 By Robert J Haller, Compaq Computer Corp

ANALYZING SIGNAL INTEGRITY IS NOT LIKE GAZING INTO A CRYSTAL BALL OR SHAKING BONES OVER A DESIGN TO DETERMINE ITS VIABILITY. YOU MUST IMPLEMENT A SET OF TOOLS, SOFTWARE, AND REPORTING MECHANISMS TO DETERMINE WHETHER A DESIGN IS ACCEPTABLE TO SHIP.

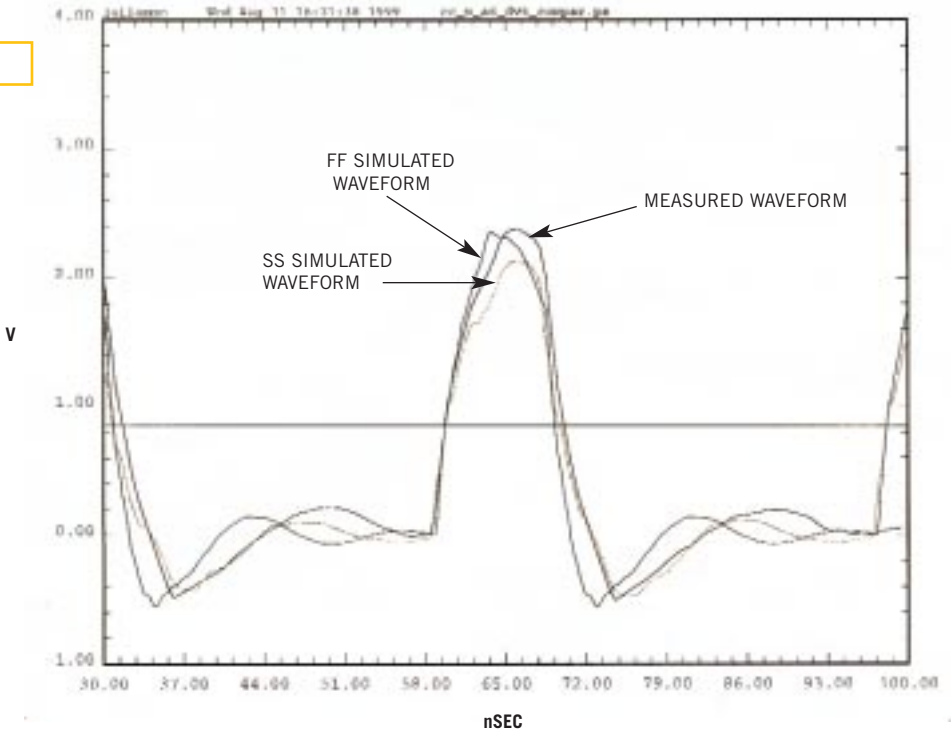
The nuts and bolts of signal-integrity analysis

As the frequency and complexity of high-performance system designs increase, signal-integrity analysis becomes exceedingly complex. High-performance challenges include 1-GHz processors; edge rates of less than 100 psec; large numbers of unique interconnected pc boards; ASIC packages; and many signaling technologies, including high-speed transistor logic (HSTL), low-voltage differential signal (LVDS), positive-emitter-coupled logic (PECL), open drain, PCI, and low-voltage transistor-transistor logic (LVTTL). Accurately predicting system performance and avoiding signal-integrity issues requires attention to details and sophisticated automated processes. Fortunately, you can use various processes and techniques to meet the challenge and obtain success in the first pass of all of your high-speed system designs.

Signal-integrity analysis aims to ensure that worst-case simulations for every analyzed network in a system encompass actual measured data. The waveforms in **Figure 1** compare measured results with simulations of a complex multidrop network. The signal traverses packages, series resistors, connectors, and a long length of etch. According to the **figure**, the measured waveform sits within the envelope of the two simulation waveforms.

To achieve this goal, dedicated and experienced engineers work to ensure that the electrical design of the pc boards meets the system-design requirements. These engineers must use robust automated signal-

Figure 1



The goal of signal-integrity analysis is for the actual measured result to sit within a widow of worst-case simulations.

integrity-design and -verification processes and accurate simulation models to ensure signal quality across all high-speed digital signals, clocks, the power distribution in pc-board modules, and interconnects. You must employ schematic-capture tools and circuit simulators for both pre- and post-layout simulations and make early decisions regarding technology and package types, signal-to-return ratios, network treeing, topology, and termination.

Next, use automated CAD tools to verify the signal nets prior to module release (**Figure 2**). These tools must include Spice and behavioral I/O-buffer-information-specification (IBIS) simulation for generating wire delays. Modeling the multiboard environment is crucial if high-speed signals traverse connector boundaries. You should generate wire-delay files and feed them into a static-timing tool to verify the module performance as well as the chip I/O performance.

Spice is the industry standard for electrical simulation because of its accuracy and the availability of free source code. Behavioral simulators have the advantage of fast runtimes, sometimes an order of magnitude faster than Spice. Some behavioral simulators also have extremely powerful "what-if" capabilities, which make these simulators attractive to both beginners and experienced signal-integrity engineers. Theoretically, models will abundantly become available that will further behavioral-model growth. Spice and behavioral simulators each have advantages. Behavioral simulators can analyze and report positive and negative overshoot, nonmonotonicities, and ring-back by virtue of excessive wire delays much faster than Spice. But Spice's accuracy is necessary when you're analyzing 1-GHz processors with little timing margin. It is essential to analyze module-level crosstalk using one of the many available tools. With the many signal-swing, etch-width, etch-spacing, and receiver-susceptibility issues, the crosstalk problem is too big to manage with simple wire and spacing rules.

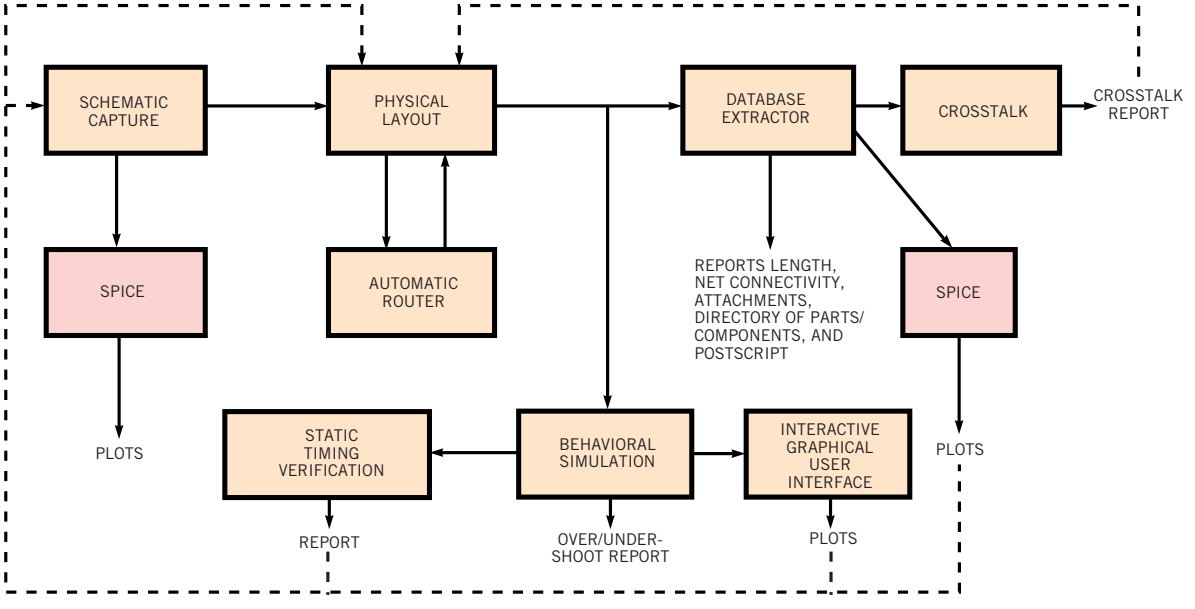
Using an automated process to extract net configurations directly from layout to feed into Spice prevents manual intervention and the possibility of human error. Although time constraints may prohibit running 1000 network boards through Spice overnight, having the automated-extraction capability enhances your ability to quickly examine critical networks on the module and frees the critical signal-integrity engineers to examine and solve real problems instead of hand constructing and disassembling Spice decks. Most companies that take signal integrity seriously perform end-to-end Spice simulations of representative critical nets and use multiline complex-package and connector models. This approach lets you look at waveform behavior from the input of the output cell to the output of the input cell.

In addition to having accurate models, a robust process flow, and talented signal-integrity engineers and module designers, you must also set up a formal module review process and strategy. Hold module reviews at least twice during the design process and at least once when the schematic set stabilizes. These reviews allow the signal-integrity engineers and their peers to review any prelayout Spice simulations, to ensure the proper implementation of the clocking system and to double check any reset or power-sequencing requirements. A final layout review should be a gate to release the module. Investing time and effort into the tools, the analysis, and the review process will simplify the design-verification process.

PLAN, PROCESS, AND PERFORM

The work necessary to perform a thorough signal-integrity analysis generally comprises the planning phase, the processing phase, and the performing phase. Planning the work is the first step. You need to figure out how many modules to analyze and how to analyze them. High-

Figure 2



The signal-integrity process flow includes schematic capture, physical layout, Spice, and behavioral simulation.

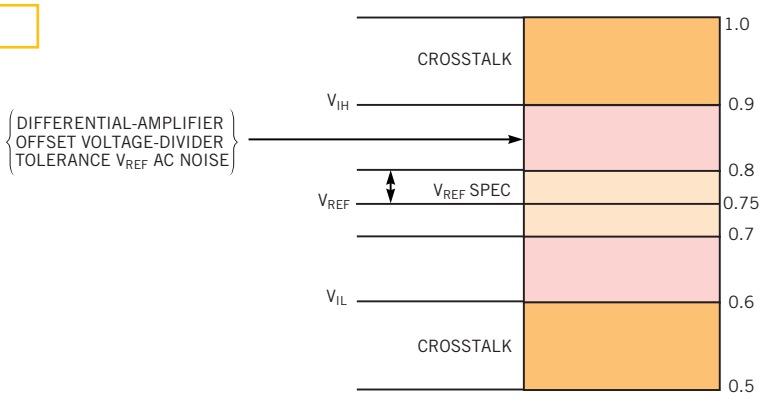
speed modules, such as CPUs, require Spice and behavioral simulation, timing and crosstalk analysis, and a manual review of both the schematic and layer plots. On the other hand, a console control-panel module may require only crosstalk analysis, limited behavioral simulation, and a manual review. Define the requirements for each module, then devise a plan and execute that plan.

Your work should focus on the system-design area that has the least amount of margin. Thus, you must plan for a margin assessment. In addition, perform a signal-to-return-ratio analysis for the package and the connectors. Armed with a map of the modules and the type of analysis they require, you can plan your engineering- and computing-resource requirements.

You will spend a lot of initial time obtaining, debugging, and verifying models. You should not underestimate the importance of model verification for accurate system analysis and, ultimately, problem-free system operation. Toward the back end, data management and revision control will consume a lot of your time. Use foresight to save time and effort during the project. Considering and properly implementing something as simple as a signal-naming convention can save time over the project design's life. Plan on one engineer for two modules, depending on the modules' complexity, and give the most complex design object to your senior signal-integrity engineer. Each engineer should have a Spice license, and at least half of the engineers should have behavioral-simulator licenses. A senior-level engineer should oversee all the modules, play backup, and work out any kinks in the process and in the models.

Make sure to document required information from each module designer and track the model-development process. Early parts lists and snapshots of the design database, including both the schematic and physical layout, are crucial to keep the signal-integrity analysis on the program's design and production schedules. You can avoid impacting the program schedule by performing early signal-integrity analysis in parallel with the design work. Obtaining models and creating process flows and up-front analyses in the early design stages minimizes the impact of signal-integrity analysis on the product-development schedule. With proper planning, appropriate resources, and a robust CAD process, the signal-integrity portion of the module-design process can take less than a week. This one-week time frame assumes that the up-front data is ready about a month before the pc board's scheduled release and that signal-integrity analysis parallels the module-design work.

Figure 3



A sample ac-noise margin for high-speed transistor logic includes V_{REF} variations, V_{IH} and V_{IL} variations, and crosstalk variations.

CREATE STANDARDS AND WIRE RULES

You must create documentation,or signal-integrity standards, for all the engineers on a project to follow. These standards should specify all assumptions and methods for signal-integrity analysis and define module-level I/O electrical parameters. Use these parameters to drive a set of simulations. Then, employ the simulation results to create a set of rules for constructing the modules (layup), for connecting the networks (layout), and for driving and terminating signals.

The standard should cover simulation corners and corner-definition rules. Ultimately, you want to perturb simulation parameters that affect hardware performance from their nominal values in a way that exacerbates certain extreme electrical behaviors. These behaviors include worst-case signal ringing (referred to as an FF corner) and worst-case signal delay (referred to as an SS corner). The initials FF and SS refer to the characteristics of the p- and n-channel FETs, respectively. That is, for an FF I/O cell simulation, you use an F, or fast, p-channel FET and an F, n-channel FET. An SS simulation means that both FETs are slow during the simulation. The initials also describe the overall simulation corners. For example, an FF simulation means not only that you are using fast p- and n-channel FETs but also that the voltages and temperatures during the simulation result in the fast circuit performance. So, FF means fast silicon and a fast environment.

Sometimes modules contain both CMOS and bipolar devices. It is important to characterize the bipolar models over temperature and characterize the supply-voltage ranges to determine how they compare with CMOS. You should put together tables of corner-definition rules that summarize how parameters must vary as a group for different types of drivers. **Table 1** is an example of a corner-process-parameter table.

You must modify the simulation-analysis thresholds from their typical or specified values to account for various ac- and dc-noise sources—such as background noise—that the simulation can't account for. You should add ac crosstalk limits to dc requirements to ensure adequate signal integrity and network performance.

You should run crosstalk software on every pc board in the system to ensure the maintenance of acceptable spacing rules between signals. Because CMOS

LVTTL voltage amplitude is significantly higher than HSTL voltage amplitude, you should observe careful module layout and verify the layout to ensure acceptable ac noise margins. Using Spice, you must obtain a simultaneous switching penalty and verify that the noise margin is adequate. Further, you must determine a simultaneous switching penalty for each package type and add or subtract this penalty to the timing margin. The automatic module-level timing verification of the chip's I/O pin must include this penalty. **Figure 3** shows a sample ac-noise margin for HSTL.

Figure 4 shows a plot of the backward crosstalk coefficient Kb versus line spacing for a typical CMOS module layup. Note that the crosstalk dramatically increases as line spacing decreases. For a nominal 5-mil line with 10-mil spacing, Kb is about 11.5%. If you decrease the spacing to 5 mils, the nominal backward crosstalk increases to almost 19%. Thus, if you have a 3.3V signal routed at 10 mils aggressing an HSTL signal, you end up with $3.3V \times 0.115 = 0.379V$ of noise, which represents the typical conditions in the worst-case direction. This amount of noise is considerably more than the entire HSTL noise margin of 0.25V that exists between V_{REF} (0.75V) and margined V_{IH} (1.V). (See chapter 4 of **Reference 1** for more information on calculating crosstalk.)

TABLE 1—CORNER VALUES FOR CMOS AND BIPOLAR DEVICES

Parameter	SS corner	TT corner	FF corner
V_{DD} (V)	3.0	3.3	3.45
V_{CC} (V)	4.5	5.0	5.5
V_{DDQ} (V)	1.4	1.5	1.6
V_{REF} (V)	0.70	0.75	0.80
Bipolar temperature (°C)	0	27 + T_{JA}	100
CMOS temperature (°C)	100	27 + T_{JA}	0
LVTTL impedance (Ω)	55	60	65
Bipolar, HSTL, and clock impedance (Ω)	45	50	66
Propagation delay (psec/in.)	0.200	0.175	0.160
Lengths	minimum	average	maximum
Termination resistance, series	maximum	nominal	minimum
Termination resistance, parallel	minimum	nominal	maximum

MODELS, MODELS, MODELS!

Accurate models of semiconductors, passive components, and parasitics are necessary to create accurate circuit- or system-level simulations. Conversely, bad models will result in bad electrical-network-simulation results. For this reason, bench verification of circuit models is crucial prior to shipping any product boards for fabrication. You should count on initially spending about half of your time obtaining, debugging, and verifying simulation models. Signal-integrity engineers spend half of their time as modeling engineers.

Start obtaining semiconductor models early, so you have time to verify that your technology and part selection are adequate for your system-design performance requirements. As soon as module designers know what part families the design uses, signal-integrity engineers must obtain, debug, and verify IBIS and Spice models. Pay attention to the models of all of the passive components in your system. It is important to correctly model connectors and packages. If your edge rates are less than 500 psec, you should probably use lossy transmission-line models.

Do not let a module designer use components that have unavailable simulation models unless extenuating circumstances exist. Unfortunately, as many engineers have discovered, collecting Spice models from third-party semiconductor vendors is one of the most difficult tasks a signal-integrity engineer undertakes when performing system-level analysis.

COMPARE MODELS WITH REAL BEHAVIOR

It is paramount to compare your model with real measurements. Even with due diligence and accurate modeling techniques, expect some imperfections because of underlying limitations in the behavioral-modeling algorithms and specifications as well as in the measurement techniques. After you look at the model in the lab, go back to the vendor to resolve any discrepancies. To remedy problems with model accuracy and build confidence in a model, be sure to develop sound methods to generate and validate the models. Some system-design companies have been doing this work themselves for decades, but it has never been cost-effective. The Signal-Integrity (SI) Reflector is an open Internet forum in which engineers discuss signal-integrity issues, including model accuracy and model verification. This forum highlights the need for system vendors to create and verify their own models. The availability of accurate models should eventually reduce the the engineers' burden of routinely performing this work.

Also, a group of engineers from various companies are working on the IBIS accuracy specification. This specification attempts to get semiconductor vendors—usually the model originators—to verify their models using a predefined set of test loads on the bench and then to document those results in a consistent format for their semiconductor customers. The specification should reduce the daunting task that signal-integrity engineers presently face: to ensure not only that all of the parts in a system operate reliably over the life of a product but also to make certain that the models actually represent the true devices (**Reference 2**).

Some engineers advocate the use of models that you derive exclusively from one part in the lab. Unfortunately, models developed from lab data cannot accurately represent process and temperature corners. Also, adequately comparing bench waveforms to Spice requires data-acquisition software and a waveform viewer that can overlay bench and simulated waveforms on the same plots. This procedure eliminates time and voltage-scale differences, which would affect your ability to ascertain whether the correlation is acceptable.

MEASURING SEMICONDUCTOR MODELS

Ensuring first-pass design success requires that you spend time verifying that the results of your simulator using semiconductor models match the bench-test results. You must use appropriate techniques when obtaining bench measure-

ments including high-bandwidth scopes and ensure that the scope's bandwidth is adequate for the signals you are measuring. Assuming a Gaussian edge, a simple relationship exists between the 10 to 90% rise time of a signal and its frequency content (**Reference 3**):

$$T_{RISE} = \frac{0.338}{F_{3dB}}.$$

If the aggregate bandwidth of the oscilloscope and the probe is not high enough for the rise time in question, the test equipment will attenuate high-frequency components of the waveform, and the measurement will be in error. The following equation expresses the measured rise time as a function of the "true" rise time, the oscilloscope bandwidth, and the probe bandwidth (**Reference 3**):

$$T_{MEASURED} = \sqrt{T_{RISE}^2 + \left(\frac{0.338}{F_{3dB-SCOPE}}\right)^2 + \left(\frac{0.338}{F_{3dB-PROBE}}\right)^2}.$$

You can use the following simple rule of thumb for a quick check:

$$BANDWIDTH\,(SCOPE + PROBE) \geq \frac{1}{0.338 \bullet T_R}.$$

For example, if your fastest edge rate is 0.285 nsec, you must have an effective bandwidth of your scope and probe of 1.2 GHz or you will be missing information. Also, if you have little glitches that are faster than your edges, you won't see these glitches if the scope does not have adequate effective bandwidth.

CREATE TEST VEHICLES FOR THE MODELS

Creating simple test vehicles to verify all of your models will help you ensure accurate modeling without wasting valuable design resources. Although these test vehicles are important, they have a high cost of engineering manpower, capital equipment, and scheduling. With a little cleverness, you can integrate multiple verification tests into one test vehicle. For example, you should place test traces of varying lengths on the same module as the connector. Use an identical layer stack-up as your system board and make sure to include test structures to look at the various parasitics of your network path, such as vias, pads, and dispersions.

For example, put two traces on your test board, with identical fixturing at each end, using SMA connectors that easily mate with a high-speed scope. Then place 20 vias of the design's predominant size in one of the traces. Measure the characteristic impedance of the test trace and the difference in delay between the trace with and without vias. Using the telegraphers' equations, $Z_0 = \sqrt{L/C}$ and $T_{PD} = \sqrt{LC}$, you can calculate the effective capacitance from the impedance and the change in the propagation delay (**Reference 1**). Make sure to measure the propagation delay at a representative rise time and not at the time-domain reflectometer's (TDR's) rise time.

$$C = \frac{T_D}{Z} = \text{TOTAL CAPACITANCE}.$$

Divide the total capacitance by 20, or the number of vias, to accurately measure the capacitance of a single via.

Connectors are a common problem in system design for both electrical and mechanical engineers. You can design a simple connector test vehicle to aid in the characterization. You should measure characteristic impedance, propagation velocity, crosstalk, and the effects of simultaneous switching on delay. You can accomplish this measurement by wiring equal lengths from an SMA to the connector. Take care to minimize, if not eliminate, module crosstalk so that you can independently examine connector crosstalk.

MEASURE CONNECTOR CROSSTALK

You can use several techniques to measure connector crosstalk. The multiple-active testing technique and the single-active testing technique have excellent correlation. Using the multiple-active technique, you simultaneously switch many lines in a connector and measure the impact on a victim under a specific set of edge-rate and signal-swing conditions. Using the single-active technique, you stimulate one line and measure the impact on the surrounding victims. Adding up the victim's voltages and using the principal of superposition provides a total crosstalk number. This approach is equivalent to the multiple-active approach, in which fixturing is more difficult to perform. Using edge rates equal to or greater than 250 psec, the principal of superposition applies to connector crosstalk.

Table 2 shows an example of this ap-

TABLE 2–CONNECTOR-CROSSTALK VALUES USING THE SINGLE-ACTIVE TECHNIQUE

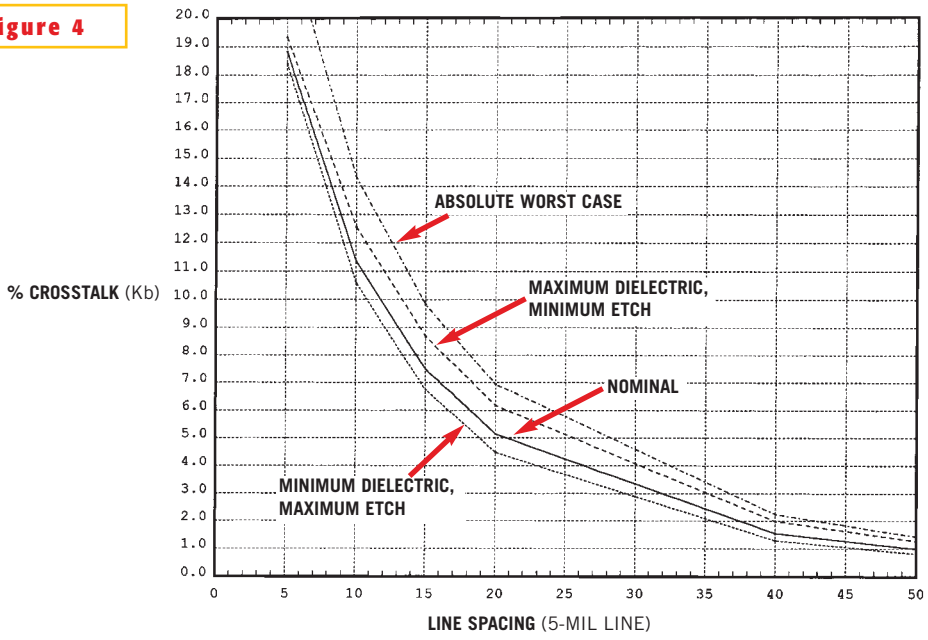
Victim nodes	Aggressor nodes							
	A_1TO1A1	A_1TO1A2	A_1TO1A3	A_1TO1A4	A_1TO1C1	A_1TO1C2	A_1TO1C3	A_1TO1C4
A_1TO1A1		11.50	5.90	3.75	7.30	3.75	2.30	1.80
A_1TO1A2	11.50		10.50	5.00	4.40	4.20	2.90	2.20
A_1TO1A3	5.90	10.50		9.20	3.00	3.10	3.90	2.80
A_1TO1A4	3.75	5.00	9.20		2.10	2.10	3.10	3.90
A_1TO1C1	7.30	4.40	3.00	2.10		6.25	2.40	1.40
A_1TO1C2	3.75	4.20	3.10	2.10	6.25		5.40	2.00
A_1TO1C3	2.30	2.90	3.90	3.10	2.40	5.40		5.00
A_1TO1C4	1.80	2.20	2.80	3.90	1.40	2.00	5.00	
Sum (mV)	36.30	40.70	38.40	29.15	26.85	26.80	25.00	19.10
Percent of signal swing (Kb)	18.62	20.87	19.69	14.95	13.77	13.74	12.82	9.79

Notes: T_{rise}=250 psec.
Step size=195 mV.

proach. The first row contains the names of the aggressor lines and the first column contains the names of the victim lines. You aggress one line and record the resultant response of the victim in the appropriate entry. Make sure you implement the appropriate grounding pattern in your connector test vehicle. It is also imperative to measure crosstalk and propagation delay at the fastest edge rate you plan on using in your system.

After you create and measure the electrical parameters of your connector, compare these parameters with your requirements for simplified connector selection. Many types of shielded connectors require a 3-D field solver unless the connector vendor provides a validated model near your required edge rates. Compare simulation with measurements to understand the accuracy of the model. Then, write a specification that guarantees that your product meets critical electrical acceptance criteria. If possible, implement a quality-assurance audit at the vendor site to ensure that the product continues to meet the criteria throughout its life. When using ASIC packages, you should also create all of the models and test vehicles and perform the verification for connectors. Using a verified, accurate vendor model can save you time and resources.

Figure 4



In a typical CMOS module, crosstalk increases dramatically as line spacing decreases.

IMPLEMENT CLOCK FORWARDING

Many design teams now use the clock-forwarding, or source-synchronous-clocking, technique. When using this technique, you simultaneously transmit data and clock from the same source chip, delay the clock half of a bit time, and then receive the clock and data at the destination. This technique has the advantage of much higher bit rates and, consequently, higher clock frequencies. It also enables you to traverse long wires with a minimal system-cycle-time penalty. This approach can also save pins. The key is to match the electrical and physical environment and minimize skew between what the clock and data signals *see* when they traverse their wire. In extremely high-speed applications, this matching applies not only to electrical lengths but also to layers, via counts, and crossovers. Be sure to use point-to-point connections for all high-speed applications. Using both edges of the clock also complicates the design and skew budget. The important contributor is not the total latency but the variation in delay from the various contributors.

To calculate a clock-forwarding budget, such as in **Table 3**, you must first understand all of your variables. Etch delays normally vary from 160 to 210 psec/inch. You can greatly reduce this range if you adhere to strict layout rules because, in this case, the difference between the clock and the data is relatively small. If you implement aggressive layout rules in a unistripline controlled-impedance environment and verify through test-vehicle measurement, you will obtain specifications such as ±1 psec/in. under certain conditions.

However, this etch-delay number comes at a cost. Make sure you really need ±1 psec/in. because it is difficult to balance etch to these tight tolerances. You quickly get into trouble if you use worst-case analysis. When doing clock-forwarding analysis, you must tabulate the delay-variation contributors and the delay variations between clock and data signals. Some of the major contributors include duty-cycle variations, skew between local copies of a clock (on ASICs), threshold mismatches, clock-versus-data delay through an output cell, simultaneous-switching effects, and package-trace skew (both driver and receiver). Other major contributors are etch mismatch between data and clock, dc offsets between transmitter and receiver, receiver-threshold variations to on-chip signal intergrity and noise, delay-line tolerance, duty-cycle symmetry, mismatch across clock-and- data-receiver cells, and clock-loading mismatches.

OUTLINE THE PROCESS

After planning, the next phase of signal-integrity analysis is to develop and debug a robust signal-integrity process (**Figure 2**). You must have an automated set of scripts and tools that allow you to verify a design without a lot of manual intervention. All designs must proceed through this standard process, or flow. Some designs may require additional

work. The time you spend developing a standard flow, libraries, and a revision-control system is essential and well spent. Your designs may change many times, but once you have a verified process and accurate libraries, reverifying design objects is greatly simplified. You must react quickly when designs change, and you must be able to go back to a previous design state. You should implement a robust revision-control scheme on your disks for software, libraries, and design objects. This measure will enable you to determine the root cause of an issue if design object changes and unforeseen process problems occur.

After you determine the general process flow and set up a working directory for each design object that uses a consistent and succinct naming convention, determine which tools each design requires. Your manager can track progress by maintaining a spreadsheet with the design-object revisions and tool requirements. Most of your high-speed designs—such as CPUs, memories, and backplanes or system boards—require Spice, behavioral, timing, clock, crosstalk, and manual checks. Other design objects require a subset of these checks. An I/O module may have high-speed sections that require a complete analysis. The slower speed sections, which may not even have models, require only a subset of the complete analysis. A console control panel that interfaces to the I/O should be sufficient to place series resistors on the outputs of any critical signal, perform a quick crosstalk run, and then manually review the plots.

Evaluate each design object and document the analysis that each object will undergo. When the analysis is complete, the design is ready to ship. This process makes signal integrity more predictable and reliable. Certain designs may require special analysis. For example, a PCI bus may require configuration testing. You can do this testing using Spice or behavioral simulation depending on model availability and design constraints. If you have accurate behavioral models and a good simulator, then a typical design may require only an hour to run through. The most important and effective use of behavioral simulators is for final board verification just prior to shipping etch artwork to the pc-board vendor. An experienced signal-integrity engineer can turn a complete board design in hours or less. If necessary, he or she can implement fixes and feed them back to layout. He or she can quickly uncover common routing, topology, termination and configuration problems and perform the optimization. The turnaround time for running Spice on every net is too long. Designs often change immediately before you release the artwork, and a quick verification of the board is essential to identify any problems that the change may introduce. If you don't analyze every net, you will likely have problems with at least one of them. Find and fix the problems to ensure first-pass success.

The signal-integrity engineer should understand the details of the behavioral simulator and models and their limitations to make appropriate trade-offs. Poorly designed or poorly decoupled power and ground planes on a board can also result in inaccurate results from behavioral simulators and Spice. You must follow good signal-integrity-board design practice for models to accurately represent the circuit in a system.

TABLE 3–SAMPLE CLOCK-FORWARDING SKEW BUDGET

Budget item	Minimum	Maximum	Total
Clock duty-cycle variation (inside source chip)	-200	200	400
Skew between local copies of clock (on ASIC)	-100	100	200
Threshold and delay mismatch of driver output cells	-100	100	200
Edge-rate mismatch between clock-and-data output cells	-200	200	400
Simultaneous switching effects (crosstalk, di/dt, and others)	-100	100	200
Package-trace skew in driver and receiver packages	-50	50	100
Etch mismatch between clock and data	-50	50	100
DC offset between driver and receiver I/Os	-50	50	100
Receiver-threshold mismatch between clock and data receivers	-50	50	100
Receiver V_{REF} variation due to on-chip noise	-50	50	100
Duty-cycle symmetry mismatch between clock and data receiver	-200	200	400
Skew between mismatch of data and clock loading	0	700	700
Total skew range			3000
Worst-case setup		500	500
Worst-case hold		0	0
Skew range + worst-case setup			3500
Note: Minimum cycle time=(skew range + worst-case setup)×2.			7000

APPLY DECOUPLING STRATEGIES

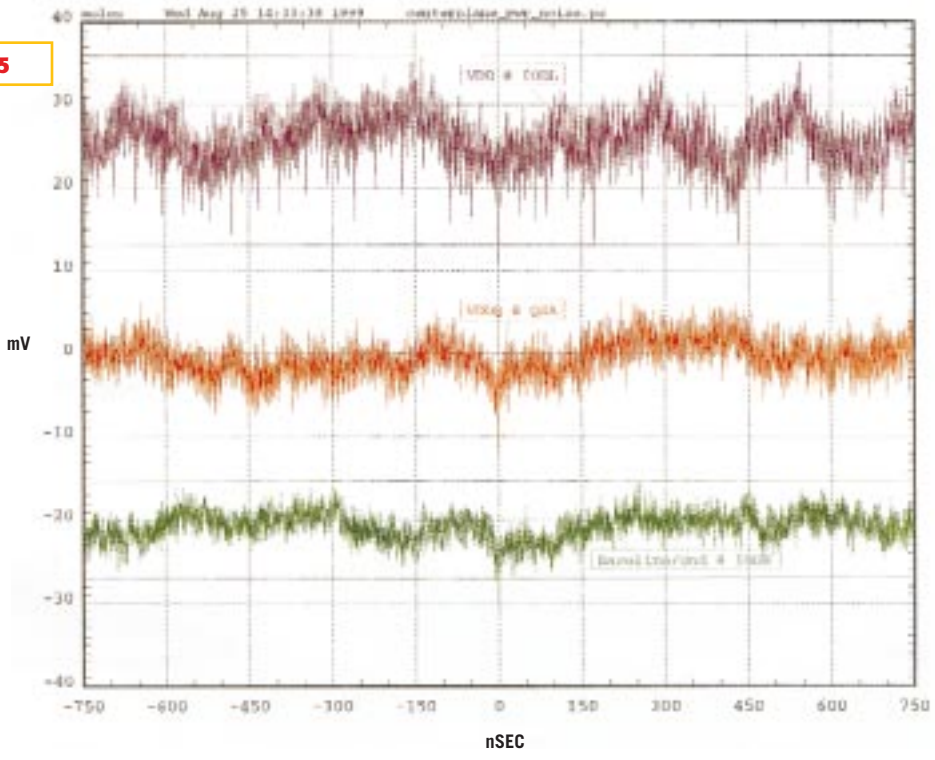
Decoupling is an important aspect of signal integrity. You should use traditional decoupling and bypassing techniques. Follow rules of thumb and measurement techniques including the use of a spectrum analyzer on the final design to avoid problems. If you can afford it, invest in a decoupling expert. The system should use several capacitive decoupling and bypassing methods to ensure clean power to all loads on the modules. You can use the following hierarchy: First, use bulk storage capacitance to provide current to the module when the converter is starting to respond to a load change. Use local decoupling for both the load and the module. These capacitors supply the high-frequency current demands of the local load. You will also require medium-frequency bypass capacitance to roll off the higher frequency switching noise that the switching loads generate. You can add these capacitors on a per-unit area to each power plane for each module. High-frequency bypass capacitance is necessary for closely coupled power to ground planes and provides an intrinsic capacitance within a module with extremely low equivalent series resistance (ESR) and equivalent series inductance (ESL). It is also critical that power planes are tightly coupled to a ground plane. Otherwise, additional low ESL and ESR capacitors may be necessary.

Careful choice of plane assignment can optimize the decoupling. Local charge depletion first occurs in the area between the power planes that are directly under the load because this capacitance is nearly ideal, with minimum self-inductance and resistance. As the charge depletion continues between the planes, the local decoupling capacitors and the current that the medium-frequency bypass ca-

pacitors provide begins to recharge the area between the planes. Finally, current from the bulk capacitors recharges the local decoupling capacitors. The converter recharges the bulk capacitors. A good suggestion for bulk decoupling is to use 100-μF tantalum capacitors evenly distributed across the board. For global, high-frequency decoupling, use one 0.01-μF multilayer ceramic capacitor in a 1206 package every 2 $in.^2$, evenly distributed across the board. For local high-frequency decoupling under the component, use one 0.01-μF capacitor per V_{SS}/V_{DD} pin pair and one capacitor per eight switching outputs, with a minimum of two per component.

As you use more advanced technologies with faster edge rates, maximizing intrinsic board capacitance and minimizing the inductance of capacitor leads and dispersion etch is critical. **Figure 5** shows a differential measurement of the power planes on an active backplane. This backplane has more than 5000 networks, 17 ASICs, 20 layers over seven unique voltages, and more than 100 high-speed connectors. The bottom trace is a ground base line with both ends of the differential probe shorted to ground on a large pad with a maximum deviation of 12 mV. The center trace is the voltage reference for the HSTL differential amplifier (V_{DDQ}) measured at the pins of the ASIC; it varies by less than 14 mV. The top trace is the supply rail (V_{DD}); it varies by less than 23 mV.

Figure 5



Differential measurements of power and ground show that using decoupling strategies results in modest noise levels.

PERFORM MULTIBOARD ANALYSIS

Today, some tools lack the ability to do software simulation across connector boundaries. However, employ this type of simulation if any part of your high-speed design traverses connector boundaries. The simple addition of ½ in. of etch on a dual-in-line-memory-module (DIMM) connector daisy-chained four or eight times can cause a memory design to fail if you don't properly model the connector. The lack of industry tools including this capability is, in part, due to the lack of a connector-model standard within the IBIS standard. But this situation is rapidly changing. A committee in the IBIS open forum is developing a connector spec that should become a standard and give all IBIS-member simulation companies the ability to perform multiboard simulation. With the advent of upgradable memory and processors, the need for connector and package model integration in the signal-integrity process flow is paramount. When you can traverse connector boundaries, you can easily and automatically perform configuration testing.

Realize that all of the simulation tools contain some basic assumptions. You must understand the underlying assumptions as well as the algorithms of the tools. One important assumption to understand about your designs is return currents— particularly their effect on behavioral simulators. If you have not adequately decoupled your design and have not analyzed your tool's assumptions, you may end up with large amount of noise induced on your signals from power planes. Many passive-component suppliers, particularly connector suppliers, provide matrices for you to use in you simulations. Field solvers create these matrices, which contain unique algorithms and assumptions. Some field solvers create loop inductances; others create so-called partial inductances. You should attempt to anticipate the correct answer before you run a simulation. If the results look suspicious, you need to investigate and determine whether you have a tool or model problem or a network problem. By itself, the use of partial inductances, such as those that connector companies often provide, reveals nothing. However, using these inductances to form a loop implements a complete network with a signal path and a return path. Avoid violating any of the tool's underlying assumptions, such as connecting ground to both sides of a connector, or the final answer may be wrong. Manually reviewing plots can verify other underlying behavioral-simulator assumptions. Most behavioral simulators analyze traces as transmission lines using 2-D field solvers. If you route high-speed signals across a power or ground split—a void area or a lack of copper on an adjacent reference plan that

a high-speed signal can cross—voids the analysis. Carefully check the plots for this type of problem and other assumptions that invalidate the tool's results.

PERFORM THE ANALYSIS

To describe signal-integrity analysis as "pushing the button" oversimplifies the final verification. Significant effort by software and hardware engineers allows you to walk up to your workstation and type in DO_IT to run the analysis.

Because behavioral modeling is so important, final verification requires other tricks. You need to create simplified package models from the fully coupled Spice models. Running Spice on a complete module can take days, so your post-route verification needs to include critical parasitics that don't overburden your run times. If you automate the topology extraction, batch submission, and delay extractions from Spice , you can set up the multiple day simulation to run in parallel with the behavioral simulation and the final-artwork-check procedure.

You can achieve significant schedule advantages by doing some behavioral simulation using the following simplification. Some behavioral simulators cannot deal with discrete RLC circuits. Thus you need to model RLC circuits with transmission lines. The trick is to select a delay value (or T_{PD}) that is a small percentage of your overall cycle time. Then, construct a behavioral model that duplicates a Spice one-pin package model using ideal transmission-line segments. In the behavioral simulation, the model becomes a drop-in replacement.

You can again use the telegrapher's equations from **Reference 1** to calculate equivalent impedance for a given inductance/capacitance and a short delay, such as 10 psec. For example, to create a simple pie network with a 3.4-nH inductor and a 1.0-pF capacitor, you need to calculate transmission-line impedance Z_0 as follows:

$$Z_0 = \frac{L}{T_D} = \frac{3.4\,nH}{0.010\,nSEC} = 340\Omega,$$

and

$$Z_0 = \frac{T_D}{C} = \frac{0.010\,nSEC}{1.0\,pF} = 10\Omega.$$

Thus, in your behavioral model, you can insert a transmission line of $Z_0=340\Omega$ and $T_{PD}=0.010$ nsec to replicate the 3.4-nH bond-wire inductance. You can also insert a transmission line with $Z_0=10\Omega$ and $T_{PD}=0.010$ nsec to replicate the 1-pF capacitance of a typical via.

REVIEW REPORTS, FIX PROBLEMS

Once you have the tools to run module and system-level behavioral simulations, you should be able to output a report that contains overshoot (both negative and positive), nonmonotonic behavior, and wire delays. Then review and fix all of the violations or obtain formal waivers from your formal signal-integrity pass/fail criteria. Justifying waivers is cumbersome and difficult, so fixing the problems is usually simpler. By fixing the problem, you may be able to design in additional margin that will minimize the effect of minor inaccuracies in some of the underlying assumptions, models, and processes. An error-free log file of overshoot, nonmonotonic behavior, and wire delay is one of the pass/fail criteria.

Next, you can feed the wire delays into your static or dynamic timing verifier. This step will enable you to uncover any timing problems or marginality in the design. The timing verifier catches any multithreshold crossings or nets with excessive length. Due diligence is necessary in examining intermediate and final log files to ensure that you analyze every targeted network in the design. Review clock-skew tables, multicycle paths, and ignored networks with the key designer of that module. Using the necessary and sufficient criteria when performing your back-end checks will assure you of first-pass success. You can also look at global slack reports, which are detailed listings of timing margins that static-timing-verification software tools automatically create. These reports determine, to some degree, how to speed your design for midlife kickers, or those design upgrades that come shortly after new-product release and are usually associated with a CPU speed up. Long networks with many contributors—components in the path, etch traces, connectors, and semiconductors—may end up near the top of the slack-report list if modeling techniques are too conservative. The same engineer that created the models implemented in the process flows and must run the tools before he or she reviews and signs off on the design. If you are modeling packages or connectors with simple single-line models, you need to include a simultaneous switching penalty in the timing-verification phase.

Ultimately, timing verification, behavioral simulation, and crosstalk are interrelated, and more robust tool suites that interact are necessary. The most challenging tasks that signal-integrity engineers will face in the immediate future are power and ground-plane, distribution, and package modeling. You can avoid crosstalk problems with simple spacing rules. However, you must use a robust process to analyze the design at the end of the design cycle. Crosstalk problems are difficult to track down in the lab because they can depend on data patterns. Thus, you are better served by avoiding the problem. Run a full post-route or multimodule crosstalk analysis on your finished pc board. This analysis is another one of the pass/fail criteria prior to etch release. You can even run this analysis on simple slow-speed boards using default logic-family edge rates and signal swings.

As timing becomes more difficult and frequencies increase, you will need to run more Spice. Remember that Spice is a behavioral simulator and a mathematical representation. Still, it is the most widely used circuit simulator. After you have weeded out all of the real design flaws, it is time to attack the other violations. You should autoextract network topologies from your board database. A few good autoextraction tools exist. You can then run Spice on a subset of the entire board. You can merge and override behavioral wire delays with Spice delays to obtain a clean timing run. It is important to create and verify timing models, but services are available to accomplish this task for as little as $1000 per board. Because most board designs have a limited number of clocks, Spice is an excellent way to ensure accurate modeling of the most critical parts of your design.

DO A MANUAL CHECK

Before you approve the design for release, perform a manual layout check.

You need to look closely at the design to verify the tool's underlying assumptions. High-speed nets should not route closely to or contact mechanical parts that would disrupt the controlled-impedance environment. Check the box on the mechanical drawing that requires the board to have controlled impedance. Otherwise, run the simulations to guarantee that controlled impedance is unnecessary. Run 2-D field simulations and verify correct layup and line widths.

At this point, you can call a formal design review and sign-off meeting. Make sure everyone who is responsible for a key piece of the design has reviewed it and approved it. Walk through the schematics and plots. Verify any schematic changes that have occurred since the schematic-set review. Check over the mechanical drawings including layup and line widths, unit assembly, and parts list and spend extra time reviewing hard copies of the layer plots. Catching a high-speed trace traversing a power split or under a handle could save days or even weeks in the lab. A qualified engineer should review the design for EMI/RFI issues, such as high-speed oscillators near cabinet openings. Then, fill out a sheet of paper that includes the initials of those who checked part of the design and the part they checked. Due diligence prevents stupid mistakes.

Finally, have the signal-integrity engineer review his analysis and highlight any risks. You must gate the final release with a module check list. If any exceptions violate the release process you have implemented, document the risk, and then ensure management signs off on a waiver. Most people will not sign-off on a design unless they have verified its accuracy. Many will complain about this formal sign-off procedure because it takes time and effort, but no one will complain when you achieve first-pass success and avoid painful months in the lab debugging signal-integrity problems that you could have easily caught in a design review.

All of the simulation you already performed can now aid in final design-verification testing. You should compare delay files and timing-verifier results with actual measurements. Make sure probe parasitics and fixturing includes any comparison between measurements and simulations. You can also compare measured waveforms with simulations. You need to perform dc and ac measurements of power and ground planes. Compare these measurements with budgets and predictions. If possible, place your system in a schmoo chamber to determine voltage and temperature margins. Increasing the clock frequency will uncover the timing margin for a random sample of the design. V_{REF} schmooing is another recent addition to design-verification testing. For this test, you place potentiometers on the resistors that set the reference voltage for the HSTL differential amplifiers. You then move the voltage up and down to determine your noise-margin window. This process provides insight into which network interfaces contain the least noise margin. Document your work. You may be the one who benefits the most. Your customer will certainly appreciate that you've done your homework.□

References

1. Blood, W R, *Mecl System Design Handbook,* Motorola Inc, Phoenix, AZ.

2. Haller, R, and G Edlund, "Constructing Accurate Models of Behavioral I/O buffers," Designcon98 proceedings, ISSN 0886-229X, ISBN 0-933-217-47-1, 1998.

3. Johnson, H, and M Graham, *High-Speed Digital Design, a Handbook of Black Magic*, Prentice Hall, Upper Saddle River, NJ.

Acknowledgement

Many thanks to Barry Maskas, Peter LaFlamme, Greg Edlund, and Paul Guglielmi for their insightful suggestions.

Author biography

Robert J Haller is a senior member of the technical staff in the Alpha Server product-development division at Compaq Computer Corp, where he has worked for 19 years. He designs boards, backplanes, and custom ICs and has spent more than 10 years developing board-level signal-integrity products. He has a BSEE from the University of Massachusetts (Amherst, MA).