Generating Accurate Behavioral Models of I/O Buffers

Thomas Fisher
Texas Instruments

Abstract

As data rates continue to rise the need for Signal Integrity simulation grows in importance. With this new interest in SI simulation comes the need to simulate accurately down to the device level. Silicon manufacturers are being asked with increased frequency for their SPICE models in order to do just that. The drawback here is SPICE models can contain proprietary information, hence difficult to make readily available. In place of SPICE models IC manufacturers are leaning more and more towards the new IBIS model. IBIS is a new behavioral model standard (ANSI/EIA-656) for modeling input/output buffers. Silicon vendors who provide accurate IBIS models for a given product line can have a positive effect on design-wins.

IBIS models can be generated mainly one of two ways, early on in the design phase from SPICE simulation or from observed measurement of a particular device. This paper discusses the engineering challenges of a methodology for generating IBIS models from the non-destructive measurement of a packaged product. The emphasis in this methodology will be describing the concerns and effort required when generating the data needed for a differential driver and receiver. Most simulators only handle a single-ended configuration so the challenge is 'how to model the differential driver so it can be properly simulated in this environment?' This paper presents recommended data collecting techniques, comparing the IBIS model generated by silicon back to SPICE, the finished model and QA.

Generation of a Behavioral Model

- Introduction
- Behavioral Models
- Performing Measurements
- Translating the Data
- Model Verification
- Summary

Introduction

Accurate circuit models are a critical part of the design process of a high-speed digital system. As these system clock rates continue to rise so does the importance of Signal Integrity simulation. With existing SI simulation technology, it is possible to simulate large systems accurately enough to prevent expensive prototypes and design reworks. One of the largest problems that still faces SI simulation is the availability of quality device models. SPICE models are the favorite among designers for doing SI simulation but the obstacle with SPICE is largely two fold, 1) very slow run times and 2)usually contain proprietary information. To get around this headache and still achieve fairly

accurate simulations a new standard has been formed. The IBIS (I/O Buffer Information Specification) Standard (EIA/ANSI 656) was established as a non-proprietary method for providing necessary buffer simulation information. In this paper we will explore the issue of generating a behavioral model from physical measurement to be used for Signal Integrity simulation.

How Behavioral Models Work (graph_1)

- DC IV Curves provide driver strength
- AC VT waveform provide switching speed and shape
- Package parasitics provide connectivity
- Package pinouts provide mapping between models and layout

Before undertaking such a task, it is first important to understand the operation of the behavioral model that is being generated. The goal is simply to create an accurate IBIS device model. Behavioral models require few kinds of data.

First, they need DC IV curves that describe the Current vs. Voltage performance of a given device at DC high, low, and high impedance states. These curves define the current sinking capacity of the device through the extended operating range. This can be tricky because it is necessary to measure the device beyond the normal operating range in order to determine the effects of overshoot protection diodes - an important effect in SI simulation. Thus, care must be taken in these areas to avoid physical damage to the device (which would, of course, somewhat effect later measurements).

Secondly, they need AC Voltage vs. Time (VT) measurements must also be made of the device. These measurements determine the transition time of drivers (rising and falling) to specify the shape of the driver transition.

The parameters of the package (L,C and sometimes R) will also impact the SI simulation. Measurements of package parameters can be obtained through TDR (Time Domain Reflectometry) technique if they are not available from the semiconductor vendor. Finally, a mapping between device models and physical pin-out must be established.

How Good are Behavioral Models? (graph_2)

- Excellent correlation between IBIS data and SPICE simulations can be shown
- Behavioral Simulation is very fast
- Model accuracy probably as good as accuracy of other data inputs
  - Dielectric constant
  - IC process skew

Are behavioral models accurate enough to use in high speed computer design? If you are able to acquire a good quality model coupled with a reasonably decent simulator the answer is yes. It is worth noting that behavioral models mimic the behavior of a circuit rather than the behavior of transistors. Feedback from reputable designers is that if the model is accurate they have been quite successful at achieving very favorable simulations.

If the model(s) are created properly they can bring the IC simulation accuracy of SPICE with a substantial reduction in simulation time and a corresponding increase in ease of use and Transmission Line simulation accuracy.

As for model accuracy, in most engineering applications your model can only be as good as your knowledge of the physical parameters of your board. How accurately do you actually know your dielectric constant or trace width? What does typical really mean? The point here is not that behavioral models are less accurate than functional models, but that SI behavioral models have been designed to focus on the parameters that are important to SI simulation.

Process for Generating a Behavioral Model

- Types of Models
- Performing the DC Measurement
- Performing the AC Risetime/Falltime Measurement
- Translating the Data to create a Behavioral Model
- The Verification Process

In defining the process for creating behavioral models, we must begin by defining the types of models we need to create. We can then define the methods by which we choose to perform the measurements. Once the physical measurements have been taken, they must be translated into an industry standard format that can be read into and interpreted by a proven state of the art transmission line simulator. Finally, the results must be tested and verified for syntax and accuracy.

Types of Models

- Receiver Model
  - Requires 1 DC Sweep
- Driver Model
  - Requires DC Sweep of High State Behavior
  - Requires DC Sweep of Low State Behavior
  - DC Sweep of High-Impedance Behavior
  - Requires Risetime/Falltime Measurement
- Bi-directional Model
  - Same as Driver Model

There are three types of pins measured for the creation of behavioral models; receivers, drivers, and bi-directional drivers. Receivers are simple, as they require just one DC measurement in their default state. Drivers and Bi-directional drivers, however, require at least two DC measurements plus a risetime/falltime measurement with a high precision oscilloscope. Although driver and bi-directional are measured exactly the same, the behavioral models generated for the bi-directional devices contain more information, such as logical switching threshold values, since they can also take on the characteristics of receivers. This is all accounted for during the translation process.

Performing the DC Measurement (Graph_3)

- Program Parameter Analyzer
  - Current Limit
  - GND/VCC
  - Sweep Range, extending beyond operating range of device
- Power up DUT Board
- Force pin to desired state (only driver)
- Perform DC Measurement recording Current vs. Voltage

When performing bench measurements most parametric analyzer will do the job but I use the HP 4145B Parameter Analyzer. The HP 4145B is a good tool for taking this data because of its overall flexibility and current limiting (100mA) design. In performing the driver DC measurement, the first step is to put the output into the proper state. (i.e. low, high or high-impedance) Next you stress the pin throughout the extended operating range recording the current at each voltage increment to a file. In most cases a voltage increment of 100mV is sufficient. Current limiting is very important, not enough current and the shunt operation of the diodes will not be seen, too much current and physical damage to the device may occur. Since diodes are linear after turn on it is not important to stress the part past this linear region. As for the receiver inputs, you basically sweep the pin over the extended voltage range capturing the current/voltage to a file.

Performing the AC Measurement

- Program Power Supply
  - Current limits

  - GND/Vcc
  - Scaleable Voltage
- Power up DUT Board
- Initialize Oscilloscope, if possible use a multi-sample mode
- Repeatedly toggle Driver Pin
- Perform AC Measurement capturing edge trigger and recording Voltage vs. Time

The AC measurement only applies to the driver. In performing the AC measurement the first step is to setup the Logic Analyzer (or generator) to toggle the outputs. Instead of a single toggle, forcing high or low state needed for obtaining the DC data, it's desirable to toggle the output pin repetitively (to allow for more accurate equivalent mode measurements). In order to get a representable rise and fall time for your model you will want to have edge rates which are specified in the product datasheet. For example, if your product has sub-nanosecond rise and fall transition times you would want to measure this voltage/time. The rise and fall times are defined as the time it takes the output to go from 20% to 80% of its final value. As for your load, I would suggest using the load that is called for in the application. The default load in IBIS is 50 ohm.

There is another option for reporting rise and fall times. It is called Rising Waveform and Falling Waveform. These waveform tables are accomplished through a series of data points (Time vs. Voltage) describing the shape the output makes during its transition.

Translating the Data

- Create Pin-out Description File from Template
  - One file referencing each PIN TYPE
- After performing appropriate Pin measurements:
  - Run High/Low/Input/High Impedance measured data (if applicable) through non-monatomic filter
  - Run Rise/Fall data through non- monatomic filter
  - Translate above input(s) through behavioral model input converter

There is other part model related data that must be hand entered into the system or captured from other database sources. This information includes package pin numbering, naming and the mapping from pin numbers to unique output and input devices.

The actual generation of the behavioral models from the measurement data is accomplished using the same programs that are used for translating from SPICE data to behavioral models and in truth, the data generated is the same as the data that would be generated if you were making these models from SPICE simulations. It is desirable, however to filter the data slightly to take out non-monatomic behavior or excessive noise. Noise filtering can also be done at the data collection side using signal averaging on the oscilloscope.

Model Verification (Graph_4)

- Compare waveform to silicon
- Overlay AC

- May have to adjust for termination resistor drift

TI's Data Transmissions have generated IBIS models and uses HyperLynx's LineSim simulator for model validation. LineSim can simulate a system board by using different transmission elements with different characteristics and connecting them together. These transmission elements can be microstrip, buried microstrip, stripline or cables to name a few. The transmission line I chose for this demonstration is HyperLynx's stackup*. The waveforms are the output of our differential driver (SN65LVDS31) and its complement , the SN65LVDS32 differential receiver using the IBIS model. In this case I am terminating with a 100 ohm resistor across the receiver's input. The transmission lines properties are:

- Impedance, 49.7 ohms -
- Delay, 0.450 ns
- Resistance, 0.121 ohms
- Inductance, 22.4 nH
- Capacitance, 9.0 pF

Although the waveform shown is not overlaid with a waveform generated from the bench, they are very close. The simulation software does handle the differential driver/receiver by using differential probes. This is a first step in validating any newly generated IBIS model.

*An alternative to modeling transmission lines one-by-one with individual cross sections is to tie the lines to a global stackup. This method has the advantage of letting you change the stackup and having all of the stackup-style transmission lines change properties automatically with the stackup.

Model Verification (Graph_5)

- Compare waveform to silicon
- Modify termination to see results

In this waveform I am using the same transmission line properties with a mismatched termination resistor of 60 ohms. The impedance of the media is 100 ohms so this should show a negative reflection. The waveform does show the initial rise in voltage at the source (driver) then a reduce voltage swing at the receiver at one τ (length of cable) with the voltage at the source reducing in amplitude at 2τ. Again, this is just a way to play with your loading configuration to see how the waveform (driver) reacts. In this case it does appear to demonstrate the proper waveform for negative reflection. You can also run this same experiment in the lab to prove out your theory.

Summary

- Accurate behavioral models are essential to design for system reliability
- Accurate behavioral models can be generated from silicon
- Models need verified and QA
- IC vendors need to provide quality models

In conclusion, physical measurement can be used to generate accurate and useful behavioral models for Signal Integrity simulation. The formation of IBIS (ANSI/EIA-656) has been a great vehicle to get around the bottleneck of proprietary information contained within SPICE models and long simulation run times that goes with them. TI is making a very strong effort in supplying (accurate) IBIS models for their various new and existing products. As for TI's Data Transmissions, all new products will have models available early on in the design phase (generated by SPICE) then validated once silicon is available. As for SI simulators, there are several out there, QUAD Design's PreVue, Mentor Graphics' ICX "Sandbox", and Hyperlynx's LineSim to mention a few. The overall goal here is to provide good accurate models of our products so designers can run their simulations with confidence.