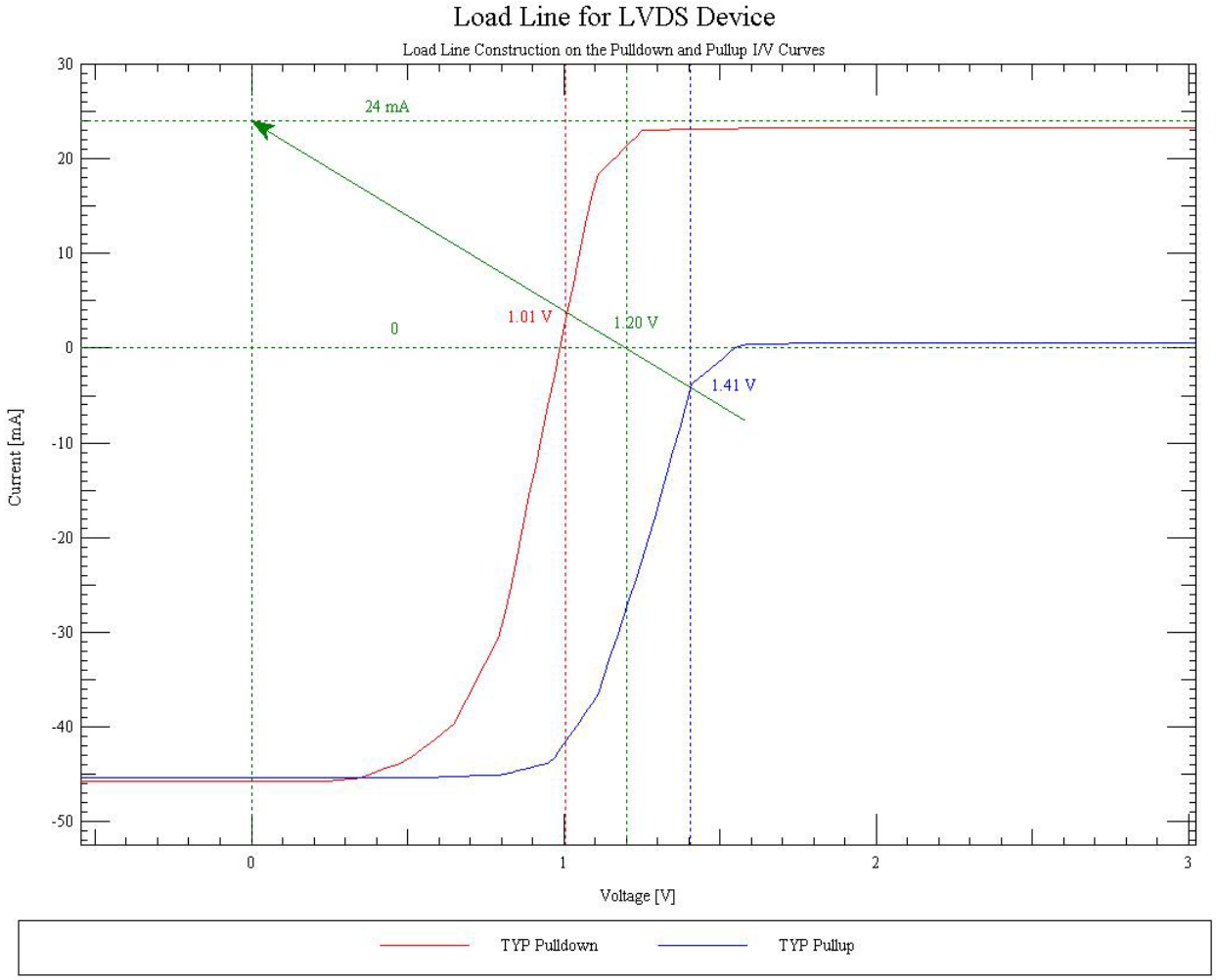

Figure 9. Load Line for Typical LVDS Device

The pulldown I/V curve is used to obtain the DC low endpoint. The test load used to generate the AC low endpoint in the V-T data is 50 Ohms to 1.2 V. (R_fixture and V_fixture keywords under the Falling Waveform data set) The reference of the load line needs to be set so that there will be zero current at 1.2 V. The second point is the current when the test load is at 1.2V, which results in 24 mA. A load line is then drawn between these two points. The load line current can be calculated by the following equation:

(1.2 – Vsweep) /50 = Load Line Current

The intersection point from the load line is 1.01 V. The pullup I/V curve is used to obtain the DC high endpoint. The same procedure to generate a load line for the pulldown curve is used to construct a load line on the pullup curve. The intersection point from the load line is 1.41 V. The calculated DC endpoints from the I/V data resulted in endpoint values of 1.01 V and 1.41 V. From either the Rising or Falling Waveform V/T data, it can be seen that the AC endpoint values match.

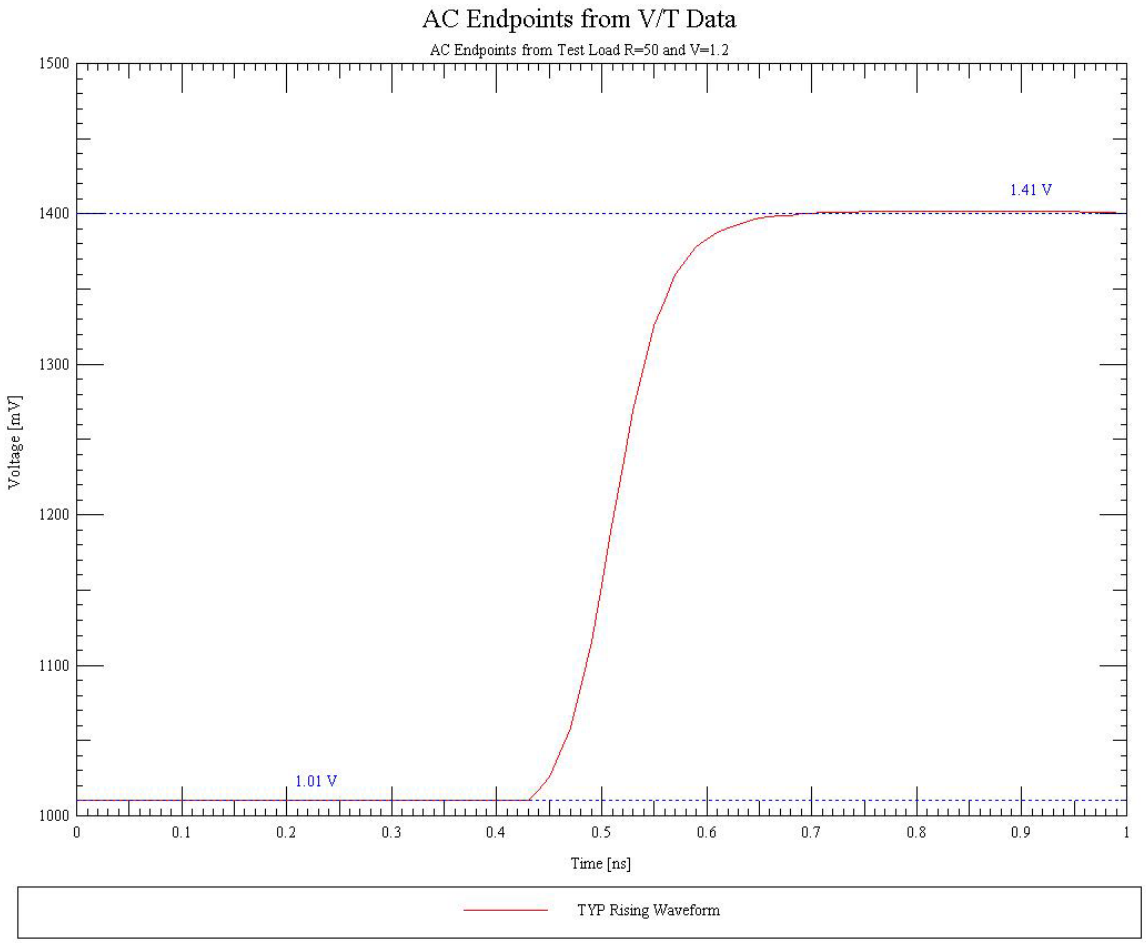

Figure 10. AC Endpoints from Rising Waveform V/T Data

Non-Monotonic Points in I/V Data

The “golden parser” checks that the data is monotonic. If a non-monotonic data point is found, the following warning message will appear:

WARNING (line 524) - Pulldown Typical data is non-monotonic

IBIS simulators like to see smooth monotonic curves. Often times the non-monotonic points are not in the “active region” of the device and can be smoothed out without affecting the accuracy of the IBIS file. For example, for a typical LVDS device with an active operating region of 1.0 V to 1.4 V, if there are non-monotonic points at –4 V, then it could be smoothed out. When the non-monotonic points appear in the “active region” of the device, then there can be accuracy issues. It should be noted that some devices do exhibit non-monotonic behavior.

Extreme Currents Present in I/V Data

The “golden parser” flags a warning if extreme currents are found in the I/V data. If extreme current is found, the following warning message will appear:

WARNING - Model 'lvds_IN': Extreme currents present in MAX GND_Clamp VI Curve (-54.956A @ -1.5V)

This is most common in the power and ground clamp curves. IBIS specifies the I/V data to be generated from –Vcc to 2 * Vcc. This large input sweep can put most devices well out of its operating region. For example, a 3.3 V TTL device might have a recommended operating voltage range of -.3 V to 3.6 V. If –Vcc were applied to the device input, it would certainly malfunction. But in SPICE simulations used to generate IBIS data, a current of –200 A might be generated. How realistic is this value? There are different approaches to solve the "beyond the rail" issue. One approach is to ensure the data is valid at least 1 V beyond where the diode starts to turn on and then extrapolate the rest of the data. Another approach is (assuming that the clamp is realistic), is to use the actual data simulated, because if the device actually went that far below ground it could produce those kinds of current. The range of data to be concerned with is 1 V or so beyond where the diode turns on. If there is extreme current in this region (this is where the "parser" will flag warnings as well), then there are probably issues with the clamp data. For data further beyond this point, it really doesn't matter what value it goes to, because the simulator will almost never go to this point. If the Clamp Current is given in the datasheet, this can be used to verify the clamp data. For the ground clamp curve below, the LVDS device has a clamp current of –18mA at -0.53 V. (Obtained from datasheet) From the graph, it can be seen that at –18 mA, the voltage is –0.68 V, which is reasonable. For the clamp curve beyond this point, it should be somewhat linear, and the current shouldn't "drop off" too fast (resulting in extreme currents.)

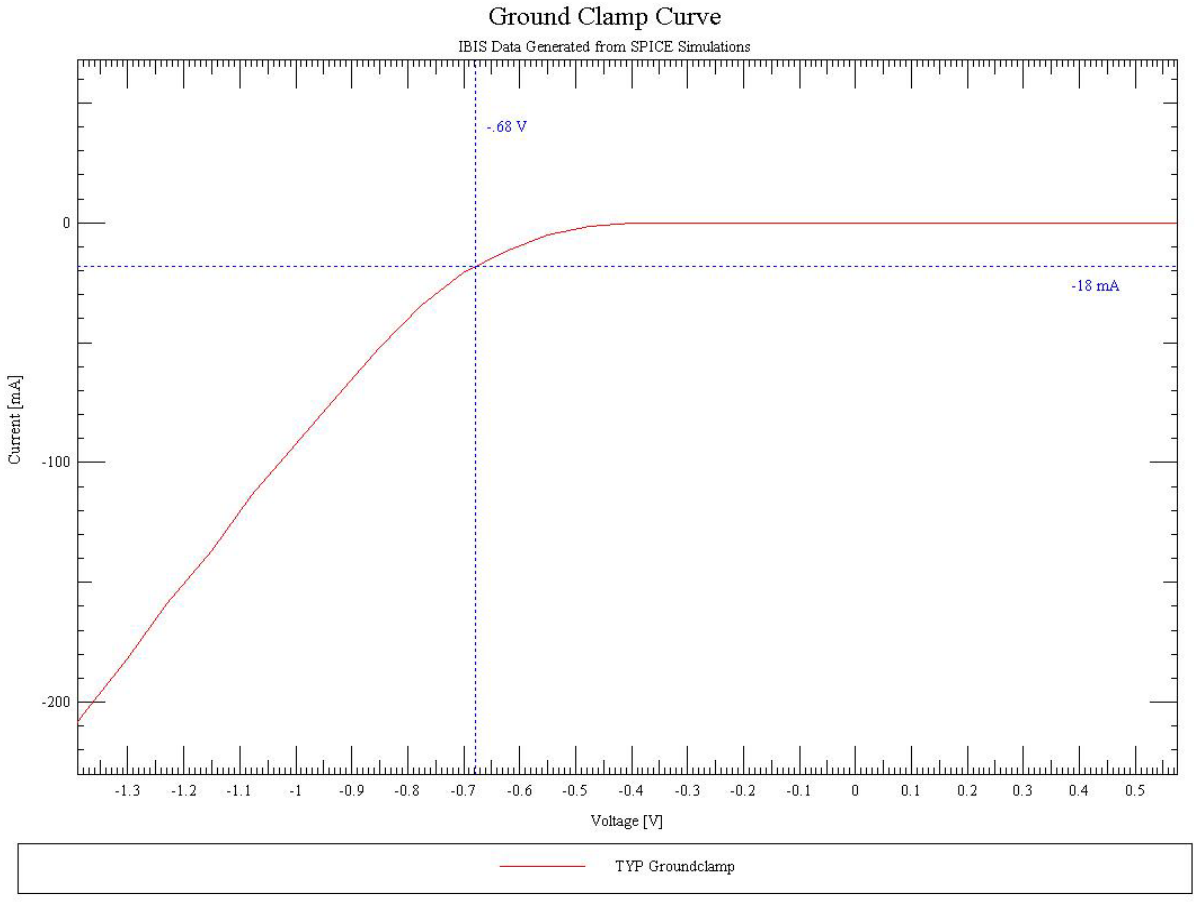

Figure 11. Validating Ground Clamp Current

Ready to Run

Once an IBIS file has been visually checked, run through the “parser”, and determined to be “good”, it’s time to load it up into an IBIS simulator. While an IBIS file is an industry standard, how different simulators use IBIS files is not. Sometimes the difference between two simulators is negligible, and sometimes the difference is extreme. Often times, it is the setup of the simulator that affects the accuracy of the simulation. The four most common IBIS simulators are Avanti HSPICE (now Synopsys), Innoveda XTK (now Mentor Graphics), Cadence SpecctraQuest, and Mentor Graphics ICX. Only the first three tools will be examined in depth in the following sections.

Using an IBIS File in HSPICE

HSPICE is an industry standard SPICE engine simulator. Some vendors, if they provide SPICE models of their devices, use HSPICE as their format. When IBIS data is generated from SPICE simulations, it may be in HSPICE or an internal special format. So it seems a logical choice to use HSPICE to simulate IBIS files. This has the advantage of correlating the IBIS file directly to the HSPICE model under the same conditions.

This being said, HSPICE has some deficiencies that might not make it the best IBIS simulator to use. HSPICE is not like the other IBIS simulation tools. The other simulation tools are part of a board level simulation package with good user interfaces. HSPICE, on the other hand, uses text based input files at a command prompt. To do complex cross-talk board level simulations, the whole board file has to be set up manually. HSPICE does have an internal 2D field solver, like the other tools. HSPICE does not automatically include the effects of the package, like the other IBIS simulators do. HSPICE has an issue with the “time window” used for the V/T data in an IBIS file. If the “time window” of the V/T data is longer than the pulse width of the input signal to the IBIS file, then the file will not run properly. For example, if a device has an fmax frequency of 1.25 GHz, this is a period of 800 psec, and a pulse width of less than 400psec. If the “time window” of the V/T data is 800psec, the file will not run properly. (The other IBIS simulators do not have this issue) This can be fixed by editing the V/T data in the IBIS file to have a smaller “time window”. This is accomplished by removing repeating leading and trailing data points.

An instance call to a LVDS IBIS file (in this example, a 3-State Output) in HSPICE looks like the following:

```
 Boutp pullup pulldown outp in enable powerclamp groundclamp
+ file = ‘lvds.ibs’
+ model = ‘LVDS_OUTP’
```

The instance call always has to begin with a ‘B’. By default, the power keyword is equal to on, which connects the pullup/pulldown/powerclamp/groundclamp nodes to voltage sources with values taken from the IBIS file. The enable node has to be set appropriately to enable the buffer. The in node has to have an input voltage source on it to stimulate the buffer. The output of the buffer can be measured on the out node. The exact name of the IBIS file has to be used for the file keyword, and the model name has to be from the IBIS file. An IBIS file treats a differential signal like it was a single-ended signal. In the above example, only the non-inverting LVDS output buffer would be simulated. A separate instance call is necessary for the non-inverting LVDS output buffer.

HSPICE runs the IBIS file through the “golden parser” before simulating. Any errors or warnings are recorded in the HSPICE output file. HSPICE also checks to make sure that for the model type to be simulated, there are the right number of nodes in the IBIS buffer instance call.

Using an IBIS File in Scratchpad (XTK)

Innoveda (now Mentor Graphics) provides a high-speed tool called Eplanner. With Eplanner, the pre-layout tool Scratchpad can be used to run simulations. The main simulator engine, XTK, can be used to simulate post-layout by simulating full board designs from such PCB tools as Power PCB. Scratchpad offers a visual interface to the XTK engine and is very easy to use. Components are dragged to the canvas, hooked up, and then simulated.

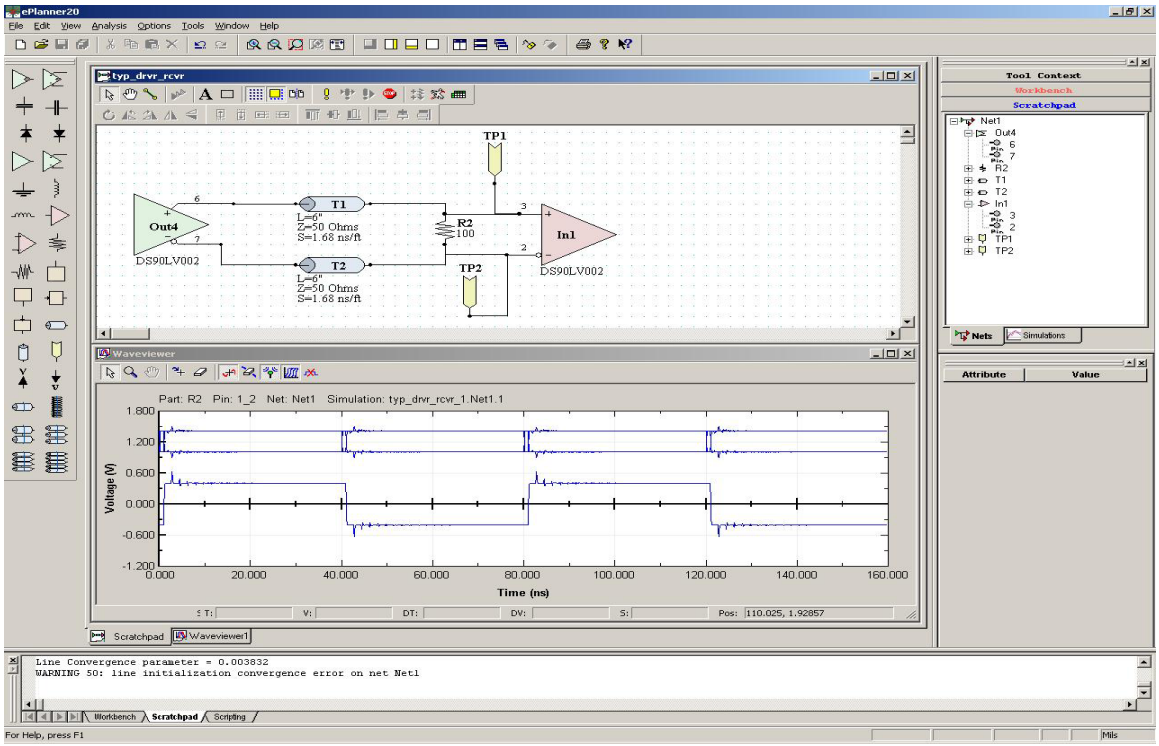

Figure 12. Scratchpad Interface

When using differential IBIS files in Scratchpad, usually only the driver outputs and the differential input signal to the receiver are plotted. If you want to see the single-ended signals going at the receiver input, you can use 'test points' to graph them. In order to use an IBIS file, it must be translated to the XTK format. When an IBIS file is associated with a driver or receiver, Scratchpad will automatically convert the IBIS file before running a simulation. Once again, the IBIS file is run through the "parser" and any warnings or errors are stored in the log file. There is an issue having Scratchpad automatically translate LVDS IBIS files. A program called ibis2xtk is used by Scratchpad (and can be used manually as well from the command prompt) to translate IBIS files. By default, ibis2xtk checks the R_fixture and V_fixture loads to ensure that they are valid for the model type. However, for a 3-State Output model, a restriction is put on the V_fixture value. It must be within .7 V of the rail voltage. This was intended for TTL like devices, where the rising and falling waveform data had a load of 50 Ohms to Vcc and 50 Ohms to Ground. LVDS typically uses a load of 50 Ohms to 1.2 V. When ibis2xtk encounters this, it automatically defaults to the Ramp data and will use that instead of the V/T data. To overcome this, the IBIS file can manually be translated at the command prompt using the following option:

ibis2xtk –no_rfchk ibisfile.ibs

The option –no_rfchk does not check for R_fixture and V_fixture values, and will translate the rising and falling waveform V/T data. The program ibis2xtk also automatically calculates the time it takes to reach Vmeas under the test load condition, as long as the timing parameters Rref,

Cref, Vref, and Vmeas are defined. The simulator then subtracts this test load delay from the system load delay in order to report correct flight time delays. Another option that can be used, dump –iv, will extract all of the IBIS data into a directory so it can be plotted in the waveform viewer.

Using an IBIS File in Signal Explorer

Cadence provides it's own high-speed simulation tool called SpecctraQuest. SpecctraQuest works very much like XTK, having a pre-layout simulation tool called Signal Explorer. Signal Explorer is a user interface to Signoise, the simulator engine. Signal Explorer, like Scrathpad, has an easy to use interface and components are dragged to a canvas and simulated.

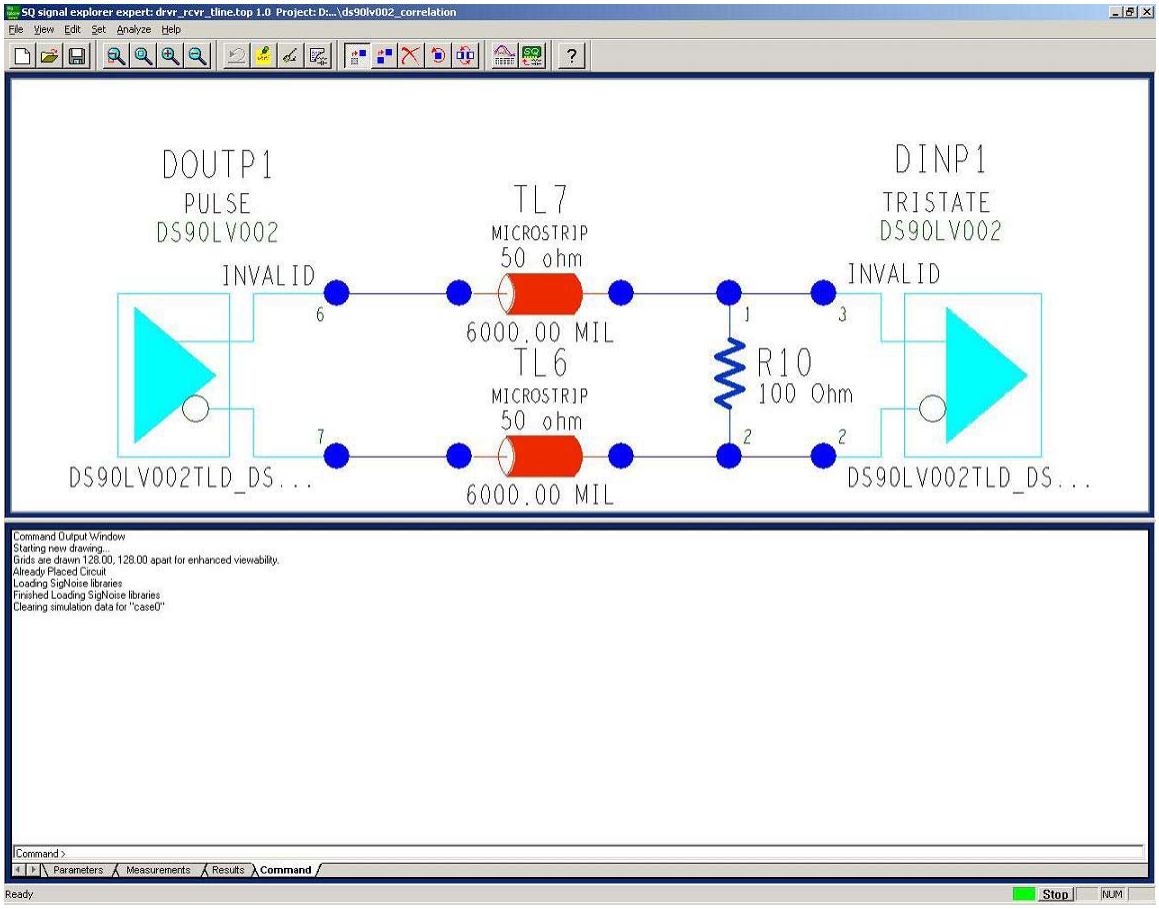

Figure 13. Signal Explorer Interface

When using an IBIS file in Signal Explorer, the IBIS file must be translated to the Cadence format, called dml (device model language). This can be done automatically through the Library Browser, or from the command prompt. If done manually from the command prompt, the curvedir option can be used and the IBIS data will be plotted in the waveform viewer Sigwave. Sigwave also has the advantage of being able to import in data from almost any other tool, including HSPICE and XTK. The buffer delay has to be calculated in Signal Explorer as well. This is not done automatically during the translation process. Under the Library Browser, the IBIS file can be selected, and the buffer delays can be measured. (Assuming the timing load keywords are in the IBIS output buffer model.)

SPICE To IBIS Comparison

The following graphs show the comparison of the original SPICE model file to the IBIS file simulated in different IBIS simulators. The circuit simulated was a point-to-point topology, with the driver driving 6 inches of 50 Ohm lossless transmission line to the receiver, terminated by 100 Ohms.

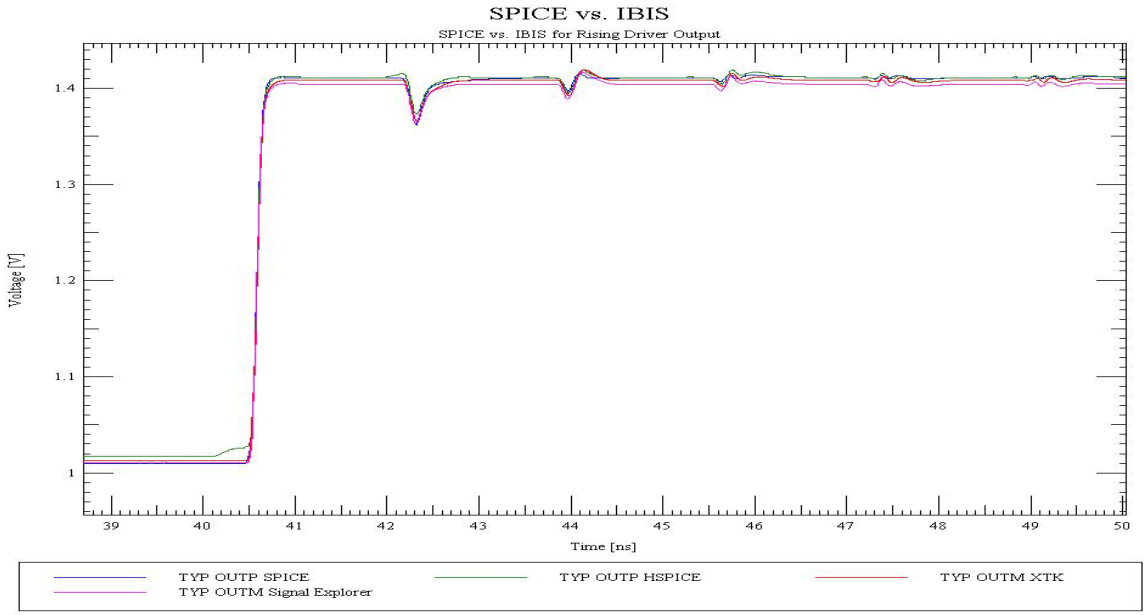

Figure 14. SPICE vs. IBIS for Single-Ended LVDS Driver Output

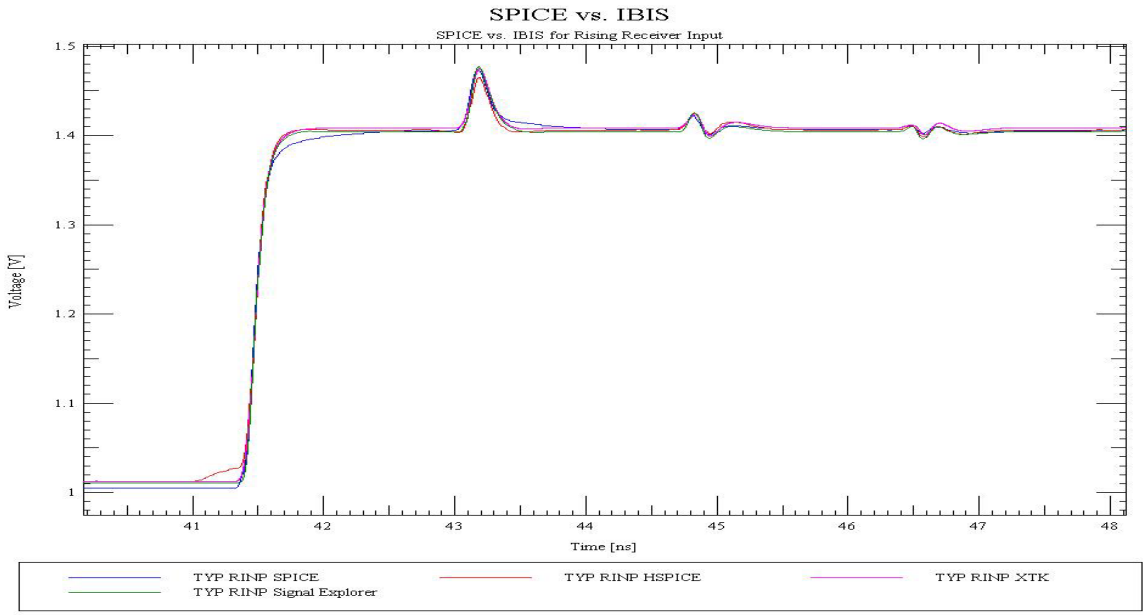

Figure 15. SPICE vs. IBIS for Single-Ended LVDS Receiver Input

As can be seen from the previous graphs, all of the IBIS simulators match up to the original SPICE model file. The IBIS file in HSPICE has a “step” on the rising signal that is not seen in the original SPICE simulation. The Scratchpad and Signal Explorer simulators are in very close correlation with the SPICE model file.

Any of these IBIS simulators are adequate for Signal Integrity simulations. Each tool has it's advantages and disadvantages, and which tool to use will come down to availability and personal preference.

A Note on Simulator Accuracy

In each IBIS simulator, the time step resolution can play an important role in the accuracy of the simulated results. If too large a time step is used, then not enough points will be simulated, and the simulator will "linearize" the results. This results in sharp edges in the transition periods rather than smooth curves. This will greatly affect the measured rise and fall times. A 10ps time step was used for all of the above simulators (including the SPICE model), except for the Scratchpad simulator, which required a 1ps time step to achieve the desired resolution.

Summary

IBIS is an important modeling standard that is used by almost every system level designer and signal integrity engineer. IBIS files can help to reduce cycle time by finding issues before a prototype is built. IBIS files have to go through a "validation" process, and by looking at the data model inaccuracies can be found. Once an IBIS file has been validated, there are a number of different simulators to use. Choosing which simulator to use is a matter of preference, and almost any simulator can be used to accomplish the task at hand. The key to getting up and running using an IBIS file is to understand the specification, understand the file, and understand the tool that is going to be used.

References

An Introduction to IBIS (I/O Buffer Information Specification) Modeling, Application Note AN-1111. National Semiconductor.

IBIS (I/O Buffer Information Specification) version 3.2, Ratified August 1999. IBIS Committee.

Introduction to IBIS Models, April 26, 2000. Arpad Muranyi, Intel Corp.

I/O Buffer Modeling Cookbook, IBIS Open Forum.

The IBIS Open Forum Website, http://www.eigroup.org/IBIS/Default.htm.

Effective Signal Integrity Analysis using IBIS Models, DesignCon 2000. Syed Huq, Cisco.

Load Line Analysis in SPECCTRAQUEST. Bradely Herrman and Pravin Patel, Design Tools.

Assessing and Improving the Quality of IBIS Models, 2001. Fisher, Katz, Moles, Figueroa, Sisoft and Apple Computer.

A Critique of IBIS Models Available for Download on the Web (Part 1), January 2002. Jim Bell and Dan Grogan, SiQual.

Star-HSPICE Manual, Version 2001.2. Avanti.

XTK User's Guide, Version 2000.4. Innoveda.

SpecctraQuest Simulation and Analysis Reference, Version 14.2. Cadence.

Innoveda Visual IBIS Editor, Version 3.0. Innoveda.