Chapter 8

SELECTING THE BEST MODEL FOR A SIMULATION

Comparing and contrasting the advantages and disadvantages of different types of models

Abstract: Several types of models are available for simulating the analog behavior of high-speed designs. Each model type has its strengths and weaknesses, which make it particularly well suited for certain simulation tasks–but not for others. IBIS achieves a good balance between simulation complexity and speed. Therefore, it is the best kind of model to use for the majority of PCB-level simulations. However, recent advances in semiconductor technology require that engineers also know about other types of models. This is especially true when dealing with high-speed digital design and high-frequency issues. When a simulation task requires another type of model, the available model can often be converted to the type needed.

8.1 FROM COMPONENT CHOICE TO MODEL CHOICE

In chapters 6 and 7, we discussed the selection of the *best component* for a design. We now discuss the selection of the best type of *model* to simulate the chosen component's behavior in a circuit. We begin by asking two questions:

- First, is a model–*any* kind of model–available at all?
  - Sometimes a model is available and is exactly what is needed.
  - Sometimes a model is available, but is not the type needed. For example, perhaps we have SPICE, but want IBIS instead.

  - Sometimes no model of any kind is available. In later chapters, we discuss what to do about that.
- Second, what do we want to learn from the simulation?
  - Before choosing a model, it is important to consider the kinds of predictions and answers we want. The more detail, accuracy, and flexibility wanted, the longer it takes the simulation to run.
  - It is also important to consider appropriateness of the simulation results. For example, are we interested in the gain versus frequency of an amplifier circuit, or the noise-margin and timing-margin of a switching network? Also, if we are laying out a PCB on which the component will be assembled, the details of the internal behavior of the IC output driver will often not be of much help.

In chapters 6 and 7, we learned that selecting the best component for the design is not always straightforward. We often need to compromise on desirable features, such as simulation speed versus output drive capability. When selecting the best *model* to simulate the component, we may also need to compromise on simulation capabilities.

8.2 QUESTIONS THAT MODELING AND SIMULATION CAN ANSWER

Typically, when simulating high-speed circuits, we seek answers to the following questions:

a. What signal is transferred from output (generator) to input (load)?
b. What signal is transferred through the load to a succeeding circuit if the load is an amplifier?
c. What signal is transferred to a neighboring circuit that is close enough for electromagnetic coupling (crosstalk) to occur?
d. What signal gets reflected and re-reflected (Signal Integrity) when time delays become significant and mismatching can occur?
e. What signal is induced on power supply lines (Power Integrity) if their source impedance becomes significant?
f. What signal is transferred to distant circuits when wavelengths get short enough for far-field radiation (electromagnetic interference and control-EMI/EMC) to occur? EMI/EMC concerns arise when interference is generated whether the source is conducted, coupled or radiate.
g. What signal is transferred when the behavior of the driver or receiver becomes non-linear? This commonly happens for large signal switching of semiconductors.

To answer those types of questions, we need to consider how the components we choose are connected and operate together. Let's review the definitions of model and system:

- A model is a mathematical representation of a physical thing, and is used to predict its behavior. The mathematical representation can be either a formula, or data to be used in a formula.
- A system is a group of components *operating together*. The common terms for systems of components operating together are *circuits* and *networks (nets)*.

8.3 TYPES OF MODELS

Figure 8-1 is a visual aid presenting the idea that there are many models and many modeling terms. Their uses overlap.

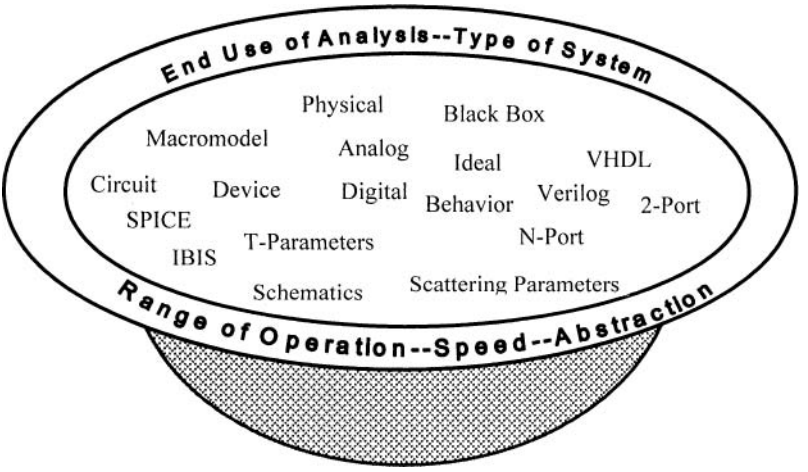

Figure 8-1. The alphabet soup of model types

Models can be classified in several ways, including:

- *Speed* of simulation versus detail of results.
- *Abstraction* versus physical detail level.
- *Analog versus logic operation equation-based models for digital circuits.*
- *Interconnection analysis* in the frequency domain versus the time domain

8.4 USING SYMBOLS AND SCHEMATICS TO REPRESENT MODELS

When people ask to *see* a model, they often mean that they want to see the model's schematic.

SCHEMATIC DEFINITION

A *schematic* is a visual map illustrating the *connection* of models, systems of models, or the internal sub-parts of models into circuits and networks. In a schematic, each model and model element is represented by a symbol. Lines represent wires and show how the elements are connected. Schematics enable one to visualize the models and their interconnections.

Schematics convey one level of information, which is the composition and interconnections of the models, systems of models, and the internal sub parts of models. A second level of information is attached to the schematic symbols. These are the mathematical relations, data and constraints that are used by the simulation engine when running the calculation.

8.4.1 Example of a Simple Schematic

This topic discusses a convention for using schematics to represent models, systems of models, and the internal sub-parts of models. Figure 8-2 shows a very simple schematic consisting of a voltage generator connected across a resistive load.

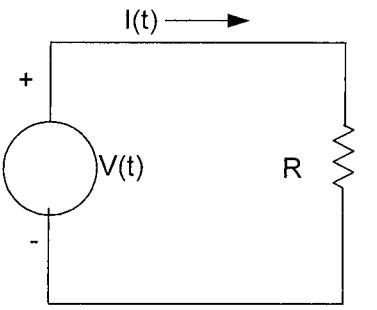

Figure 8-2. A voltage generator (driver or output) and a resistive load (receiver or input)

The generator's symbol is a circle and the resistive load, R, is a zigzag line. A wire is connected from the + output port of the generator to one input port of the resistor. A second wire is connected from the – output port of the

generator to the other input port of the resistor forming one continuous circuit loop.

Using a schematic to represent a system of models helps designers solve many issues in the design of high-speed circuits. With the aid of schematics, we can solve for signals around and across the system or circuit. In this simple schematic, we assume the following conditions:

- Time delays of signals through the circuit are negligible
- The generator internal characteristics are negligible
- The load has no associated parasitics or resonances, and neighboring circuits are far away, thus having no influence

The behavior of this simple system is described by: $I(t) = V(t)/R$. This formula is Ohm's Law. It tells us that the current that flows around the circuit is equal to the voltage impressed across the resistor divided by the value of the resistor. Analyzing the behavior of such systems with a computer involves doing the following tasks:

1. Attaching formulas and data to the symbols.
2. Following the conventions for describing the connections to a computer as is done in a SPICE netlist file.
3. Programming the simulation software to follow certain rules for how electrical circuits behave (Kirchhoff's Laws and Ohm's Law).

Specifically, Kirchhoff's Laws state that:

- Voltages around a current loop must sum to zero
- Currents entering and leaving a node (connection of two or more wires) must sum to zero.

Both of Kirchhoff's Laws are a consequence of the conservation of energy.

8.4.2 A More Complex Schematic

Figure 8-3 shows an example of a more complex schematic. Getting any specific information requires zooming in on portions of the schematic. Showing the entire network makes the point that some circuits are very complex. Adding symbols for vias, connector pins, and corners increases the complexity.

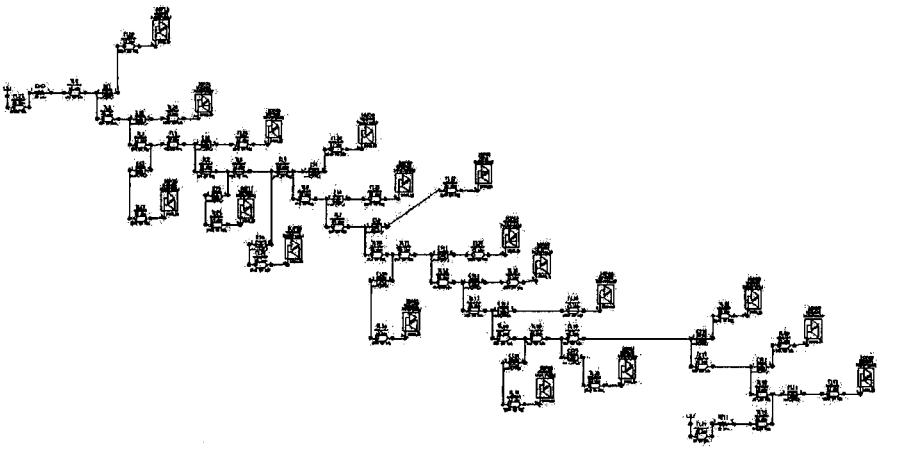

Figure 8-3. An 18-drop daisy-chain backplane network

Additional layers of complexity are associated with high-speed circuit modeling. High-speed circuits are usually characterized by the following conditions:

- Time delays of signals through the circuit are significant.
- Neighboring circuits are close enough to influence each other through coupling and emission phenomena.

Additionally, high-speed circuits may be characterized by the following conditions:

- Generator internal characteristics are significant.
- Associated parasitics or resonances in the load, power supply systems, PWBs, and equipment enclosures are significant.

Simulating for these effects requires additional modeling that takes into account circuit layout, equipment enclosures, and high-frequency effects that change how energy is transmitted down a conductor,[1] power delivery systems, and much more. When modeling the IC, we consider not only the circuit it is part of, but also how the circuit is structured, materials properties of the circuit, neighboring circuits and structures, and non-ideal (zero impedance) power supply systems.

In the case of more complex net connections, more critical timing, and higher switching speeds, it becomes necessary to verify our previous

[1] For example, high-frequency skin effect losses.

schematic-driven simulation results with a layout-driven simulation. For added accuracy in modeling the interconnections, the simulator field-solves and models the transmission line sections, vias, pin fields, and connector pins.

8.5 MAJOR TYPES OF MODELS

We now look at the schematics of major types of models to see how they would be represented in a circuit or network. They include:

- *Two-Port Matrix models.*
  Historically, 2-Port matrix models were one of the first model types applied to active semiconductor circuits. They are now mainly confined to passive networks.
- *S-Parameter models.*
  In high-frequency analog RF and microwave circuits, S-Parameter models are applied to linear active semiconductor circuits. In high-speed non-linear digital switching circuits, this model type is confined to passive networks.
- *SPICE models.*
  This model type includes a wide variety of active and passive device models applied to linear and non-linear analog and digital circuits.
- *IBIS models.*
  This model type includes a wide variety of active device models applied to non-linear digital switching circuits.

8.5.1 Two-Port Matrix Models

The black-box N-Port *Matrix Model,* with N from one to a very large number, is among the earliest circuit models. Before semiconductors were invented, 2-Port models were already in use. The schematic in Figure 8-4 shows a 2-Port (input-output) network with input and output voltage and current signals. The box is opaque. That is, we do not know exactly what it contains, and how what it contains is connected. If the internal circuit is known, we can derive its transfer functions and input and output load impedances. We can then model the box with these functions and its external connections, as in *Macromodeling*, and use this model to find the output signals as a function of the input signals.

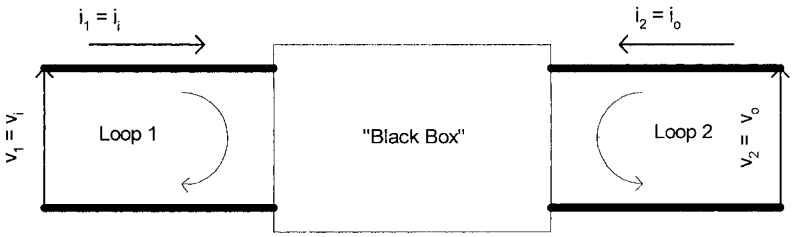

Figure 8-4. Two-Port network

Alternatively, we can measure the output signals as a function of input signals. For example, we can measure currents as a function of voltage as in behavioral modeling, and use this model to find the output signals as a function of the input signals. Further, we can *parameterize* the measured behavior as a set of model elements, one form of which is shown in Figure 8-5, and embed this model in its bias, generator, and load circuitry. The entire circuit can then be solved using Ohm's Law and node and mesh equations. There are only four parameterized elements inside a 2-Port Matrix model. ICs and more complicated networks that use 2-Port models lose some of their correspondence between model elements and device behavior compared to simple single-stage amplifiers and switches.

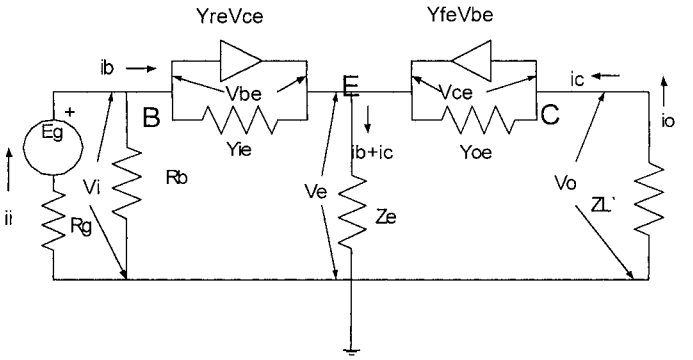

Figure 8-5. Common emitter (CE) amplifier in CE y-parameters with external circuit elements Rg, R_b, Z_e, and Z_L

Matrix models are most often used for linear, small-signal analysis. Table lookups can extend the model for use in non-linear, frequency-dependent, temperature-dependent, and voltage-dependent problems. But limitations are imposed by measurement expense, computer memory expense, and in grasping the key features of the model. Computer memory is getting cheaper

fast, but there is a new problem. The new problem is that both 32- and 64-bit calculations are hardware and software dependent.

8.5.2 Scattering-Parameter Models

A modification of the matrix model is the Scattering-Parameter model, shown in Figure 8-6. It was specifically developed to deal with measurement problems imposed by the impracticality of applying ideal measurement shorts and opens. Short circuit and open circuit conditions are applied when measuring transfer and load impedances. At very high frequencies, ideal shorts and opens applied to current and voltage signals become difficult to establish. Instead, transmission line matched generator and load conditions are applied to the model (component). Then transmitted and reflected waves are measured. The main benefit of the Scattering-Parameter model is to extend model parameter measurement validity to very high frequencies. However, the same cautions about changing bias and temperature or driving circuits non-linear apply. As with all matrix models, we have to re-measure the model parameters when we change bias, temperature, or signal swing.

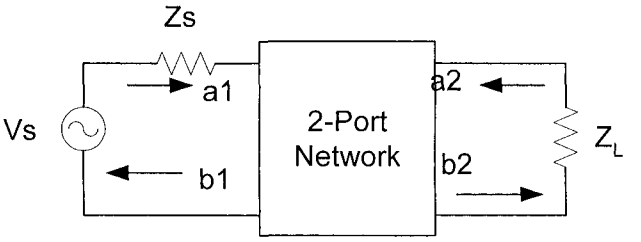

Figure 8-6. A typical 2-Port Scattering-Parameter network

The measurement of the S-Parameters uses the definitions in Table 8-1:

Table 8-1. S-Parameter definitions

Description	Definition	Condition
Input reflection coefficient	$s_{11}=b_1/a_1$	$a_2=0$ (output properly terminated)
Forward transmission coefficient	$s_{21}=b_2/a_1$	$a_2=0$ (output properly terminated)
Reverse transmission coefficient	$s_{12}=b_1/a_2$	$a_1=0$ (input properly terminated)
Output reflection coefficient	$s_{22}=b_2/a_2$	$a_1=0$ (input properly terminated)

where: Zs = Zo sets $a_1 = 0$ and Z_L = Zo sets $a_2 = 0$.

8.5.3 SPICE Models

The difficulty in dealing with changes in bias, temperature, and signal swings in black-box models, plus the need to understand and model the internal, physical workings of transistors, lead to the development of various *Device-Physics Models* as shown in Figure 8-7, for example.

Elements in the model can be modeled with their original device-physics equations. Non-linearities, manufacturing variations, effects of temperature, and voltage can be built into the model. But the common approach is to measure the model parameters under particular conditions and *parameterize* their values in the model. Unless tables of parameter values are used, this approach limits the model's power to handle variation in such parameters.

The circuit diagram shown in Figure 8-7 is simplified compared to real, modern devices. Such BJT models contain more than 40 to 60 model elements. Even with the simplified circuit shown, it would be tedious to solve for its node voltages and mesh currents. The SPICE computer program was developed to deal with the difficulty of solving the circuit equations.

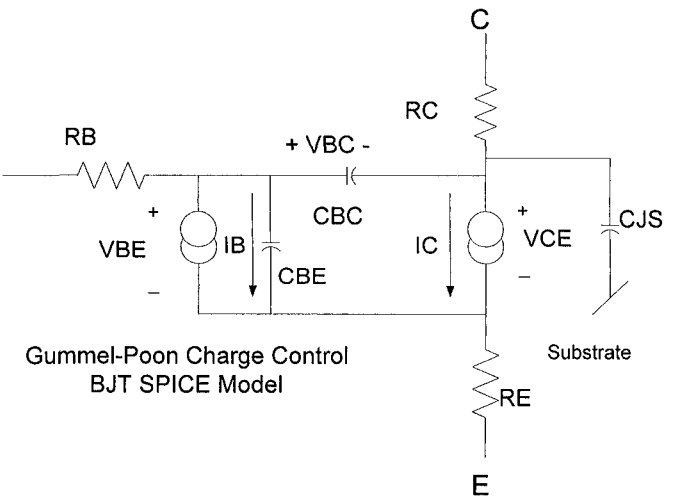

Figure 8-7. BJT equivalent circuit: Source [84].

The SPICE *program* has become synonymous with the SPICE *model.* SPICE is inherently a very good model. But a competitor to a semiconductor supplier can reverse-engineer it to provide information about the inner design of the device.

We can have the following issues with SPICE models:

- The semiconductor supplier may not want to provide this proprietary model.
- The model will run very slowly because the program is solving for all the internal device behavior.
- The internal device behavior information cannot be used and is not wanted at the board level, where we are primarily concerned about signals on the interconnections.
- The non-linear switching behavior is our primary interest and SPICE parameters are not normally measured and calculated under those conditions.

8.5.4 IBIS Models

To resolve SPICE issues, an industry committee developed and adopted IBIS, a new model based on well-established modeling principles. IBIS is an acronym for Input-output Buffer-Information Specification [142, 143]. Inputs (Enable and Signal) are digital, while the Output pad is analog. IBIS is an example of a mixed-signal conceptual model that is usually implemented as a purely analog model.

Measuring the behavior of a black box and creating a model for it is a well established modeling procedure. So is modeling a generator (output) driving a load (input). This is particularly useful when our primary interest is with the signals on the interconnections between the driver and receiver.

The IBIS approach uses the following strategies:

- Data measurements are not "parameterized" as internal model elements.
- Data measurements are swept over the linear and non-linear ranges of switching behavior.
- I-V and V-T data are stored in table format in the model file
- Pin connection information directs a software parser to the correct I-V and V-T tables for a particular I/O.

With this data, we have almost all the information that we need for a basic IBIS model. Adding internal parasitics, switching thresholds, and other information extends the abilities of the model.

This is a simplified explanation of the main structure of the IBIS model. Additional details and refinements extend the IBIS data exchange format capabilities to more complex parts. The IBIS Spec provides a standard for formatting and exchanging model data used in HSDD simulations.[2]

---

[2] Additional IBIS references are [55, 56, 57].

What does the schematic of an IBIS model look like? The Output, or Generator, side is shown in Figure 8-8, and is followed by the Input, or load, side in Figure 8-9.

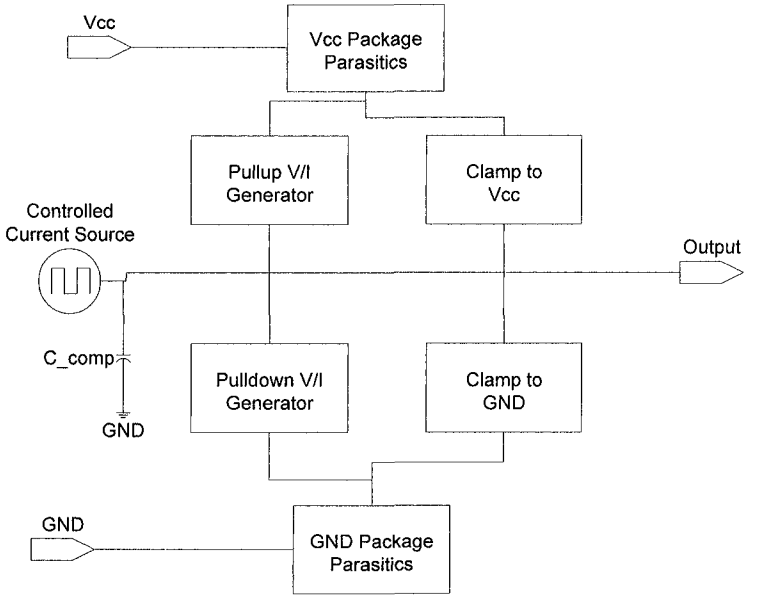

Figure 8-8. IBIS model output (driver) side

The IBIS model has the same limitations as all behavioral models. When bias, temperature, and other conditions change, parameter data for behavioral models have to be re-extracted.

The IBIS Specification now incorporates extensions that allow it to interface with SPICE, VHDL, and S-Parameter hardware description (modeling) languages. A companion spec [56] covers complex connectors and IC packages.

Various proposals [58] are being advanced by both IBIS and the European community to include modeling of the power-supply interconnections for Power Integrity issues. These proposals basically model the in-package and on-die parasitics with an internal I/t curve representing simultaneous switching behavior.

These modeling extensions do not effectively address the issue of EMI/EMC from the IC package. This issue has more recently received some attention in the literature but is still very much in its infancy versus need [13, 119]. Some suggestions have been put forward regarding the issues of noise emissions on power and ground and the conducted emission of core

switching noise. These suggestions are discussed in "Chapter 22, Future Trends in Modeling."

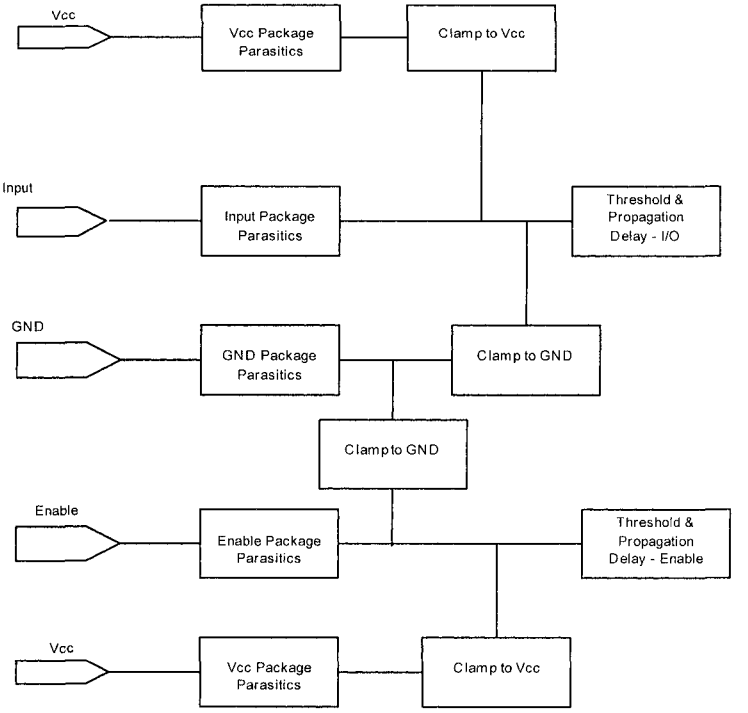

Figure 8-9. IBIS model input (receiver) side

8.6 COMPARE MODELS BY SIMULATION PERFORMANCE

We again list some criteria that can be used to compare and contrast the simulation performance of models, including:

- *Speed* of simulation versus detail of results.
- *Abstraction* versus physical detail level.
- *Analog versus logic operation equation-based models for digital circuits.*
- *Interconnection analysis* in the frequency domain versus the time domain

We are now ready to discuss these topics in greater detail.

8.6.1 Speed of Simulation Versus Detail Level of Results

Table 8-2 compares models by speed. A model can be inserted in a larger system of models for which the overall performance can be calculated.

Table 8-2. Speed of simulation versus detail level in results

Type of Model	Purpose	Simulation Speed	Results
Behavioral models	Usually used when scanning a large design for the first time. At that point, speed is desired over detail.	Fastest	Least detailed
Macromodels	Used early in top-down design approaches when behavioral models are getting decomposed into sub-function blocks		
Ideal-generic models	Used early in the detailed product development when part-specific models have not yet been obtained and only estimates are needed.		
Detailed physical models	Usually used in the verification of a (virtual) physical board design. Accurate and detailed models are particularly important for critical circuits	Slowest	Most detailed

Table Notes:

Behavioral models are made from the measured responses at the input and output ports of a black box containing a circuit. Some formulas and characteristics (actual or implied) of input and output impedance and port-port transfers are derived.

Macromodels are models of a circuit (system) composed of individual parts that have been previously modeled as a whole, unique part or subsystem. Some modeling simplifications and assumptions are usually made at this stage. Equations (actual or implied)[3] can be derived for the following characteristics:

- Input impedance and output impedance
- Port-to-port transfer

The equations are presented as the model for the complete system or subsystem. The input-output impedance and voltage and current transfer ratios of a resistor divider network is a simple case in point.

[3] SPICE macromodels are actually solved as structured netlists instead of actual transfer functions. This is what is meant by implied equations.

Equation-Based Models are input-output equations (transfer functions) derived directly from a block of circuitry. These equations can include internal capacitance and inductance, voltage and current gain, and other parameters.

AMS Models are analog mixed-signal models. AMS models allow both analog and digital inputs and outputs. The analog section of a model uses equations, while the digital section uses logic (bit/byte) operations. Analog threshold crossings affect digital operations, and digital transitions affect analog operations.

Ideal models are models that are simplified to save simulation time and ignore 2nd, 3rd, and higher order effects. Ideal models also describe ideal components, such as capacitors with no internal inductance that are unrealizable in real life. In spite of this, they can be very useful in analysis. Examples include an ideal current source and an ideal operational amplifier.

Ideal models can be useful as building blocks inside macromodels or in the early stages of a design when ideas are being checked for feasibility. It is interesting to note that models can be at once behavioral, macromodel, ideal, and detailed, as in the case of an OpAmp. It depends on the level of detail and abstraction in the model versus the level of detail and abstraction needed in the analysis task.

Detailed physical models are sometimes not as detailed as expected. Modern, competitive, high-speed digital parts usually have a SPICE model generated by the semiconductor process engineer in charge of their manufacture. That engineer is best situated to know the latest and most detailed component data. However, the engineer in charge typically has no need to know some of the SPICE model parameters used only for circuit simulation and may not include that information in the published model.

In most of the BJT examples in this book, only 24 out of a possible 40 available SPICE parameters are characterized.[4] The remaining parameters are defaulted. Often, only about six SPICE parameters are characterized in a given BJT model.[5]

8.6.2 Abstraction Versus Physical Detail Level

Table 8-3 compares models by abstraction level.

---

[4] Exactly how many and what specific BJT parameters are in the model depend on which SPICE version is used.

[5] For more detail, see Table J-3 Small Signal General Purpose Amplifiers-& Switches in "Appendix J, Device Physics."

Table 8-3. Abstraction versus physical detail level

Type of Model	**Description**	**Correspondence to Structure of Device Modeled Or Circuit Schematic**	**Abstraction Level**
AMS Model	Linear or Non-Linear equation-based mixed signal model	Structure usually completely obscured, but circuit synthesis can restore it.[6]	From very detailed to very simplified
Behavioral Model	Measured or simulated characteristic curves, data tables, and combinations of these	Not physically modeled from structure. Observed at the terminals of a device.	Completely abstracted, but can be very exact
Macromodel	Functional Model: circuit simplification, circuit build-up, parameterized model methods, and combinations of these	Building-block circuit primitives incorporate simplified functional elements. Equation-based, dependent, controlled sources are used to represent effects that are not modeled by actual components.	Simplified and abstracted
Transistor level circuit model	Primitive Model: devices represented by detailed physical model equations. Schematics aid visualization.	Close to one-to-one correspondence with physical devices and structures	Not abstracted

Table 8-3 Notes:

In *macromodeling,* the necessary equations are extracted from the building blocks available and their topology, or interconnection. In macromodeling, use is made of controlled sources to model non-ideal effects. For example, input leakage current in an op amp. Also in macromodeling, use is made of additional components. For example, in the transistor macromodel to capture higher order effects.

In *behavioral modeling,* after extracting the input-output data (behavior), we go directly to extracting the model properties. We extract these properties without necessarily knowing the internal structure of the black box. Behavioral modeling can apply to low-level primitives or a top-level abstracted, complex hierarchy of interconnections and components.

Transistor modeling is also based on measurements or process simulations. This type of model applies to the lowest level primitives, such as transistors and diodes. Generally, the model is a set of parameters for an

[6] Synthesis from equations is difficult. Filter synthesis is one practical application of the procedure.

existing set of equations, such as the SPICE bipolar transistor model discussed previously.

8.6.3 What Does Physical Versus Behavioral Mean?

The concept of modeling abstraction involves going from a detailed to a more general description. This can occur in a continuous series of steps similar to a purification process. At any stage of analysis whether a model is behavioral or physical depends on the user's philosophical point of view.

It is important to know the level of abstraction that is appropriate for both a simulation task, and for modeling the underlying detail or causes. Seeking to understand the design at the deepest level possible builds comfort with the results and subsequent decisions. But the more detail desired, the longer the simulation takes. And what will be done with that detail? Modeling what is going on inside an IC may be comforting to engineers. But the average product design engineer will never make any use of that data. Therefore, the level of modeling and simulation detail chosen should be appropriate to the task.

Models predict behavior, or in other words, performance. All models that describe an object bigger than sub-atomic particles are behavioral in some sense. At the same time, if the variables in the model are perceived as physical variables, the model is also perceived as physical. At almost any level of abstraction, a model can be considered to be both physical and behavioral.

Figure 8-10 shows, in a very generalized way, the process of model abstraction from very detailed to very simplified. The process of circuit model abstraction is presented in more detail in "Chapter 20, The Challenge to IBIS."

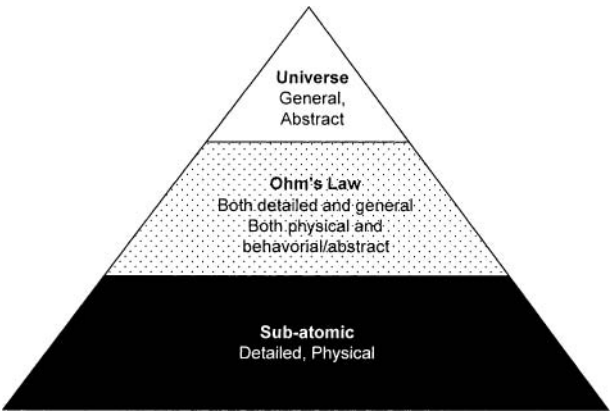

Figure 8-10. Abstraction levels (detailed to general)

Consider the behavioral equation, Ohm's Law. Ohm's Law was derived empirically. In it, the term resistance was originally a constant (fudge factor) that made the equation meaningful. Consider $v = ir$, where:

v = voltage drop across some device
i = current through the device
r = a measured constant of proportionality, resistance

When Ohm's Law was being formulated, resistance was the term chosen to represent the concept of a resistance to current flow when a forcing voltage was applied across a device. v = ir is the behavioral model of the resistor. This behavioral model postulates a physical property that is known as resistance. At the higher-abstraction level, the resistor is a physical thing. If variables (properties) in a formula are physical quantities, the model is said to be a *physical* model.

Resistance is a useful property because, after measuring resistance at one voltage and current pair, we can substitute it in the equation and predict v or i at some other conditions.

As knowledge and experience deepens, we begin to make educated guesses (predictions) about the way that the physical world operates. We then began to talk about first principles in science and engineering. But at root, all our first principles are attempts to explain observation.

Consider now what happens as we seek to deepen our understanding of what resistance is. We develop the equation:

r = (length)/(area)*(conductivity)

Conductivity is thought of as a material property. Then we discover that it may depend on dopant concentration, carrier mobility, temperature, surface roughness, and modulation of channel width by applied voltage.

Now, when we extract the result that (length)/(area)*(conductivity) = r, we say that r represents the *behavior* of underlying physical phenomena. Therefore, the concept of *resistance* is being used in both a physical and behavior sense. At almost any stage of top-down decomposition or bottom-up combination, whether a model is behavioral or physical depends on the user's philosophical point of view.

8.6.4 Level of Physical Detail Modeled

Table 8-4 compares both semiconductor circuit models and device models to the amount of physical construction detail in the models.

Table 8-4. Level of physical detail modeled

Circuit or Device Model	Description	Correspondence to Structure of Device	Level of physical abstraction
Ebers-Moll [34] Gummel-Poon [48][7]	Device Physics Based	One-to-one correspondence. Bipolar Junction Transistor (BJT). Circuit level application.	Lower
Shichman-Hodges [109], Grove-Frohman [42]	Device Physics Based	One-to-one correspondence. MOSFET Transistor (MOSFET). Circuit level application.	
Hybrid-π [83]	Circuit visualization of Gummel-Poon and Ebers-Moll	One-to-one correspondence. Bipolar Junction Transistor (BJT). Circuit level application.	
SPICE, [44, 47] and SPICE derivatives – For example, SABER [104, 105]	Computer implementation of Gummel-Poon, Ebers-Moll, etc.	One-to-one correspondence. Devices include Diodes, BJTs, JFETs, MOSFETs, OpAmps, R-L-C, voltage sources, etc. Modern versions may include many other types of devices. Circuit level application.	
Example: Boyle [19] Operational Amplifier	Macromodel (Also black-box)	Abstracted to and visualized at circuit level: OP1 and OP2 and variants. Circuit level application.	
T-Parameter [36]	Behavioral Model (Also black-box)	3-Terminal circuit visualization of elements. Behavior measured at terminals. Circuit level application	
2-Port Parameters: H, Y, Z, G, ABCD [65, 126]	Behavioral Model (Also black-box)	Input-Output Port model measured at terminals with open and short-circuited loads and generators. Circuit visualization of elements. Behavior measured at terminals. Circuit level application	Higher
N-Port	Generalization of 2-Port to N-ports in H, Y, Z, G, ABCD or S Parameters (Also black-box)	Particularly useful for microwave circulators, and cross-coupled package models for IBIS, connectors. Circuit level application.	Higher (Abstract)

[7] See also [33, 53, 64, 66, 71, 110, 111].

Table Notes:

Black-box models are usually considered to be behavioral models. In one case, the model data is visualized as lumped elements inside the black box. In that case, it looks more like a macromodel. In the behavioral case, the model data is presented as behavioral tables measured at the ports of the model (IBIS). As presented, both types are totally empirical in nature. That is, they come from taking data measurements at the input and output of the device being characterized.

Both black-box and behavioral model viewpoints normally mask the details of construction and the materials inside the device being measured. Consequently, both models run faster in simulations. But both models lose visibility into internal effects, such as simultaneously changing temperature and voltage. Behaviors at different bias points, frequencies, temperatures, and voltages can be represented by curves, sets of curves, data tables, and so forth. But there is a limit to this approach, which still does not match the flexibility of a model like SPICE that is visualized at the physical level.

Black-box models that have been visualized and parameterized as a set of internal elements can be incorporated in a circuit and analyzed with Ohm's Law and Kirchhoff's Law, leading to closed form equations for entire amplifiers, oscillators, and other types of circuits.[8] There are two common ways of doing this visualization: matrix models and SPICE models. Four-element matrix models are simpler than SPICE and more easily analyzed in closed-form equations.

Parameterized black-box model elements, including S-Parameters, are more commonly measured small signal and linear. They are great for amplifier design in the frequency domain, but not much use in the time domain for non-linear switching behavior. For non-linear large-signal switching behavior in the time domain, it is often more convenient to leave the measurement data in the form of lookup tables like IBIS.

8.6.5 Models Oriented to Interconnection Analysis

Table 8-5 compares black-box models. One example is signal reflection variation with load matching.

---

[8] For more detail on some very useful black-box models, see: "Chapter 4, Measuring Model Properties in the Laboratory," and "Chapter 19, Deriving an Equation-Based Model from a Macromodel."

Table 8-5. Models oriented to interconnection analysis

Black-Box Models	Description	Correspondence to Structure of Device	Mathematical Analysis Domain
Scattering Parameter (S-Parameter)	Behavioral Model	Same as 2-Port and N-Port, but with model measured at terminals with matched loads and generators. No circuit visualization of elements. Instead, reflection and transmission coefficients. Circuit level application.	Frequency
IBIS	Data Exchange Protocol	A combination of elements: Behavioral characteristic curve elements, circuit elements, etc. Device behavior is measured (or simulated from SPICE) at the input and output terminals into known loads. Behavior from input to output of a single device is not modeled.	Time

Table Notes:

S-Parameters were developed for use in RF-Microwave (high-speed analog) applications. Analysis is conducted in the *Frequency Domain*. S-Parameters can be applied to Time Domain problems if transform/inverse-transform methods[9] are applied during simulation. Application to IBIS models requires transforming the I/O output driver signal into the frequency domain for simulation.

IBIS models were developed for use in HSDD applications. Analysis is conducted in the *Time Domain* for large-signal non-linear circuits. IBIS models can be applied to Frequency Domain problems if transform/inverse-transform methods are applied during simulation.

8.6.6 Equation-Based Models for Analog and Digital Circuits

Consider whether a model is analog or digital, particularly at the system level. The next higher level of abstraction takes us to the system, System-On-Chip (SOC), hybrid module and multi-chip module levels. Here the correspondence with the physical structure of a single device has pretty much disappeared. The individual structure and elements of the circuit or

[9] Fourier/Inverse-Fourier and Laplace/Inverse-Laplace Transforms are particularly useful tools in this process.

network, including interconnections, are replaced. The entire network is then treated as a single device in an even larger network or system.

At the system level, it often makes sense to abstract much of the analog nature of the circuits to a digital level. For example, a multiplexer or an encoder/decoder might be abstracted from hundreds of transistors to a single digital block, perhaps with a state-dependent delay. At the same time, critical analog functions, such as a low noise input amplifier or a high-power output amplifier might need to still be modeled at the transistor level. An important part of modeling at this level is to make sure all interfaces are correctly modeled: the digital sections load and drive analog sections, and the analog signal changes cause digital data to change state with the expected delay.

Table 8-6 compares system-level analog and digital models.

Table 8-6. Equation-based models for analog and digital circuits

System-Level Models	Description	Correspondence to Structure of a Subsystem
Analog Examples: SPICE, Verilog-A	Linear and non-linear circuit operation. AMS languages support C-like math functions; Laplace transforms, previous values and slopes can also be used in calculations. SPICE is limited to controlled-source functions.	Represents an analog section as a black box. Equations for sections of the system or network are derived using standard circuit theory. There may be several equations, each representing a functional block, within the analog section
Digital Examples: VHDL, Verilog	Digital circuit operation. Node values and timing delays can have continuous "analog" values. Events are scheduled based on timing delay calculations.	Represents a digital section as a black box. Time delays are derived using standard circuit theory. There may be several equations, each representing a functional block, within the digital section
Mixed-Signal Examples: VHDL-AMS, Verilog-AMS, SABER-MAST	Analog circuit responds to state changes in digital section. Digital section responds to threshold crossings in analog section.	Represents the combination of digital and analog sections within a black block. Analog events can cause digital events to be scheduled, while digital events can cause analog values to change. When a digital event occurs, the analog and digital halves of the model are synchronized in the simulator.

8.6.7 Introducing the Characterization Model

When selecting a part to perform in an application, look at its characteristics before modeling and simulating its performance. There is a

connection between modeling and simulation and the performance depicted on a data sheet. The idea of the "Characterization Model" of a device is discussed in "Chapter 7, Using Data Sheets to Compare and Contrast Components."

The *characterization model is another term for a device's data sheet.* The data sheet provides information (as applicable) on a device's logic functionality, power-handling capability, curves of beta versus collector current, gain bandwidth, and more.

We present the characterization model as a new concept. To characterize something means to depict or describe it. The characterization model helps in making good selections of devices to perform in particular designs. A typical use would examine how a device's gain would hold up under maximum current demand.

To make a choice of the best device for an application, it is common to compare data sheets of different devices. Once the choice is made, it is usually followed by a request for the model (SPICE, IBIS, or S-Parameter). The activity of simulating and analyzing the device in the circuit then predicts the performance of everything operating together as a system.

In the low-speed analog and digital circuits of the 1970s, the device-system performance could usually be simulated from circuit topology (schematic) alone. RF was an exception. Today, so many circuits operate so fast, that physical structure and layout on the PCB (and its power planes and enclosure) has to be routinely included in the modeling and simulation.

8.7 ADDITIONAL MODEL COMPARISONS

8.7.1 Behavioral Model Compared to Macromodel

For analog applications, a distinction is usually made between a behavioral model and a macromodel. A macromodel is composed of components and elements. Sometimes these components are blocks of circuitry or function blocks. A macromodel is made by combining models of the individual components or elements. The components in turn can be other macromodels, behavioral models, or a combination of them. We can then say that we can model and simulate the internal behavior of macromodels. In contrast the behavioral model is concerned with the behavior presented to the outside world at the input-output ports of a component.

8.7.2 Digital Compared to Analog in Behavioral Modeling

Digital components have an abstracted model that describes their logic functions. But that does not describe their electrical analog I/O performance. The IBIS behavioral model is used to model the analog behavior of digital parts. However, this too simplistic distinction does not always describe the interactions between digital and analog. First, consider that the analog performance of a digital circuit affects its digital logic performance. Second, logic functions are now used to control analog switching behavior.

A case in point is the bus-hold Intel function. This innovation required an addition to the IBIS Spec to model the new functionality. Another example is programmable drivers, where the presence or absence of a plug-in-board will change the driver, thus the source termination, to a network. It became clear that hardware description and analog macromodeling languages were needed in the IBIS Spec. The IBIS Spec now includes support for various Analog Hardware Description Languages (AHDLs). SPICE itself is one kind of AHDL-like language that is supported in IBIS. However, SPICE is not a true HDL since users cannot edit it.

8.7.3 Behavioral Compared to Physical Modeling Accuracy

As the simulation conditions and the model data approach a true reflection of the actual conditions, the accuracy of a measured behavioral model can approach or even exceed that of a physical SPICE model. This statement is especially true if too little care was exercised in generating the SPICE model. It is also true when the SPICE model is used outside of its assumptions and limitations.

A behavioral model becomes inaccurate when bias conditions, temperature, voltage, and frequency differ significantly from model-measurement conditions for the application being simulated. Many behavioral models do not account for variables like temperature and voltage that change the behavior.

8.7.4 Modeling and Measurement Accuracy Compared

Measurement is usually taken as the most accurate data. But, be careful in such judgments. There is no question that measurement is used to catch modeling and manufacturing mistakes or verify that everything is as expected. But measurement, especially high-speed Signal Integrity and RF/EMI measurements, can be full of mistakes. Performing them is a science in itself. Measurement should verify that the design came out as

intended. If environmental factors (like temperature or EMI, or the statistical variations of components and boards) are not appropriately accounted for, it's trusting in luck that the simulation or the measurements will be correct.

8.7.5 IBIS and Other Models in Accounting for Variation

IBIS provides a data exchange format that accounts for some the statistical spread between members of a population of parts with typical, minimum and maximum data. Plus, it provides guidelines for generating the minimum and maximum data. But these guidelines are not requirements. Only typical data is a requirement. However, population spread data in SPICE, S-Parameters, or any other modeling method are strictly an ad hoc addition. There are no standards to set expectations or requirements on such data.

Modeling accuracy and measurement verification is affected by the statistical probability of:

- Outright mistakes in the modeling and/or manufacture of the component or the design.
- Random variation present in all manufacturing processes.

Random variation present in all manufacturing processes also affects whether a given prototype is representative of what performance will be seen in test measurements on any given day or manufacturing over the long run. Unless considerable care and expense is expended we have no way of knowing from what part of a population distribution components came from that went into building the given prototype. Unless we purposely build prototypes from carefully selected components we can only say that what goes into their construction is determined by random chance. Statisticians tell us that a random sample size of less than 25 of anything gives results that match the population distribution only by pure chance.

8.8 RECOMMENDATIONS FOR MODELING

8.8.1 Recommendations for Design Tasks

Only IBIS, SPICE, and Scattering-Parameter models will be rated because widespread technical knowledge, acceptance and availability make them the reasonable choices.

In conclusion, the best choice for the majority (but not all) of HSDD tasks is IBIS. But before dismissing SPICE, S-Parameters, and other models, let's broaden our perspective to include more of the high-technology design process, as shown in Table 8-7.

Table 8-7. Recommendations for design tasks on high-speed boards

Task or Criteria	IBIS	SPICE	S-Parameters
Large signal switching	Excellent	Fair to good	Poor to fair
Linear I/O impedance of drivers and receivers	Fair to good	Excellent	Excellent
Non-linear I/O impedance of drivers and receivers	Excellent	Fair	Poor
Model parameter variability due to process	Good to excellent	Poor to fair	Poor to fair
Model parameter variability due to bias, temperature, etc.	Fair	Excellent	Poor to fair
Effects of non-ideal power-ground paths	Fair	Good to excellent	Good
IC die, package, and pin interconnection parasitics	Good	Good to excellent	Excellent
IC die, package, and pin interconnection crosstalk	Good	Poor to fair	Excellent
Special IC technology	Fair to good	Good	Poor or N/A
Time domain analysis	Excellent	Good	Poor
Frequency domain analysis	Poor or N/A	Good to excellent	Excellent
Wide bandwidth analysis	Good	Good to excellent	Fair to good
Very high frequency analysis	Fair	Fair to good	Excellent
Simulation speed	Excellent	Slow	Slow
Model availability	Fair to good	Fair	Poor to fair

8.8.2 Recommendations for Selected Design Tasks

In making a recommendation for using a particular type of model and type of simulation to use, the authors have to keep in mind that simple answers are the most helpful. To paraphrase: "An analysis should be as possible–but no simpler." In the case of serial data busses operating at several GHz/GBITs, simplistic answers do not work.[10] Simulating serial data bus at GHz/GBIT data rates is a good example that one model type and one modeling method are no longer sufficient for handling today's design challenges. Table 8-8 shows the authors' recommendations of the type of model used to accomplish various design tasks.

[10] For more information, see "Transform Methods for Gigabit Serial Nets" (page 237) in this chapter and "SPICE-based Macromodeling Templates by Cadence" (page 602) in "Chapter 20, The Challenge to IBIS."

Table 8-8. Recommendations for accomplishing selected design tasks

To Perform This Task	Use This Recommended Model or Method	Or This Secondary Choice
Design devices	TCAD Software	Device physics models
Select devices	Selection guides and data sheets	Design re-use
Analog	Behavioral models and transform methods	Macromodels
Macromodeling: analysis of behavioral tendencies	2-Port	Various simplified generic forms: For example, a voltage amplifier, etc.
Control system simulation	Behavioral models and transform methods	Macromodels
Analog simulation: amplifiers, oscillators, etc	SPICE	2-Port. Also, various simplified generic forms : For example, a voltage amplifier, etc.
High frequency analog simulation: RF and microwave	Scattering-Parameters	SPICE
Signal Integrity simulation	IBIS	SPICE
Serial data bus at GHz/GBIT data rates	Special case requiring a combination of models and methods	SPICE, IBIS, S-Parameters and Transform methods
EMI simulation: transmitter-receiver portion	Scattering-Parameters	SPICE, IBIS
Power, thermal and reliability simulations	SPICE + thermal resistance	2-Port + thermal resistance
Digital system simulation	Verilog and VHDL	–

8.8.3 Summary of Model Recommendations

SPICE, IBIS, and S-Parameters[11] are the most important types of models in today's circuit simulators.[12] Most simply summarized:

- *SPICE* can be very accurate (if care is taken), very detailed (in predicting non-ideal and subtle effects), and very flexible in being able to represent what happens when temperature and bias conditions change. A SPICE model file is compact. A SPICE model is used by designers of semiconductor devices, semiconductor process engineers, and is the most readily available type of model. Other models are most often simulated and extracted from the original SPICE model.

[11] S-Parameters are a form of 2-Port Matrix model.

[12] There are other types of models, but they play almost no role in user interaction with today's simulators.

The negatives are that: There are no standards for SPICE, there are many different varieties of SPICE and they are often not fully compatible. SPICE can be reverse engineered, difficult to obtain and encrypted when obtained. Designers of semiconductor devices and semiconductor process engineers do not share the same concerns as circuit design engineers and the SPICE model they produce often do not pay correct attention to characterizing SPICE parameters used in circuit design but not in process control.

SPICE models can contain over 40 circuit nodes compared to an average of 4 for an IBIS or S-Parameter model and they simply take too long to simulate for most HSDD tasks.[13]

- An *IBIS* model can be very accurate, but only under bias, load and temperature conditions close to the conditions under which it was measured or extracted. IBIS makes use of data lookup tables for key parameters. These data-lookup tables deal with non-linear switching efficiently, but make for large model files. IBIS simulates very fast. IBIS strikes an excellent balance between speed and accuracy on HSDD simulation tasks where interconnections dominate. IBIS data exchange follows an industry-backed standard and IBIS models are very portable between simulators.

  The downside of the IBIS canned approach to a system of templates (Keywords) to represent I/O is slowness to respond to rapidly evolving I/O. Modern I/O designed for dynamic clamping, flexibly programmed drive strength and other features need more of the output circuitry included in the model than the traditional IBIS approach..

  Consequently, SPICE is now used in over 20% of HSDD modeling, especially gigabit SERDES® links. The IBIS Committee is in the process of evolving the IBIS Specification[14] to take fuller advantage of external circuit calls, macromodeling, and equation based modeling to respond to these issues.

- *S-Parameter* models do an excellent job of handling high-frequency related effects of modeling and simulation. Particularly, these effects include loss and dispersion on interconnects for nets with signal frequencies into the 10s of gigahertz. Thus, S-Parameters are familiar in their application to passive interconnections, but unfamiliar in general,

---

[13] Simulation run time increases as the cube of the number of nodes.

[14] Power Integrity and SSN/SSO issues are also receiving attention.

and specifically to active I/O devices. S-Parameters are data-table based, inherently small signal linear, and frequency domain models. But S-Parameters probably have a growing role to play since the design issues of HSDD and microwave signaling are tending to converge.

8.9 CONVERTING A MODEL TO ANOTHER TYPE OF MODEL

Got the wrong type of model? Not to worry. Many models can be converted to other types. Here is how:

8.9.1 Two-Port Conversions for Z, Y, H, ABCD, and S

Two-port matrix model conversions are commonly used. The matrix conversion formulas can be found in many standard texts on circuit theory or matrix math. N-element square matrices are fairly simple mathematical constructs. Generally, they are an arrangement of n equations in independent variables used, for example, to solve for the currents through a circuit given the voltages across its elements.

For a black-box input-to-output model of a semiconductor device, the N-Port models simplify down to two equations in two independent variables, or, a 4-element, 2-Port matrix.

Conversions Between the Matrix Model Z, Y, H, ABCD, and S

Table 8-9 shows the matrix formulas for converting the model parameters of Z, Y, H, and ABCD.

Table 8-9. Two-Port matrix conversions

	Given			
Find	**Z**	**Y**	**H**	**ABCD**
Z	Z_{11} Z_{12} Z_{21} Z_{22}	$Y_{22}/\lvert Y\rvert$ $-Y_{12}/\lvert Y\rvert$ $-Y_{21}/\lvert Y\rvert$ $Y_{11}/\lvert Y\rvert$	$\lvert H\rvert/H_{22}$ H_{12}/H_{22} $-H_{21}/H_{22}$ $1/H_{22}$	A/C Δ_8/C 1/C D/C
Y	$Z_{22}/\lvert Z\rvert$ $-Z_{12}/\lvert Z\rvert$ $-Z_{21}/\lvert Z\rvert$ $Z_{11}/\lvert Z\rvert$	Y_{11} Y_{12} Y_{21} Y_{22}	$1/H_{11}$ $-H_{12}/H_{11}$ H_{21}/H_{11} $\lvert H\rvert/H_{11}$	D/B $-\Delta_8/B$ -1/B A/B
H	$\lvert Z\rvert/Z_{22}$ Z_{12}/Z_{22} $-Z_{21}/Z_{22}$ $1/Z_{22}$	$1/Y_{11}$ $-Y_{12}/Y_{11}$ Y_{21}/Y_{11} $\lvert Y\rvert/Y_{11}$	H_{11} H_{12} H_{21} H_{22}	B/D Δ_8/D -1/D C/D
ABCD	Z_{11}/Z_{21} $\lvert Z\rvert/Z_{21}$ $1/Z_{21}$ Z_{22}/Z_{21}	$-Y_{22}/Y_{21}$ $-1/Y_{21}$ $-\lvert Y\rvert/Y_{21}$ $-Y_{11}/Y_{21}$	$-\lvert H\rvert/H_{21}$ $-H_{11}/H_{21}$ $-H_{22}/H_{21}$ $-1/H_{21}$	A B C D

where:
$|Z| = Z_{11}Z_{22} - Z_{21}Z_{12}$
$|Y| = Y_{11}Y_{22} - Y_{21}Y_{12}$
$|H| = H_{11}H_{22} - H_{21}H_{12}$
$\Delta 8 = AD - BC$

Scattering-Parameters are not measured with AC shorts and opens on the black box model ports, but rather with transmitted and reflected waves. The 2-Port S-Parameter matrix follows.

$$\begin{matrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{matrix}$$

Therefore, equation sets (8-1) to (8-7) derive the conversions between S-Parameters and Z, Y, H, and ABCD Parameters.

Equation set (8-1) derives S-Parameters from z-parameters. The lower case z is used to indicate normalized values; thus $z_{11} = Z_{11}/Zo$ and $\Delta 1 = (z_{11}+1)(z_{22}+1)-z_{12}z_{21}$

$$\begin{aligned} S_{11} &= \frac{(z_{11}-1)(z_{22}+1)-z_{12}z_{21}}{\Delta 1} \\ S_{12} &= \frac{2z_{12}}{\Delta 1} \\ S_{21} &= \frac{2z_{21}}{\Delta 1} \\ S_{22} &= \frac{(z_{11}+1)(z_{22}-1)-z_{12}z_{21}}{\Delta 1} \end{aligned} \qquad (8\text{-}1)$$

Equation set (8-2) derives S-Parameters from y-parameters. The lower case y is used to indicate normalized values; thus $y_{11} = Y_{11}Zo$ and $\Delta 2 = (y_{11}+1)(y_{22}+1)-y_{12}y_{21}$

$$\begin{aligned} S_{11} &= -\frac{(y_{11}-1)(y_{22}+1)-y_{12}y_{21}}{\Delta 2} \\ S_{12} &= -\frac{2y_{12}}{\Delta 2} \\ S_{21} &= -\frac{2y_{21}}{\Delta 2} \\ S_{22} &= -\frac{(y_{11}+1)(y_{22}-1)-y_{12}y_{21}}{\Delta 2} \end{aligned} \qquad (8\text{-}2)$$

Equation set (8-3) derives S-Parameters from h-parameters. The lower case h is used to indicate normalized values; thus $h_{11} = H_{11}/Zo$, $h_{12} = H_{12}$, $h_{21} = H_{21}$, $h_{22} = H_{22}Zo$ and $\Delta 3 = (h_{11}+1)(h_{22}+1)-h_{12}h_{21}$

$$
\begin{aligned}
S_{11} &= -\frac{(h_{11}-1)(h_{22}+1)-h_{12}h_{21}}{\Delta 3} \\
S_{12} &= -\frac{2h_{12}}{\Delta 3} \\
S_{21} &= -\frac{2h_{21}}{\Delta 3} \\
S_{22} &= -\frac{(h_{11}+1)(h_{22}-1)-h_{12}h_{21}}{\Delta 3}
\end{aligned}
\tag{8-3}
$$

Equation set (8-4) derives S-Parameters from abcd-parameters. The lower case abcd is used to indicate normalized values; thus a = A, b = B/Zo, c = CZo, d = D, and $\Delta 4 = a + b + c + d$

$$
\begin{aligned}
S_{11} &= \frac{a+b-c-d}{\Delta 4} \\
S_{12} &= \frac{2(ad-bc)}{\Delta 4} \\
S_{21} &= 2/\Delta 4 \\
S_{22} &= \frac{-a+b-c+d}{\Delta 4}
\end{aligned}
\tag{8-4}
$$

Equation set (8-5) derives abcd-parameters from S-Parameters. The lower case abcd is used to indicate normalized values; thus a = A, b = B/Zo, c = CZo, and d = D.

$$
\begin{aligned}
a &= \frac{(S_{11}+1)(S_{22}-1)+S_{12}S_{21}}{2S_{21}} \\
b &= \frac{(S_{11}+1)(S_{22}+1)-S_{12}S_{21}}{2S_{21}} \\
c &= \frac{(S_{11}-1)(S_{22}-1)-S_{12}S_{21}}{2S_{21}} \\
d &= \frac{(S_{11}-1)(S_{22}+1)+S_{12}S_{21}}{2S_{21}}
\end{aligned}
\tag{8-5}
$$

Equation set (8-6) derives z-parameters from S-Parameters. The lower case z is used to indicated normalized values; thus $z_{11} = Z_{11}/Zo$ and $\Delta 5 = (S_{11}-1)(S_{22}-1)-S_{12}S_{21}$

$$\begin{aligned} z_{11} &= -\frac{(S_{11}+1)(S_{22}-1)-S_{12}S_{21}}{\Delta 5} \\ z_{12} &= 2S_{12}/\Delta 5 \\ z_{21} &= 2S_{21}/\Delta 5 \\ z_{22} &= -\frac{(S_{11}-1)(S_{22}+1)-S_{12}S_{21}}{\Delta 5} \end{aligned} \tag{8-6}$$

Equation set (8-7) derives y-parameters from S-Parameters. The lower case y is used to indicated normalized values; thus $y_{11} = Y_{11}Zo$, and $\Delta 6 = (S_{11}+1)(S_{22}+1)-S_{12}S_{21}$

$$\begin{aligned} y_{11} &= \frac{(S_{11}-1)(S_{22}+1)+S_{12}S_{21}}{\Delta 6} \\ y_{12} &= -2S_{12}/\Delta 6 \\ y_{21} &= -2S_{21}/\Delta 6 \\ y_{22} &= \frac{(S_{11}+1)(S_{22}-1)+S_{12}S_{21}}{\Delta 6} \end{aligned} \tag{8-7}$$

Equation set (8-8) derives h-parameters from S-Parameters. The lower case h is used to indicated normalized values; thus $h_{11} = H_{11}/Zo$, and $\Delta 7 = (S_{11}-1)(S_{22}+1)+S_{12}S_{21}$

$$\begin{aligned} h_{11} &= \frac{(S_{11}+1)(S_{22}+1)-S_{12}S_{21}}{\Delta 7} \\ h_{12} &= 2S_{12}/\Delta 7 \\ h_{21} &= -2S_{21}/\Delta 7 \\ h_{22} &= -\frac{(S_{11}-1)(S_{22}-1)-S_{12}S_{21}}{\Delta 7} \end{aligned} \tag{8-8}$$

8.9.2 Conversions Between BJT Configurations

The three circuit configurations, in which a BJT transistor can be used, are Common Emitter (CE), Common Base (CB), and Common Collector (CC). Table 8-10 is an example of measuring a set of 2-Port admittance parameters in the CE configuration and computing the values that would be measured in the CB and CC configurations.

Table 8-10. CE-CB-CC conversion formulae

CE Measured	CB Computed from	CC Computed from
y_{ie}	$y_{ib}=y_{ie}+y_{fe}+y_{re}+y_{oe}$	$y_{ic}=y_{ie}$
y_{fe}	$y_{fb}=-(y_{fe}+y_{oe})$	$y_{fc}=-(y_{ie}+y_{fe})$
y_{re}	$y_{rb}=-(y_{re}+y_{oe})$	$y_{rc}=-(y_{ie}+y_{re})$
y_{oe}	$y_{ob}=y_{oc}$	$y_{oc}=y_{ie}+y_{fe}+y_{re}+y_{oe}$

We could also present other conversion tables to show how useful these models are in fast computers. However, it's best to have the level of detail remain in and be handled by computer programs.

8.9.3 SPICE to IBIS Conversion

IBIS Golden Parser

The IBIS Golden Parser checks for correct syntax and the presence of required model elements. It is a free software program available from the IBIS Open Forum. Each version of the Golden Parser is backward compatible with earlier versions; that is, ibischk4 can check version 1 files. The most recent version should always be used since it will incorporate all the latest parser bug fixes.

The Parser is available at: http://www.eigroup.org/ibis/tools.htm and is maintained by the IBIS Committee. The website also provides other useful links and free tools. For instance, a SPICE to IBIS conversion utility from North Carolina State University (s2ibis3) can be used to create IBIS models from SPICE. There are also downloadable IBIS tools from EDA providers that make it easier to validate IBIS models, including graphically viewing IBIS tables.

The Golden Parser should be run during file construction on a check-as-you-build basis. Before running the Golden Parser for the first time, it is necessary to download and install the program. Be sure to choose the correct IBIS version (DOS or UNIX) for your platform. The program is run from the

DOS prompt by typing: `ibis_chk <filename>`. The most trouble-free approach is to run a copy of the parser in the same subdirectory as the IBIS model file.

Determine how many unique buffer designs and how many package types are to be tried in the part to be modeled. Each unique buffer and package combination requires a different [Component] model. Find out whether the device will be used in a mixed power-supply environment. For example, the ESD diodes may be referenced to a different supply than the buffers.

In a large device of several hundred pins or more, the construction of a complete IBIS model file can be a lot of work. Consider consulting with the end user to find out if some pins are not going to be simulated and can be ignored or given a default model. For example, static control pins rarely get simulated. According to the IBIS Quality Checklist, a complete IBIS file should have a model for every pin, but this is not required for Signal Integrity simulations early in a design development.

"Chapter 13, Using EDA Tools to Create and Validate IBIS Models from SPICE," shows the use of one commercially available EDA tool for deriving an IBIS model from a SPICE model. The example in this chapter uses Model Integrity® from Cadence Design Systems. Competing EDA tools are available.

Overview of the SPICE Simulation Method

SPICE models are usually considered to be proprietary and are not commonly available to customers from semiconductor suppliers. That is one reason the IBIS Spec was created! To get a SPICE model, especially with process distribution data, a customer usually needs to be doing considerable business with the supplier and sign a Non-Disclosure Agreement. If a SPICE model is obtained, the customer may need to generate the IBIS behavioral tables. If the SPICE model came without min-typ-max data, the customer needs to do the following tasks:

1. Put in estimates of the +/- variation from the typical or nominal values of the SPICE model elements and simulate under those conditions to get IBIS min-typ-max results. In the IBIS file, the IBIS format arranges the results data typ-min-max from left-to-right.
2. Simulate the I-V and V-T tables.
3. Consult with the semiconductor supplier to get a sense of how much their process varies.
4. Most often assume +/- 10% on the current for the I-V MIN-MAX tables and +/- 25% on the time for the V-T MIN-MAX tables.

Today, simulating complex I/O, such as a driver with pre-emphasis, requires capabilities beyond those of traditional IBIS models. Encrypted SPICE models are being supplied under NDAs. But this is not an optimum solution as is discussed in "Chapter 20, The Challenge to IBIS." For one, simulation times with SPICE are much longer.

I-V Tables

To start, use typical process parameters. To get pullup and pulldown tables for a buffer:

1. Set the control and enable inputs to turn on and turn off each pullup and pulldown device as required to go to output logic high and low.
2. Set the SPICE simulator for a DC sweep analysis. This is done by connecting a voltage source to the output node, sweeping it over –Vcc to +2Vcc and measuring the current into and out of the output node. Output clamp diodes, when present, must be disconnected in the SPICE model to obtain the pullup (pulldown) I-V tables. Next, reconnect the diodes and re-simulate to get the total turn off (turn on) I-V tables. After subtracting the pullup (pulldown) portion, the ground (power) clamp I-V current portion is left for inclusion in the IBIS model. Remember that the I-V behavioral table datums are stored (if present) in for portions -- pullup, ground clamp, pulldown, and power clamp. These datums are re-added appropriately by the simulator depending on which part of the –Vcc to +2Vcc range the voltage is at.
3. Most of the time simulate the min-max tables as well as typ. Let us say that the min voltage range is the same as the typ voltage range minus 5 %. Then, adjust the sweep voltage range by the same amount when the simulation is run.

V-T Tables

To start, use typical process parameters. To get rise and fall waveforms for an output buffer:

1. Set the control and enable inputs to turn on and off each pullup and pulldown device as required to go to output logic high and low. Refer to the IBIS Spec for specifying the output load or test fixture to drive.
2. Set the SPICE simulator for a transient analysis. Connect a voltage supply to the output node of Vcc. Remember to reconnect any output clamp diodes in the SPICE model if they were disconnected to obtain the pullup and pulldown tables.

3. Suppose this is a single discrete transistor being modeled. Then, drive the SPICE model input with a pulse waveform having a rise and fall time at least as fast (but, not a lot faster) as the fastest expected rise and fall time of the device being simulated. Also, the amplitude of the input drive should be enough to drive from hard on to hard off and vice versa. Most of the time the output buffer of an IC is being modeled. Then, use the inherent internal process rise and fall time for the driver as provided by the semiconductor supplier.
4. The simulated rise and fall time is supposed to be the rise and fall of the buffer itself. So, remove internal pin and package parasitics from the SPICE model before running the simulation. And, remember to use the V-T data at the internal die connection point node in the IBIS file.
5. Remember to connect the load to the proper rail when running the simulations. Suppose the standard 50 Ω load is used. On pullup connect it to the Vcc rail and on pulldown connect it to ground. For open pullup or pulldown devices and ECL use the load and voltage used by the semiconductor supplier when they specified propagation delays.
6. Remember that the rise and fall slew rate interval for IBIS is 20 % to 80 % of the normal full output swing, if ramp (slew) rate information will be used instead of full V-T data.
7. Remember that serious ringing and non-monotonicities tend to invalidate simulations because they are hard to interpret, repeat, reproduce and verify with bench measurements. Clean output waveforms are needed for reliable model data.

Usually simulate min-max tables as well as typ. Let us say that the min voltage range is the same as the typ voltage range minus 5%. Then, adjust the sweep voltage range by the same amount when running the simulation.

8.10 TRANSFORM MODELS FOR SYSTEMS

In solving today's extremely high speed switching networks, transform methods are a very cutting edge method.

Transform methods often allow analysis to be easily switched between time and frequency domains. State machines for logic analysis can be represented using transform methods. Analysis of mixed analog digital systems with sampled data is also facilitated.

8.10.1 Domain Transformations Assist Problem Solving

The boundary between analog and digital is becoming increasingly blurred. For example a recent article discussed the use of spread-spectrum methods to suppress (double-sideband suppressed carrier, Phase-Lock-Loop) the carrier of a high-speed digital clock! This is definitely an analog technique being used in a digital system. Another reason for covering something about system-level modeling subjects in a discussion of semiconductor modeling is that the future holds more System-On-Chip (SOC) designing, where such modeling will be employed.

The design of serial busses operating at 10 GHz frequency rates and above requires a combination of models and modeling methods. In the macromodeling template approach, the template amounts to a structured SPICE-like netlist that can program-call various types of models as best fits the simulation task–SPICE, IBIS, convolved S-Parameter impulse response, or AMS-equation-based type modeling. For more information about the macromodeling template approach, see the "SPICE-based Macromodeling Templates by Cadence" topic in "Chapter 20, The Challenge to IBIS."

Because of these reasons and that transform methods are now being used in high-speed serial data busses, it is important to briefly review them. By transforming (mapping) a problem into a new mathematical domain the mathematics and solution of problem can often be made simpler. The inverse transform, back into the original domain, is usually performed to yield the final answer.

The transform method most immediately familiar to engineers is the Fourier Transform and its inverse between the Time-Domain (physical space-time) and the Frequency-Domain (a new, mathematical domain). We can visualize a single sine wave frequency and think that we are conceptualizing a physical entity. But that is a misconception. The single sine wave frequency is still a time-domain signal. The point is that the frequency domain is a mathematical transformation, or mapping, from the time domain.

System-level transfer function modeling is not a new subject. For example, this method has been used for a long time in the design and development of analog hybrid OpAmps. Such models in conjunction with Bode Gain-Phase graphs are used for designing and optimizing feedback loop compensation circuits. For a given OpAmp chip, the objective is to achieve the maximum gain-bandwidth for an unconditionally stable hybrid thick-film OpAmp, when the bare chip itself is potentially unstable. Transfer-function modeling is used in control system design and has been in use in top-down filter synthesis and design since very early in the 20th century.

Transfer functions can be developed *bottom-up*. That is, from a circuit analysis of the detailed network. But system level transfer function modeling is the usual starting point for analog *top-down* design [81]. Once the transfer function is derived from system constraints, there are various was to physically implement it, such as the Cauer realization for a filter. Also, once the transfer function is derived, it can be *decomposed.*[15] The overall system function-block is usually referred to as a *macromodel.* This block can often be split down into simpler, more physically realizable blocks, such as into an amplifier gain block (function) followed by an interstage filter block (function).

The decomposition of an analog system transfer function and its implementation in hardware is explained well in [81]. In addition to analog systems, there are mixed analog-digital systems employing sampled data.

8.10.2 Transforms in Mixed Technology – Mixed Signal Systems

Consider a gurney carrying a patient into a Magnetic Resonance Imaging (MRI) scanning machine. The gurney is physically moved by a set of electromechanical servomotors driving appropriate mechanical gear trains. There may be hydraulics involved, particularly if a certain amount of inertia in table motions have to be taken into account. The motors will be driven by some kind of low frequency analog electrical power circuit with amplifiers, and filtering. Feedback regarding positioning in the MRI machine will involve a laser-optic, high frequency RF detection link. Data from the positioning link will be converted to digital for processing in a digital circuit controlled by a software program. Control signals from the digital computer will be fed-back to the servo loop to move the patient back and forth into perfect alignment for their test scan. A human operator observing through a TV display will probably intervene in the process.

Long before the first piece of hardware or software is designed and built, this entire system can be, and usually is, modeled and simulated as a virtual design in a computer. Top-down system level design is rapidly advancing just as systems are rapidly becoming more complex

The system design is decomposed down into working on specifically analog, digital, software, communications and communications interface portions. For a digital portion, the IBIS model addresses the analog behavior resulting from fast switching digital signals. VHDL-Verilog (Verilog is an alternative to VHDL) addresses the timing analysis of the digital circuits.

---

[15] For instance, the Heavyside Expansion Theorem method.

8.10.3 Transform Methods for Gigabit Serial Nets

Transform methods convert a mathematical problem from one frame of reference (such as space-time for distributed networks) into another (such as frequency) where the problem is often easier to solve. This process is usually followed by an inverse transformation of the solution back into the original frame of reference. For example, a Laplace transformation of a differential equation, derived from circuit analysis, reduces a time-domain calculus problem into an algebraic (s-domain[16]) problem where it is easier to solve.

A lumped-element circuit equation in the time domain usually results in an ordinary differential equation in time (variable t, as in dV/dt). A distributed-element circuit, typically a transmission line, equation in space and time usually results in a partial differential equation. Examples are variables x and t, as in $\partial^2 V(x, t)/\partial x^2$ in the transmission line equation. Using Laplace transforms of the partial differential equation, for a transmission line in the time domain converts it into the s-domain. There it becomes an ordinary differential equation with respect to x [145]. Thus, formulations of transmission line equations, lossless or lossy, can be readily solved using transform methods.

Above 1 GBIT, S-Parameters (a frequency domain model) provide the best representation of (frequency-dependent) lossy, dispersive interconnections and their impedance discontinuities. Full-Wave solvers are often required in deriving and extracting accurate S-Parameter models for lossy, dispersive interconnections, even for Signal Integrity design.

The task in HSDD, however, is to compute the transient, time-domain response of signals using the frequency-domain S-Parameters. Transformation can be applied to these S-Parameters models if they are linear to bring them into the time-domain. If the models are non-linear, one method is to use the Impulse Response of the network, which is convolved, with the specific spectrum of the signals (direct and reflected) to derive the results.

Other mathematical methods, such as the Harmonic Balance and Envelope Detection methods, are more commonly used in RF, microwave, and EMI/EMC design. In those circuits, the frequency response of signals, wavelengths, and resonance modes are more informative. Because Signal Integrity, Power Integrity, and EMI/EMC are challenges for HSDD, a designer of GBIT serial SERDES nets has to be informed of time-domain, frequency-domain, and transform methods [4].

---

[16] S-domain is distinctly different than S-Parameter. See Laplace transforms in Table 8-11.

Table 8-11 summarizes the various transform models and methods.

Table 8-11. Transform methods

Transform Name	**Definition of Transform Type**	**Transforms/Inverse-Transforms (variable range)**
Fourier	Frequency Domain	$F(\omega) = \Phi[f(t)] = \int f(t)e^{-j\omega t}dt$ ($t = -\infty \rightarrow +\infty$)
		$F(t) = \Phi^{-1}[F(\omega)] = 1/2\pi \int F(\omega)e^{-j\omega t}d\omega$ ($\omega = -\infty \rightarrow +\infty$)
	Comment: Expresses a complicated time-domain function as a frequency spectrum with magnitudes and phase.	
Laplace	Complex Frequency Domain	(s-plane, $s = \sigma + j\varpi$) $F(s) = L[f(t)] = \int f(t)e^{-st}dt$ $t = 0 \rightarrow \infty$ $f(t) = \left(\frac{1}{2\pi j}\right)\oint F(s)\ell^{st}ds = L^{-1}F(s)$
	Comment: Converts differentiation and integration into algebraic operations. Models for R, L, C and G map directly into the s-plane. Very useful for continuous transfer functions and closed loop system stability analysis.	
Z (a bi-linear transforma-tion from/to the s-plane)	Complex Frequency Domain for time Sampled Data Signals	(z-plane, $z = e^{Ts}$) $F(z) = Z[f(t)] = \sum f(nT)z^{-n}$, $n = 0 \rightarrow \infty$
	Comment: Additional uses for continuous and sampled data transfer functions and closed loop system stability analysis. Nyquist plots are used.	
W (a bi-linear transforma-tion from/to the z-plane)	Complex Frequency Domain for time Sampled Data Signals	For communications (w-plane, $w = (z-1)/(z+1)$) $z = (1+w)/(1-w)$ For control systems (w-plane, $w = (2/T)(z-1)/(z+1)$) $z = (1+(wT/2))/(1-(wT/2))$
	Comment: Enables and eases use of Bode diagram and Nichols chart for continuous and sampled data closed loop system stability analysis.	

Now that the authors have introduced transform methods as relevant to HSDD, we will say a few more things about transform methods.

A mathematical frame of reference is called a domain, as in space-time domain and frequency domain. A mixed domain uses more than one frame of reference simultaneously. Mixed domain techniques for modeling and simulation are becoming popular in EDA tools.[17]

[17] The Advanced Design System (ADS2004A)–E8885A Convolution Simulator–from Agilent EEsof EDA is one such tool.

High-speed switching is a transient, non-linear, large-signal problem and it is best handled in the time domain. Simultaneously, high frequency effects such as conductor and dielectric losses, and dispersion, are best modeled in the frequency domain. This is because they are frequency dependent in ways that often involve non-linearities. Losses, dispersion, and decrease of dielectric constant increase with frequency. S-Parameters are the method of choice for such frequency dependent models. The S-Parameters for lossy networks become complex variables, with attenuation and phase information. In other words, the variable contains a real number term and imaginary number term, $\alpha + j\beta$.

Also, to *convolve* something literally means to *fold* it. In this case, the mathematics folds, or maps, the impulse response of the S-Parameter models out of the frequency domain and into the time domain. The convolved impulse response is the transfer function of the network in the time domain. When the convolved function is multiplied by the input waveform function, the output response is produced [4, 146, 149]. Initially, before reflections occur, the switching signal out of the driver sent towards the receiver is the input signal.

With data rates well above one GBIT, the entire signal path from driver output pad to receiver input pad must be simulated to get waveforms, timing delays, and timing margins correct. Both pads are inside their respective IC packages.

To show what can be accomplished with today's sophisticated EM simulators and S-Parameter model extractors, consider a 25 GHz connector. Figure 8-11 shows a partial assembly view of a proof-of-concept 25 GHz connector. The connector was modeled with Sonnet Software's CST Microwave Suites® 3D modeler. This software can extract a full set of S-Parameters: S_{11}, S_{22}, S_{12}, and S_{21}. There are two center pins, each surrounded by a ring of return signal pins, in a side-by-side arrangement. The pin diameters are 2 mils. The impedance of this arrangement was 51.34 Ohms and showed little variation, plotted on a Smith Chart, over the frequency range of interest. Figure 8-12 shows one result, the $|S_{11}|$ magnitude in dB, extracted by the modeler from this arrangement.

The S-Parameter model for this 25 GHz connector can be used in a time domain simulation of a multi-gigabit switching network by employing transform and convolution methods.

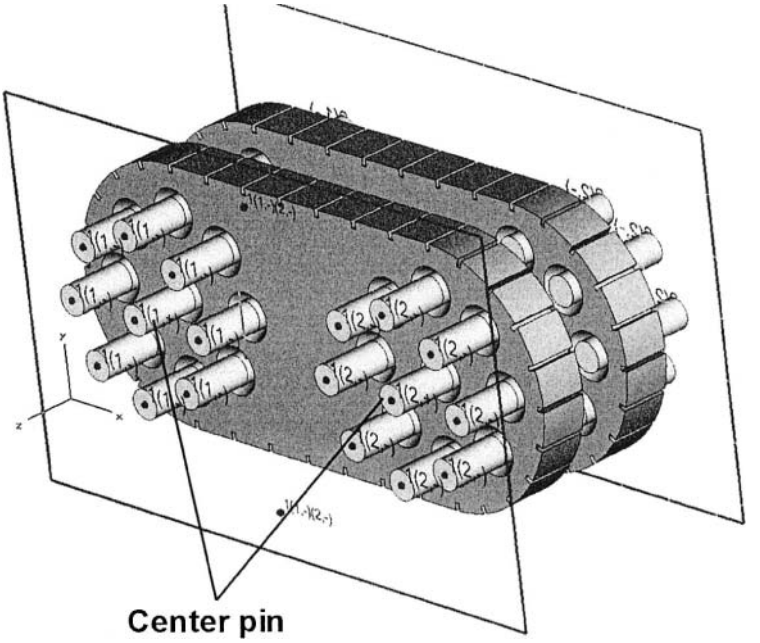

Figure 8-11. Test connector for 25 GHz signal.
Used with permission of Cinch Connector

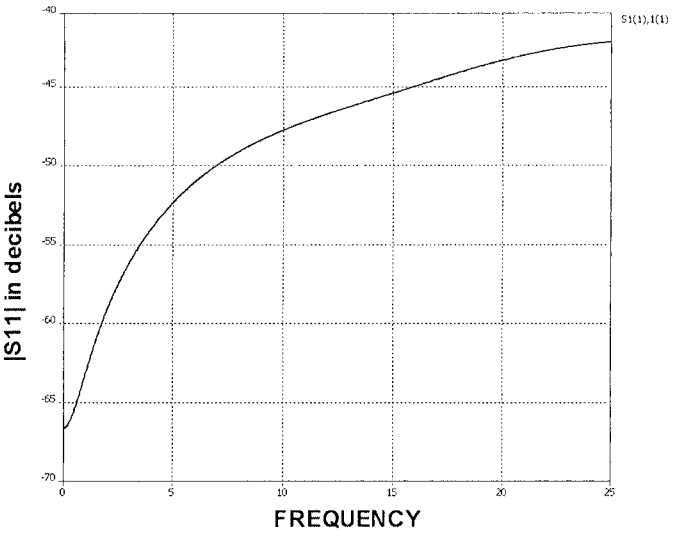

Figure 8-12. $|S_{11}|$ magnitude in dB for the Cinch 25 GHz connector.
Used with permission of Cinch Connector

8.11 SUMMARY

After selecting the best component to use in a design, the next step is to choose the best model to analyze how that device will behave in the circuit. There are several types of models to choose from. Each model type has certain strengths and weaknesses for the necessary analysis tasks. The major types are SPICE, IBIS, and S-Parameters models. SPICE models are very complex and good for tracking physical effects. IBIS models are good for PCB-level high-speed switching on interconnections, but do not track bias changes. S-Parameter models handle high-frequency effects well, but do not model time-domain switching well. Mixed-domain simulations, involving time, frequency, and Laplace mean a combination of models will be used. Mixed analog-digital simulations will involve W- and Z-Transforms.