How to Use the IBIS Model

Roy Leventhal

Last Update: 3/9/14

Table of Contents

Introduction

Scope

In preparing this document I have referred primarily to IBIS 2.1 functionality. The added functionality of IBIS 3.2 will be addressed at some later date. What I have intended was to capture as much practical example material of using IBIS models for Signal Integrity (SI) simulation as the reader may find useful.

History and Motivation of IBIS Modeling

IBIS is an acronym for "I/O (Input-Output) Buffer Information Specification." It is a template (standard, data-exchange format, etc.) for exchanging modeling information between semiconductor device suppliers, simulation software suppliers and end users of this information.

IBIS models are not models in the more traditional meanings of that word where a modeling language and/or schematic symbol, and/or nth-order polynomial representation of a device and/or its internal structure is being represented. Such models are referred to as *physical* models because the physical elements are being described. But, the IBIS model is a model in the sense that the *behavior* of a device is being represented. However, the distinction between physical and behavioral is often philosophical and usually conveys something about the level of abstraction of the model. Models imitate the real thing in appearance and/or behavior and allow you to predict (simulate) the performance of the device they model in the application you intend to use them in.

The most popular of such previous models is SPICE, which is a computer implementation of device physics based models (hybrid-π, Gummel-Poon, etc.). Other models include the 3-node T-parameter and 2-Port types (h-parameter, y-parameter, etc.) up to and including Scattering (S) – Parameters. I mention these other models, because in IBIS the input-output behavior is not generated from model inputs and properties as is done with them. The radical departure for IBIS is to catalog the output and input behaviors of devices in the form of curves, or actually, tables of data that catalog how an output or input behave electrically.

It is assumed that some **event** triggers an output to switch high or low launching a wave down a transmission line to a set of inputs. The essential signal integrity question is how the transmission line structure (topology, terminations, drivers, receivers, coupling to other lines, etc.) responds to that stimulus.

Added to the I/O cell Voltage-Current (V-I) and Voltage-Time (V-T) behavior are several other elements that allow more complete signal integrity modeling of a complete IC package. This is information about model I/O cell and power and ground assignments for each pin and the parasitics for each pin, and possibly internal coupling between pins. Information about pins that act as differential pairs is also included.

The IBIS model is computer parsable through its keywords in keeping with its computer simulatable mission. The model also has very specific syntax rules based on generating a file in ASCII format.

The IBIS Open Forum was formed in May, 1993 with about 10 members. In early 1994 it became affiliated with the Electronic Industries Alliance (formerly the Electronic Industries Association). It has grown to about 30 members. The IBIS model was originated to enable users to obtain Signal Integrity simulatable data from suppliers without obtaining data that would allow them to reverse engineer the supplier's device and thus copy it.

The IBIS Open Forum meets every 3 weeks via teleconferencing and e-mail and occasionally in person to ratify changes and set future directions. Most activities are handled via the email reflector at:

Ibis@eda.org

To add your name, send your email address to:

Ibis-request@eda.org

To suggest changes and improvements, submit a "BIRD (Buffer Issue Resolution Document)" to this email address.

Current State of IBIS Modeling

IBIS is considered an emerging standard. Version 2.1 was ratified in 1995 as ANSI/EIA-656. Version 3.2 was ratified in January, 1999 as ANSI/EIA-656-A.

IBIS models were supposed to be readily available and to free up the simulation logjam created by a lack of models and an adversarial relation between supplier and user. Such has not been the case. In the opinion of this writer, the reasons are:

Cost issues:

First, and foremost, getting and publishing data costs money. Inventing a new way to do that doesn't change that. In some ways the IBIS model is less costly to generate and use than the traditional SPICE model. But, the infrastructure to support IBIS doesn't yet exist and will require an investment to create. Plus, there is still the "per-unit cost" to generate the data. Users have a long tradition of being unwilling to pay for modeling data from suppliers. Users have also not established a cost-benefit tradition of simulating product behavior vs. build-and-debug.

Second, semiconductor suppliers are not going to replace SPICE modeling (which they can relate to process modeling, device engineering and yield improvement) with IBIS modeling (which they cannot relate to their processes). IBIS simply represents an added cost with no benefit yet firmly established for them. I don't think many semiconductor sales have yet been lost for lack of an IBIS model.

Adoption issues:

SPICE is the modeling of choice at the current time. It has a long tradition of use, understanding and consequently trust of its advantages and limitations. IBIS is brand new. People forget that it took SPICE twenty years (from 1956 to the mid-70s) to catch fire.

Then also, the IBIS model suffers the same fate of all documents written by committees over an extended period.

Legitimacy issues:

Wide adoption of IBIS will aid its achieving legitimacy. Second, the realization that it is not a standard in being, nor very firm in its expectations (it's full of loopholes and caveats) will aid in understanding of what it can accomplish. Further, individual IBIS files are full of the usual manufacturer's disclaimers. For a clear agreement and understanding of what is expected, write a purchase specification control drawing and negotiate its acceptance with your supplier.

IBIS only provides a format for the exchange of data to be used in modeling signal integrity.

Adversarial Issues:

IBIS addresses one set of adversarial issue between buyer and seller. Namely, the copying of a product design which it prevents fairly effectively.

It does not address other, equally important, adversarial issues. These have to do with the setting up and fulfillment of reasonable and legitimate expectations. I'll leave aside discussions of product guarantees, disclaimers, specifications, incoming inspection, etc., and instead address what was intended to be accomplished by the exchange of data allowing for the simulation of signal integrity.

The most that the supplier of the data can hope for is that the users will apply their skill and not force unreasonable specifications on them and not misapply the device being supplied and blame the supplier.

The most that the user of the data can hope for is that the data supplied is accurate, complete, and useful and fairly represents what is being supplied.

There is a long and dominant tradition of suppliers specifying much wider tolerances than they can deliver and users countering with setting much narrower tolerances than they need and not exercising their skill in applying the product. The question continually occurring is: Why beat your brains out being clever in your design when there is a long history that beating up the supplier is more visible, appreciated and cost effective?

Whither IBIS?

It is only when suppliers and users have learned to exchange data for their mutual success that the adversarial relationship will change. Simulation of IBIS models is an enabling technology for doing that.

Knowledge of how to generate and use IBIS models is growing slowly. So long as the process and the IBIS specification remain obscure, unreliable and difficult, the demand for software that can run that process will lag. That is an issue that only the software suppliers can really address.

Symbolic Representation of an IBIS Device – Input Side

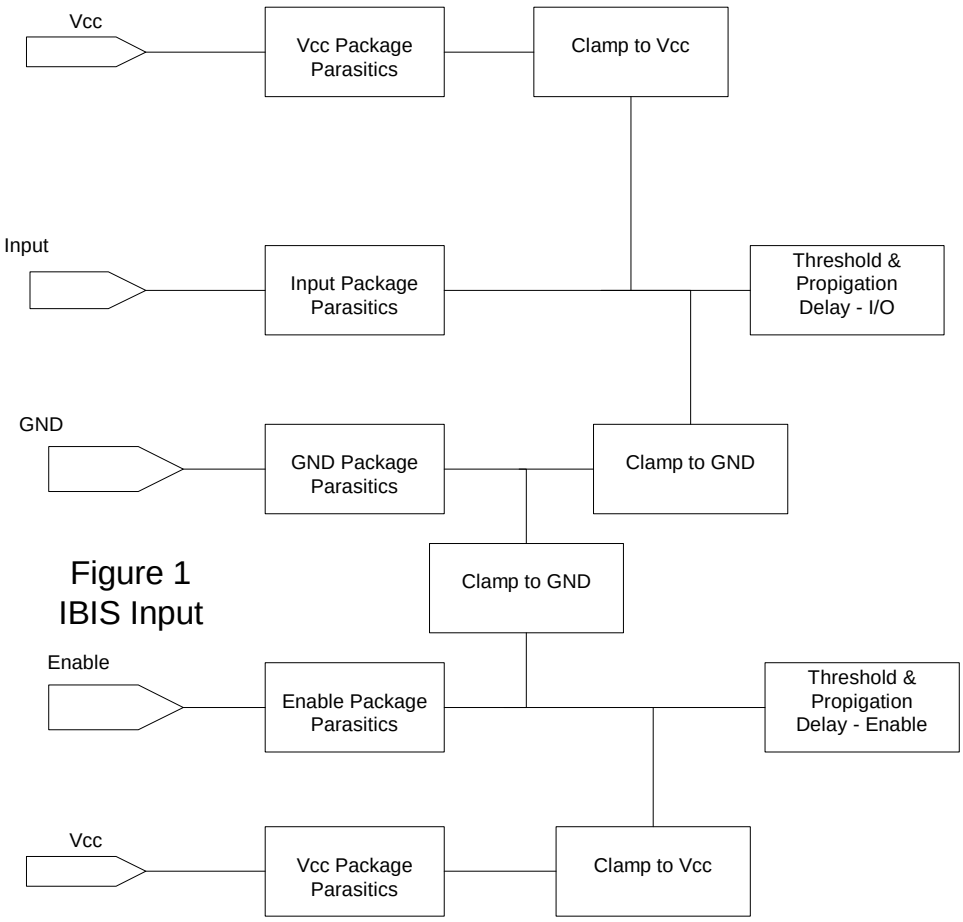


Figure 1
IBIS Input

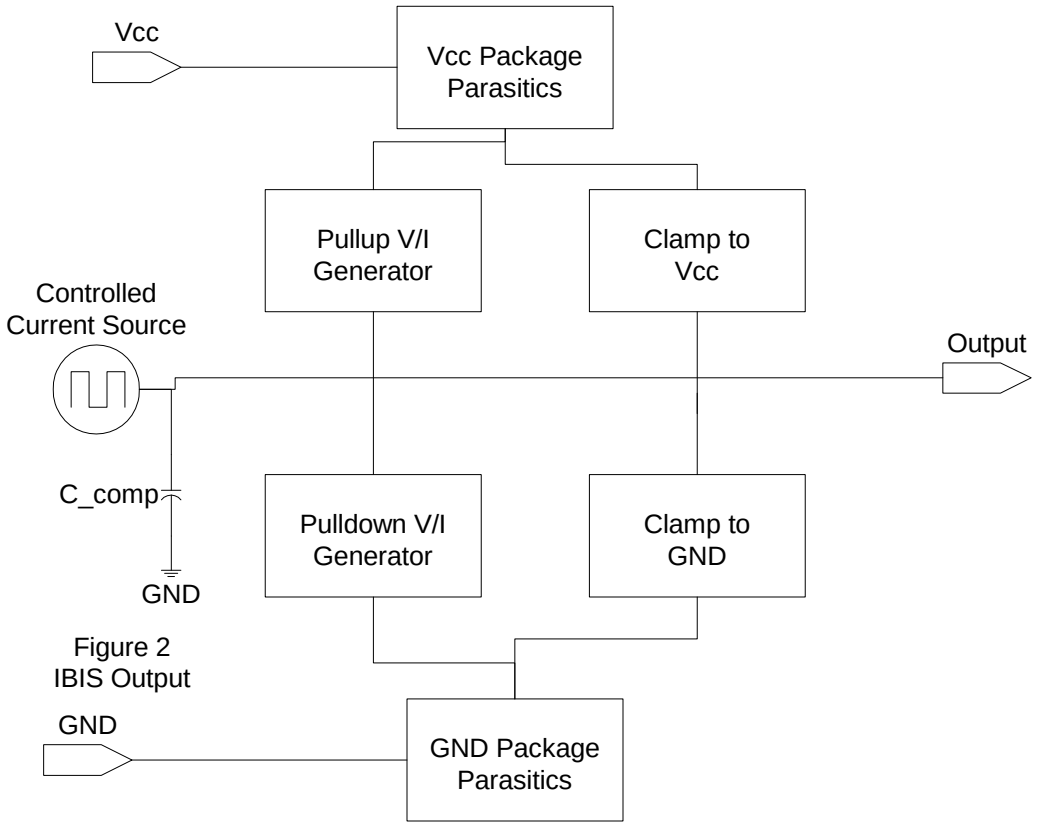


Figure 2
IBIS Output

Symbolic Representation of an IBIS Model – Output Side

Model Types

Typical data must be supplied in the following. Min and Max data is optional. Model_type is a required parameter.

Input

An Input model functions only as a receiver. Vinl and Vinh (input threshold levels) must be defined. Power and/or Ground Clamp V-I curves must be define if they are supplied in the device.

Output

An Output model functions only as a Driver. Voh and Vol (output high and low limits) are not part of the IBIS spec. Most simulator companies put such information to good use, however. Power and/or Ground Clamp V-I curves must be defined if they are supplied in the device. Pullup and Pulldown V-I curves, as present in the device, must be supplied. This model always sources and/or sinks current and cannot be disabled.

Buffer switching speed information in the form of output rise and fall slew rates or V-T rise and fall curves must be supplied. If V-T curves are supplied they supercede the slew rates.

I/O

This is a type of model where the pin is connected to device cells that can function either as a driver or a receiver depending on the enabling logic.

3-state

This is a type of driver model. It indicates that an output can be disabled. That is, put into a high impedance state.

Open-drain

This is a type of driver model with an open pullup side. This name is retained for backward compatibility.

I/O_open_drain

This device indicates a combination of I/O and Open_drain behavior.

Open_sink

This is a type of driver model with an open pullup side. The user supplies a pullup resistor and power rail connection.

I/O_open_sink

This device indicates a combination of I/O and Open_sink behavior.

Open_source

This is a type of driver model with an open pulldown side. The user supplies a pulldown resistor and/or ground/pulldown rail.

I/O_open_source

This device indicates a combination of I/O and Open_source behavior.

Note: ECL is an acronym for “Emitter Coupled Logic.” **PECL** is an acronym for “Positive Emitter Coupled Logic.” These types of technology follow some different conventions for pulldown than the previous types.

Input_ECL

Output_ECL

I/O_ECL

3-state_ECL

Terminator

This model is an input-only device that can have an analog loading effect on the circuit being simulated but has no digital logic threshold. Examples are resistors, diodes, capacitors, etc.

V-I Data

Range of V-I Data

The V-I curve data is presented as a table of current measurements taken at a series of output voltages that covers a range of –Vcc to +2Vcc. The current sourced or sunk by the device is measured with currents into the device being positive by convention.

Measurement Conditions

The data is measured placing a stepped/swept voltage source on the output and measuring (simulating) the current into or out of the pin. The device is allowed to settle to semi-quiescent conditions. Allowing a buffer to reach quiescent conditions is a lot easier to measure, but this can be misleading in modeling dynamic switching behavior. Adding pin parasitic information better models the high speed/high current switching behavior.

For further details refer to the IBIS spec, especially the “data derivation” section.

Simulation From SPICE

IBIS data is most likely to be simulated from a SPICE deck rather than being measured. The same technique of hooking up a (virtual) constant voltage source, sweeping it and measuring output

current is used. Often the SPICE model assumes zero diode resistance on clamps. It is not uncommon to see clamp currents of 10^{20} amps in the data tables as a consequence. But, this is not real data and needs to be corrected when found.

The Four V-I Curves: Pullup, Power Clamp, Pulldown and Ground Clamp

The current at an output is the total of the pullup / pulldown and the clamp curves, when present. But, these curves are stored in 4 separate data tables (not counting min-max conditions) in the model. IBIS includes an explanation of measurement techniques for separating out these individual elements. When simulating from a SPICE deck the same effect is achieved by disabling / disconnecting the various elements in turn in the simulation.

The IBIS Voltage Convention

The voltage in the table (ECL and PECL parts can be exceptions) for pullup and power clamp curves is:

$$V_{table} = V_{cc} - V_{output}$$

And, is therefore, the voltage across the pullup circuit referenced to Vcc and not to ground, by convention. At the same time, the pulldown curves are normally referenced to ground. The range –Vcc to +2Vcc is chosen because those are the limits of the voltage swings that can be seen on a transmission line with high reflection (high mismatch) coefficients when switching a driver from Vcc high state to 0 V low state.

Receiver Input Impedance and Matching

In the following discussions of impedance, reflection coefficient, etc., we slip effortlessly back and forth between V-I curves and impedance and waveforms and reflections. Remember that impedance and transmission lines speak of circuitry while waveforms speak of signal generation and content. Signal generation and content is not generally discussed until we get to ramp rates and V-T curves. Impedance, parasitics and matching address the response of a circuit to a digital signal.

Almost all receivers have a very high input impedance. The input impedance changes when clamping diodes are present and the input voltage overshoot turns them on. Once the clamping diodes turn on they ideally present a short circuit to the clamping rail, be it Vcc or ground.

Within the active range the input impedance, Zin, consists primarily of the loading capacitance that the device presents to the circuit as a first approximation. This capacitance usually consists C_comp from input pin to ground. Pin and package parasitics can add a couple of pF to C_comp.

Matching the receiver input impedance to the transmission line impedance, Zo, will often consist of putting a shunt termination equal to the transmission line impedance in parallel with the receiver.

More exactly:

$$Zo = Zload = (Ztermination)(Zin)/(Ztermination + Zin)$$

Where Zload represents the total adjusted input impedance.

The degree to which the input impedance (with or without matching termination) does not satisfy this equation is the degree to which energy from the incident switching wave will be reflected back towards the source. More specifically, we can analyze the effect mathematically by looking at *reflection coefficient*, ρ:

$$\rho = (Zload – Zo)/(Zload + Zo)$$

ρ can vary from +1 to –1, or +100% to –100%. The + sign means that the reflected wave will be of the same polarity as the incident wave and the – sign means that the reflected wave will be of opposite polarity than the incident wave.

The next level of sophistication in looking at input impedance is to add in the effects of pin parasitics per the following circuit:

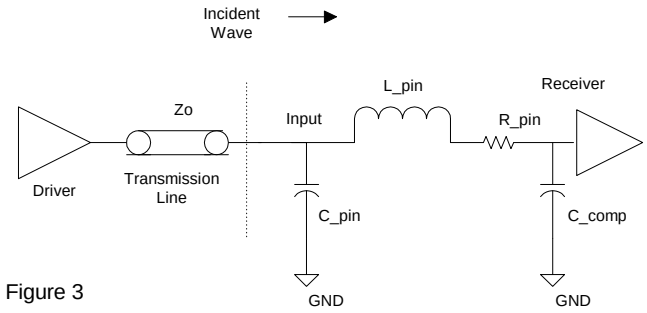


Figure 3

Now, when figuring out the input impedance to the receiver you are considering the effects of C_pin, R_pin, L_pin and C_comp in the topology shown. This will become your new Zin in the above equations.

An even more detailed look at input impedance would replace the single line pin parasitic model consisting of C_pin, R_pin and L_pin above with an N-by-N RLGC matrix set that would include the effects of mutual coupling and conduction between pins and between pin and package.

Driver Output Impedance and Matching

The output impedance of a driver is significant in two ways. First, it is of interest in terms of how it matches the Zo of the transmission line it is connected to. Second, it is of interest when understanding what the magnitude of the wave it launches down the transmission line will be.

IBIS does not include any parameters for driver (or receiver) impedance directly. But, such information is implicit in a device's V/I curves. Think of the Zout of the driver chip itself as the slope of those curves:

$$Zout = \Delta V/\Delta I$$

The V/I curves can be quite non-linear. But, in the active, unclamped midrange they can usually be approximated by a straight line calculation of:

$$Zout = |(V2 - V1)/(I2 - I1)|$$

With the absolute value signs, | |, indicating that Zout will be real, non-regenerative, for most any situation encountered in signal integrity analysis.

Zout is usually quite low and less than the Zo of the transmission line. Thus, it's quite common to see a series terminating resistor matching a driver to a transmission line where the object is to have:

$$Zo = Ztermination + Zout$$

When a driver sees the reflected energy wave returning from a receiver (or impedance discontinuity) it can be in an on or off state. If on, its Zout is given by the above. If off, it goes to a high impedance state. You would be more likely to parallel terminate if the round trip reflection propagation time delay is large enough (greater than the pulse width) for the driver to have turned off.

The effects on matching Zo of C_comp and pin parasitics (C_pin, R_pin and L_pin) are added in the same manner as discussed under Receivers above. Refer to figure 4 below for how this topology is connected.

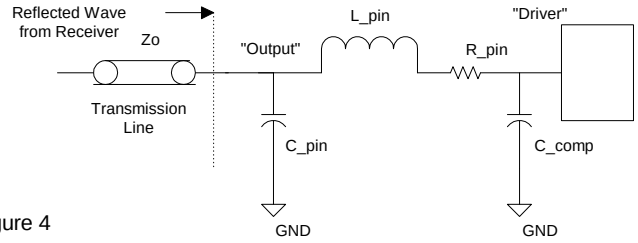


Figure 4

As mentioned above, Zout is also of interest when analyzing its effects related to a low impedance and/or heavily loaded line. So far in our discussion, we have been assuming that our driver is a constant voltage generator. A constant voltage generator assumes a zero or insignificant internal generator impedance, Zg.

The condition that Zg be insignificant (especially in relation to Zo, loaded Zo and parallel terminated (at the driver) Zo) is not always met. Since Zg = Zout we can look to see if it is insignificant. The fraction of the driver output voltage swing launched down the line will be given by:

$$Vout' = (Vout)(Zo')/(Zo' + Zout')$$

Where Zout' = driver impedance plus internal parasitics, and Zo' = net transmission line impedance presented to the driver including the effects of lumped distributed loads and parallel termination at the driver. Likewise, the voltage sent out of a series terminator at a driver will be reduced because the terminator will act like an added internal generator resistor.

Waveforms at the Die vs. Waveforms at the Pins

So far, we have been discussing waveforms, reflections and matching as things appear on the etch trace connecting a driver package and a receiver package on a PWB. For understanding what waveforms appear at, or on, the die's connection pads consider that:

The die bondwire, package pin, etc. are usually physically too small and too short electrically to act as transmission line structures.

The values of C_pin, R_pin, L_pin and C_comp can be large enough to act as a filter to waves entering or leaving a die as that wave propagates through that package filter structure.

Curves, Behavioral Models and Data Table Lookup Models

The IBIS model is then interpreted as a “behavioral” model in the sense that the V-I (+ slew rate response) curves represent the devices’ behavior once an output switching event occurs. But, no curves are stored in the data. The more precise definition of the model is that it is a “table data lookup” model. And, no behavior or output response curve is predicted, say by a polynomial, given an input variable. Rather, the simulator looks up and uses the output or input current given a voltage across the output.

The current data from pullup, pulldown, power clamp and ground clamp is summed together at the voltage being used as lookup value. If there is a crossover zone in the pullup/pulldown power/ground clamp current data it will be detected and source/sink and leakage currents will be summed.

Examples of V-I Curves

The following data has been copied from the Cadence Design Systems, Inc., IBIS model “CDSDefaultIO.dml” file for illustration:

```
(PullDown
   (ReferenceVoltage
    (minimum 0 )
    (typical 0 )
    (maximum 0 ) )
 (VICurve "-5.0 -215ma   -210ma   -225ma
        -4.0 -212.0ma -207.0ma -217.0ma
        -3.0 -207.0ma -202.0ma -212.0ma
        -2.0 -188ma   -183ma   -193ma
        -1.0  -70ma    -65ma    -75ma
         0.0    0ma      0ma      0ma
         0.5   70ma     65ma     75ma
         1.0  127ma    122ma    132ma
         1.5  164ma    159ma    169ma
         2.0  188ma    183ma    193ma
         2.5  203ma    200ma    208ma
         3.0  207ma    202ma    212ma
         3.5  210ma    205ma    215ma
         3.0  207ma    202ma    212ma
         3.5  210ma    205ma    215ma
         4.0  212ma    207ma    217ma
         4.5  214ma    209ma    219ma
         5.0  215ma    210ma    220ma
        10.0  220ma    215ma    225ma" ) )
  (PullUp
   (ReferenceVoltage
    (minimum 5 )
    (typical 5 )
    (maximum 5 ) )
   (VICurve "10   -142ma   -137ma   -147ma
         5   -137ma   -132ma   -142ma
         4.5 -133ma   -128ma   -138ma
         4   -128ma   -123ma   -133ma
         3.5 -123ma   -118ma   -128ma
         3   -118ma   -113ma   -123ma
         2.5 -110ma   -105ma   -115ma
         2    -98ma    -93ma   -103ma
         1.5  -83ma    -78ma    -88ma
         1    -64ma    -59ma    -69ma
         0.5  -38ma    -33ma    -43ma
         0     0        0        0
        -1     64ma     59ma     69ma
```

```
      -2    98ma    93ma   103ma
      -3   118ma   113ma   123ma
      -4   126ma   121ma   131ma
      -5   137ma   132ma   142ma" ) )
 (GroundClamp
  (ReferenceVoltage
   (minimum 0 )
   (typical 0 )
   (maximum 0 ) )
  (VICurve "0 0 0 0
      -0.1 0 0 0
      -0.4 -0.1ma -0.1ma -0.1ma
      -0.5 -0.5ma -0.5ma -0.5ma
      -0.6 -1.2ma -1.2ma -1.2ma
      -0.7 -2.4ma -2.4ma -2.4ma
      -0.8 -6ma -6ma -6ma
      -0.9 -13ma -13ma -13ma
      -1.0 -25ma -25ma -25ma
      -5.0 -293ma -293ma -293ma" ) )
 (PowerClamp
  (ReferenceVoltage
   (minimum 5 )
   (typical 5 )
   (maximum 5 ) )
  (VICurve "0.0 0 0 0
      -0.1 0 0 0
      -0.4 0.1ma 0.1ma 0.1ma
      -0.5 0.6ma 0.6ma 0.6ma
      -0.6 1.2ma 1.2ma 1.2ma
      -0.7 2.4ma 2.4ma 2.4ma
      -0.8 6ma 6ma 6ma
      -0.9 13ma 13ma 13ma
      -1.0 25ma 25ma 25ma
      -5.0 293
```

Note that the clamp curve currents above typ-min-max are all equal at a given voltage.

In Cadence Design Systems, Inc’s., GUI these curves can be observed as follows:

Figure 5: Default Model Ground Clamp

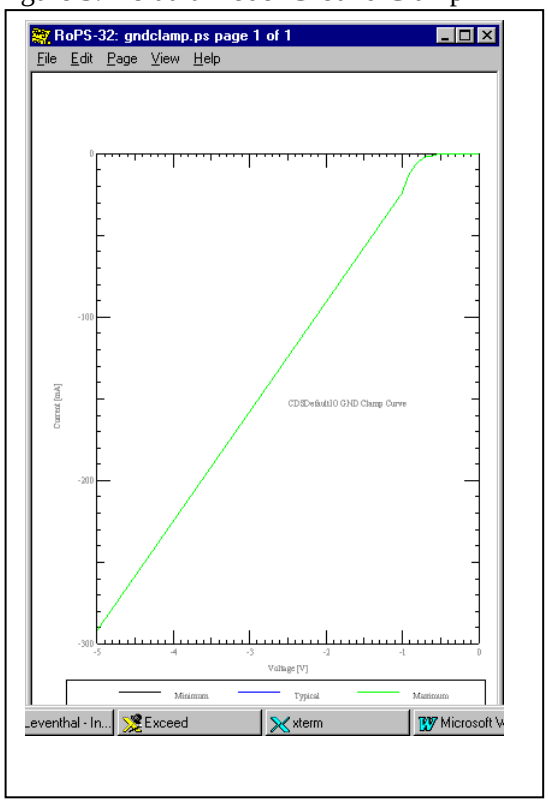


Figure 6: Default Model Power Clamp

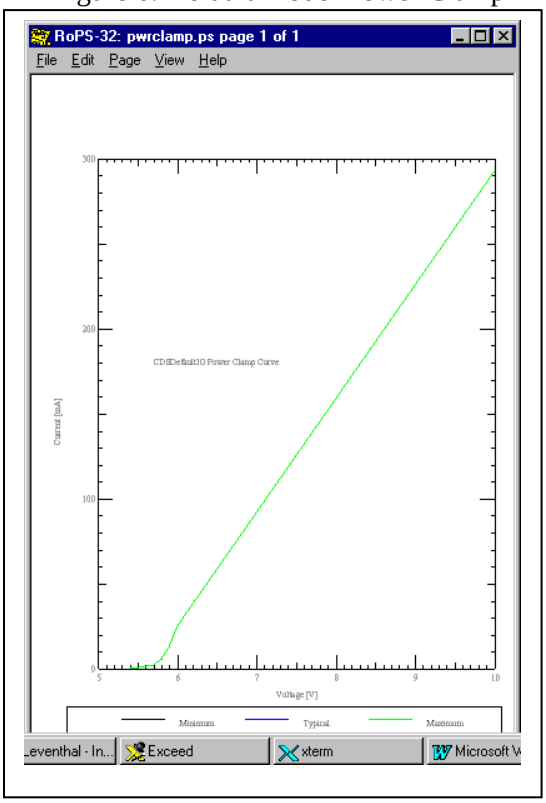


Figure 7: Default Model Pulldown

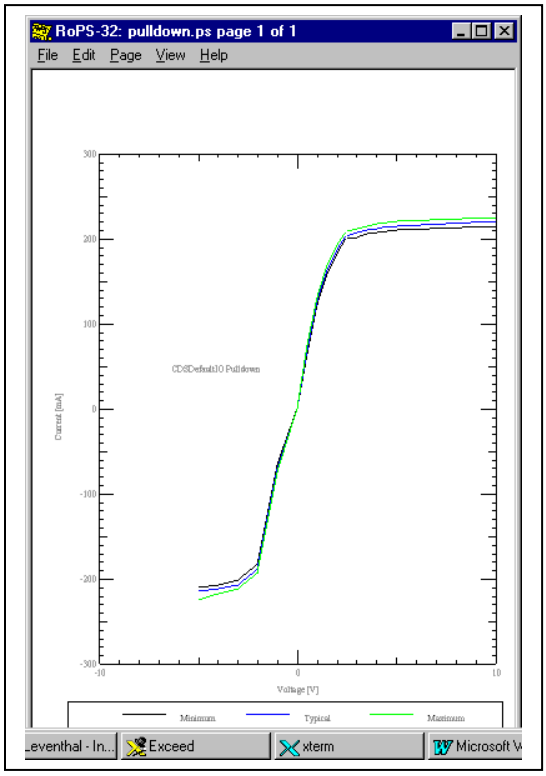
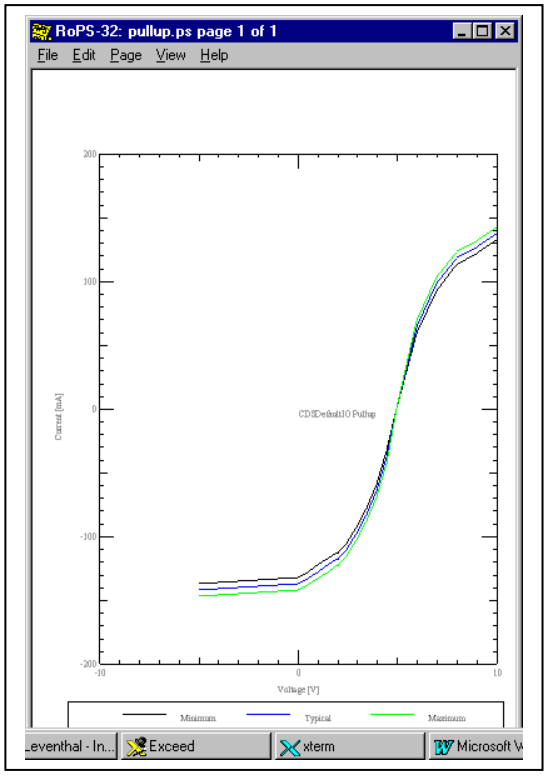


Figure 8: Default Model Pullup



Scaling V-I and Simulating the Results

We will look next at one aspect of the effect of driver characteristics on simulation results. That is, how changes in V-I curves change performance. One way to think about the analysis is to think about the output impedance of the driver remembering that it is non-linear, i.e., it changes with voltage level.

The model above has an output impedance in the range of 20 ohms on the pullup side and 10 ohms on the pulldown side. We will drive lines perfectly terminated in their characteristic impedances (no reflections from the receiver) of Zo1 = 72 ohms and Zo2 = 18 ohms and look at the results. Next, we will multiply the currents at a given voltage times 4 (makes the driver output impedance ¼ as big as originally) and again simulate and look at the results.

Figure 9: Default Model driving 72 Ohms – Vout ≈ 4.13 V. Driver Zout estimated at about 18 ohms.

The first pulse train is the driver at 50 MHz. The 3nS delayed (t_{pd} of transmission line = 3nS) pulse train is the receiver.

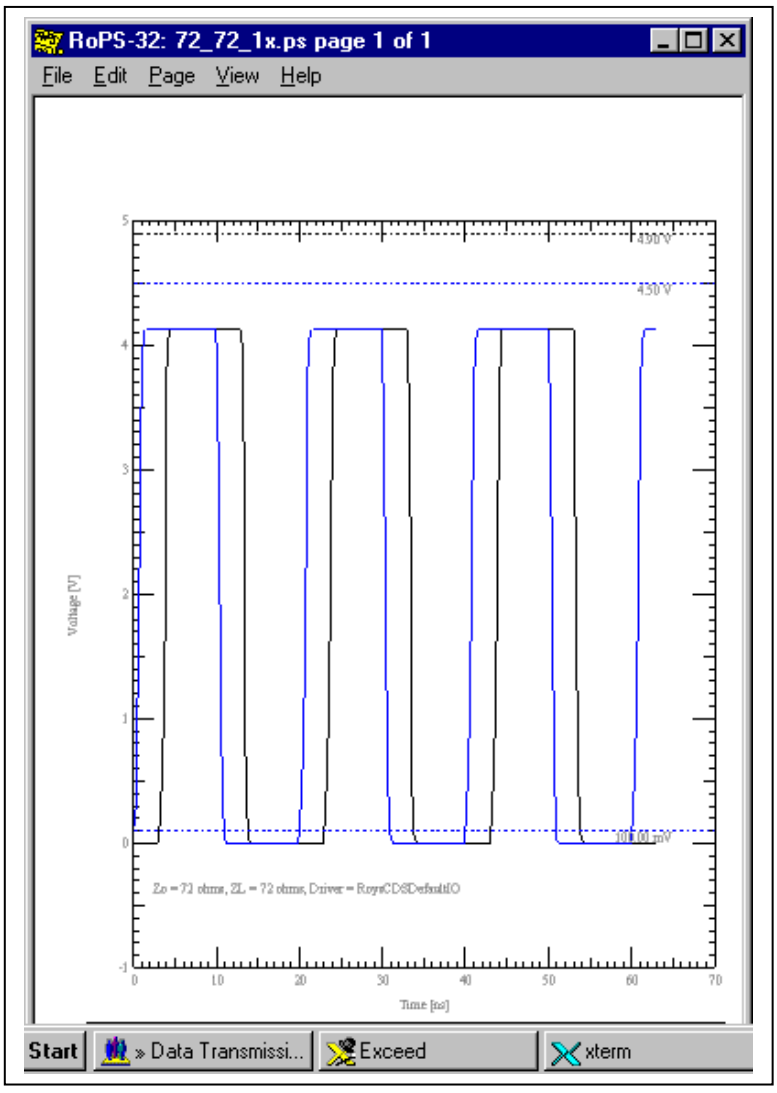


Figure 10: Default Model driving 18 Ohms – Vout ≈ 2.1 V ≈ ½ our original Vout.

This is another indication that our Zout is effectively about 18 ohms.

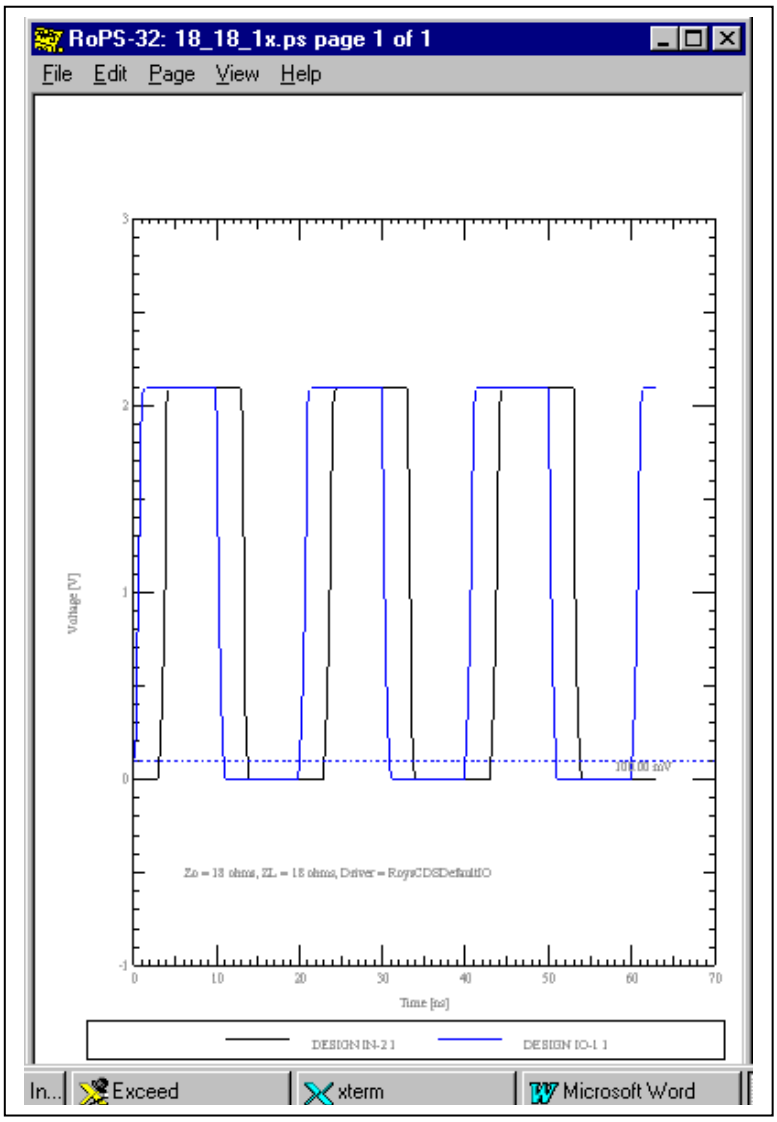


Figure 11: 4X V-I Model driving 72 Ohms – Vout ≈ 4.8 V.

This is close to the full output capability of the driver ideal internal voltage generator. Here, our Zout is about 4.5 ohms or only about 6.25% of the 72 ohm load on the driver.

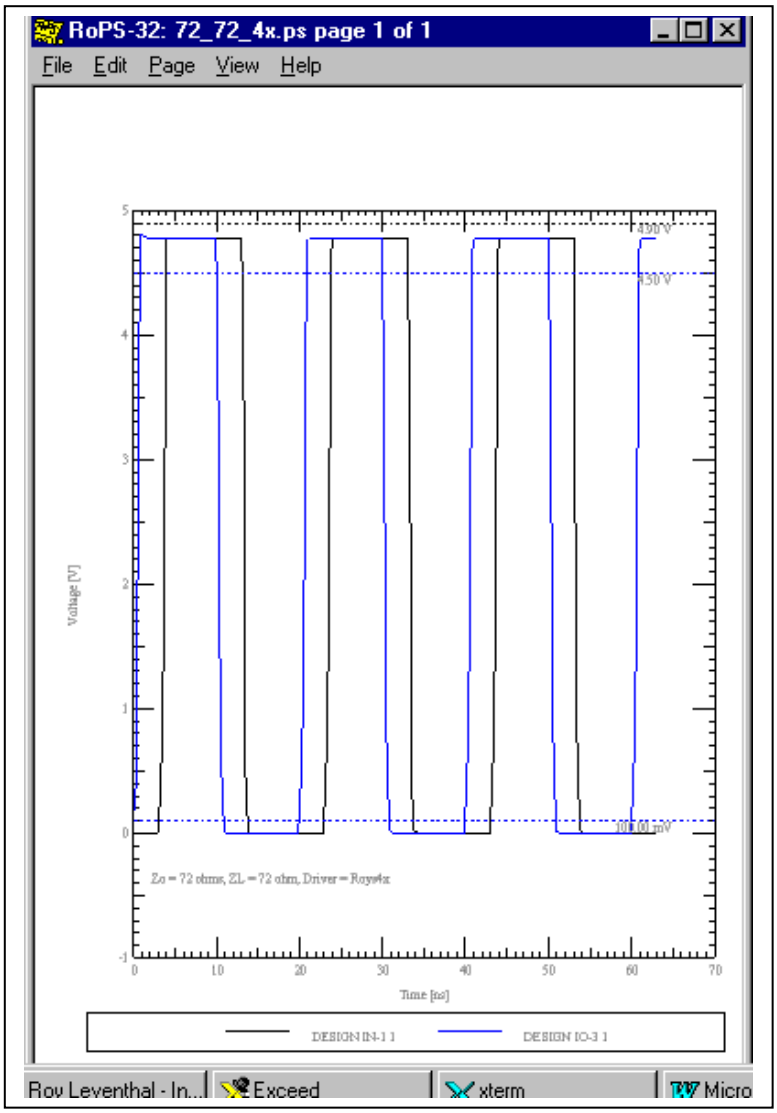
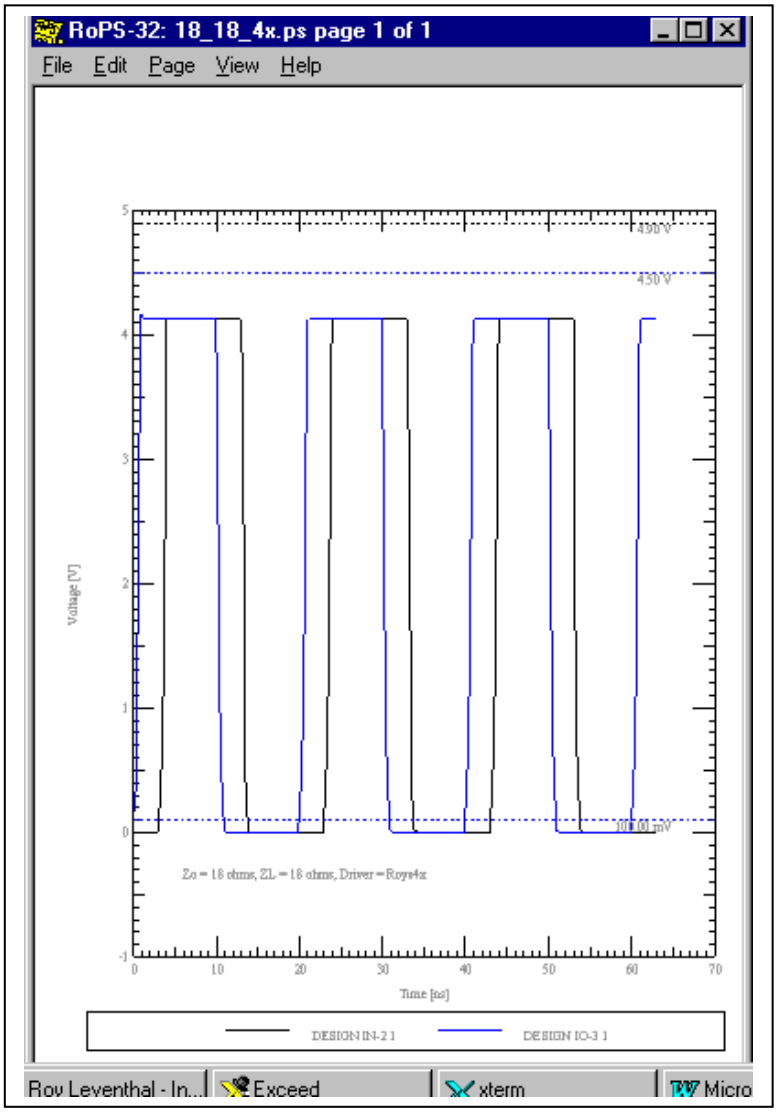


Figure 12: 4X V-I Model driving 18 Ohms – Vout ≈ 4.13 V

Note that driving a line with ¼ the Zo of our first simulation with a driver having 4 times the drive capability as was originally used, gives the exact same output swing. Just like ohms law claims!



V-T Data

What Happens Electrically when a Buffer Switches?

Recall that:

$$i = C \bullet dV/dt$$

Which can be solved to give:

$$dV/dt = i/C$$

What happens when a driver switches is that a constant current generator is switched on or off (sources or sinks current) and proceeds to charge or discharge its output (i.e. die capacitance or C_comp) capacitance.

In the absence of other loading on the output this means that the IBIS ramp / slew rates are set by:

$$dV/dt = i/C\_comp$$

Where i = current drive capability of the pullup or pulldown circuitry.

C_comp

Thus far we have discussed C_comp in relation to its three effects: 1) The effect of C_comp on the matching impedance presented to a transmission line; 2) The effect of C_comp on the filtering characteristics, and; 3) The setting of ramp rate as covered above.

IBIS ignores propagation delay (= flight time) from input to output through a device. An event occurs, somehow, that tells the output buffer to turn-on/turn-off. The turn-on and turn-off time of a buffer because of its finite slew rate behaves nearly identically with propagation delay. Thus, the term *buffer delay*.

It is important to remember that the effect of C_comp on dV/dt is already included, else how would you get a ramp? Wrong answers will be calculated if a simulator uses the C_comp data as an additional load on the output and adjusts the ramp rate accordingly. This will be “double-counting” the effect of C_comp. Of course, the actual ramp rate will be affected by the loading on the output of the driver. This is primarily due to the capacitive loading of the transmission line and the lumped loading of any termination circuitry placed on the driver. It would be nice to be able to modify the value of C_comp and simulate its effect on signal integrity and switching of a

net. But, if a software simulator supply company isn't careful they could end up double-counting C_comp's effect on the driver side.

On the receiver side the problem of double-counting C_comp's effect doesn't exist.

In the section on "Scaling V-I and Simulating the Results" we used a 72 ohm load to ideally terminate a 72 ohm line. The 72 ohm load was constructed by taking an Input model and reducing its V-I curves to a single characteristic of a 72 ohm resistor while reducing its C_comp to zero. The same approach can be used to construct an open circuit receiver load and adding back in various values of C_comp. This creates an idealized receiver model without clamps in which we can look at the effects of C_comp.

In our default model $t_{r(historical)}$ = (.8/.6) $t_{r(IBIS)}$ = 1.3333(.6 nS) = .8nS (typical) = edge-rate.
A critical frequency of interest is 1/edge-rate, which is 1.25 GHz

Another critical frequency[(1)] of interest is F_{knee} = .5/edge-rate, which is 625 MHz

Which is the frequency below which most of the digital pulse energy concentrates.

Now reactive impedance, Xc = $1/j2\pi fC$

At our two frequencies and two values of C_comp of interest we get the magnitude of Xc as:

Frequency (MHz)	|Xc| (ohms)		
	2.5 pF	5 pF	10 pF
625	101.92	50.96	25.48
1250	50.96	25.48	12.73

And, Reflection Coefficient, ρ, is given by:

$$\rho = (ZL - Zo)/(ZL + Zo)$$

And, on our 72 ohm line we get:

Frequency (MHz)	ρ		
	2.5 pF	5 pF	10 pF
625	.172	-.171	-.477
1250	-.171	-.477	-.7

In the first three simulations below you will see the results of simulating a pure capacitive load (no clamping) of 2.5 pF (Fast Simulation), 5 pF (Typical Simulation), and 10 pF (Slow Simulation). Following those three simulations is a Slow Simulation with a standard default CMOS Input (6 pF) at 50 MHz.

It's clear that what you are seeing is the effects of the reflection coefficient, frequency content of the edge rate and frequency rolloff of the input capacitance in the first three simulations plus, in the fourth simulation the add-in effects of clamping.

Figure 13: Ideal Open Circuit Receiver with C_comp = Cin = 2.5 pf, Fast Driver

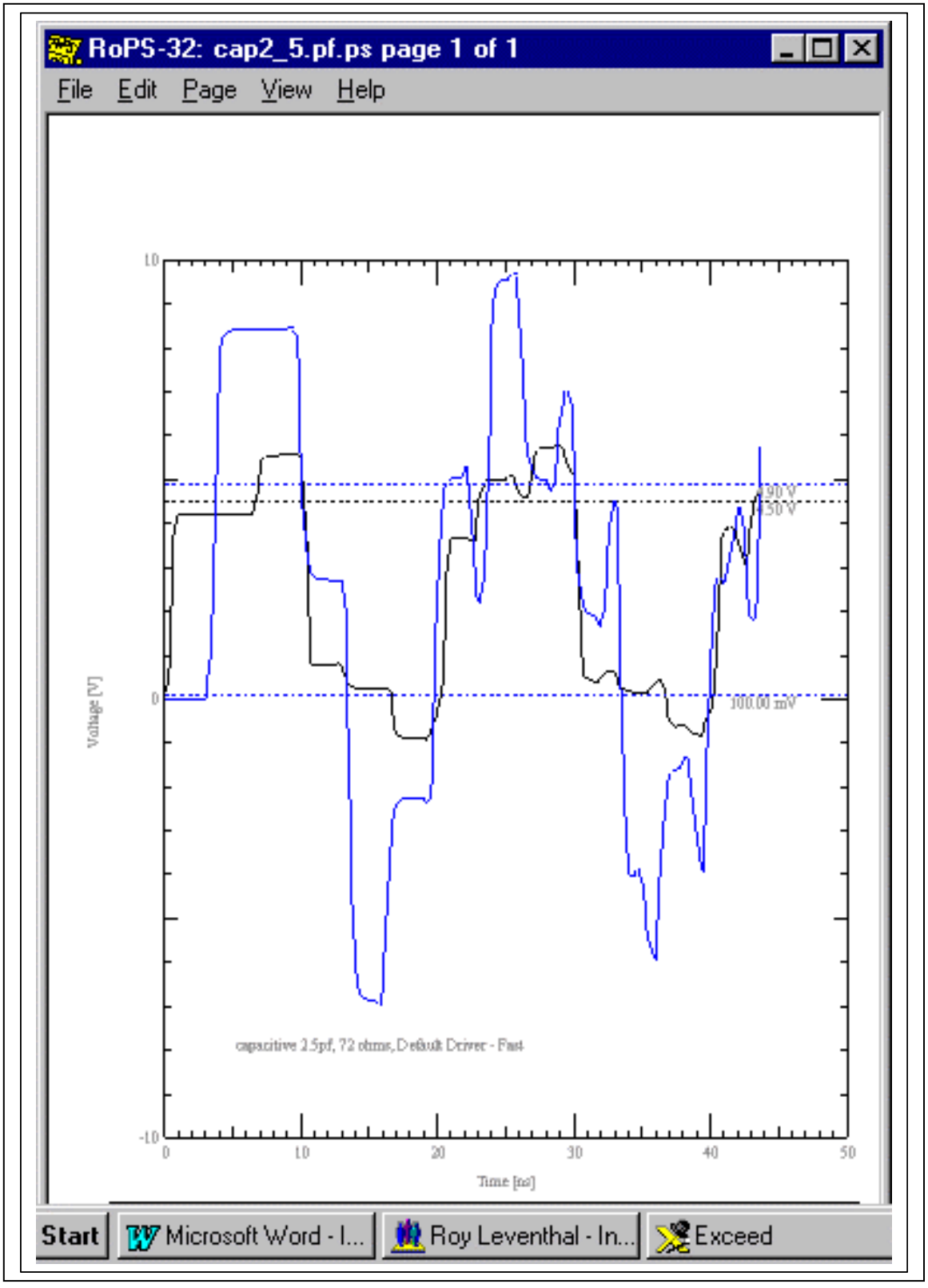


Figure 14: Ideal Open Circuit Receiver with C_comp = Cin = 5pf, Typical Driver

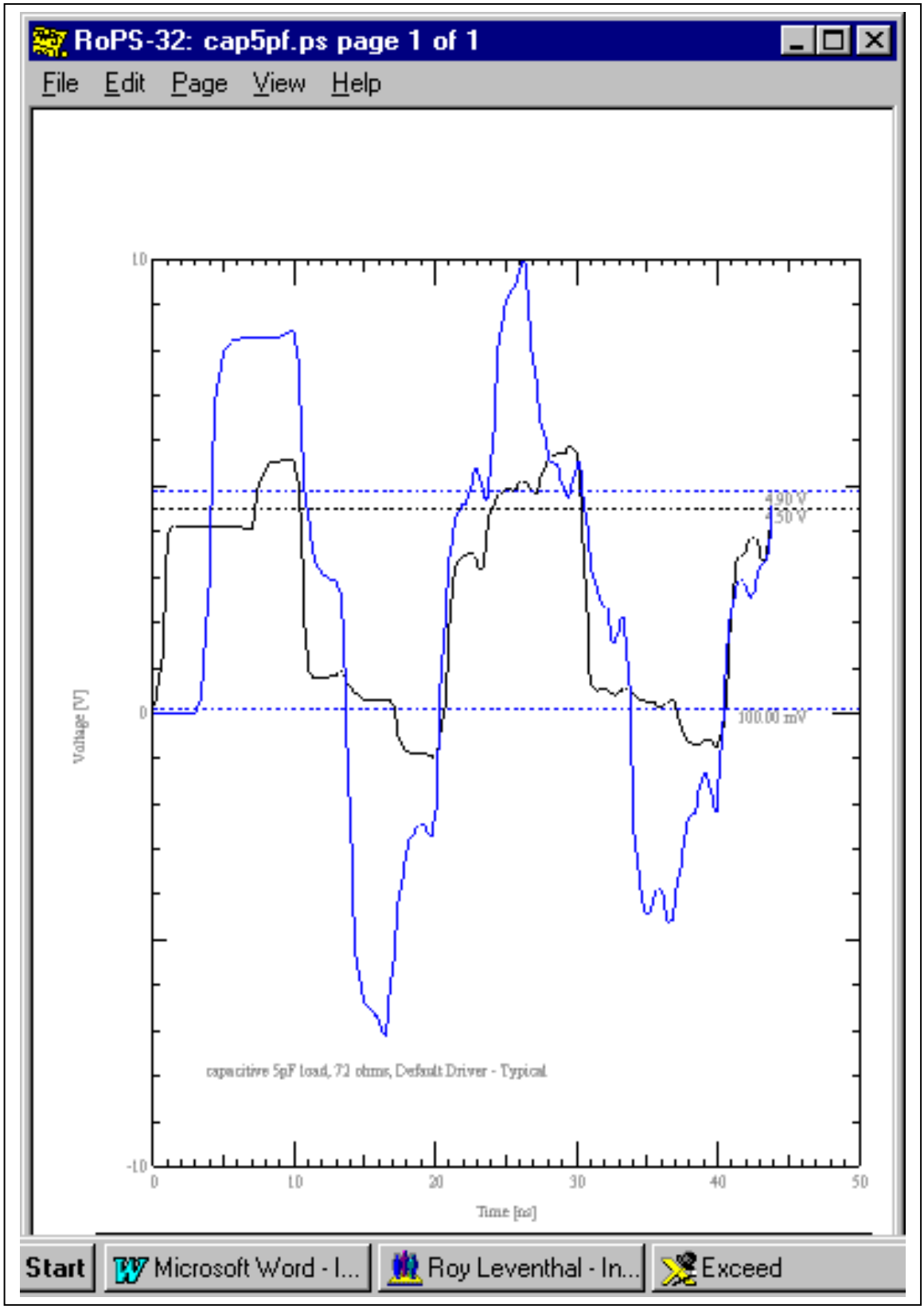


Figure 15: Ideal Open Circuit Receiver with C_comp = Cin = 10pf, Slow Driver

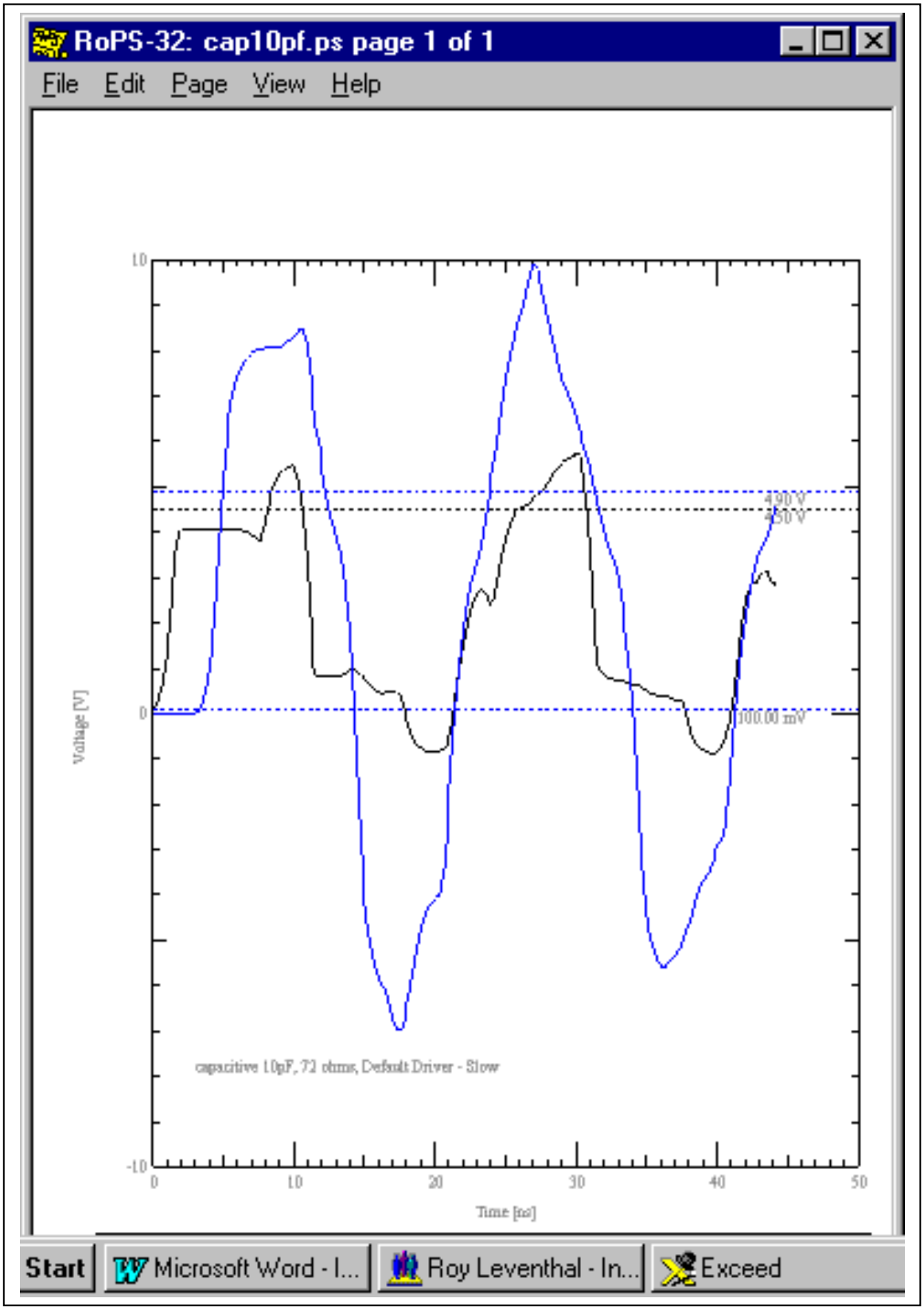
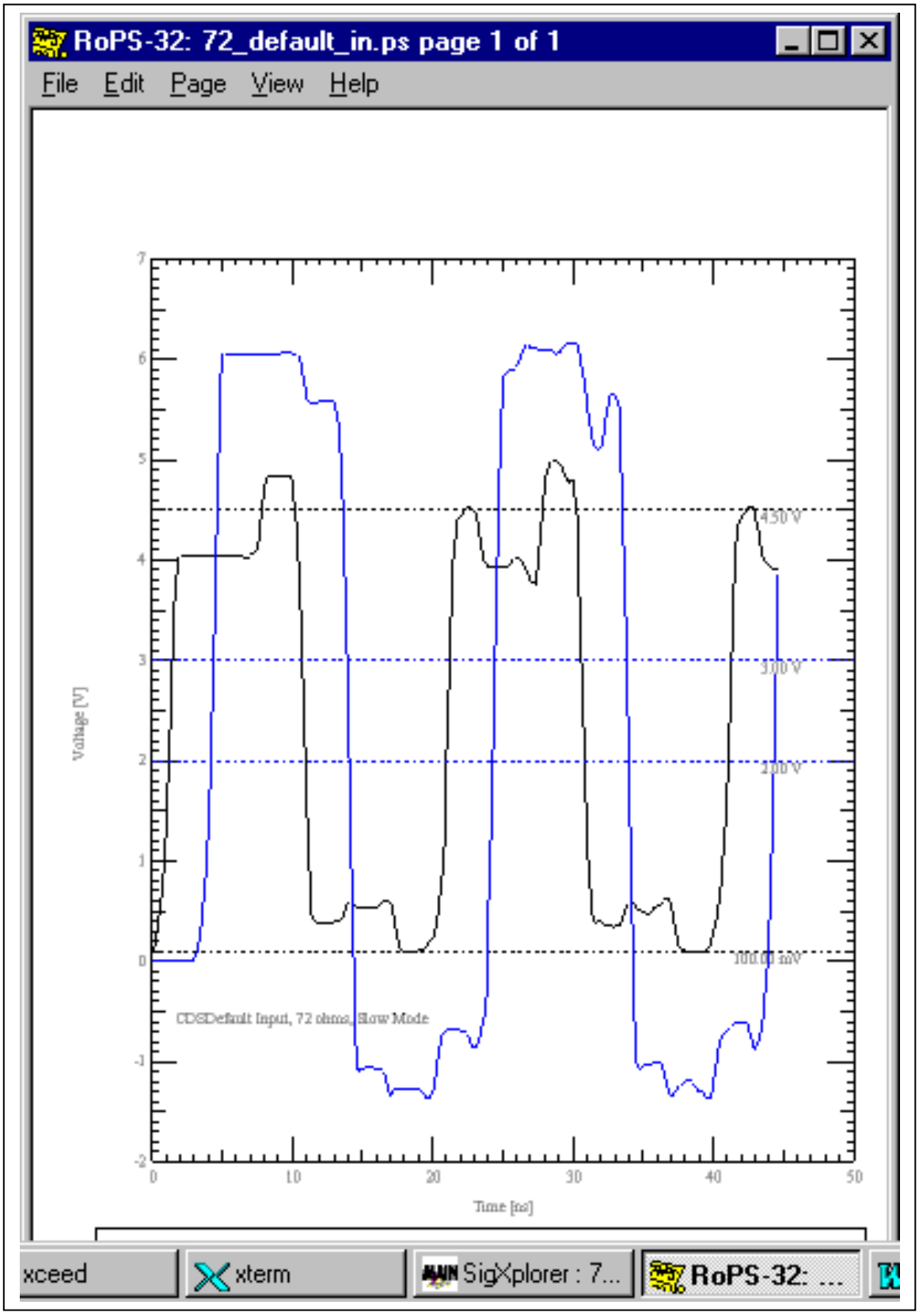


Figure 16: Default CMOS Input Receiver with C_comp = Cin = 6 pf and Clamping, Slow Driver



Scaling the size of C_comp in an IBIS model and re-simulating is just a matter of editing the IBIS file. This is nice for doing “what-ifs” regarding its effects. But, in a real device C_comp will be interdependent with slew rate so you can only draw general conclusions from your manipulation. The current drive capability of a driver can be increased by making it larger. This increases C_comp also. So, you end up with some intrinsic figures of merit for drivers made with a given technology that can be hard to improve upon.

Buffer Delay

Propagation delay of a buffer is assumed to be entirely due to turn-on and turn-off delay in the output cell itself. Actual flight time from input to output of a device is assumed to be negligible.

The turn-on and turn-off delay of a buffer can be re-simulated using the values of the buffer delay test fixture given its element values per the IBIS spec. This can be used to verify that the simulator produces the same answers as the data sheet. Or, small corrections can be made given that simulation and measurement will always have small errors between them. Likewise, buffer delay can be corrected for the effects of loading.

In simulating First Switch, Final Settle and flight time measurements it can be wise to correct buffer delay. If the original buffer delay was for a heavily loaded case and the current simulation is a lightly loaded case you can get negative delay times when extracting wire propagation delays! Such answers are obviously math artifacts caused by the buffer switching faster under light loads than the data stored in the simulator.

Ramp Data: Slew Rate

The various definitions and terms used to describe switching speed can be easily confused, because:

1) They have changed depending on usage, for example the definition of Rise Time dV range.

   And:

2) Minimum Rise Time corresponds to Maximum Rise Slew rate (or, alternatively edge rate) column which:

   Corresponds to Fastest Switching.

   And, Minimum Fall Time - - - and so forth.

Also, an IBIS file arranges its data columns Typ-Min-Max left-to-right instead of Min-Typ-Max which latter format most simulators and data books follow anyway. Formatting can get confused in the translation from databook to IBIS file to simulator file to simulator user interface.

The user has to understand the basic switching speed terms and definitions and remember to keep a few things straight:

Switching speed:

- "Fast Rise Time" corresponds to minimum (Min) Rise Time (t_r) and maximum (Max) Slew Rate (dV/dt_r) and, for parts normally held low, minimum drive high/turn on time. Similar definitions hold for the fall time/turn off time. Note that holding a logic part low usually means that it is turned on and drawing current!

- "Typical Rise Time" corresponds to typical (Typ) Rise Time (t_r) and typical (Typ) Slew Rate (dV/dt_r) and, for parts normally held low, typical drive high/turn-on time. Similar definitions hold for the fall time/turn-off time.

- "Slow Rise Time" corresponds to maximum (Max) Rise Time (t_r) and minimum (Min) Slew Rate (dV/dt_r) and, for parts normally held low, typical drive-high/turn-on time. Similar definitions hold for the fall time/turn-off time.

Rise/Fall Definitions:

- Historical Definition of Rise Time (and similarly Slew Rate) was the time to go from 10% to 90% of the rail-to-rail (usually 0 to Vcc) swing at the output of a device. Similarly for Fall Time. This is still the convention followed by most of the electronics industry and data books in particular.

- IBIS Definition of Rise Time (and similarly Slew Rate) is the time to go from 20% to 80% of the total transition when the output is loaded by R_load to Vcc for falling transition. R_load defaults to 50 ohms if it is not defined. ECL model types use R_load to Vcc - 2 V for both edges.

Scaling Slew Rate and Observing the Results in Simulation

We next look at the effect of ramp rate on some simulations. To do this we will not make use of Slow-Typical-Fast or Min-Typ-Max since the usual practice is to simultaneously vary C_comp, driver output impedance, etc. Instead, the Typ value of dt will be edited in the CDSDefaultIO model to take on the values of 0.9nS, 0.6nS and 0.48nS in turn. Also, we will set up a 79 ohm terminating resistance (10% higher than a perfect match) on a 72 ohm line. This is so we can set up some small reflections (ρ = -.046) and see what happens:

Figure 17: Reflections @ dV/dt = 3/0.9nS

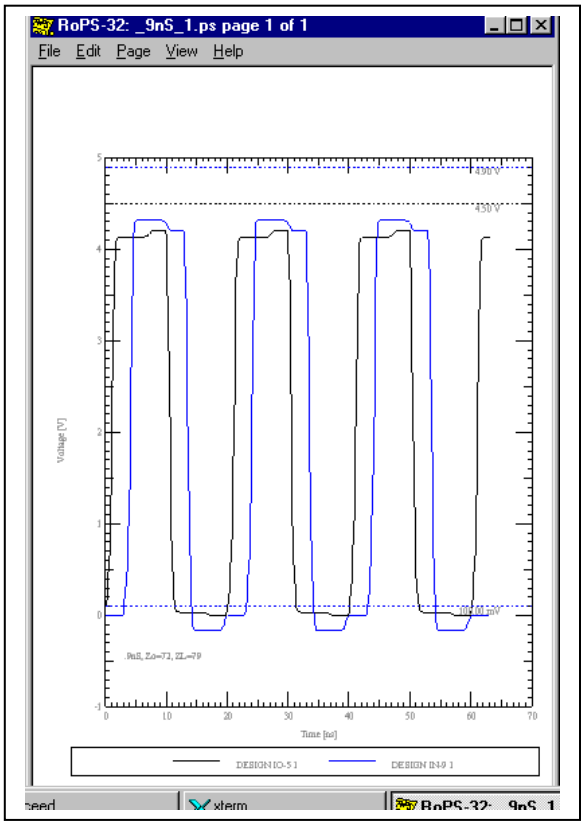


Figure 18: Reflections @ dV/dt = 3/0.6nS

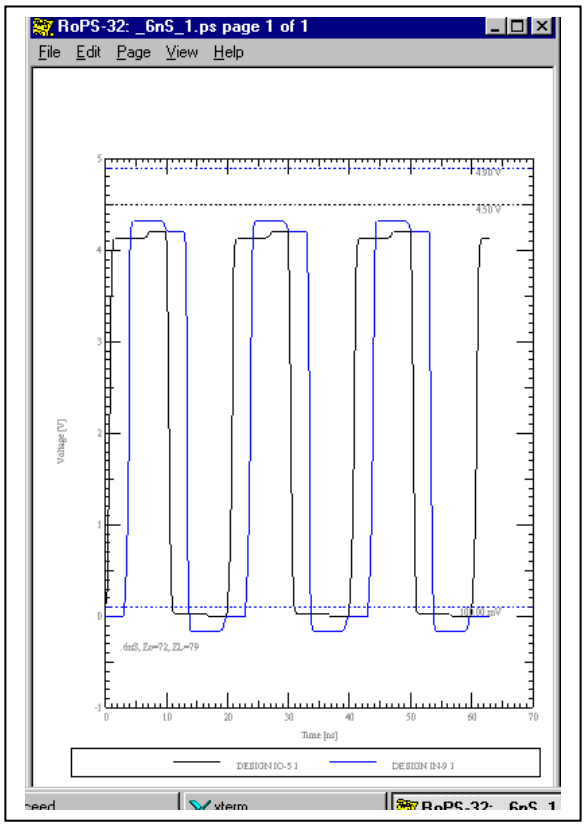
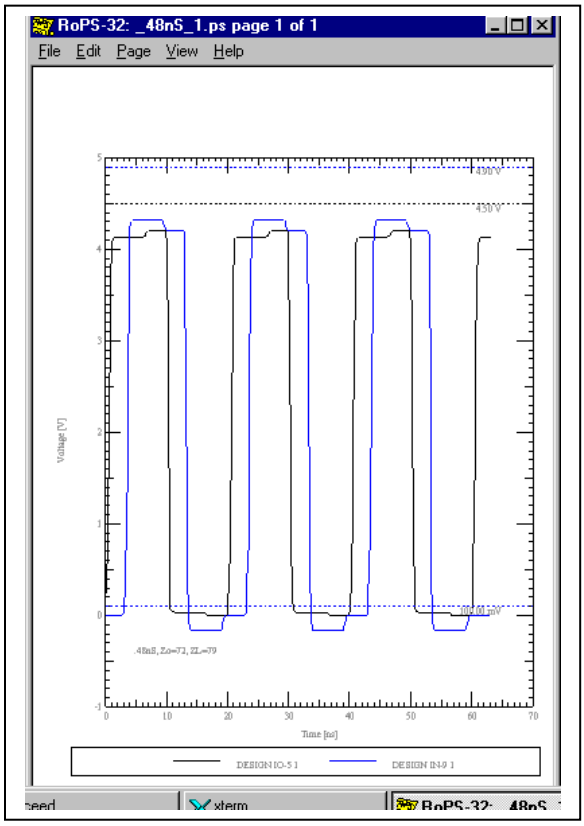


Figure 19: Reflections @ dV/dt = 3/0.48nS



As you can see, nothing much happened at all (This is like Dullsville, dude.). Actually, with a pure resistive mismatch, clock rate and edge rate changes do not affect reflection coefficient at all. So, we really won't see much change beyond an almost undetectable, at these scales, change in edge rate.

The topology was (as used in several examples elsewhere in this document) an unpackaged default CMOS IO cell driving a 3nS 72 ohm characteristic impedance line into a pure resistive load.

But, look next at what happens when we add a shunt parasitic capacitance (frequency variable element) of 5pf across the 79ohm load:

Figure 20: Reflections @ dV/dt = 3/0.9nS

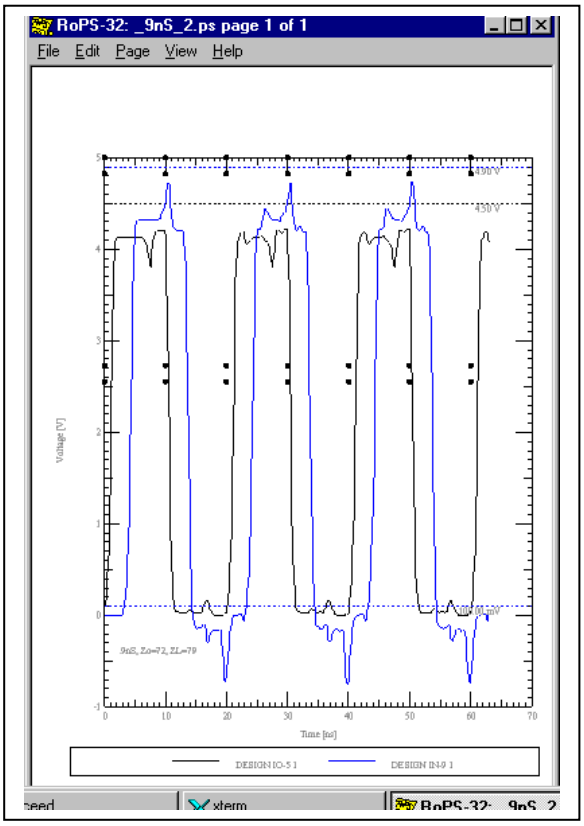


Figure 21: Reflections @ dV/dt = 3/0.6nS

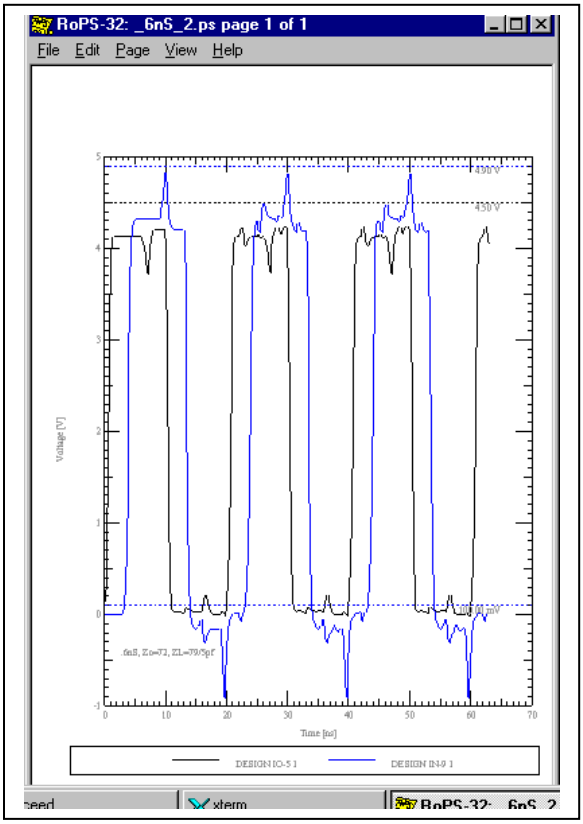
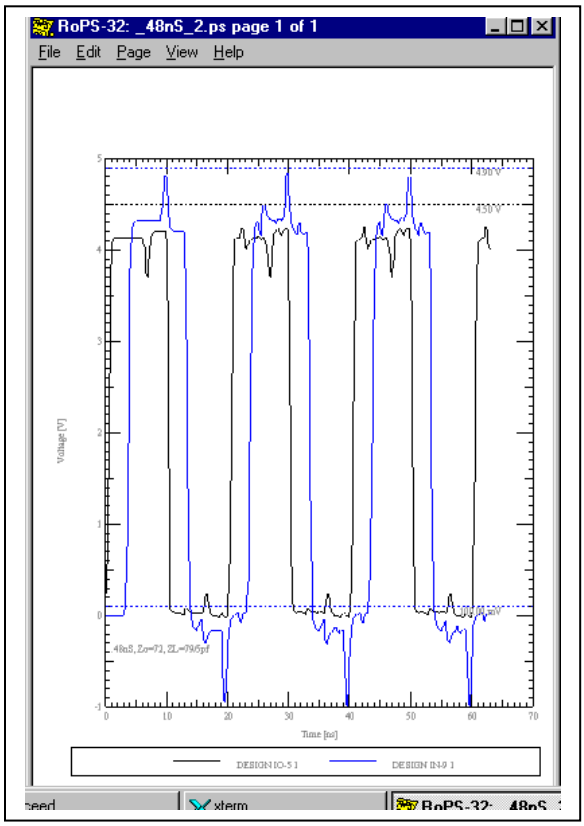


Figure 22: Reflections @ dV/dt = 3/0.48nS



Now, you can see that the interaction of edge rate frequency content and reactive load element very much does change results.

V-T Curves

V-T curves are nothing more than the plots of voltage vs. time of the rising and falling waveforms of the output of a driver into a specified test fixture. They take a little more labor to measure and then enter the data than the simple expedient of measuring the slopes of the rising and falling waveforms (dV/dt) and entering that data.

Advantages of V-T Curves: dV/dt Non-Linearity

Simplifying assumptions are made when the slopes only, dV/dt, of the rising and falling waveforms of a driver are used in simulations. The assumption is made that the waveforms are

linear (straight lines) from 0% to 100% of the output swing. The straight line curves then make sharp turns when they reach the rails.

The simplifying assumptions are simply not true for most real devices, which usually show rounding where the rising and falling waveforms are close to the rails and, sometimes, other non-linearities.

The effect of sharp corner V-T curves vs. round corner V-T curves is to inject more high frequency edge rate related energy into the simulations than is sometimes warranted. This can result in overly conservative designs and problems in correlating simulated vs. measured results. Some simulators recognize this and make “adjustments” in how they use ramp data. But, exactly what is done is usually a closely guarded secret.

In the next two sections we will examine these issues.

Examples of V-T Curves

Two sets of V-T curves are now presented. The first set is purposely linear with sharp corners on the rising and falling waveforms in contrast to the rounded corners of the “real” V-T curve model. This is done to mask out any rounding algorithms the simulator may be using on ramp data and contrast the non-linear effects directly.

Linearized V-T Curves

Figure 23: Linearized (Idealized) Rise Waveforms for Simulation in Following Example

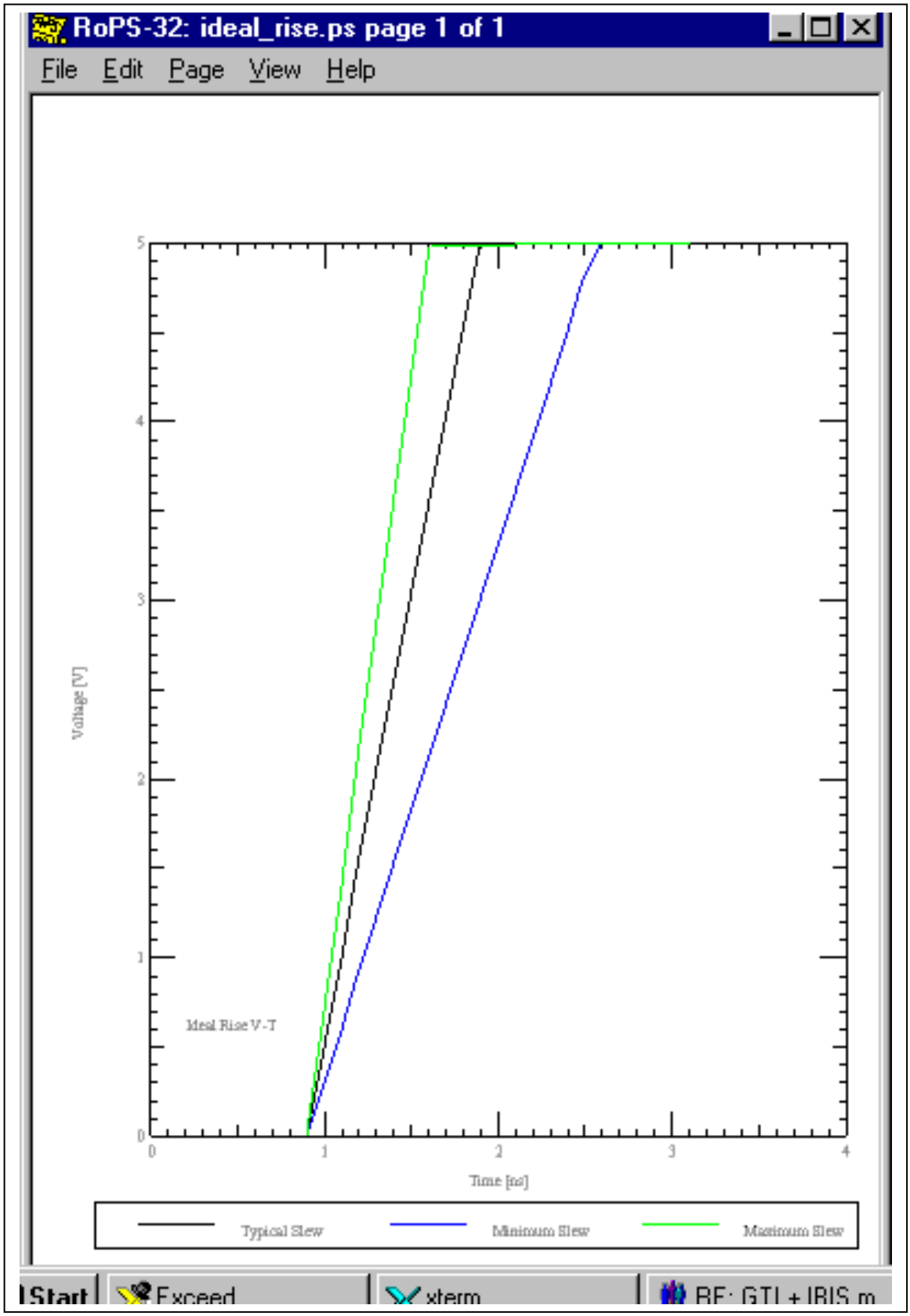
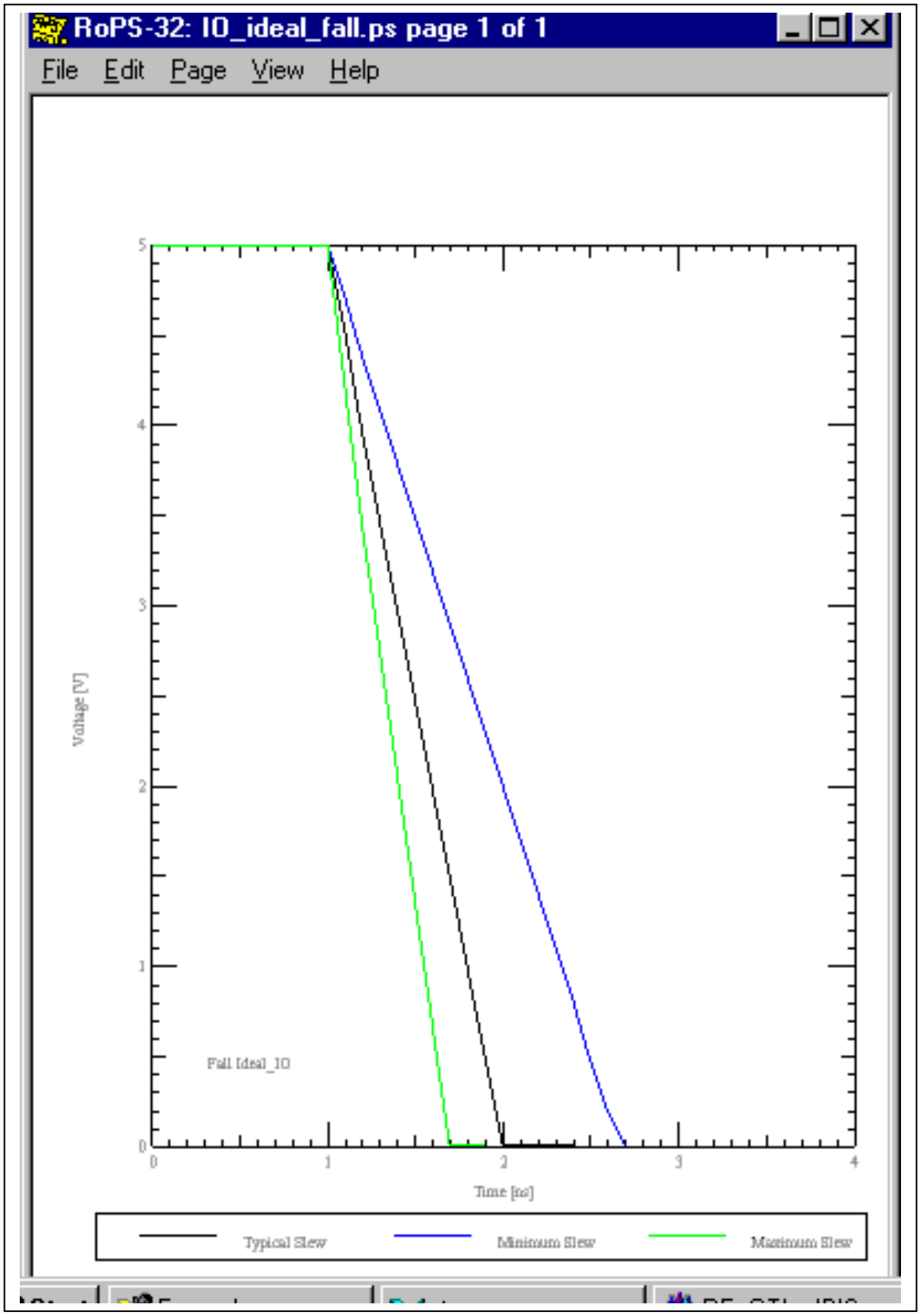


Figure 24: Linearized (Idealized) Fall Waveforms for Simulation in Following Example



Real V-T Curves

Figure 25: Non-Linear (Real) Rise Waveforms for Simulation in Following Example

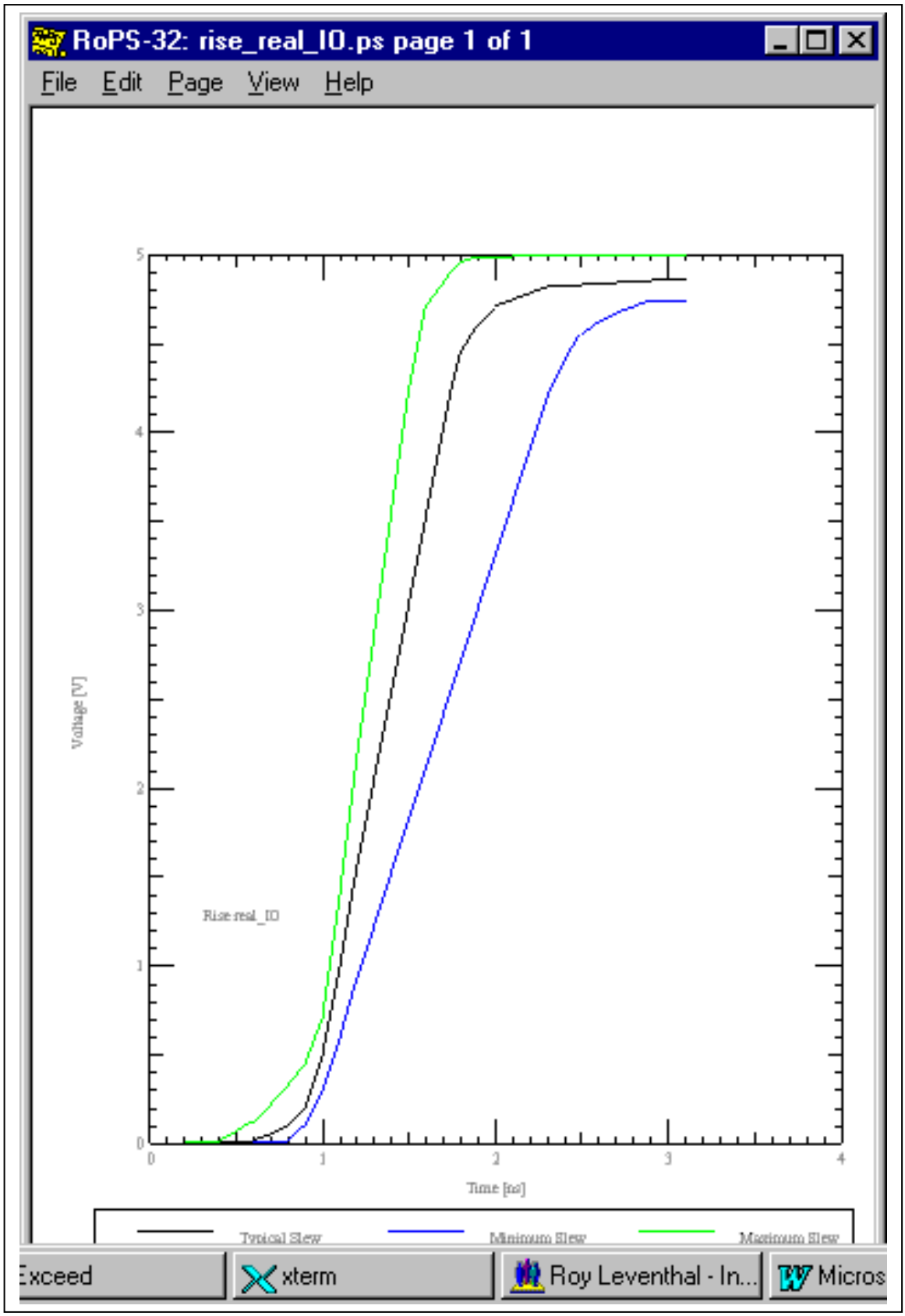
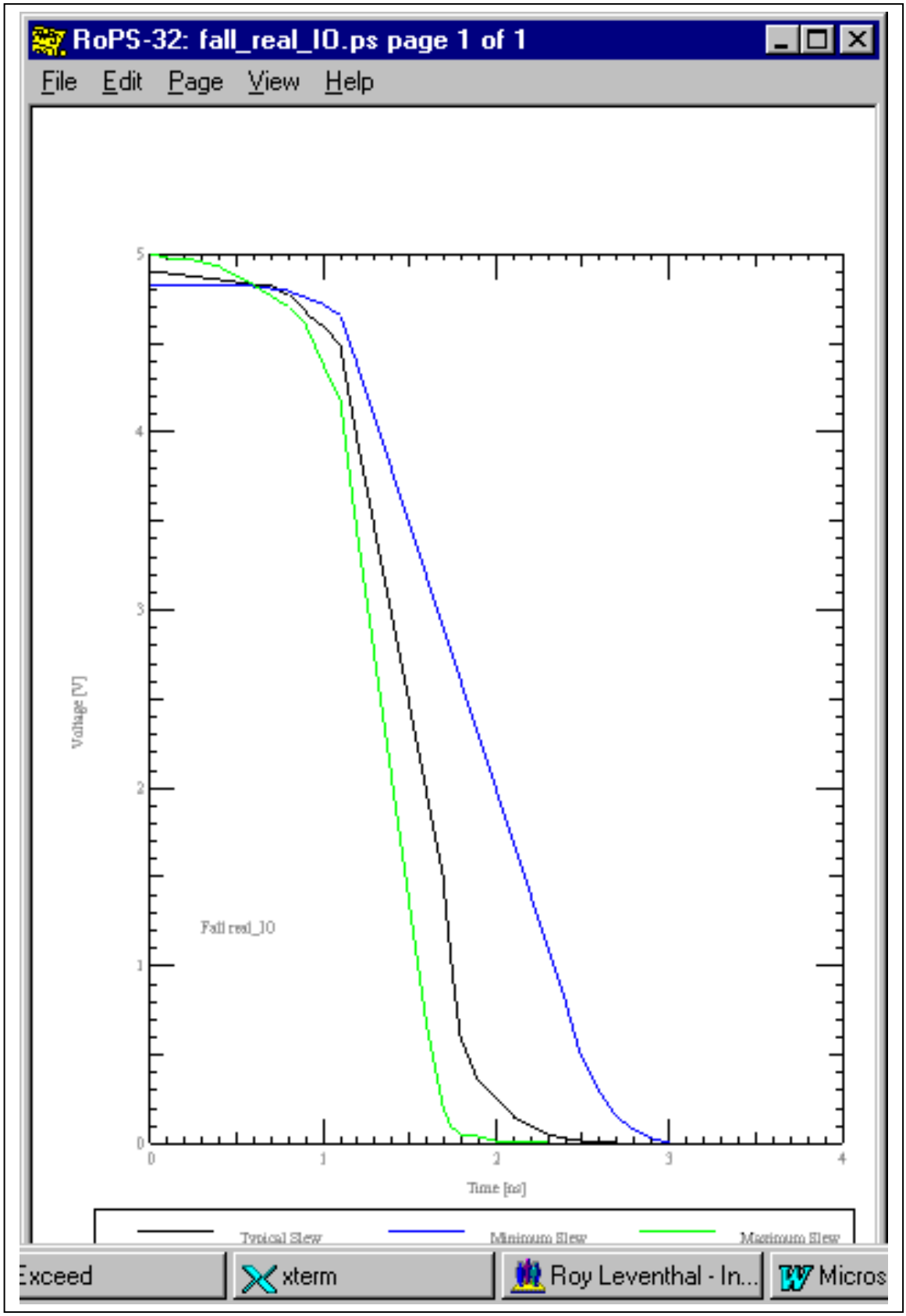


Figure 26: Non-Linear (Real) Fall Waveforms for Simulation in Following Example



Inserting V-T Curve Non-Linearity and Observing the Results in Simulation

The following pairs of simulation results are from driving the topology of Figure 27 with a CMOS default I/O model that first incorporates the “ideal” V-T curves and then uses the “real” curves illustrated above. The pairs contrast the results of “Slow” simulation, “Typical” simulation, and “Fast” simulation in turn.

Differences can be seen in the results. These differences would be significant for verification studies where it is desirable for measured and simulated results to agree within a couple of percent. But, these differences are small for the frequencies and circuit conditions given and should not enter into design decisions unless you are designing right “up to the edge” on noise margin. It is better by far to make design choices such as adding termination, shielding and use of slower edge rates that result in 50% and 100% improvements in noise margin. Fussing over a couple of percent difference due to V-T curve non-linearity is important for correlation, but the wrong approach for design. However, the results from a controlled ramp rate - soft turnon/turnoff part on a heavily loaded distributed bus can be marked, as the "V-T Curves: a Real Example" section following this one will show.

The best approach for measuring the differences in the simulation results is to look at peak-peak overshoot-to-undershoot for the two cases.

Figure 27: Topology for Contrasting Linear & Non-Linear V-T Curves

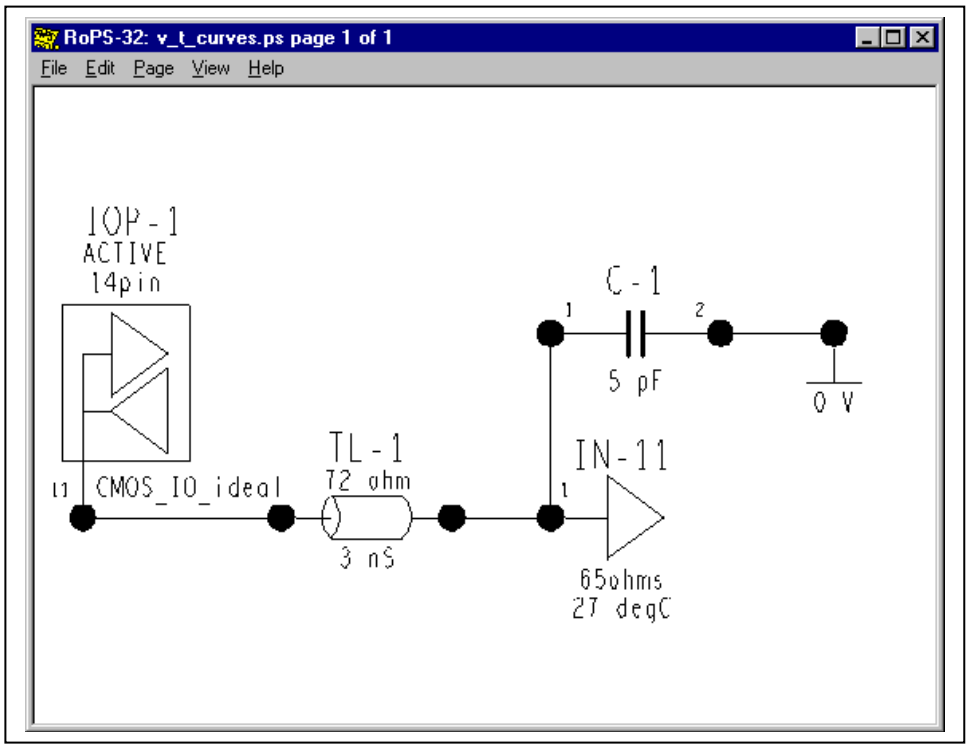


Figure 28: Ideal-Slow Driver

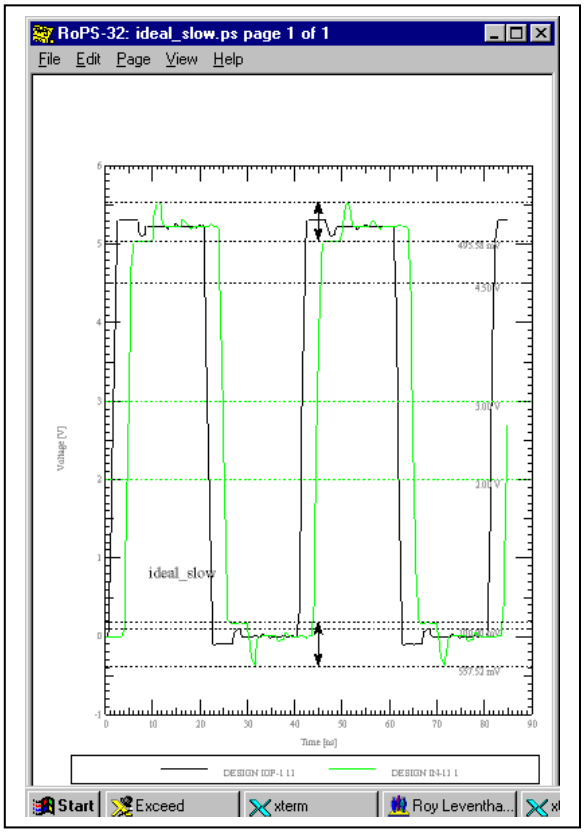
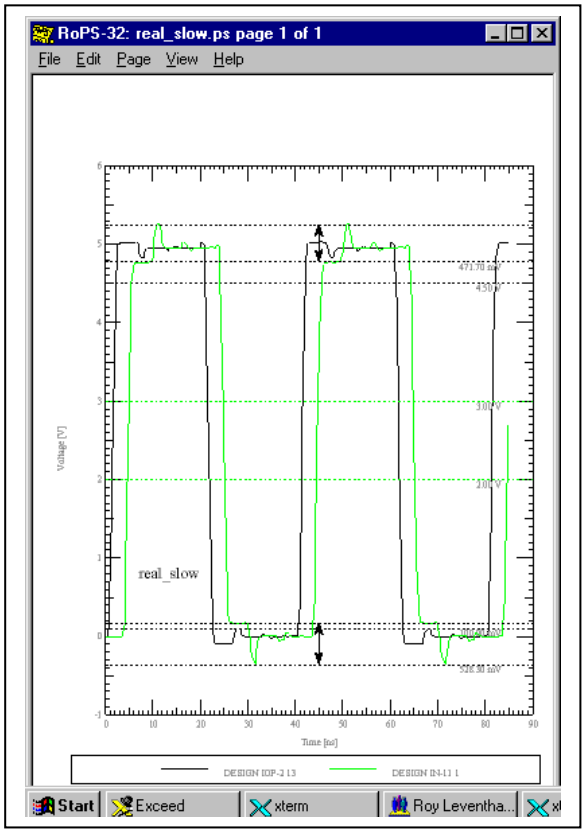


Figure 29: Real-Slow Driver



Peak-peak noise – high – ideal: 495.58 mV
Peak-peak noise – low – ideal: 557.52 mV

Peak-peak noise – high – real: 471.70 mV
Peak-peak noise – low – real: 528.30 mV

Figure 30: Ideal-Typical Driver

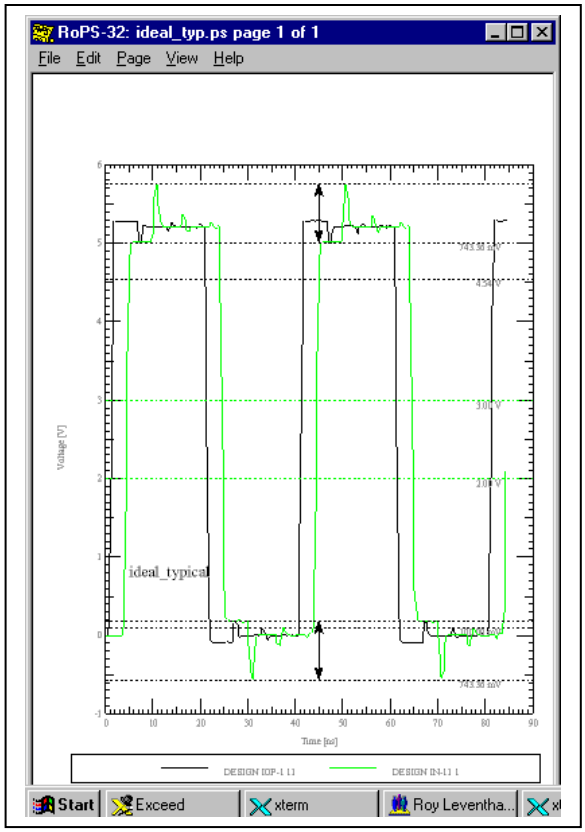
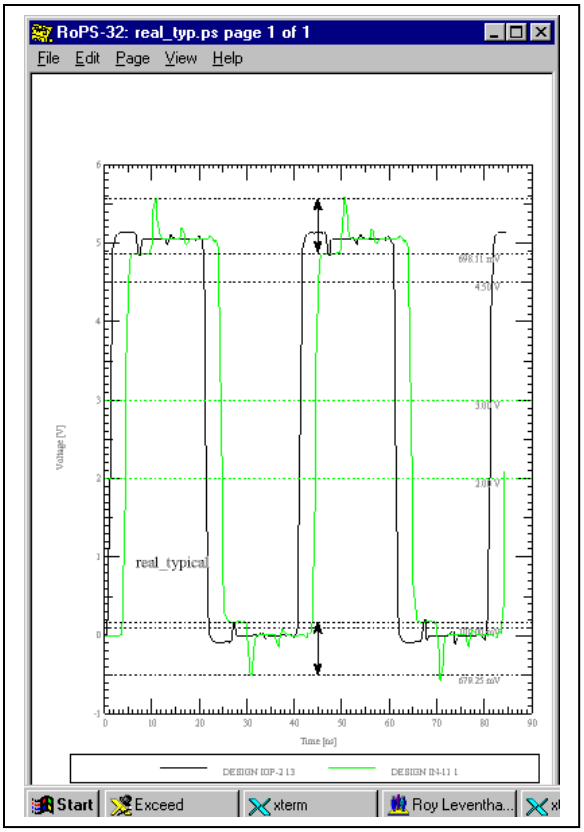


Figure 31: Real-Typical Driver



Peak-peak noise – high – ideal: 743.36 mV
Peak-peak noise – low – ideal: 743.36 mV

Peak-peak noise – high – real: 698.11 mV
Peak-peak noise – low – real: 679.25 mV

Figure 32: Ideal- Fast Driver

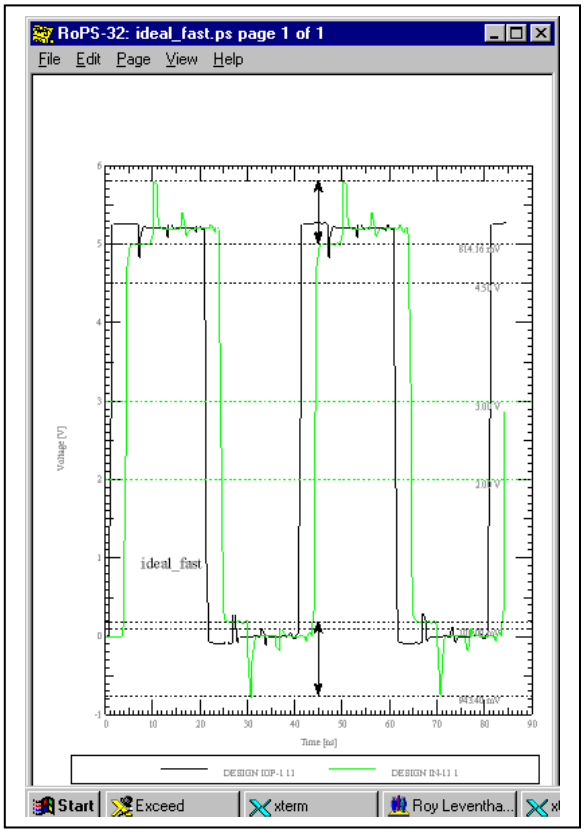
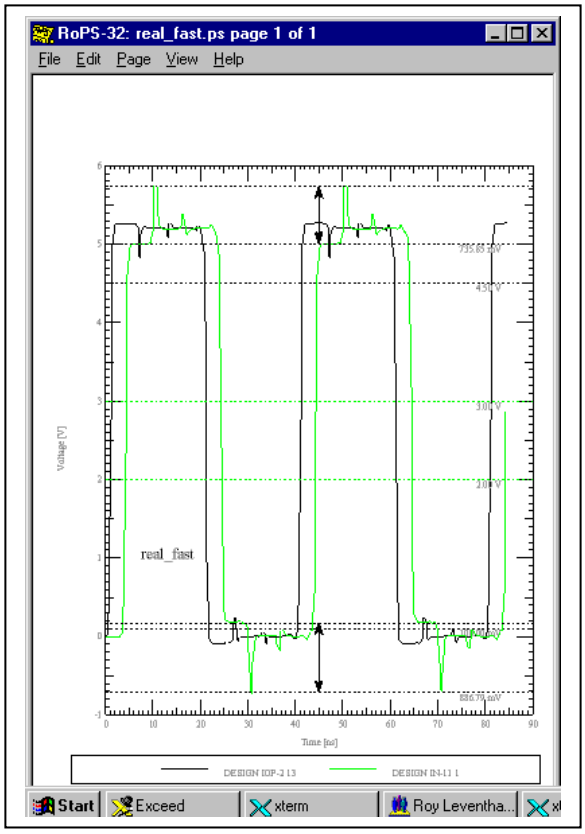


Figure 33: Real-Fast Driver



Peak-peak noise – high – ideal: 814.16 mV
Peak-peak noise – low – ideal: 943.40 mV

Peak-peak noise – high – real: 735.85 mV
Peak-peak noise – low – real: 886.79 mV

Summing up:

The “real” IBIS V-T curves result in an average 5% less maximum overshoot - undershoot across the board than the “ideal” IBIS V-T curves for this relatively well behaved example. Not a big deal. But, the effect described here can be really pronounced such as in the case of edge-controlled GTLP devices. These parts use active feedback to achieve “soft” turn-on and turn-off and edge rate control. Their reflections are much easier to tame.

V-T Curves: a Real Example

A multidrop TDM backplane bus with variable loading was studied to come up with recommendations for the design engineers.

Some of the givens were:

- Clock frequency: 8 MHz
- Backplane: 957 mils pitch between connectors, Zo = 63 ohms, 80 mils separation to closest aggressor nets on same layer.
- Nets: 56 TDM lines with 2 to 18 boards (variable) teed into each net at the connector in daisy chain fashion. But, removing or inserting a board did not interrupt the backplane bus continuity.
- Daughter cards: Each daughter card was a bi-directional I/O, that is each could be driver or receiver as determined by switching logic.
- Termination: Termination at the ends of the backplane bus would be OK, termination on each daughter card was less desirable.
- To Be Designed: Termination details, stub lengths and technology choices.

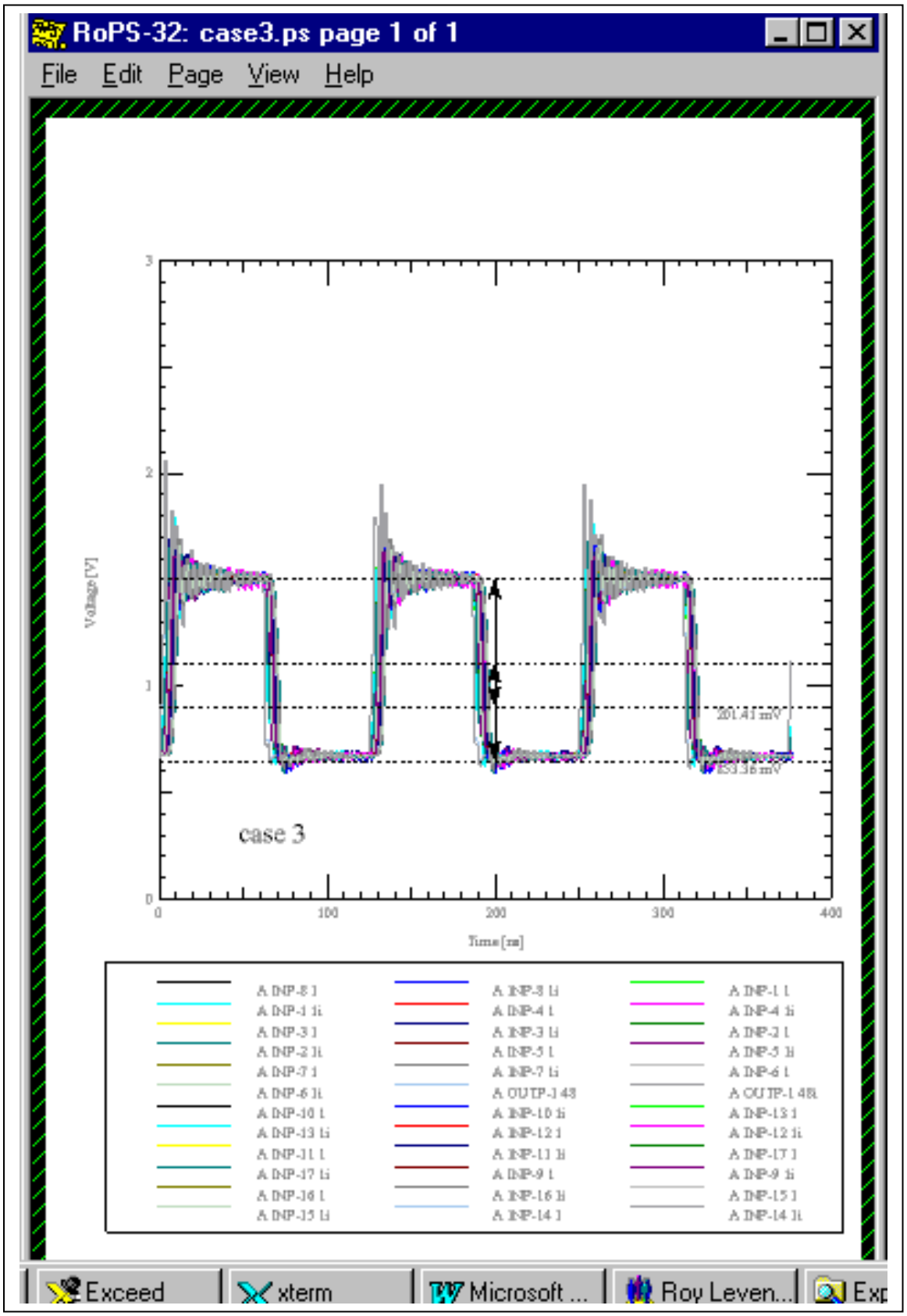
Some Results were:

- Stub lengths were optimized at 1000 mils.
- Technology choice was open-drain GTLP I/Os.
- Termination was one 33 ohm pullup resistor at each end of the nearly 18 inch long backplane bus.

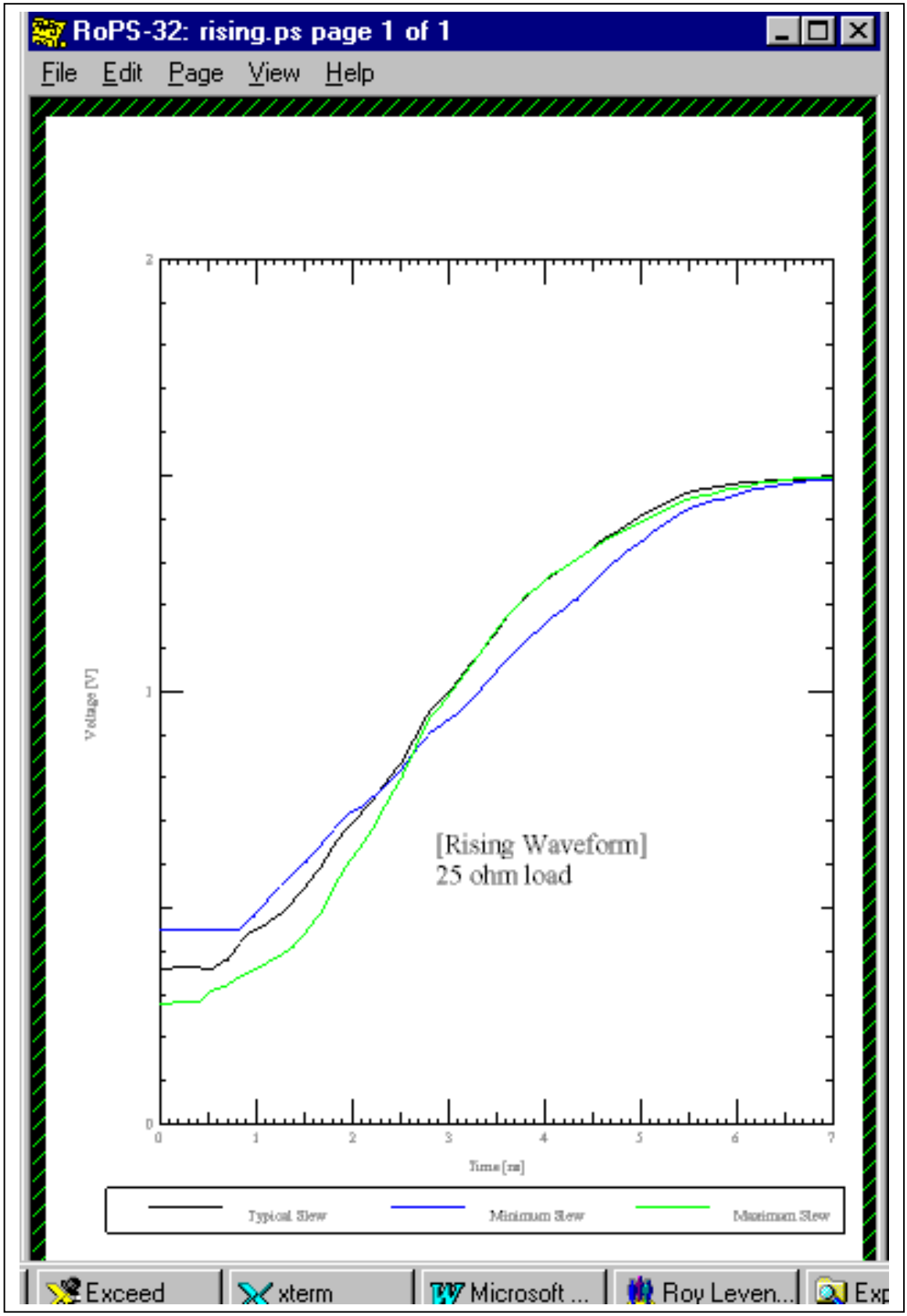
One of the most interesting results was the completely different conclusion I came to about the GTLP technology choice as opposed to my first impressions of it. In between I spent considerable time optimizing BTLP technology (which was the most recommended at the time) and investigating several others. What happened to change my mind?

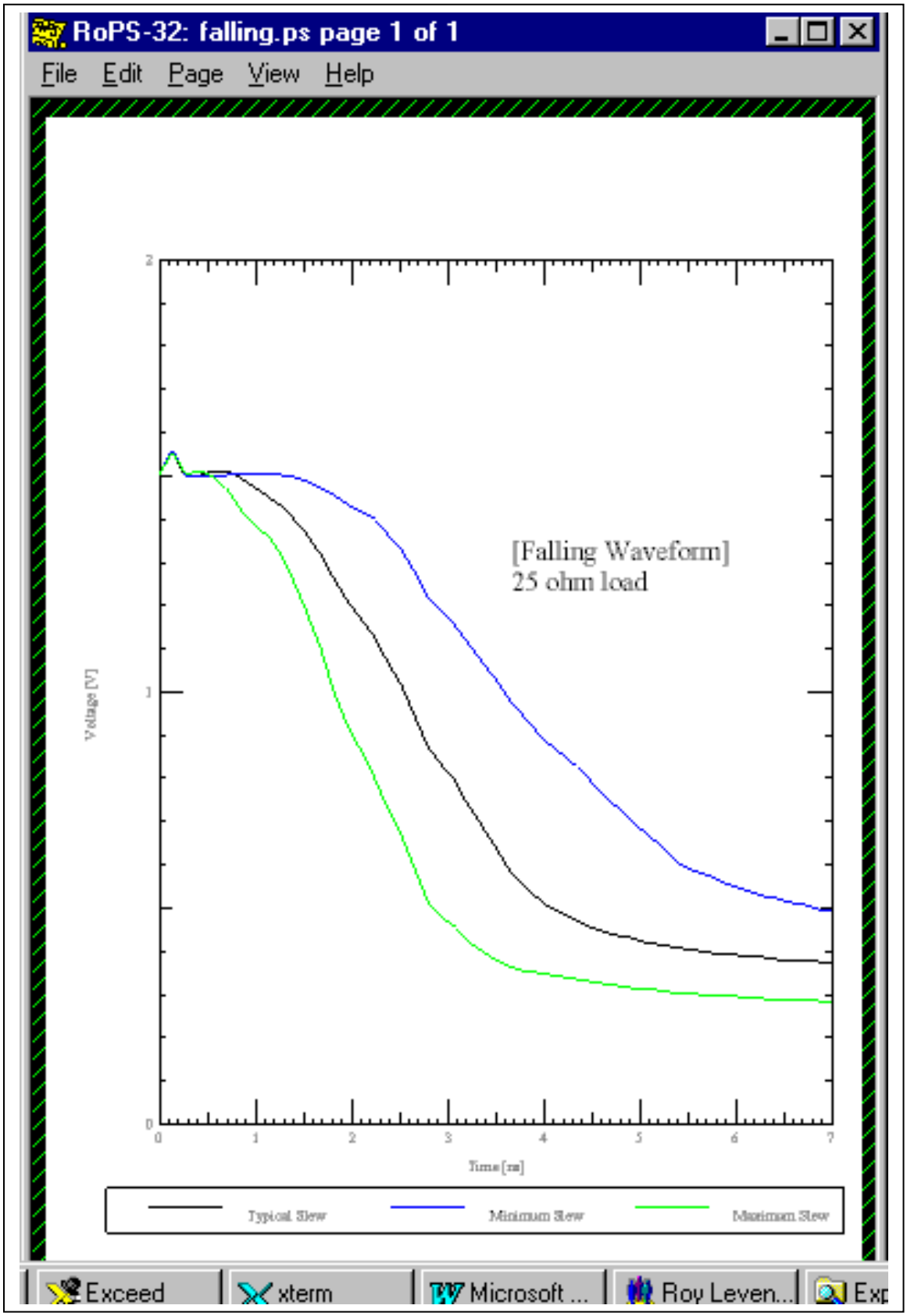
Well, the first time I simulated the GTLP model it was an IBIS 1.1 version that did not contain V-T curves. Here is what that simulation looks like in the final optimized topology:

Figure xxx: TDM Bus Envelope of Responses: 18 Boards

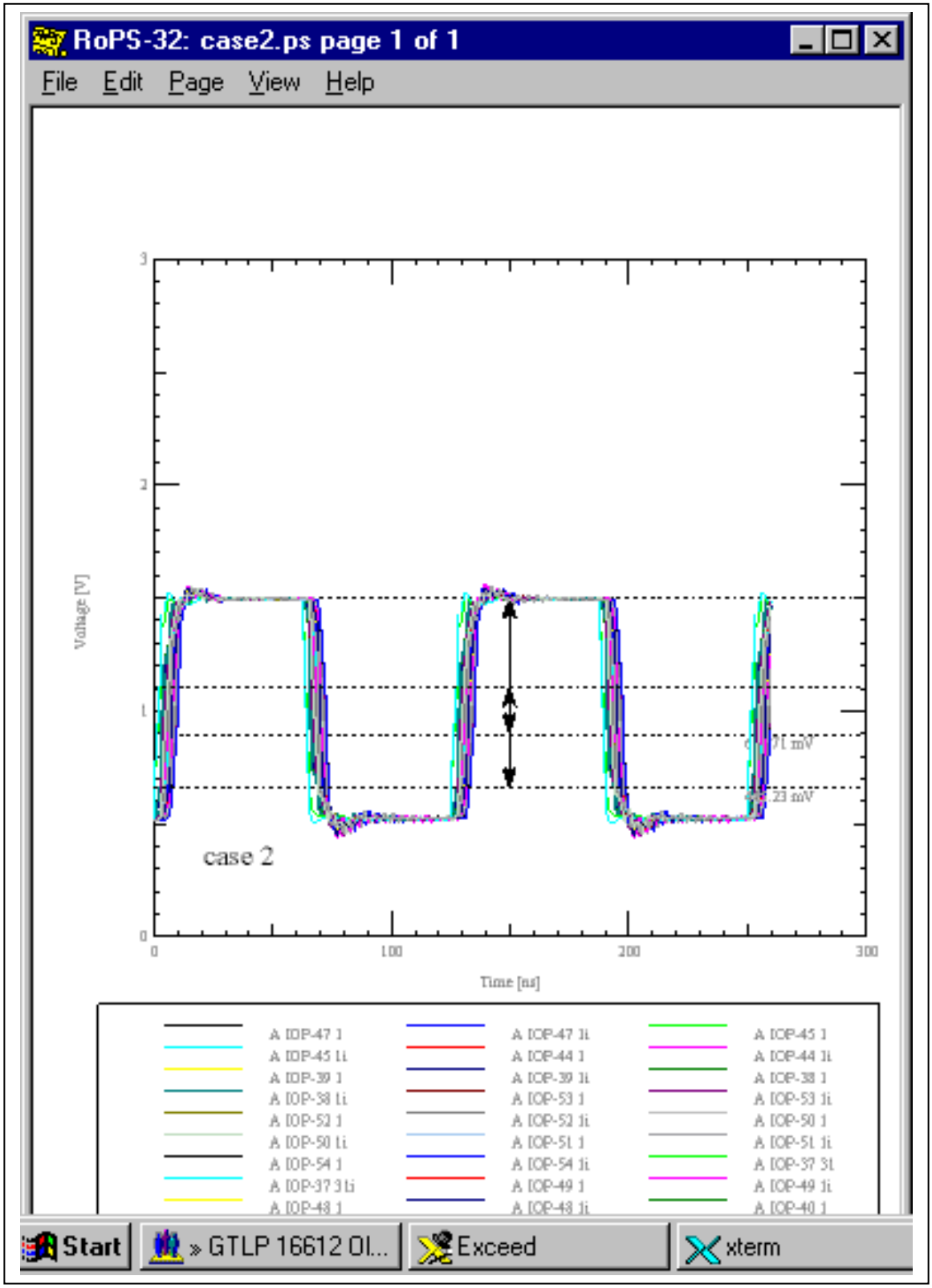


But, this part has a an active feedback (Gunning Transistor Logic) controlled ramp rate and soft turnon/turnoff. Look at the following properly modeled V-T curves from the IBIS 2.1 model:




RoPS-32: falling.ps page 1 of 1
File Edit Page View Help
[Falling Waveform]
25 ohm load
Voltage [V]
Time [ns]
Typical Slew
Minimum Slew
Maximum Slew
Exceed
xterm
Microsoft ...
Roy Leven...

And, here are the results of using the corrected model in the simulation:
Figure xxx: TDM Bus Envelope of Responses: 18 Boards



```
################################################################################
#  DF/SigNoise 13.0
#  (c) Copyright 1997 Cadence Design Systems, Inc.
#
#  Report:  Standard Reflection Summary Sorted By Worst Settle Delay
#          Mon Jan 25 16:18:16 1999
################################################################################

CASE 2: 1000 mil stubs, 63 ohm system, no RC termination,
              8 MHz, 33 ohm pullups, 18 boards
              load5.top

*****************************************************************************************************************
Delays (ns), Distortion (mV), (Typical FTSMode)
*****************************************************************************************************************
XNet    Drvr         Rcvr        NMHigh NMLow OShootHigh OShootLow SwitchRise SwitchFall SettleRise SettleFall Monotonic
------- ------------ ----------- ------ ----- ---------- --------- ---------- ---------- ---------- ---------- ---------
1 A MD7  A IOP-37 31  A IOP-54 1  373.7  344.9  1542       452.5     10.12      10.06      10.98      10.82      PASS
1 A MD7  A IOP-37 31  A IOP-53 1  376.4  347.9  1543       453.2     9.757      9.708      10.61      10.46      PASS
1 A MD7  A IOP-37 31  A IOP-52 1  382.6  357.3  1539       466.7     9.358      9.311      10.24      10.08      PASS
1 A MD7  A IOP-37 31  A IOP-51 1  372.3  343.6  1543       458.4     8.975      8.928      9.846      9.687      PASS
1 A MD7  A IOP-37 31  A IOP-50 1  381.5  354.5  1548       455.8     8.552      8.501      9.481      9.289      PASS
1 A MD7  A IOP-37 31  A IOP-49 1  382    336.4  1560       439       8.12       8.088      9.06       8.863      PASS
1 A MD7  A IOP-37 31  A IOP-48 1  385    323.1  1544       451.8     7.748      7.685      8.646      8.449      PASS
1 A MD7  A IOP-37 31  A IOP-47 1  385.3  303.9  1540       459.6     7.367      7.293      8.267      8.045      PASS
1 A MD7  A IOP-37 31  A IOP-46 1  381.4  305.5  1547       456       6.915      6.861      7.818      7.6        PASS
1 A MD7  A IOP-37 31  A IOP-45 1  387.3  294.8  1545       447.1     6.534      6.456      7.368      7.162      PASS
1 A MD7  A IOP-37 31  A IOP-44 1  369.2  288.3  1547       450       6.106      6.023      6.972      6.735      PASS
1 A MD7  A IOP-37 31  A IOP-43 1  369.3  350.8  1526       475.5     5.642      5.567      6.537      6.295      PASS
1 A MD7  A IOP-37 31  A IOP-42 1  356.4  268.3  1538       458.4     5.241      5.165      6.066      5.84       PASS
1 A MD7  A IOP-37 31  A IOP-41 1  358.4  344.3  1531       472.6     4.825      4.741      5.695      5.431      PASS
1 A MD7  A IOP-37 31  A IOP-40 1  356.3  331.6  1521       478.7     4.396      4.321      5.254      4.987      PASS
1 A MD7  A IOP-37 31  A IOP-39 1  356.4  336.8  1518       488.5     3.952      3.826      4.823      4.53       PASS
1 A MD7  A IOP-37 31  A IOP-38 1  349.7  327.2  1515       490.8     3.51       3.39       4.383      3.992      PASS
-----------------------------------------------------------------------------------------------------------------------

*********************************************************************
Simulation Preferences
---------------------------------------------------------------------
Variable                Value
 ------------------------  ---------------
 Pulse Clock Frequency     8MHz
 Pulse Duty Cycle          0.5
 Pulse Step Offset         0ns
 Pulse Cycle Count         2
```

So, with a more badly behaved topology a more sophisticated and correct model of the V-T behavior makes all the difference. All the T-junctions where the daughter boards tap into the bus, the stub lengths and the distributed lumped - loaded, heavily loaded effects on the line cause noise. Eliminating as much high frequency content, especially edge rate and hard turnon/turnoff from the part as possible and properly modeling it makes all the difference. This has the effect of making the clock more trapezoidal - even sinusoidal - and (we would already have guessed) much easier to tame.

Parasitics & Packaging

Pins, Connections, and Model Assignments

IBIS provides syntax for calling out which model types are connected to which pins. Digital ICs are multi-pin devices containing more than one input and/or one output almost by definition. The types of models attached to each pin and the pins where the rails are connected to need to be defined.

The IBIS Keywords [Package], [Pin], [Package Model], [Pin Mapping], and [Diff Pin] are all concerned in whole or in part with the internal connections and model cell assignments of a device. These Keywords allow for increasing levels of detail regarding connections and cell types.

[Package] allows for a default set of parasitics (R_pkg, C_pkg & L_pkg) to be globally assigned to all the pin-to-semiconductor die connections of a device.

[Pin] allows all of the pins (actually I/O cells on the semiconductor die) to be assigned their proper model type. [Pin] also allows pin-to-semiconductor die connection parasitics (R_pin, C_pin & L_pin) to be individually called out by pin. R_pin, C_pin & L_pin will override the values in R_pkg, C_pkg & L_pkg.

When the R_pin, C_pin & L_pin properties are measured they obviously are for a specific semiconductor die + package combination. Depending on how the device was set up to be measured, test socket or PWB, the user should be aware that a given set of pin parasitics are specific to a given testsocket/padstack/via combination also.

There are some cases where bare die behavior (V-I & V-T curves) can be inserted in different package types and the effects of the pin parasitics added in making characterization information “portable.” Certainly, “what-if” simulations can be run varying pin parasitics data.

[Package Model] allows the creation of a separate file (optional) *packagemodel*.pkg in which additional detail can be supplied. This file must be in the same directory as the *ibismodel*.ibs file. Else, include the [Package Model] Keyword in the .ibs file. This model allows for the description of cross coupling between pin connections and between pin connections and package body. This is done through a set of R, L, & C matrices that describe the self and mutual values of the connections. IBIS does not officially support a G (conductance) matrix. Most often the values of G are extremely small and are approximated to zero.

In a large package model of N pins (N-by-N matrix) these matrices can be very large. Their combination into an RLGC matrix and solution can be computer-intensive. Therefor, IBIS is set up to handle sparse and banded matrices where coupling between non-adjacent and shielded pins is insignificant.

The Package Model has been adapted by some simulator companies to handle multi-pin connectors. Also, it should be noted, the RLGC matrix handles the case of a set of cross-coupled transmission lines (etch) or cable wires.

[Pin Mapping] allows a set of I/O cells to be associated with particular ground and power pins (busses). Today, it is often the practice in large, high speed digital packages to provide multiple power and ground pins that serve particular sets of I/O cells. Also, multiple pullup and pulldown rail voltages can be assigned to those busses.

[Diff Pin] allows for pairs of I/O pins to be set up as differential pairs. Offsets in driver launch time delay or input threshold voltage (mutually exclusive) can be specified.

Individual device types can contain from eight pins to many hundreds of pins. A packaged semiconductor device containing 1000 pins is not unheard of. The case where many hundreds of pins exist can present a bit of a dilemma. It may be necessary to analyze a few critical nets attached to this device for signal integrity while ignoring most of the many hundreds of other pins.

The IBIS spec clearly states that all pin information is to be given. Since spelling out all the pin information can be labor intensive and tedious, observing the letter of the law on the IBIS spec can be counterproductive. But, all the pins may need to be specified in translating an IBIS model to a file that is to operate with a particular software company's simulator. You can make the unused pins temporarily NC = No Connect, if necessary, if you are in a hurry to check out a device.

Package Parasitics

Package parasitics really refers to the global assignment of pin parasitics. C_pkg does represent the total stray capacitive coupling of a pin to ground through the package. C_pkg does not represent the stray capacitive coupling of the package itself to ground.

Pin Parasitics

A manufacturer most often estimates package and pin parasitics (R-L-C). A few manufacturers characterize their bare die and add in parasitics. Some vendors, especially modeling services, de-embed the die behavior and the parasitics from a characterization of packaged devices.

The parameter G (GMATRIX) is normally non-existent in most IBIS models including interconnect models because it is normally so small it is most often approximated to zero. It represents leakage conductance.

The only place it might appear is in very high frequency simulations beyond 5GHz. The value of G might be involved in a field solver simulation where it could be used in calculating the high frequency bandwidth limiting effects of loss tangent and frequency dispersion on the PC board itself. The dimensions of ICs, connectors and discrete components are usually too small to involve loss tangent and frequency dispersion all the way into the 50 GHz^{+} range.

Examples of Pin Parasitics

Following is a bit of text from an IBIS file giving the default pin (package) parasitics with some fairly typical values:

	typ	min	max
R_pkg	0.071	0.063	0.077
L_pkg	4.129nH	3.799nH	4.627nH
C_pkg	0.923pF	0.803pF	1.143pF

We shall use these values in the next section to take a look at the effects of pin parasitics on simulation results.

Simulation of Package Parasitics

In this section we are primarily interested in looking at how the magnitudes of R-L-C in pin parasitics change the results of simulation. We shall again use the expedient of editing the Typ(ical) values for these parameters so as to avoid changing a bunch of other properties at the same time. We will increase and decrease (together) pin parasitic R-L-C at both receiving and driving ends of a transmission line. And, we should probably change edge rate because L & C are reactive elements. Maximum edge rate with maximum R-L-C and minimum edge rate with minimum R-L-C .

The topology will be (as used in several examples elsewhere in this document) a default CMOS IO cell driving a 3nS 72 ohm characteristic impedance line into a pure resistive load. The load impedance will be 79 ohms to set up a small reflection.

Figure 34: Topology For Pin Parasitics Example

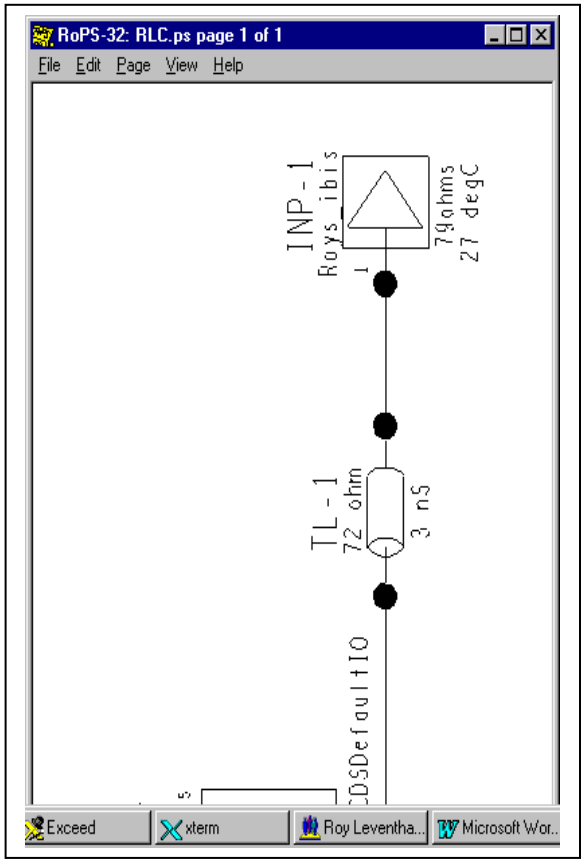


Figure 35: Reflections @ dV/dt = 3/0.9nS
Minimum Pin Parasitics R, L & C

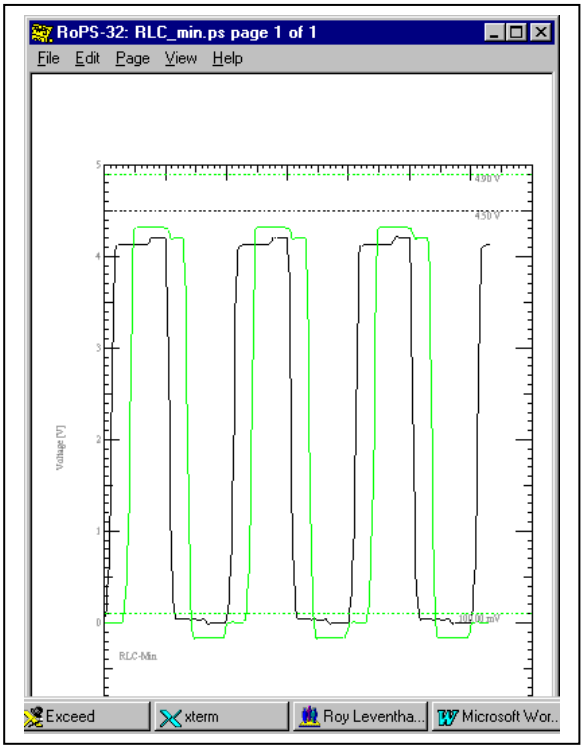


Figure 36: Reflections @ dV/dt = 3/0.6 nS
Typical Pin Parasitics R, L & C

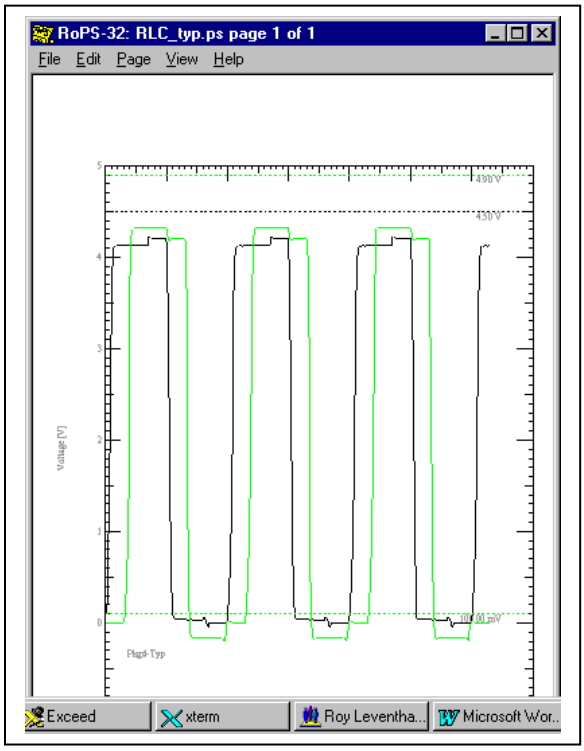
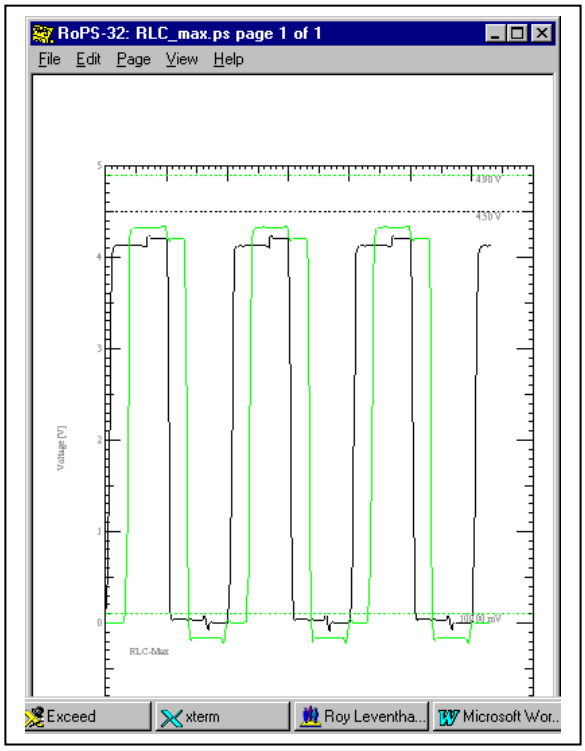


Figure 37: Reflections @ dV/dt = 3/0.48 nS,
Maximum Pin Parasitics R, L & C



The effect of package parasitics and ramp rate and +/- 10% changes in their values worst case can easily be seen in the above simulations. But in this example, at the edge rates and RLC magnitudes relative to Zo, they are not a major source of signal integrity problems.

The actual values varied in these simulations were:

	typ	min	max
R_pkg	0.071	0.063	0.077
L_pkg	4.1nH	3.8nH	4.6nH
C_pkg	0.9pF	0.8pF	1.1pF

	typ	min	max
dV	3.0	3.0	3.0
dt	.6nS	.9nS	.48nS

Environmental Variables

Voltages, Temperatures, and Population Spreads

Power supply voltage and ambient temperature are not independent, accessible variables with regard to the IBIS model. This is discussed in the next two sections. However, when running a simulation on an open pullup part you can connect a resistor to a pullup voltage and vary the voltage. This has validity and meaning. As well, you can place various termination networks in your topology and vary power supply voltage and other variables as valid modifications.

Another place that you will see voltage, temperature, etc., is in the IBIS model properties such as Vol, Voh, Vil, Vih, the V-I curves, slew rates, etc. It is important to recognize that in most instances these parameters represent statistical data of the population distribution run at a combination of variables that will result in worst case results.

The Min and Max V-I curves, for instance, are supposed to incorporate the corners of the process population distribution as well as the effects of temperature and voltage. That is, the effects of temperature and voltage over the range of the [Temperature Range] and [Voltage Range] Keyword variables of the IBIS model.

While min and max data may appear and may have been taken with worst case population units plus temperature conditions plus supply voltage, etc., there is no direct connection between setting those parameters in the simulator and the results predicted. All such effects are subsumed under the min-typ-max (sometimes called fast-typical-slow by the simulator software) data given in the model as typical behavior and worst case behavior. Whatever conditions of high-voltage, low-temperature, process-corner result in those worst case responses – they are subsumed under the min-typ-max data. There is no way to "de-embed" the effect, say, of the temperature variable and extend the range or precision of its predictions.

Example: Drive Capability – "Worst-Case"

In the following three figures are the simulated results of driving a perfectly terminated Zo = 72 ohm transmission line using "Fast-Typical-Slow" versions of the default model. For a line with no reflections such as this we can understand "Fast-Typical-Slow" to mean Min-Typ-Max. We correspondingly get Vout = 4.05 V, 4.13 V and 4.20 V respectively. These differences are due to minimum currents being about 5 ma less and maximum currents being about 5 ma more than the typical case. The Min-Typ-Max values are supposed to represent a worst case combination of varying temperature, supply voltage and population sample.

This set of curves reflect a case where the driver Zout is small with respect to the line impedance, Zo. About 18 ohms vs. 72 ohms, but, not an insignificant percentage of Zo.

See “Scaling V-I and Simulating the Results,” for more discussion on this

Figure 9x: Slow Default Model driving 72 Ohms – Vout ≈ 4.05 V

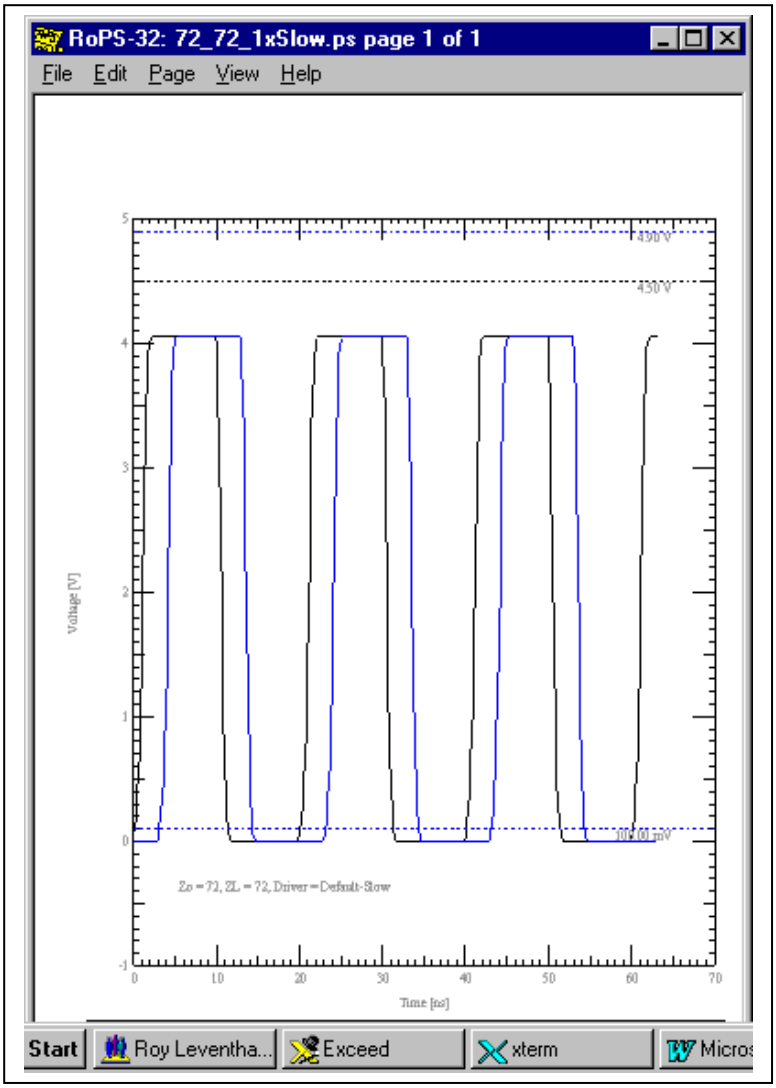


Figure 9x: Typical Default Model driving 72 Ohms – Vout ≈ 4.13 V

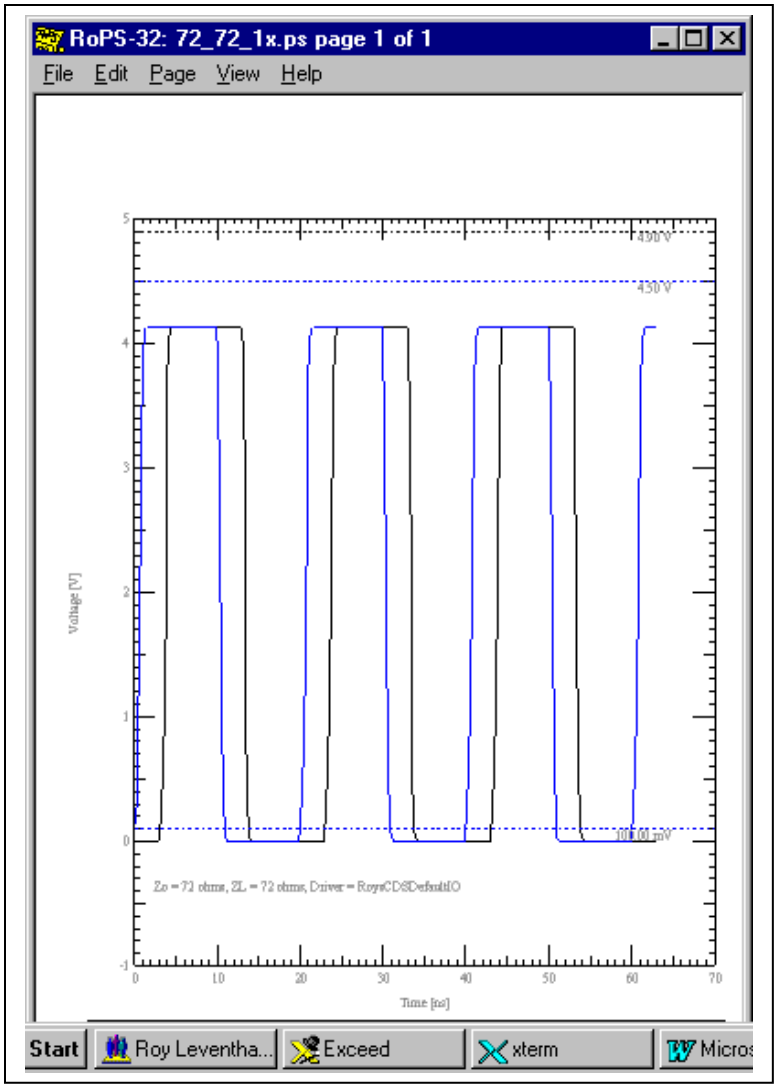
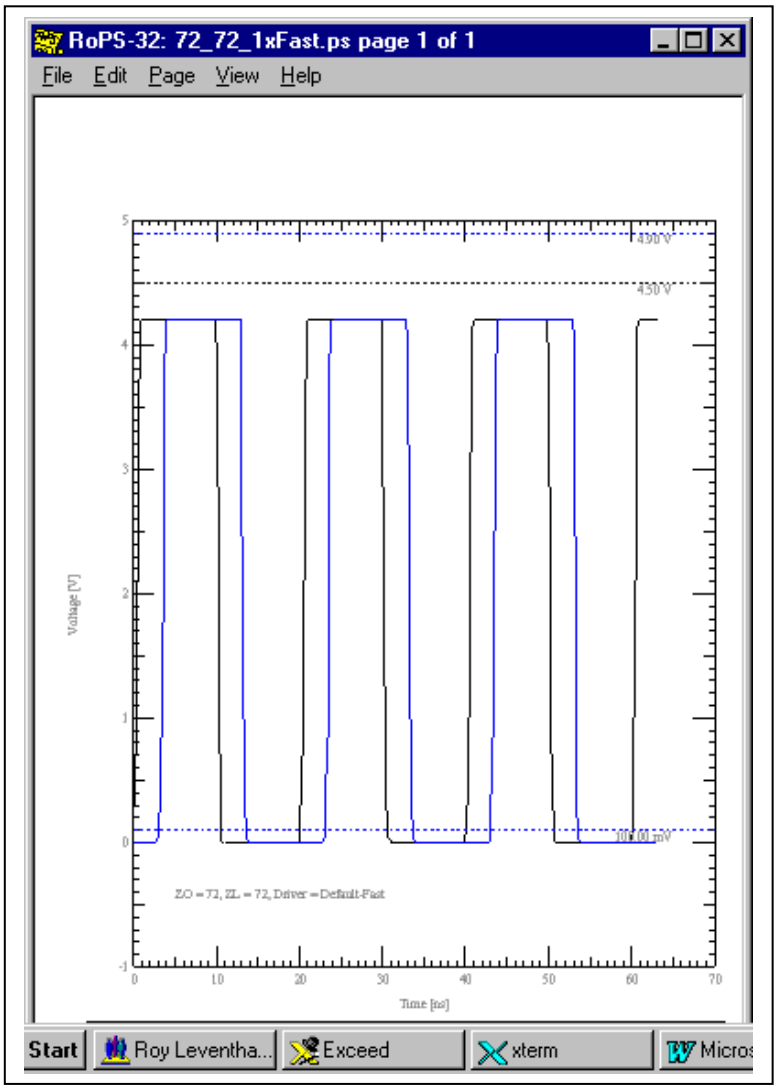


Figure 9x: Fast Default Model driving 72 Ohms – Vout ≈ 4.2 V



Can the Effects of Environmental Variables be Simulated?

In a word – no. The Min-Typ-Max values on the V-I and V-T curves, parasitics and other parameters already have embedded in them worst case conditions including variations in voltage, temperature, process, etc. The voltage, temperature and other variables are not directly accessible and variable, per se. Some simulators may add features and "wrappers" that make their capabilities look different. But, remember that IBIS is not a "physical" model like SPICE where temperature, voltage, etc., are directly accessible variables.

IBIS is a behavioral model. Vcc can be varied by +/- 5% if the data is in the model. But, not by much more. You cannot, for instance, simulate a 5 V model of a part at 3.3 V and get meaningful results. The V-I curves and other data will default back to the 5 V values. To simulate at 3.3 V you have to generate a 3.3 V model.

Interactions Between the IBIS Model and Board Simulators

Introduction

This entire section is written to provide greater insight into IBIS models and what should be included in a given IBIS model. This section does not address the IBIS model directly. But, the "end game" of using an IBIS model is to simulate signal integrity issues inherent in boards, connectors and cables.

To illustrate: If I assert that output pulse frequency content is a critical consideration for signal integrity, unless I have some understanding of transmission lines, reflection coefficients and capacitive loading, I will not have much understanding of the merits of the various elements of the IBIS model.

SPICE Model Elements and SubCircuits

If there are SPICE subcircuits attached at the driver/receiver points their currents will be solved for in and summed with the driver/receiver currents. Most of the core circuit simulation engines used in signal integrity analysis are SPICE based. All component structures are assumed to be lumped element SPICE solvable circuits connected by transmission lines as appropriate.

The behavior of the transmission line sections joining these SPICE circuits are handled by a type of analysis known as transmission line reflection diagrams.

Time and Frequency

An amplitude Vs time electromagnetic (EM) waveform can be observed when an observer with an oscilloscope probes a point on a circuit. As well, this EM wave propagates along the interconnects from its point of injection to other points in the system.

We do not directly experience the frequency domain, or dimension, as we do with space (distance) and time. But, to gain understanding about traveling EM waves, transmission lines, and how the IBIS model is applied, we often need to convert our thinking to the frequency domain. The mathematics that helps us make the transition back and forth between a time waveform and its frequency spectrum is called Fourier Analysis.

What we are aware of as electrical engineers is that any time waveform can be thought of as a particular combination of a set of discrete frequency sine/cosine time waveforms of various amplitudes and phases.

Once we make this transition we can start to answer how reactive impedance, characteristic impedance, skin effect, losses, dispersion, wavelength, reflections, electromagnetic radiation and interference and many other properties behave with frequency. And, having found these answers in the frequency domain, we can construct what is going on in the time domain.

Time, Distance and Frequency: Lumped and Distributed Circuits: Transmission Lines

We all recall that the velocity of an electromagnetic (EM) wave in air, or in free space is (denoted by):

$$C = 3 * 10^{10} \text{ cm/sec} = v_p$$

This wave may be partially reflected and partially absorbed (transmitted, received) when it reaches another medium or discontinuity, as in radar and microwaves.

If the time delay for the reflected wave to arrive back at the signal source is very short, the time element drops out except in terms of charging capacitors, establishing currents in inductors and turning devices on and off. Everything happens "instantaneously" around a given circuit and the mathematics for analyzing what is happening simplifies. Ohm's law and Kirchoff's Laws become the basis for thinking about what is happening. Such circuits are called "lumped" because their properties and behaviors can be modeled in terms of *lumped* elements (resistance (R), inductance (L), conductance (G), and capacitance (C)) with connections between them as ideal, property-less wires.

Propagation velocity and distance give the time, t_p, for an electromagnetic wave to travel from source to receiver:

$$t_p = \text{Distance}/ v_p = D/ v_p$$

For any reflected wave to arrive back at its source it takes twice t_p to make the round trip.

But, an electromagnetic wave will not normally propagate into a dielectric (non-conductive) medium. To do so, for starters, the source would have to begin acting like an antenna. This begins to occur when the dimensions of the source become 1/10 of a wavelength, λ , long or longer. The length of an electromagnetic wave, λ, in a propagating medium is:

$$\lambda = v_p /\text{frequency} = v_p /f$$

For lower frequencies electromagnetic propagation is pretty much confined to conducting wires and R, L, G, and C elements. At higher frequencies (radio, microwave, x-ray, etc.) wave propagation is chiefly by radiation of the wave. Losses in conducting elements become high at these higher frequencies and one is better off without such elements. Energy can propagate through dielectrics and / or be guided by dielectric structures very effectively until the x-ray frequencies.

In between the lumped element region and the pure radio/microwave region lies the region of chief interest to signal integrity engineers. In this region electromagnetic energy is normally guided along conductive paths but, is strongly affected by the properties of the surrounding dielectric medium. The connections that guide the energy become known as *transmission lines*. The energy penetrates only a short distance into (skin effect) the conductor and the electric and magnetic fields appear chiefly in the surrounding dielectric. The transmission lines can be modeled as lossless at the lower frequency end of this range and lossy at its higher end.

Propagation in a dielectric medium slows down. This fits intuitively because the electric property (per unit volume) of capacitance is increased substantially over that of air. And, there is a time delay associated with energizing and de-energizing this element. Expressing this in a dielectric:

Velocity becomes lower

$$v_{p`} = C/(\varepsilon)^{1/2}$$

where ε = dielectric constant

or, as we should really consider for our dielectric medium:

$\varepsilon = \varepsilon_{r`}$ = effective relative dielectric constant

where effective, $\varepsilon_{r`}$ refers to whether the field lines are in dielectric (stripline) or partially in dielectric and partially in air (microstrip), and relative, ε_r, refers to relative to air

and, time delay increases

$$t_{p`} = D/ v_{p`} = (D/ v_p) * (\varepsilon)^{1/2} = t_p * (\varepsilon)^{1/2}$$

Working out the numbers:

$t_{p`} = 85 * (\varepsilon_{r`})^{1/2}$ picosecond per inch

and, wavelength becomes smaller

$$\lambda` = v_{p`} /f = v_p /(f * (\varepsilon_{r`})^{1/2})$$

Our discussion of time, distance, frequency and wavelength leads to this question: what is the *electrical length* of a waveform feature that will start to interact according to transmission line behavior with the physical length of an interconnect? The electrical length of a feature, for example t_r, is given by:

$L = t_r / t_{p`}$

Systems small enough for all points to act in near unison to an EM wave traveling on them are lumped systems. Systems larger than that are transmission line systems. Reflections and re-reflections that travel back and forth on these systems can become a problem because they introduce noise on the waveform being transmitted.

The rule of thumb[(2)] is that systems with:

$D <$ (waveform feature electrical length)/6

Assuming a stripline with $\varepsilon_r = 4.6$ we get $t_{p`} = 182.3$ pS/inch.

With a $t_r = 1$ nS we get L = 5.5 inches.

A trace length of 915 mils is expected to begin behaving like a transmission line.

Note 2: My reference for this rule of thumb is found on page 7 of:

"High Speed Digital Design: A Handbook of Black Magic"
H. W. Johnson & M. Graham
Prentice-Hall c1993
ISBN 0-13-395724-1

Transmission Line Equations

Using the methods of differential calculus (the transmission line is divided into more and more, smaller and smaller sections until we can approximate its distributed behavior with cascaded lumped constant RLGC sections) we can derive:

$$Zo = [(R + X_L)/(G - 1/X_C)]^{1/2}$$

With $X_L = j2\pi fL$ and $X_C = 1/ j2\pi fC$

For a lossless transmission line this simplifies to:

$$Zo = (L/C)^{1/2}$$

Where L and C are the *per unit length (or, per unit volume in a 3-D sense)* values of the transmission line inductance and capacitance.

Since we have already said that capacitance per unit volume in a dielectric medium increases it follows that microstrip Zo for a given structure is higher than for stripline. This is because only part of the EM wave travels through dielectric.

The reflection coefficient, ρ, expresses the fraction of the amplitude of an incident wave that gets reflected at a load or discontinuity. The load can be the intended receiver or the driver when a

reflected wave reaches it. The load consists of the receiver (driver) – usually its C_comp - and any bias or terminating resistances. Discontinuities can be vias, corners, different Zo on different layers, stub T-points, etc.

$$\rho = (Z_L - Zo)/(Z_L + Zo)$$

Transmission Lines, Reflection Diagrams, and the Bergeron Method

The transmission lines are solved and simulated by closed form algorithms, or in the better simulators, a field solver. “2-1/2 D” field solvers are most common. These simulators solve and derive the RLGC matrices for a given section of interconnect where physical dimensions, etc. do not change. One of the “end products” of this geometrical field solution of a transmission line is its characteristic impedance, Zo. Other “end products” are coupling impedances to neighboring lines, propagation velocity, and propagation time.

The concept of a reflection diagram solution of the reflections and re-reflections at each end of a transmission line with linear terminating impedances, and the sum of these reflections on the line, is fairly straightforward and familiar to most people with an electrical engineering background. The Bergeron method adds the (pre-computer, pre- data-table-lookup and interpolate) capability of constructing such a diagram for non-linear terminating impedances on the line. The non-linear driver and receiver impedances are represented by their V-I characteristics. The transmission line impedance is represented as a load line on the V-I graph and the resultant voltage/current and reflected waves can be found. This load-line method of solution given non-linear device characteristics is also familiar to most people with an electrical engineering background.

Mr. Bergeron gets the credit for developing this technique. He was a mechanical engineer who needed to analyze the water-hammer effect in steam lines well before the era of electronics. For an excellent explanation of the method applied to transmission lines, see:

Ch. 4: Non-Linear Sources and Loads
“Transmission Lines in Computer Engineering”
Sol Rosenstark
McGraw-Hill c1994
ISBN 0-07-053953-7

Computer based digital simulators do not use these graphical methods directly. The IBIS model and its data-tables of V-I points to look up are combined with SPICE solutions for the total effects – transmission line, termination network, bias resistors, etc. - at a node being analyzed. Some simulators also modify the transmission line RLGC matrices with frequency when instructed to and thus deal with dispersion and loss effects.

Timing Simulations

Timing information is not “officially” supported in IBIS. However, most simulators take the IBIS data and are able to simulate First Switch, Final Settle, Noise Margin, Buffer Delay, Propagation Delay and Clock Skew with it.

Timing driven simulations references timing information from logic simulations to determine the sequence of how a set of drivers turns on and off. This information improves the simulation results for **crosstalk** and **Simultaneous Switching Noise**, SSN (ground bounce), rather than the simplistic assumption that all drivers turn on and off together. This will probably give an absolute worst case induced switching noise result. But, this approach can end in an overly conservative design.

Board Parameters

Board properties and parameters are not a part of IBIS. They can be very much a part of a particular signal integrity simulation you are running. Dielectric constant, etch width, board stackup, resistor value (SPICE), none of these are part of IBIS.

RLGC Matrices

An RLGC matrix can easily represent the electrical model of any linear structure. This structure could as well be a transmission line, a connector, or the wire bond or other connections inside an IC package. Such a matrix can be a so called “single-line” model or one that includes the mutual coupling effects between package interconnects, or connector pins, or a group of etch lines on a board. Again, IBIS does not officially support the G matrix.

A full discussion of RLGC matrices only applies to the parasitics information included in an IBIS model under the [Package Model] keyword.

In general, you should know that in the [Package Model]:

- The matrix will be N x N where N = number of pins in the package.
- All the self-parasitic elements, a_{ii}, are on the main diagonal.
- All the mutual-parasitic elements, a_{ij} or a_{ji}, are off the main diagonal.
- In a large package with many pins Banded_matrix and Sparse_matrix are used to limit the complexity.
- The G matrix (leakage conductance) is usually zero or approximated to zero since its elements are usually extremely small.
- Most matrices will be symmetric and thus easier to mathematically manipulate.
- The R, L, G, and C matrices are combined into an impedance, Z, matrix by the simulator.

It is the impression of this author that the effects of most pin self-parasitics are inconsequential above 1 nS edge rate and most mutual pin-parasitics are inconsequential above 200 pS edge rate.

Algorithms & Field Solvers

I use the term algorithms to loosely refer to results calculated from a formula as opposed to, in this instance, first solving for the electromagnetic field in a transmission line structure and then extracting the RLGC parameters of the field solutions on the way to finding Zo, etc.

Field solution results have been shown to correlate within a percent or two to measured results. Using algorithms often results in answers that are off by +/- 5% and sometimes as much as much as +/- 10% - depending on whose algorithm you use. Obviously, it's better to use a simulator that uses only a field solver for extracting Zo, propagation velocity, coupling capacitance, etc.

The formulas that were developed to solve PCB transmission line structures stem mostly from empirical curve fitting research that started in the early 1950's. This was before the availability of sufficient computing power to go the field solution route. The chief problem with their accuracy is that they only apply to situations that closely match those in which the curves were generated.

Most simulators use a "2-1/2 D(imension)" field solver to save on computing time and power. The field is solved in two dimensions in an x-y plane perpendicular to the etch in this approach. The velocity of propagation along the etch is then used in the z-direction. This limits the solver's accuracy at etch transitions, corners, vias, split planes, etc. But, this is not of practical concern until you begin to switch at edge rates in the 100 pS range or less.

Vinl, Vinh, Vol & Voh

You need to have a clear understanding of the meanings of properties such as Vol, Voh, Vil and Vih. These variables are defined as:

Vol: The maximum driver output low voltage. All units are supposed to have a low state output voltage no greater than Vol. Vol should be defined for a given current.

Voh: The minimum driver output high voltage. All units are supposed to have a high state output voltage no less than Voh. Voh should be defined for a given current.

Vil: The minimum receiver input threshold voltage. All units are supposed to require an input threshold switching voltage of at least Vil. If you rise above this value some units will switch on.

Vih: The maximum receiver input threshold voltage. All units are supposed to have switched on by or below this voltage. If you drop below this voltage some units will turn off.

Note that a device at the low state will have inputs below Vil and at the high state will have inputs above Vih. The actual switching will occur between Vil and Vih.

Note that Vil and Vih are IBIS model properties while Vol and Voh are not. Yet, most simulators will make use of Vol and Voh in calculating noise margins and switching speeds.

Switching Delay Times

Three popular switching delay measurements for determining if a physical implementation will meet its logic timing constrains are buffer delay, propagation delay, first switch delay and final settle delay.

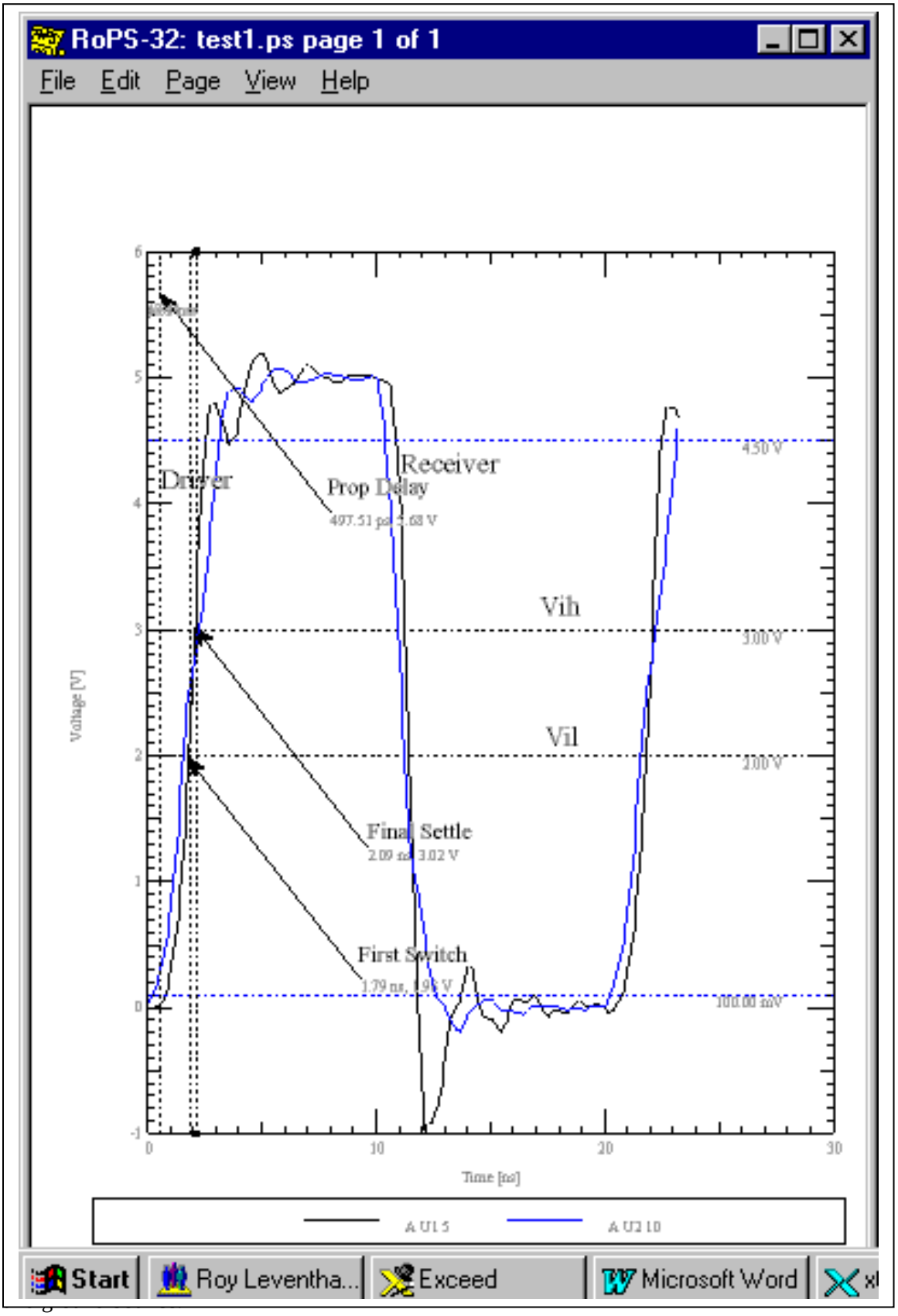
Buffer Delay	Buffer delay is the time for a driver to turn on or off, otherwise known as rise and fall times. The IBIS spec contains dV/dt or equivalent data that inherently gives buffer delay. See comments below about buffer delay.
Propagation Delay	Transmission line wire delay. The time required for a wave to propagate from driver to receiver. Propagation delay is usually arrived at by taking the absolute value of first switch delay and compensating for buffer delay. “Compensating for” means subtracting from.
First Switch Delay	First switch delay is found by measuring the delay when an input rising (falling) wave first crosses the Vil (Vih) threshold. This is when a few members of the receiver population will first be able to switch high (low) at their output. First switch measurements reported by the simulator may be compensated or uncompensated (absolute).
Final Settle Delay	Final settle delay is found by measuring the delay to when an input rising (falling) wave last crosses the Vih (Vil) threshold. This is when the last few members of the receiver population will finally be able to switch high (low) at their output and stay there. Final settle switch measurements reported by the simulator may be compensated or uncompensated This measurement is especially sensitive to ringing that causes the input waveform to cross and re-cross the Vih (Vil) threshold.

Comments on buffer delay:

The rise/fall time of a buffer will vary according to the definition of how it is being measured. Is it being measured per the traditional 10% - 90% rule, the IBIS 20% - 80% guideline, a manufacture’s “transition time” definition of 30% - 70%, or what?

Secondly, the actual rise/fall of a buffer will vary according to the loading placed on it. IBIS provides for supplying rise/fall test fixture data which the simulator can use to adjust buffer delay to correlate with the loading conditions you are actually simulating Vs the test fixture the rise/fall data was taken (simulated) in.

Lastly, rise/fall time looses its credibility if the measurement is taken on a badly ringing waveform. It can be next to impossible to verify measurements under such conditions.


RoPS-32: test1.ps page 1 of 1
File Edit Page View Help
Driver
Receiver
Prop Delay
497.51 ps, 2.68 V
Vih
3.00 V
Vil
2.00 V
Final Settle
2.09 ns, 3.02 V
First Switch
1.79 ns, 1.98 V
100.00 mV
4.50 V
Voltage [V]
Time [ns]
A U1 5
A U2 10
Start
Roy Leventha...
Exceed
Microsoft Word

Crosstalk is the EM energy that can get coupled between neighboring transmission lines. In digital circuits it is unwanted because it is an additional source of noise on the digital signal of the victim line. The coupling impedances, the frequency and amplitude of the aggressor signal and the characteristic impedances of the coupled lines primarily control crosstalk. The board geometry and stackup primarily set up the coupling and characteristic impedances. Switching sequences and logic states affect the final results with groups of mutually coupled nets affecting each other. As such, it would seem for crosstalk to not involve the IBIS model that much.

But, remember that the IBIS model sets up the starting conditions of the pulse waveform: its amplitude and frequency content. As well, reflections and clamping are simulated per the IBIS model. Reflections are a second order effect that can be important at times. Reflections represent high frequency energy that is bouncing around in the circuit, not being efficiently absorbed in the load, and presenting additional interference with the victim line.

Ground (power) bounce represents the additional source of noise interference that can come from ground not remaining at zero volts and Vcc not remaining at its voltage. Which I/O cells are affected by which bouncing voltages depends on the bus structure and connections inside the IBIS package. As well, switching sequences and logic states affect the results.

As switching speed and power goes up, voltages begin to be induced on points that are ideally supposed to have zero impedance and zero parasitic effect. So, if the path to true ground contains sufficient R, L, and C in series we get:

$$V\_bounce = R(i) + L(di/dt) + 1/C(\quad i*dt)$$

In this equation consider the switching current to be the forcing function and V_bounce to be the response.

Of course, the exact form of the V_bounce equation will depend on what parasitics are present in the ground (power) return path and how they are connected. The portion of V_bounce that is due to the etch connection to the power and ground planes and the parasitics of those planes themselves is outside the scope of IBIS. The V_bounce due to the POWER and GND pin parasitics of the IBIS model are accounted for if that data is present.

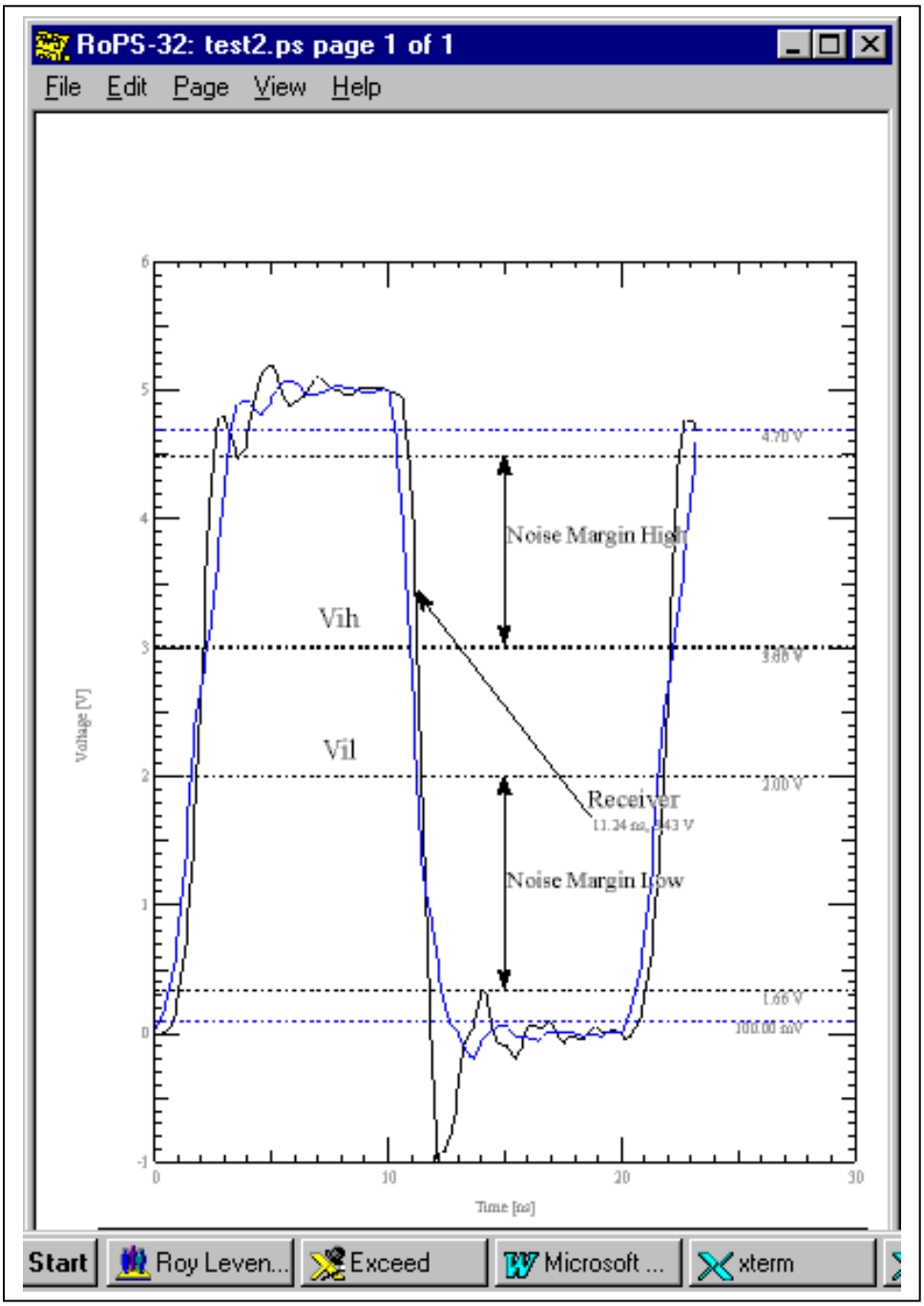
How to Tell if the Effect of C_comp is being Double-Counted

The simulation of an IBIS driver model is supposed to include the effect of C_comp in the slew rates and/or V-T curves as mentioned previously. Varying C_comp is supposed to have no effect on driver rise and fall as opposed to adding a variable capacitance across the output.

One way to test this is to reduce C_comp to zero and then raise it to a very large value, say 50 pF. C_comp may be getting “double counted” in a simulation if you see the buffer rise time change as it is varied. If you see such results, I suggest you simulate the rise and fall time of the IBIS test fixture with the correct values of C_comp entered in Min-Typ-Max to see if you reproduce the IBIS slew rate and/or V-T curves. Remember to use the IBIS values, as the data sheet values are usually “guard-banded.”

Noise Margin

Noise margin is one of the most important considerations in signal integrity. It is a measure of the difference between an input switching threshold and the voltage on the input pin that may cause that input to switch again – unwantedly, due to noise on the waveform. There are two noise margins: Noise Margin High (logic state) corresponding to Vih, and Noise Margin Low (logic state) corresponding to Vil.



Verifying Your Simulator and IBIS Model

The switching speed data simulated for verification purposes below was done with ramp rate data. V-T curve data could as well be used.

Rise and Fall Times, Slew Rates, V-T Curves and Verification

The definition of Rise and Fall Times and Slew Rates and Switching Speed has already been covered in "Ramp Data: Slew Rate."

Rise and Fall Times – IBIS - 20% - 80% of the full output Final Settle swing by definition.
Rise and Fall Times – Traditional - 10% - 90% of the full output Final Settle swing by definition.

Again, rise times are defined for a given R_load in IBIS. The data sheet often does not assume a particular R_load except for ECL technologies.

Rise and Fall times for an IBIS model can be verified for an IBIS model by simulating the Data Sheet test fixture with the IBIS model and seeing that the measured rise and fall correspond to the data sheet. Sometimes you will want to have the simulator report the rise and fall time directly. At other times you will want to measure the rise and fall times off the simulator waveform curves. Many simulators have cursors that can be placed on the waveform to aid in this process. What follows is an example of measuring the rise and fall off the waveforms.[1]

The waveform measurement method was chosen in the following example to contrast the IBIS and traditional (data sheet) methods of defining rise and fall times. In the case in point, yet a third definition (1.3 to 1.8 V output or about 30% to 70%) was used on the data sheet for a measurement called "t_t" or "transition time" with a Min = 0.5 nS and a Max = 4.5 nS. In addition, it was already known that the IBIS model had been corrected to give slightly slower rise and fall times after the data sheet had been published. Still, the simulations provide a good reality / verification check.

When the model is simulated to yield the rise and fall times you usually have to make any "transmission line" to the receiver, i.e., "test instrument" so short as to not be a transmission line. That is so no reflections of any consequence would be set up as in a real test fixture.

The network topology is shown first. After it a pulse waveform is shown that was actually taken after the series of rise and fall waveforms was run. This latter simulation was taken with package parasitics added in and the pullup current adjusted to 100 ma (pullup resistor = 11 ohms) instead of the 71 ma of the switching test fixture. The very slight amount of rising edge ringing is due to these extra parasitics and pullup current.
This amount of ringing is not significant.

But, the Vol changed from 0.93 V to 1.0 V when the pullup current was changed from 71 ma to 100 ma. The data sheet calls for a Vol of 1.15 V Max at 100 ma. In this instance the manufacturer

[1] The purpose of the Ramp or the V-T tables is to produce the best information for simulating the device, not specifying it. The lower impedance R_loads are of the same order of magnitude as real transmission lines. Information under these conditions is more accurate for the purposes of simulation.

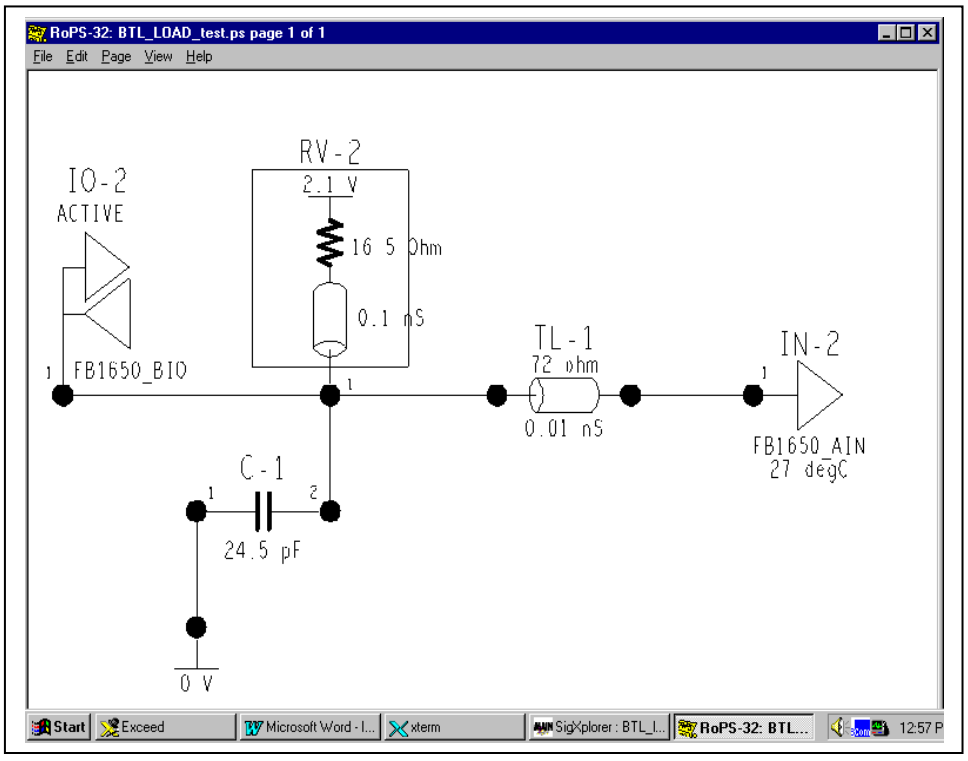
"guard bands" the values in the data sheet. This can lead to instances of overly conservative design if not understood correctly.

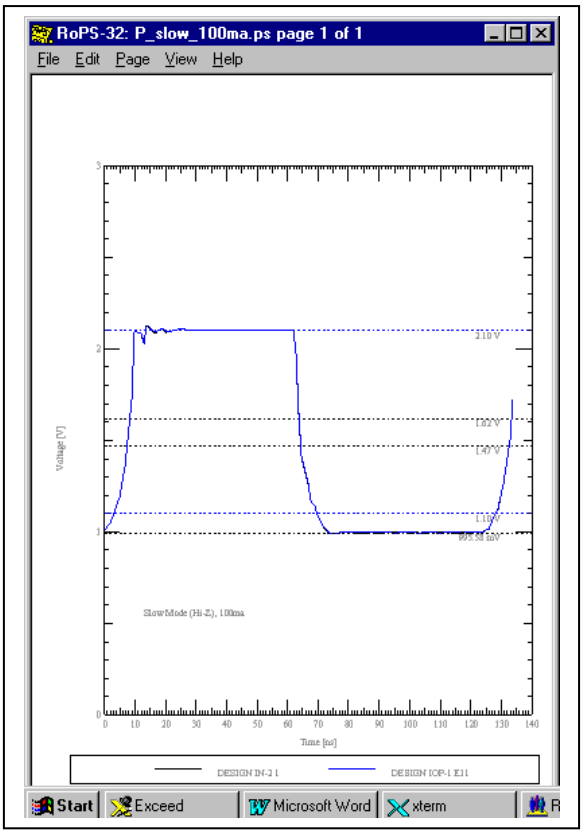
The difference between 1.47 V and 1.15 V vs. 1.47 and 1.0 V represents a 47% increase in low state noise margin. Is the IBIS noise margin real? This, and related issues, are investigated in the next section: V-I Curves.

The 72 ohm transmission line in the topology below is so short that it need be considered as a wire connect only and not possessing a property called characteristic impedance. This wire is connected to a high impedance input receiver which presents only its input capacitance, C_comp (= 5.5 pF Min, Typ, & Max) to the circuit. The end result is to present a lumped discrete SPICE circuit to the simulator consisting of 30 pF to ground and a 16.5 ohm pullup resistor to 2.1 V, driven by the 1650.ibs model. This reproduces the data sheet t_t test fixture.

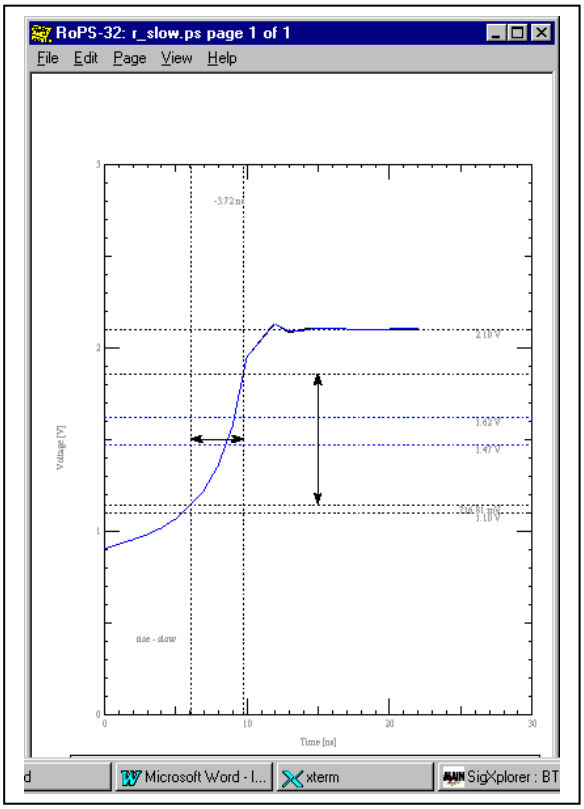
t_t = "Transition Time"
= rise/fall from 1.3 V to 1.8 V
= rise/fall from 30% to 80% of the output swing
= yet another definition of rise/fall time different than either the IBIS or the traditional definition

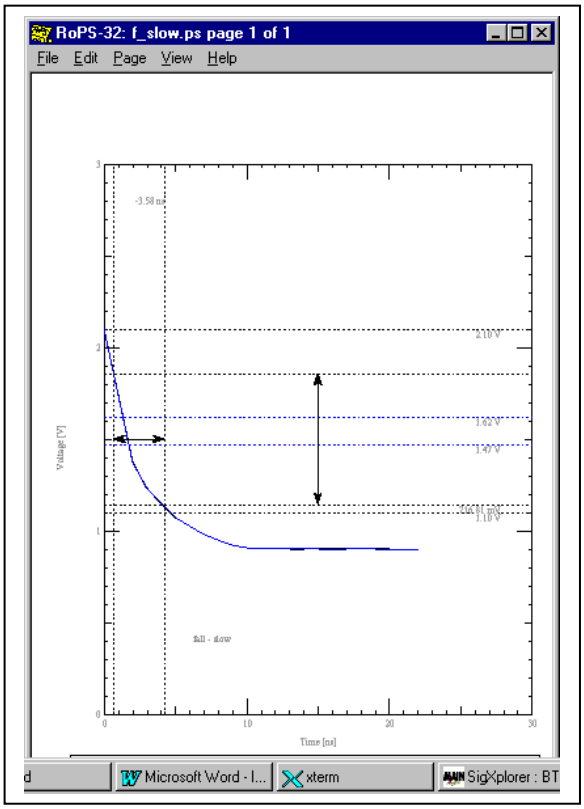
Transition Time t_t: Test Circuit From Data Sheet



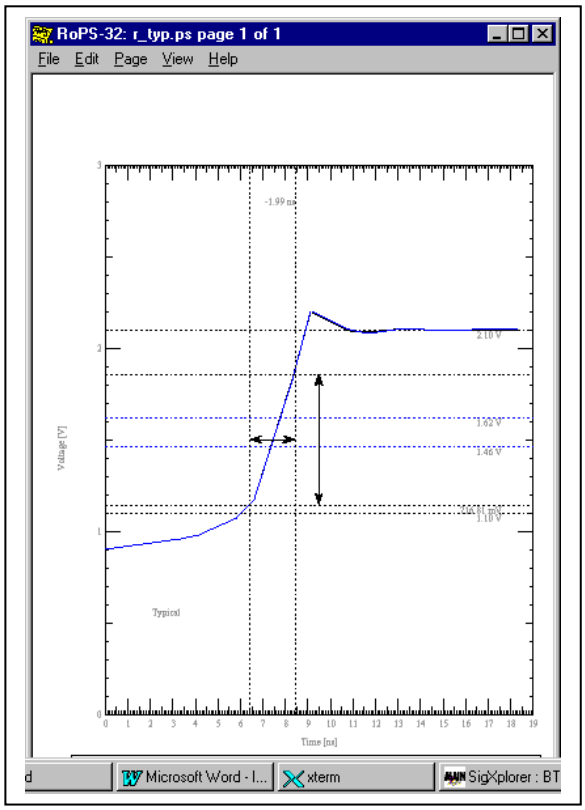

RoPS-32: P_slow_100ma.ps page 1 of 1
File Edit Page View Help
Voltage [V]
Time [us]
SlowMode (Hi-Z), 100ma
Start
Exceed
Microsoft Word
xterm

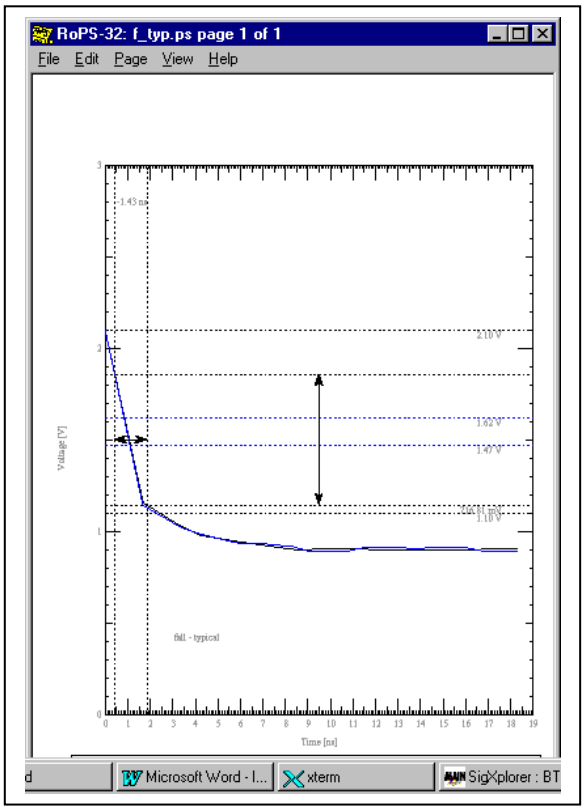
Slow Rise and Fall: IBIS



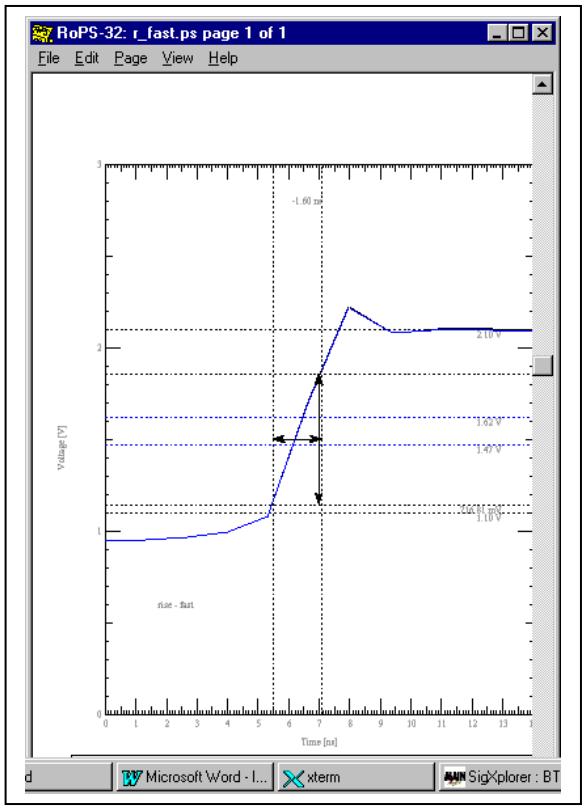


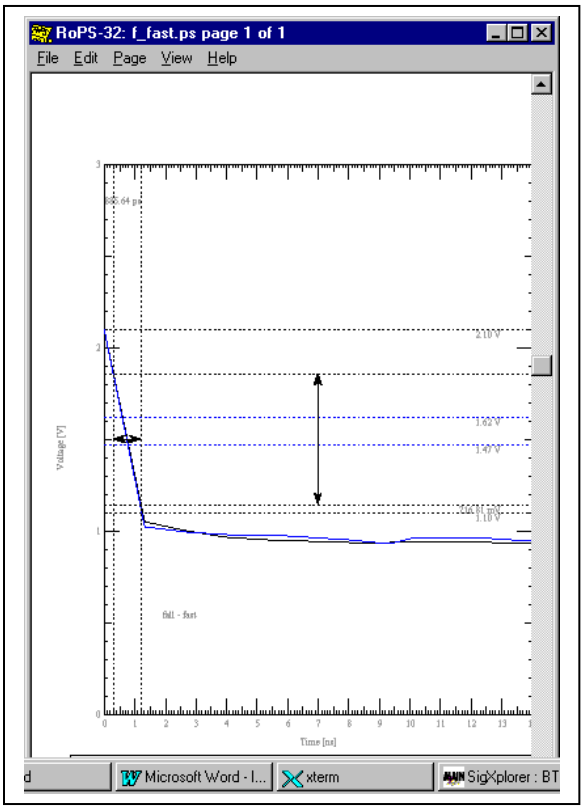
Typical Rise and Fall: IBIS



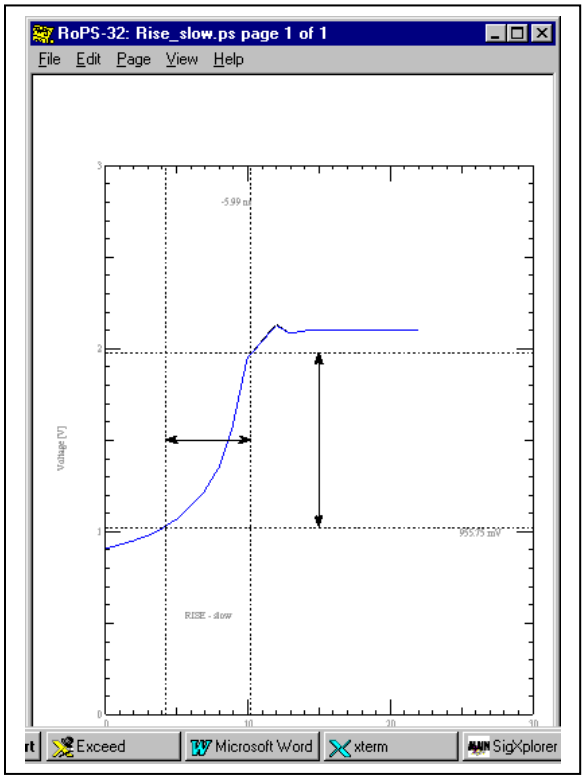


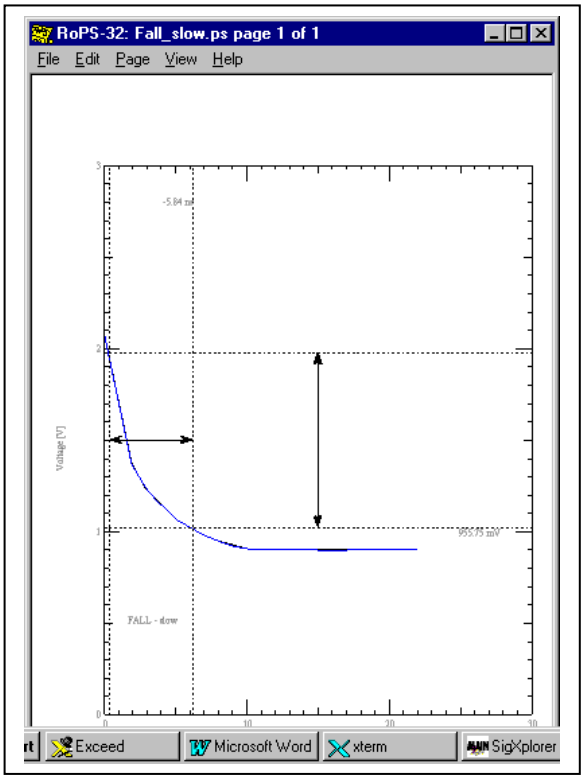
Fast Rise and Fall: IBIS



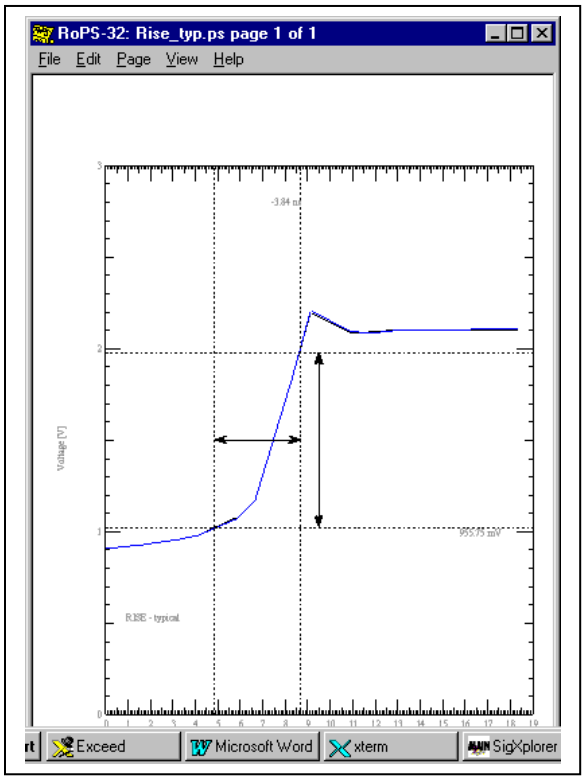


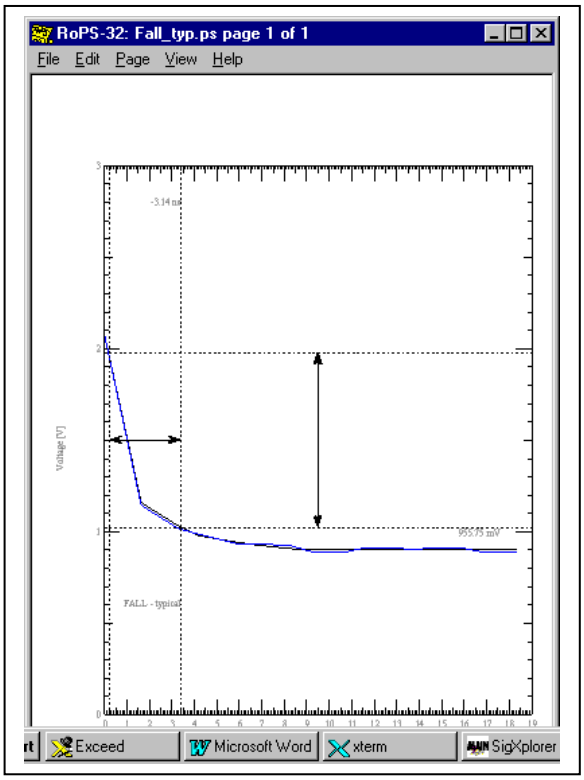
Slow Rise and Fall: Traditional



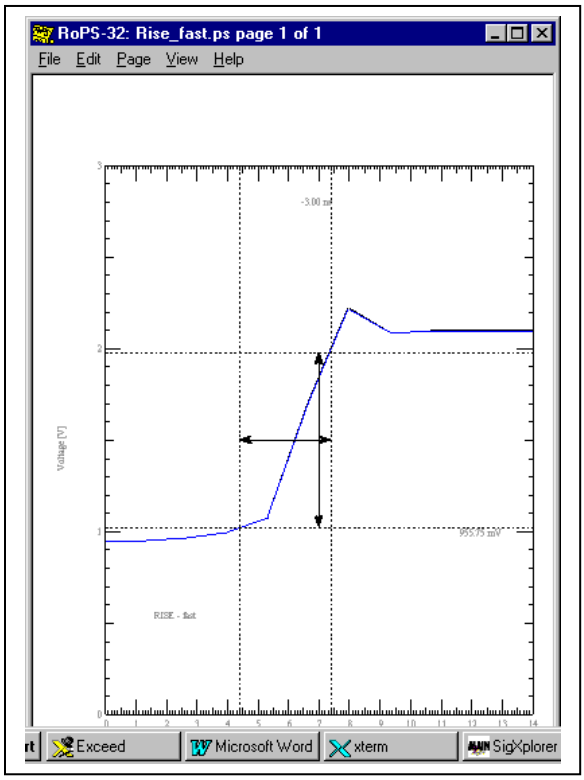


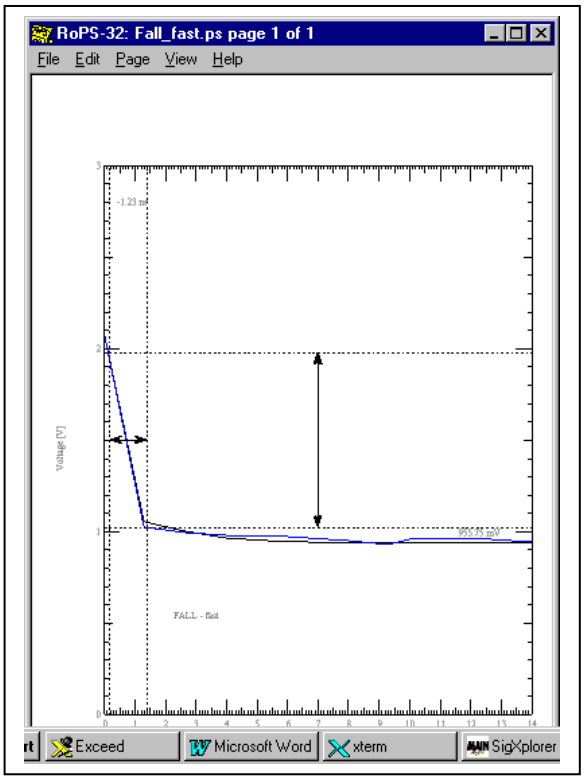
Typical Rise and Fall: Traditional





Fast Rise and Fall: Traditional



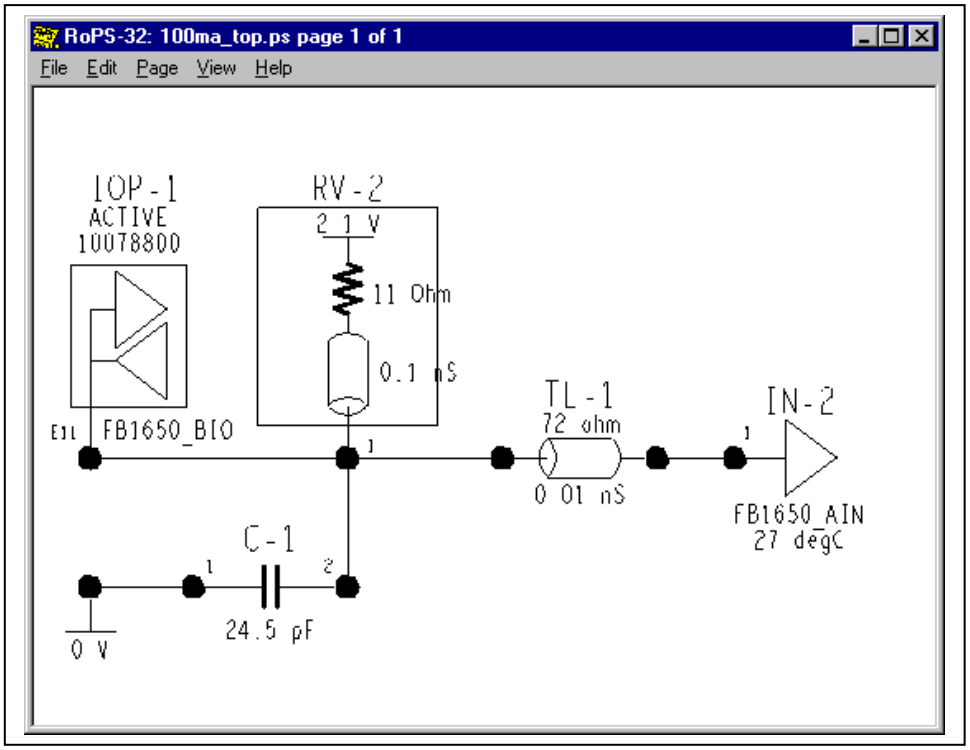


Simulation Model Mode	Simulated and Measured IBIS Model				Data Sheet	
	20%– 80% IBIS Definition		10% - 90% Definition		1.3 – 1.8 V - t_t - 30%-70% Definition	
	Rise (nS)	Fall (nS)	Rise (nS)	Fall (nS)	Rise (nS)	Fall (nS)
Slow	3.72	3.58	5.99	5.84	1.99	1.69
Typical	1.99	1.43	3.84	3.14	1.23	.913
Fast	1.6	.886	3.0	1.23	1.16	.627

V-I Curves and Verification

Following is one example of verifying V-I type information. More will be added as they are developed.

The following circuit was simulated to verify a manufacture's data sheet value of Vol Vs the results of using their IBIS model. A pullup current of 100 ma was established to match their data sheet conditions. The transmission line to the receiver was kept too short to enter into the result thus duplicating the discrete test fixture conditions. In arriving at the 11 ohm pullup resistor value to set up the 100 ma current the simulator was used to "empirically" arrive at the result. This is expected, since the output is not represented by a simple, linear saturation resistance.



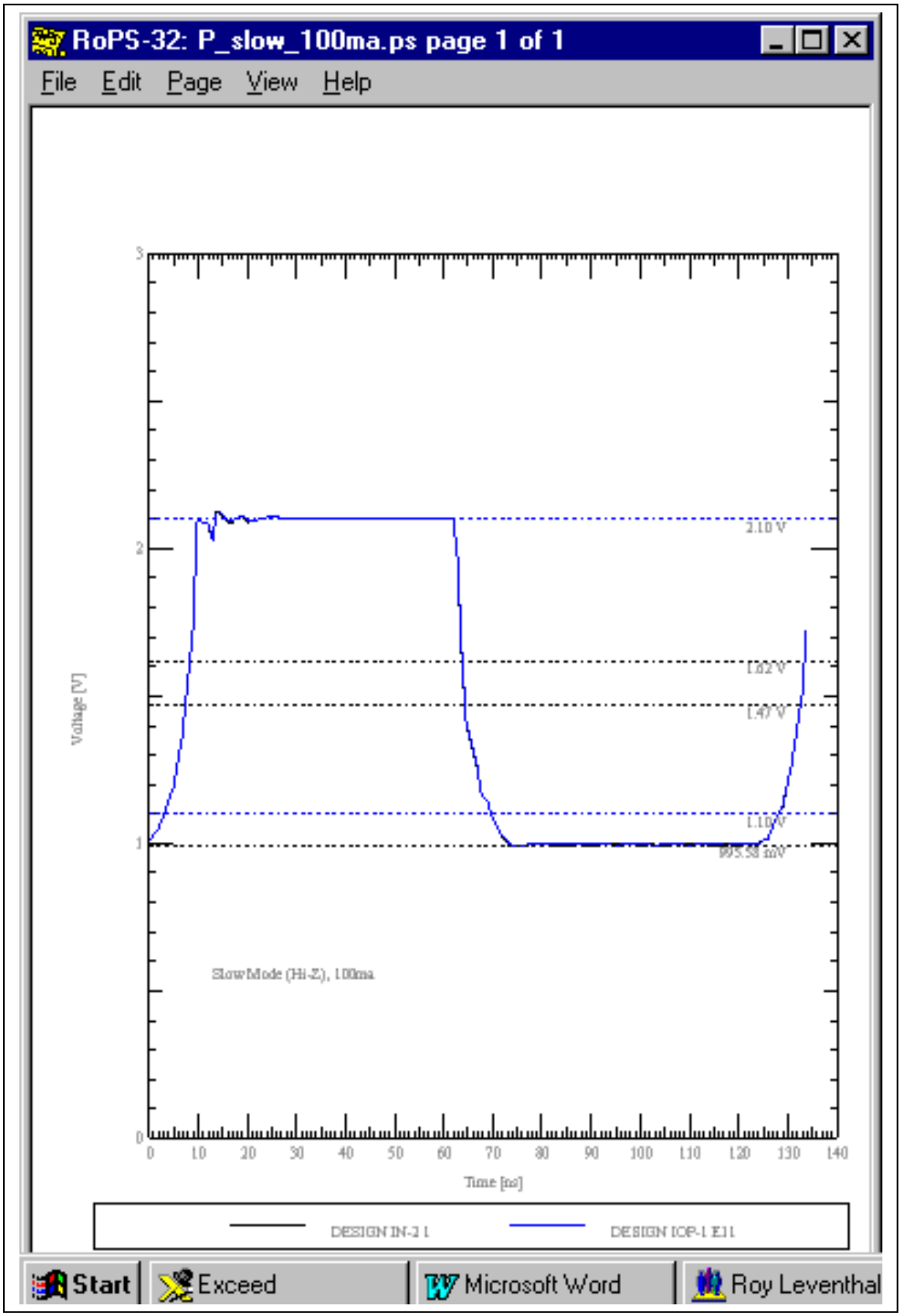
The pulse waveform is displayed below. It shows an output low state voltage at 100ma Iol of approximately 1.0 volts vs. the data sheet specification of 1.15 volts Vol. This represents an increase of 47% in low state noise margin for the IBIS model over the data sheet. Is the IBIS model as received too optimistic, or is the data sheet too pessimistic? What result should we design to? Calling the manufacturer revealed that the data sheet spec was guard banded by 200 mV.

I expect IBIS models extracted from SPICE models or from real parts to exceed data sheet specifications. My experience indicates significant variance.

The conservatism of the data sheet derives from the suppliers' fear of being sued. Unfortunately, it tends to subvert the whole modeling process.

It's obvious that good supplier-user relationships, trust and communication could improve this situation. Beyond that, I advocate the application of Statistical Design and Robust™ Design[2] techniques and due diligence by all involved.

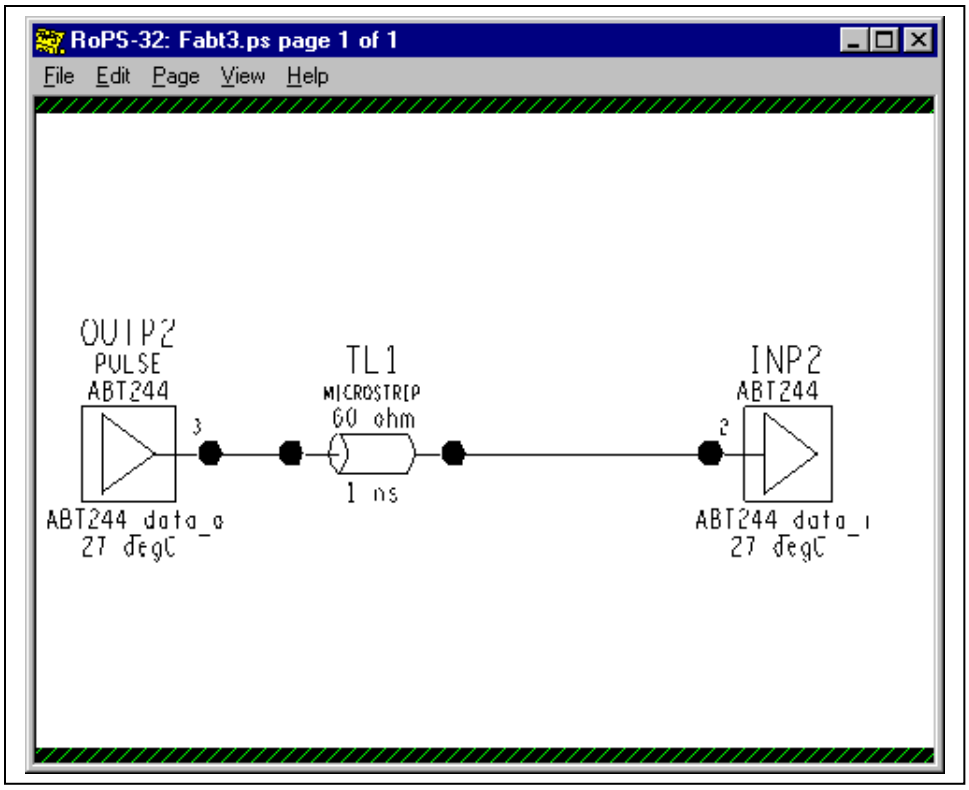
[2] Robust Design is a registered trademark of the American Supplier Institute, Inc.


RoPS-32: P_slow_100ma.ps page 1 of 1
File Edit Page View Help
Voltage [V]
Time [ns]
2.10 V
1.82 V
1.47 V
1.10 V
SlowMode (Hi-Z), 100ma
DESIGN IN-2 1
DESIGN IOP-1 E11
Start
Exceed
Microsoft Word
Roy Leventhal

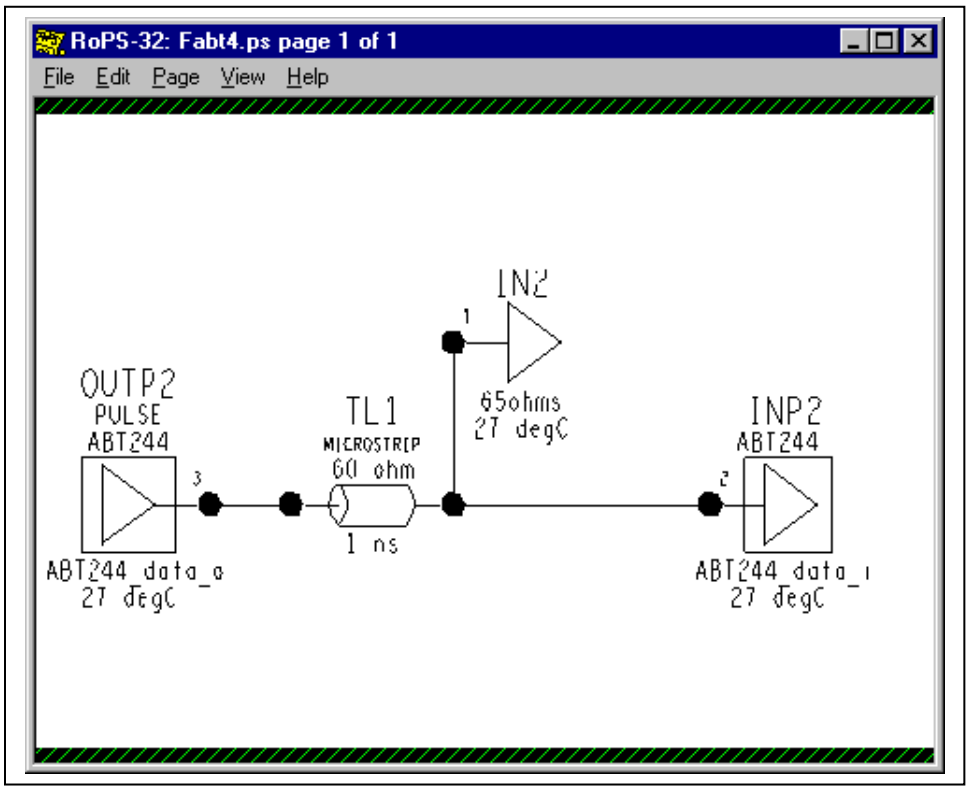
Different Technologies: Different Models Really Do Give Different Results:

In the following sets of simulation results we can illustrate the effects of different V-I and V-T curves, the presence or absence of clamps, etc. I call your attention as to how the two choices, picked for illustration, handle large and small reflections from unmatched and nearly matched receiving ends.

First, the unterminated and terminated circuits:

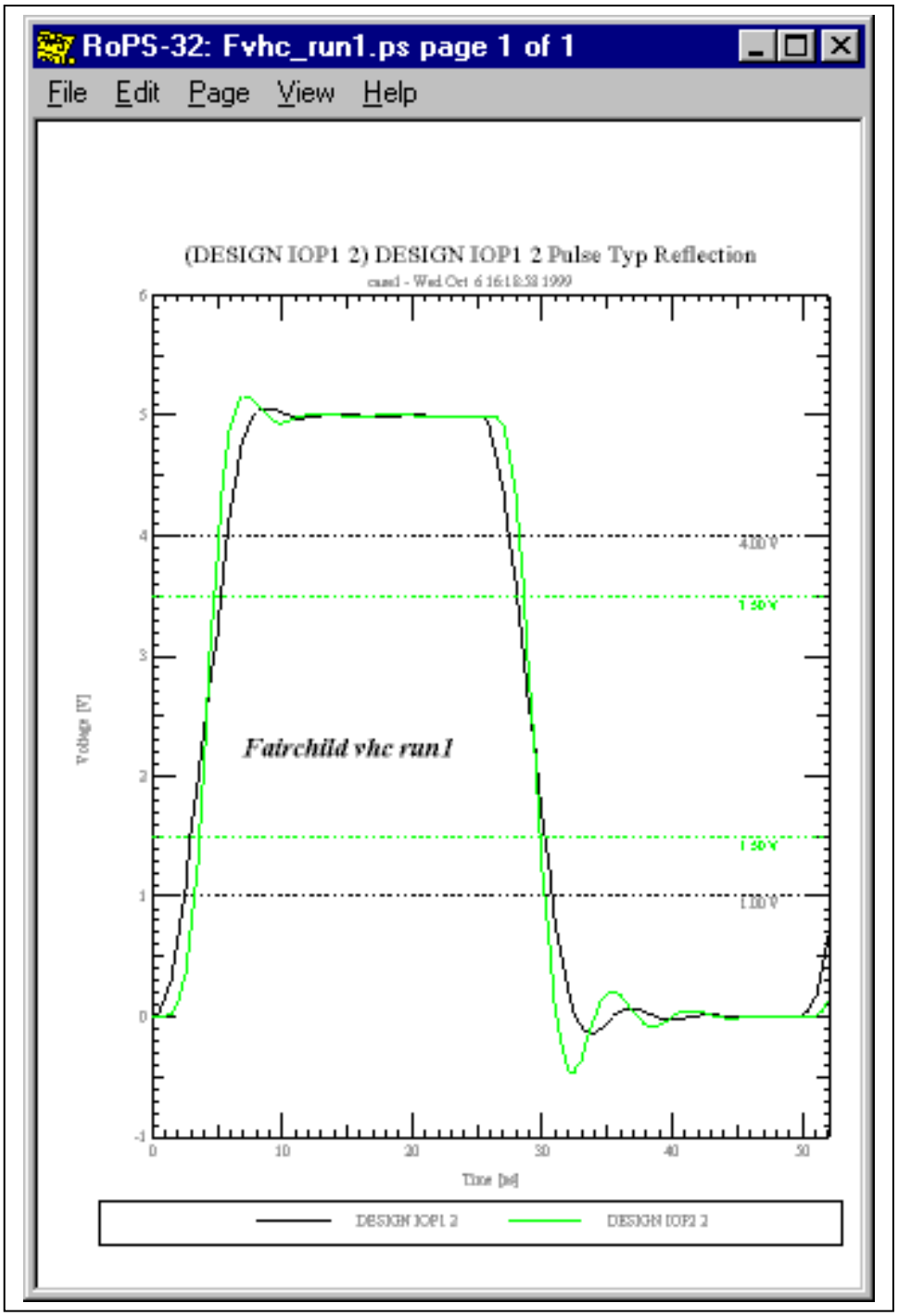


Unterminated

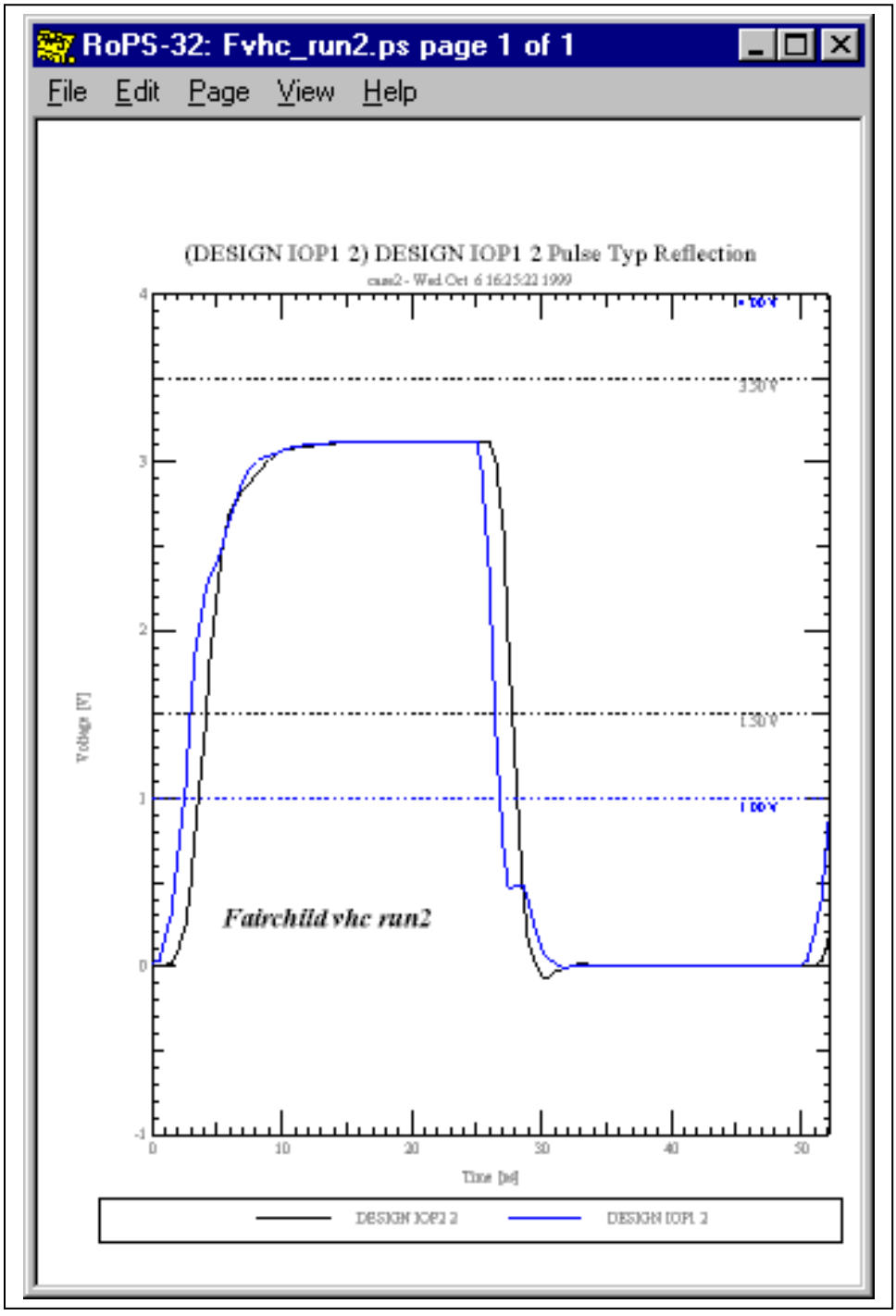

RoPS-32: Fabt4.ps page 1 of 1
File Edit Page View Help
IN2
1
65ohms
27 degC
OUTP2
PULSE
ABT244
3
TL1
MICROSTRIP
60 ohm
1 ns
INP2
ABT244
2
ABT244 data_o
27 degC
ABT244 data_i
27 degC

Terminated

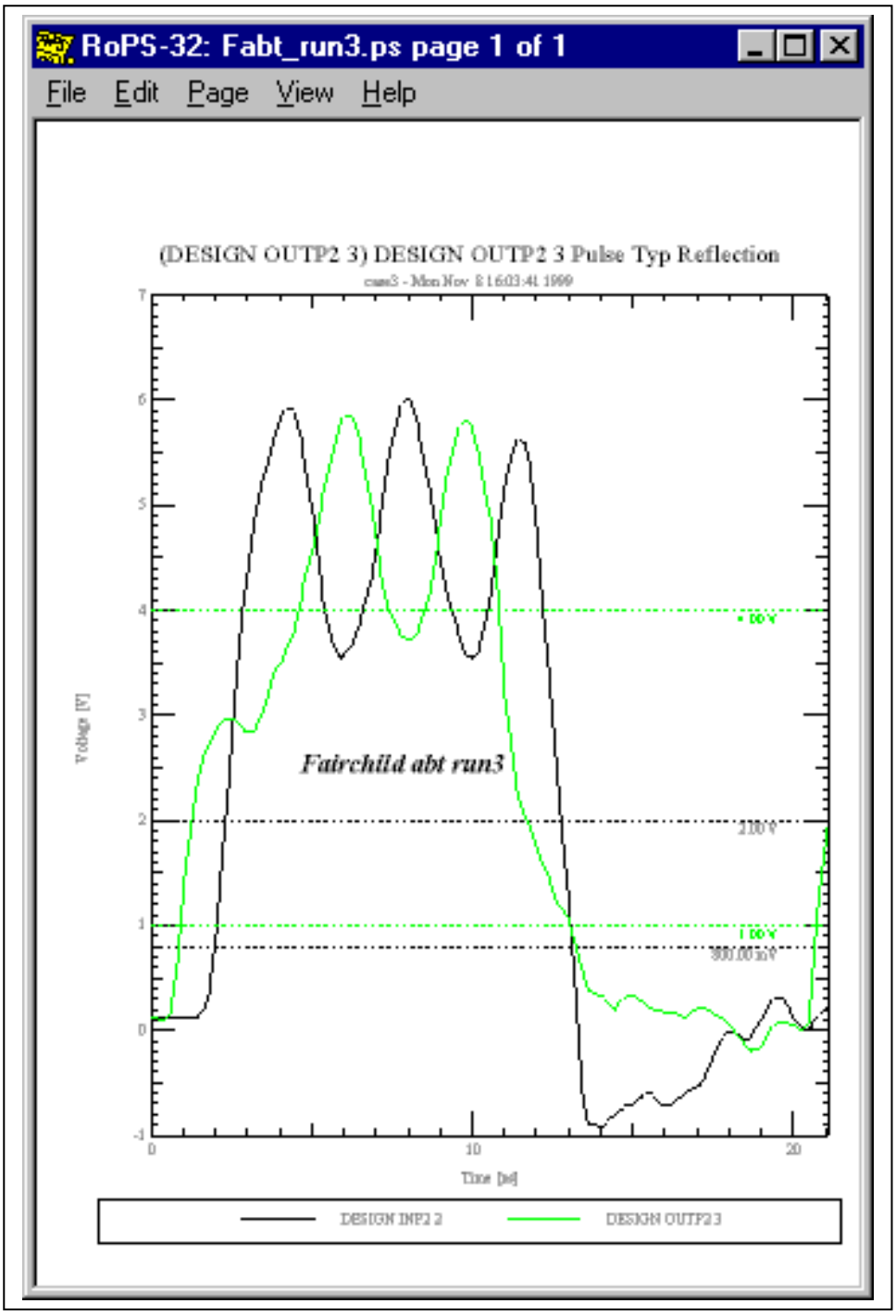
Unterminated



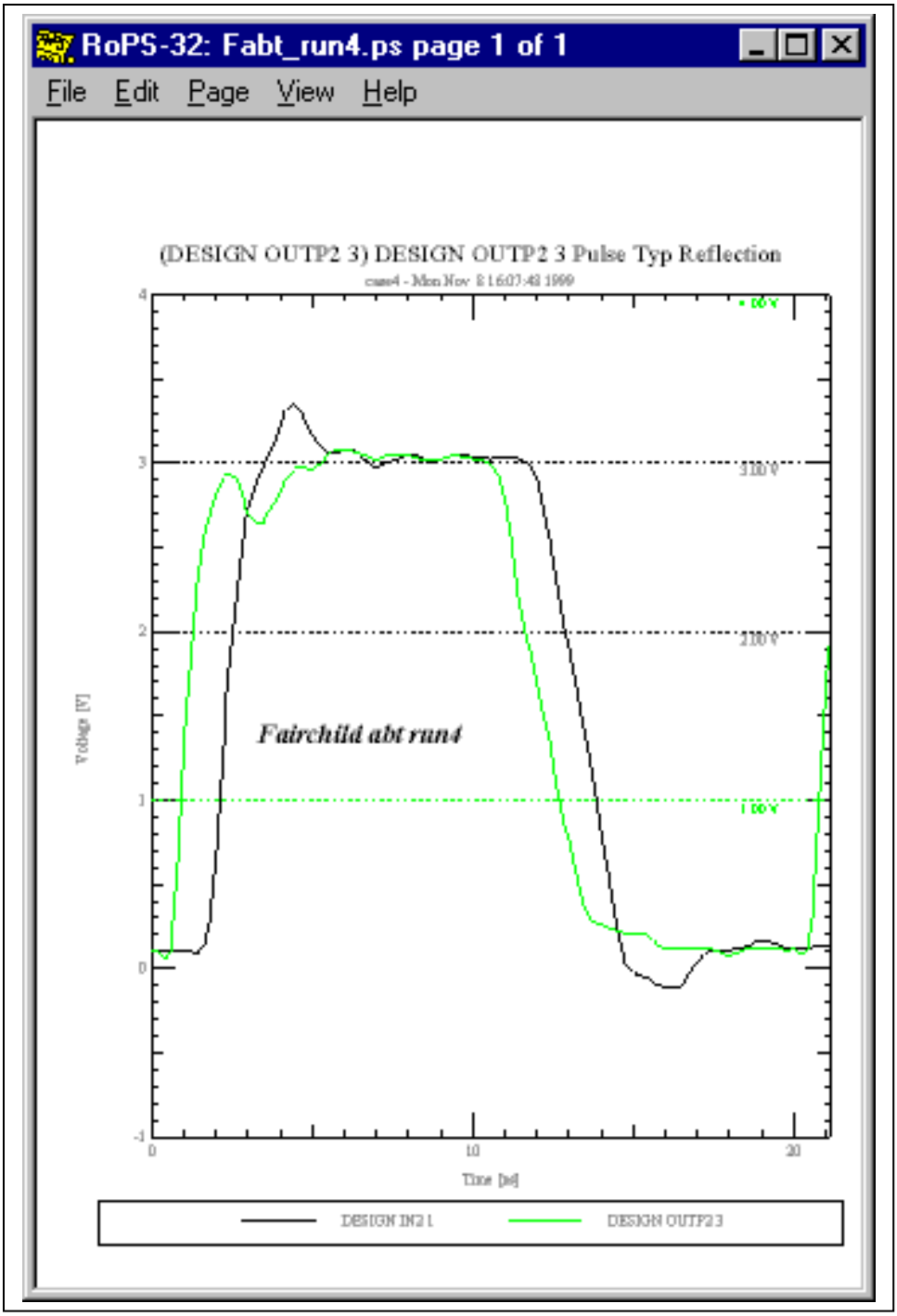
Terminated


RoPS-32: Fvhc_run2.ps page 1 of 1
File Edit Page View Help
(DESIGN IOP1 2) DESIGN IOP1 2 Pulse Typ Reflection
Voltage [V]
Time [ns]
Fairchild vhc run2
DESIGN IOP2 2
DESIGN IOP1 2

Unterminated



Terminated


RoPS-32: Fabt_run4.ps page 1 of 1
File Edit Page View Help
(DESIGN OUTP2 3) DESIGN OUTP2 3 Pulse Typ Reflection
Fairchild abt run4
Voltage [V]
Time [ns]
DESIGN IN21
DESIGN OUTP23

Relation to High Frequency RF Analysis

Behavior that occurs from input to output of a device (such as amplification, etc.) is normally ignored. The focus and use of an IBIS model is what happens from Driver Output to Receiver Input across a transmission line. Behavior with frequency, gain amplification, phase, bandwidth, feedback, dispersion, etc., is inherently not a part of IBIS. The impedance matching and pulse shaping techniques available to RF engineers in the frequency domain are not readily available here. They require different models, different simulators and different mathematics.

EMI

EMI – Electro-Magnetic Interference – is not addressed at all by IBIS. EMI concerns and signal integrity issues are unified in board simulations that take into account reflections, crosstalk, exposed etch length, radiation patterns, etc. IBIS contains no direct connection to EMI properties. IBIS also doesn't contain package dimensions, lead dimensions and other information that would enter into an EMI simulation. The tacit assumption is made that the package is too small and the edge rates too slow to create EMI radiation.

General

IBIS Syntax

IBIS syntax follows ASCII rules on allowable characters, line length, etc. In addition there are reserved words and keywords of allowed length and many other syntax rules. These rules are scattered throughout the 40-plus pages of the spec, not organized or gathered together for easy reading. Therefore, I have prepared a roadmap into IBIS, "**IBIS Model Syntax,**" which is available to the user.

I highly recommend the use of software tools to check suspect IBIS files for syntax errors. The free "IBIS Golden Parsers" mentioned in the IBIS Model Checking Procedure and repeated here are highly recommended:

http://www.eia.org/eig/ibis/tools.htm

Advantages of the Data Lookup Model

The strength of this data table model is that a great amount of large-signal, non-linear, population unit and environmental condition dependent time domain switching behavior can be summed up and accessed efficiently. Further, it is basically impossible to reverse engineer this data and is, supposedly, much more (will be much more?) readily available from silicon vendors. The IBIS

data exchange format supposedly brings together the interests of silicon supplier, software tool supplier and end product user design engineer.

The jury is still out on that.

Data Monotonicity and Convergence

Most simulators tend to choke when data is fed to them that is non-monotonic. While these simulators (often? usually?) produce signal integrity results that are non-monotonic due to ringing, noise and the like, they don't like being fed IBIS models that are non-monotonic (often due to a SPICE simulation to produce the IBIS model). Some simulators will "filter" a non-monotonic model and smooth out or just eliminate the offending data points. Some simulators can tolerate small model non-monotonicities. But, in general they all have serious convergence problems with non-monotonicities.

The IBIS spec recognizes this and includes a serious write-up of what is allowable. Because non-monotonic V-I data can exist in real devices, it is not disallowed in IBIS. Warning messages are provided. However, not all simulators will be able to process the data. Non-monotonic V-T tables are allowed as well. Some simulators will reject this. Others will process the tables.

Examples: A Model IBIS Model File and Curves

An example IBIS model file, g612mtd.ibs, is attached (with the permission of Fairchild Semiconductor, Inc.) at the end.

Resources

IBIS Web Sites:

IBIS directory of publications:
http://www.eda.org/pub/ibis

ANSI/EIA-656 IBIS Home Page:
http://www.eia.org/eig/ibis/ibis.htm

ANSI/EIA-656 IBIS – Tools:
http://www.eia.org/eig/ibis/tools.html

ERL IBIS Project:
http://www2.ncsu.edu/eos/project/erl_html/erl_ibis.htm

IBIS Cookbook:
http://www2.ncsu.edu/eos/project/erl_html/ibis/cookbk_ToC.htm

ERL Software:
http://www2.ncsu.edu/eos/project/erl_html/erl_software.htm

HyperLynx:
http://www.hyperlynx.com/main.htm

References

"An Introduction to IBIS Modeling"
Interface Data Book – Section 13
S. B. Huq
National Semiconductor c1996

Signal Integrity

"High Speed Circuit Design – A Handbook of Black Magic"
H. W. Johnson & M. Graham
Prentice-Hall c1993
ISBN 0-13-395724-1

"Handbook of Digital System Design" 2nd Ed.
Wen C. Lin
CRC Press c1990
ISBN 0-8493-4272-4

"Digital Design Principles and Practices" 2nd Ed.
J. F. Wakerly
Prentice-Hall c1994
ISBN 0-13-211459-3

"Transmission Lines in Computer Engineering"
S. Rosenstark
McGraw-Hill c1994
ISBN 0-07-053953-7

"Signal and Power Integrity in Digital Systems: TTL, CMOS & BiCMOS"
J. E. Buchanan
McGraw-Hill c1996
ISBN 0-07-008734-2

Motorola Semiconductor c1988

"High Speed VLSI Interconnections –Modeling, Analysis & Simulation"
A. K. Goel
J. Wiley & Sons c1994
ISBN 0-471-57122-9

"Transmission Line Handbook"
B. C. Wadell
Artech House c1991
ISBN 0-89006-436-9

"Foundations for Microstrip Circuit Design" 2nd Ed.
T. Edwards
John Wiley & Sons c1992
ISBN 0-471-93062-8

"MECL System Design Handbook" 4rth Ed.
W.R. Blood

Model Sources

Aptos Semiconductor:
http://www.aptos.com/ibismain.htm

Fairchild Semiconductor:
http://www.fairchildsemi.com/models/ibis

IC Works Semiconductor:
http://www.icworks.com/IBIS

IDT Semiconductor:
http://www.idt.com/products/logic/logic_models.htm

Intel:

wysiwig://45/http://developer.intel.com/design/i960/SWSUP/INDEX.HTM

Mentor

http://www.mentor.com/icx/modeling/ibis_modeling.html

Mitsubishi Semiconductor:

http://www.mitsubishichips.com/data/files/download.cgi?ibis.htm

National Semiconductor:

http://www.nsc.com/models/ibis/

PMC Sierra Semiconductor:

http://www.pmc-sierra.com/Ibismodels/default.cfg

Quality Semiconductor:

http://www.qualitysemi.com/main/device.htm

TI Semiconductor - Logic:

http://www.ti.com/sc/docs/asl/models/ibis.htm

TI Semiconductor – Data Transmission:

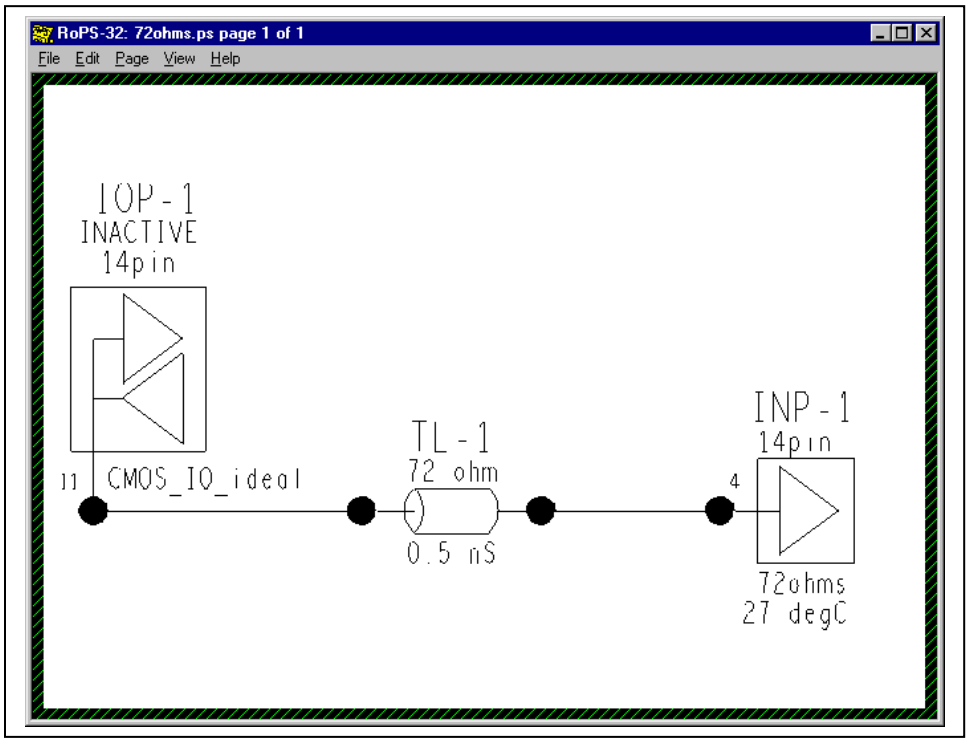
http://www.ti.com/sc/docs/msp/datatran/ibis.htm

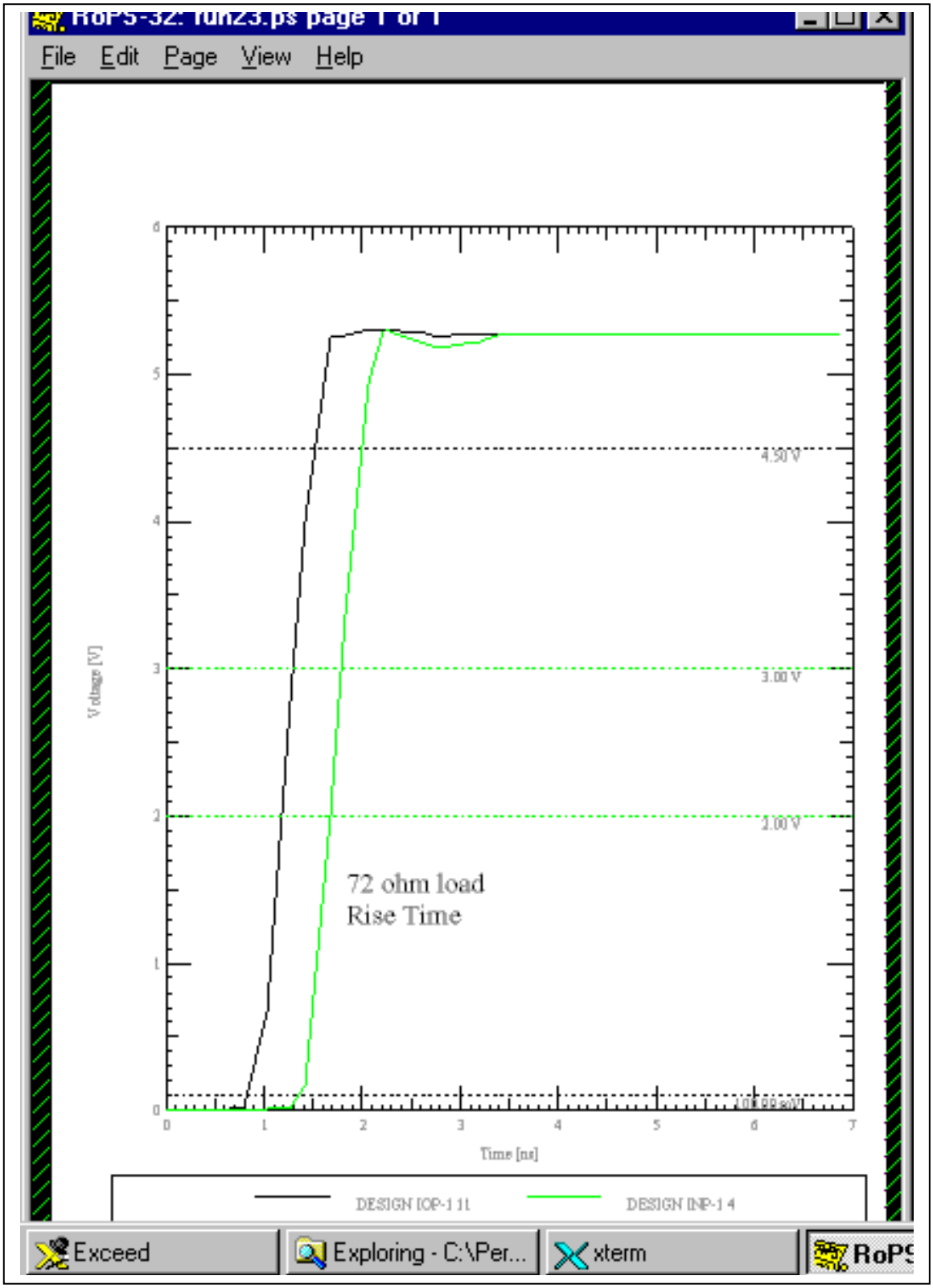
Zeelan Technologies/ICX:

http://www.mentorg.com/icx/icx_models

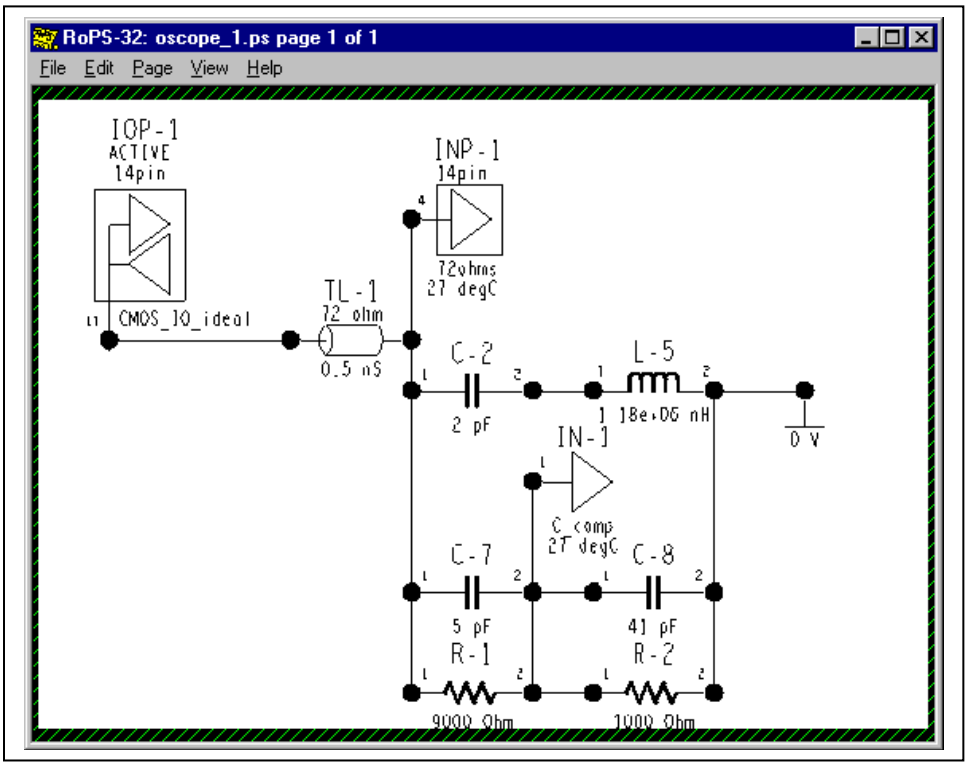
Modeling Oscilloscope Probes for Simulation Results

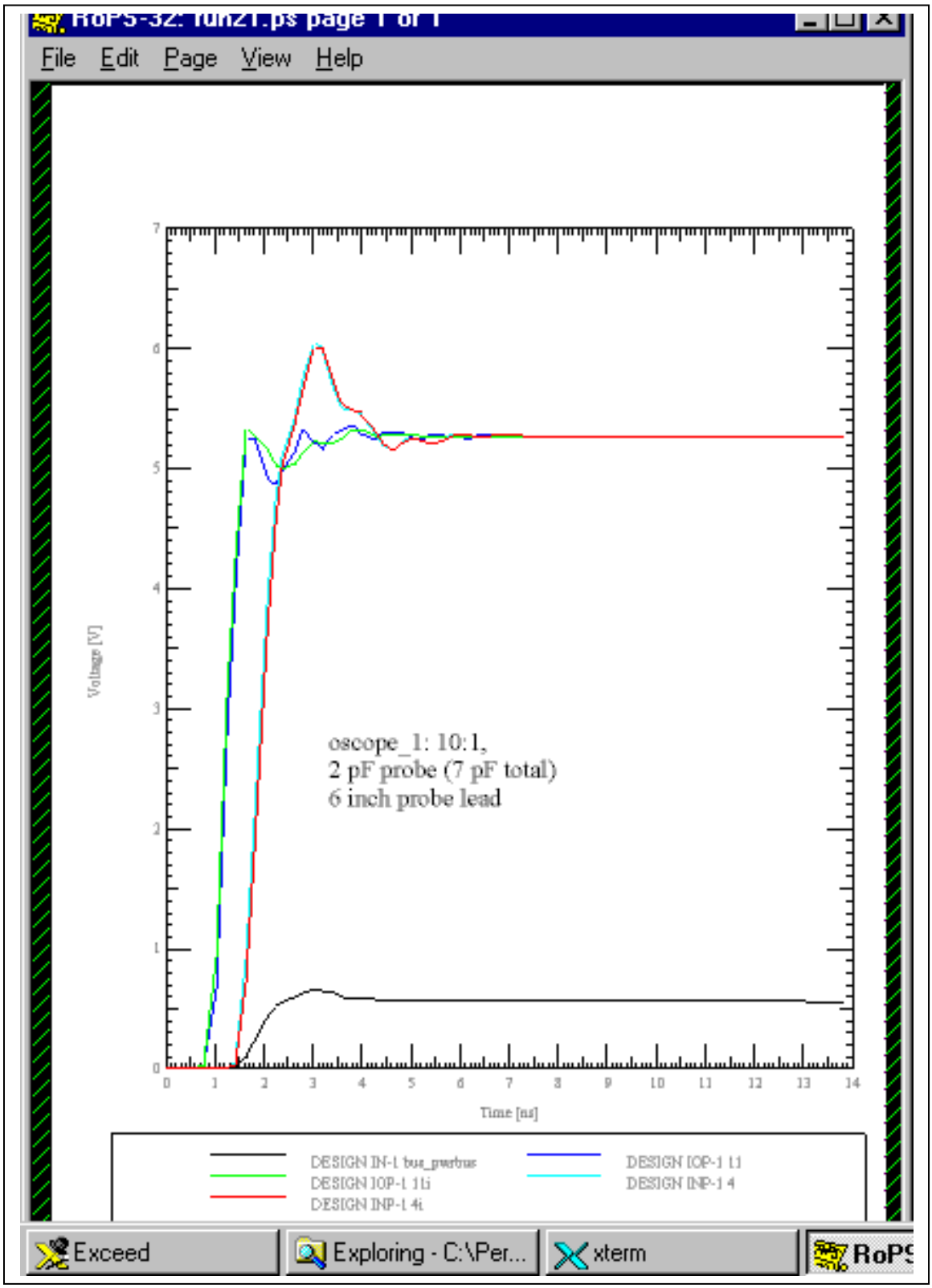
The following topologies were simulated to illustrate the effect an oscilloscope probe might have on a circuit. The simulations were run to show rise time. Clock frequency was 50 MHz and the driver model used was the CMOS_IO_ideal model (presented before -contains V-T curves with sharp corners) under Fast simulation conditions:




File Edit Page View Help
4.50 V
3.00 V
2.00 V
72 ohm load
Rise Time
Voltage [V]
Time [ns]
DESIGN IOP-1 11
DESIGN INP-1 4
Exceed
Exploring - C:\Per...
xterm

The first virtual oscilloscope modeled shows how a passive input probe of 2 pf input capacitance (7pf total loading capacitance) and a 10:1 divider ratio might look. Adding the extra receiver and its input circuit created this. Probe lead inductance for a 6 inch ground lead was modeled but, not the EMI noise pickup effects.

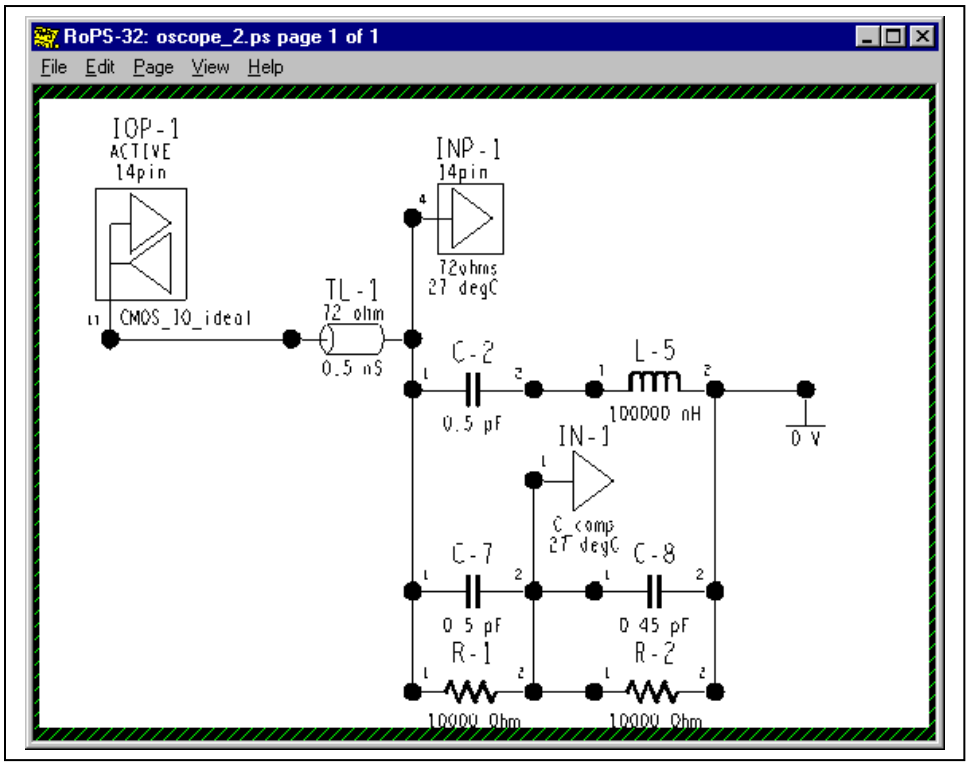



File Edit Page View Help
oscope_1: 10:1,
2 pF probe (7 pF total)
6 inch probe lead
Voltage [V]
Time [ns]
Exceed
Exploring - C:\Per...
xterm

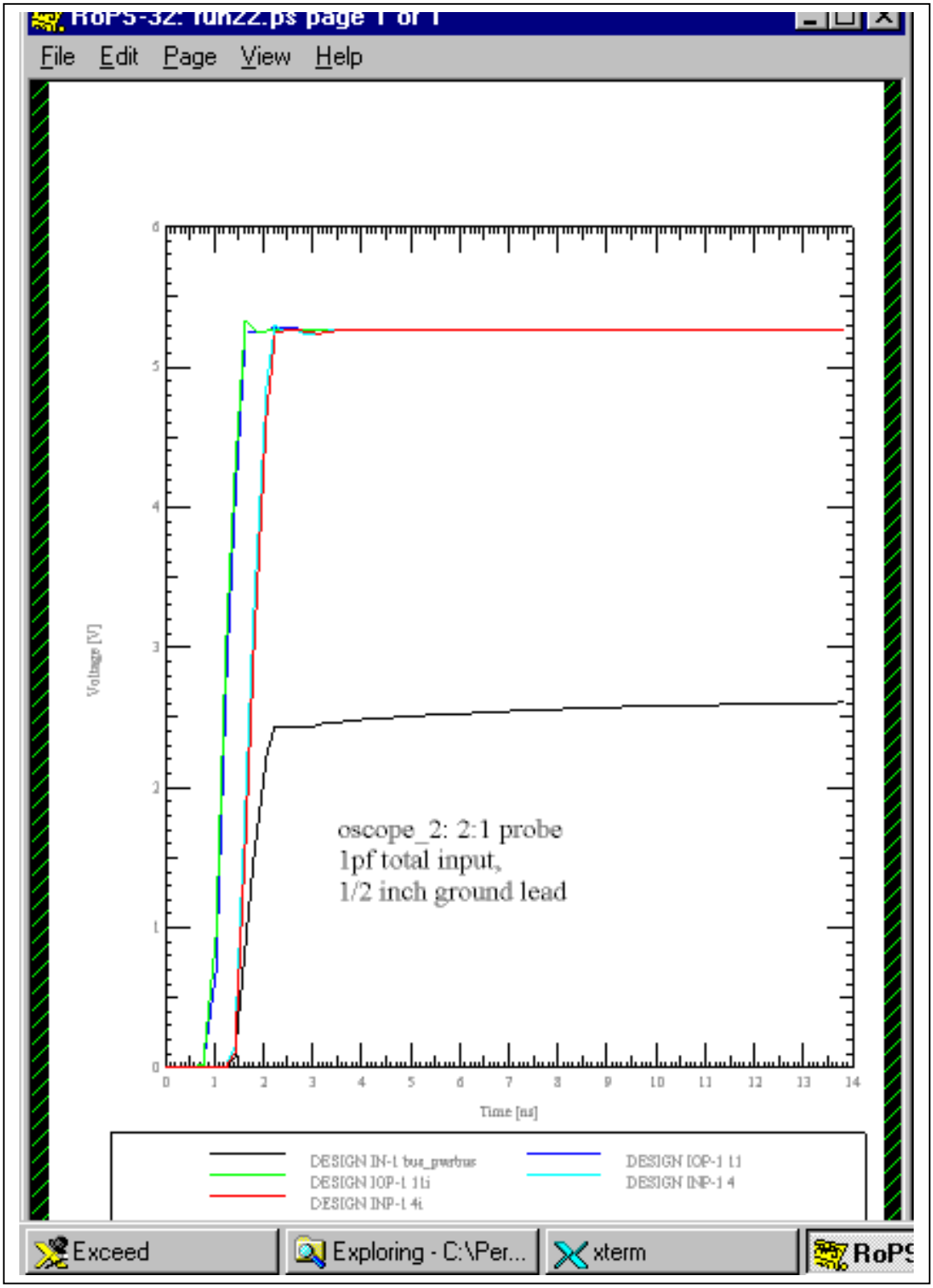
Pretty Amazing – What!

You can easily see the miss-matching of the line due to the probe and the rolloff of the scope.

The second virtual oscilloscope modeled looks more like how an active input probe of 1 pf total loading capacitance might look. A 2:1 divider ratio was used to give a better look at what the viewed waveform would look like. Probe lead inductance for a 1/2 inch ground lead and no EMI noise pickup effects was modeled.



You may note that the probe may not be perfectly compensated. But, it's pretty close.


File Edit Page View Help
oscope_2: 2:1 probe
1pf total input,
1/2 inch ground lead
Voltage [V]
Time [ns]
DESIGN IN-1 bus_pwrbus
DESIGN IOP-1 1i
DESIGN INP-1 4i
DESIGN IOP-1 1i
DESIGN INP-1 4
Exceed
Exploring - C:\Per...
xterm

Model Accuracy, Precision and Verification

You can reproduce the V-T waveform tables, if provided, of your simulator to verify the accuracy of the simulator. That is, the waveform tables can also serve as "golden waveforms" to check the simulator with. Since the load conditions that produced the tables should be provided, the simulator should be able to reproduce those waveforms using the specified loads. The simulator can be said to be verified if the IBIS waveforms and the simulator waveforms agree.

A good verification study would have simulated and measured results agreeing within a few percent. To accomplish this you would need accurate model and measurement results within a few percent. You would also need to account for all primary, and at least secondary, effects at your switching speeds of interest.

This exercise does not verify the IBIS model, however. For that, the IBIS model should be checked for consistency with the databook/datasheet but, especially, verified with test measurements. The EIA IBIS Open Forum Committee has begun to address this issue by formulating an "**IBIS Accuracy Specification**." As of this date, 10/7/99, it is still a draft document. We have copies and I have referenced it in "**Creating an IBIS Model**" and "**IBIS Model Requirements and Verification**."

Remember that high speed waveform measurement is a complex and precision science. It is easy for stray EMI pickup and other effects to mess up your measurements.

Plus, it is very difficult to ensure that the device you model is the device as built into the board you are measuring. For such reasons I advocate a statistical approach to verification studies. By that I mean you simulate an envelope of results using a population (model population) sample and see if the measured results fall within that envelope.

Why Do You Need to Know All This?

Why is it important to understand this? Because, understanding it is key to understanding how the data is derived, how a simulator uses it and what is appropriate and correct for an IBIS model. These modeling and design method concerns are summarized next.

Summary: Design Guide

Methodology

Designing for signal integrity in high speed digital circuits should be done using a Top-Down methodology.

It is extremely difficult to fix such designs, if need be, if you haven't already gained considerable insight in what to do and haven't already designed out as many problems as possible when you reach the breadboard stage. Modern designs are too complex, tricky and design cycles are too short to fly by the seat-of-your-pants and wait until you get copper before you worry much about the design. So, you might as well take good advantage of the Top-Down high speed digital signal integrity simulation methodology. A brief review is in order:

An ideal sequence of processes that the Signal Integrity Engineer works through is:

- Start with the System Designer / Logic Designer having done a schematic capture design and having produced a schematic, netlist, bill-of-materials and timing analysis.

- Simulate reflections, along with what-ifs, of the nets/netlist topologies in a suitable simulator. Design the topologies and their terminations. Produce topology templates with electrical and physical constraints to pass to the Board Designer.

- Place critical parts and rooms of parts and pre-route critical nets in a floorplanning layout tool. Investigate EMI issues with a placement-driven EM control rules checker. Do what-if simulations of timing, crosstalk and EMI with tools that can run from placement and possible routing scenarios. Provide for bypass capacitors, faraday shields, etc., in the design as appropriate. Update the electrical and physical constraints to pass to the Board Designer.

- Design a preliminary board stackup and extract critical pre-routed nets into the signal integrity simulator. Verify performance for timing, reflections, ground bounce and possibly crosstalk and EMI.

- After the Board Designer returns a routed board and stackup database verify its signal integrity performance as required. Include timing, reflections, crosstalk, simultaneous switching noise and EMI as appropriate.

The Signal Integrity Engineer will apply design skills in ground/power plane design and bypassing, shielding and crosstalk, transmission line reflections and termination, topology and device loading and timing design, device selection and noise margins, and EMI radiation issues at a minimum.

What to do if You Don't Have an IBIS Model to Simulate With

Start by simulating with anything to see if your topology/design will work at all. Start with its topology if you don't have a routed board with a defined stackup. Start with the simulation software's default generic driver/receiver model. Progress to a generic model of the process technology you intend to use ASAP.

The Right Model for Each Design Phase and Design Challenge

So, what is the right model for each design phase and design challenge? One that is accurate enough. To quote:

> "The accuracy of an experiment is a measure of how close the result of the experiment comes to the true value." An engineer might amend his definition as follows: "The accuracy of a simulation is a measure of how close the result of the simulation comes to the true value." In the case of behavioral modeling of high-speed digital circuits, the *true value* is what one accurately measures in the lab, and the behavioral simulation is a theoretical prediction. A highly accurate behavioral simulation is one in which the difference between simulation and lab data is small.

From the book, "Data Reduction and Error Analysis for the Physical Sciences," author Philip Bevington.

This author wishes he knew what hard and fast advice to give for all situations. But, the truth is that the particulars vary too much from case to case to do that. It is very much a matter of inference unless a previously used model and simulation are very similar to your new design problem.

Of course, if you can do some design reuse, you can save yourself a lot of work, provided: that the original design, parts, models, databases, etc., have been thoroughly simulated, verified and documented. And, the original parts/processes are still purchasable.

Beyond that, here is the process I recommend:

- Start simulating with a generic part/model in the process technology (LVTTL, CMOS, GTLP, ECL, etc.) speed and drive range you think you will need.
- Modify (do "what-ifs," worst-case, etc.,) the (remember to save yourself an original "virgin" copy of the IBIS model) speed, drive, pin parasitics, etc., parameters that you think your design will be sensitive to. Gain a quantification of how sensitive the design signal integrity performance is to model properties.
- Try to de-sensitize your design to the most sensitive parameters. Good routing, shielding, termination and heat sinking are a few obvious techniques. Devices with more/less speed and output drive capabilities and inherent noise margins are others.

- Simulate your design with an IBIS model of the actual part you will be using as soon as the model becomes available.

- Push on the provider of the IBIS model for a verification of its accuracy (or a better or more complete model) if you are close to violating certain design specifications, e.g., noise margin. In parallel, add requirements to the specification requirements for the IBIS model as outlined below. Such add-ons necessitated by the need for greater accuracy are things such as V-T curves, individual pin parasitics, and MIN-MAX data on parameters and lab measurement verification data. Use some judgement. For instance, it's one thing to come close to violating a un-guardbanded Vih threshold. It's a different matter if you've specified a conservative, guardbanded noise margin. Be prepared, worst case, to do your own lab verification measurements yourself and adjust your design if necessary.

The Minimum IBIS Specification

We have developed a minimum (Standard) IBIS Model Specification Requirement for 3Com Carrier R&D. It is covered in detail in our "**IBIS Model Requirements and Verification.**" This specification is summarized below:

General Property Parameter Requirements

Keywords: IBIS Ver., File Name, File Rev, Date, Source, Copyright, Manufacturer, Component, Voltage Range, Temperature Range and End.

The keywords: Comment Char (new), Notes and Disclaimer are optional.

Package Model Parameter Requirements as Present in the Device

Keywords: Package, Pin, model_name, Rac, Cac, Rgnd, Rpower and Diff Pin. Also, the Diff Pin sub-parameter inv_pin.

I/O Cell V-I Curve Parameter Requirements as Present in the Device

Keywords and sub-parameters: Model (type), Polarity, Enable, Vinl, Vinh, Vol, Voh, C_comp, Vref, Vmeas, Rref, Cref, Technology, Pullup, Pulldown, POWER Clamp and GND Clamp.

I/O Cell Ramp Rate / Switching Speed or V-T Curve Parameter Requirements as Present in the Device

Keywords and sub-parameters: Ramp, dV/dt_r, dV/dt_f and R_load.

The above are required in the 3Com Standard IBIS Model when present in the device.

Adding Min & Max Worst Case Information

Still open for discussion is whether or not to make Min & Max worst case information a requirement of the 3Com standard spec or an add-on requirement.

Adding More Connection and Parasitics Information

The following package/pin parasitic and cell connection IBIS model property data can be added to the specification requirement based on the design engineer's judgement:

Keywords and sub-parameters: R_pin, C_pin, L_pin, signal_name, Package Model, Define Package Model, Manufacturer, OEM, Description, number of Pins, Pin Numbers, R-L-C Matrices, Row, Bandwidth, End Model Data, Pin Mapping, pullup_ref, pulldown_ref, gnd_clamp_ref, power_clamp_ref, vdiff, tdelay_typ, tdelay_min and tdelay_max.

Adding V-I Curve Related Data

The following IBIS model property data affecting V-I curve performance can be added to the specification requirement based on the design engineer's judgement:

Keywords: Pullup Reference, Pulldown Reference, POWER Clamp Reference and GND Clamp Reference.

Adding V-T Curve Data

The following IBIS model property data affecting V-T curve performance and switching speed can be added to the specification requirement based on the design engineer's judgement:

Keywords and sub-parameters: Rising Waveform, Falling Waveform, R_fixture, L_fixture, C_fixture, R_dut, L_dut, C_dut, V_fixture, V_fixture_min and V_fixture_max.