Conquering Noise in Deep-Submicron Digital ICs

KENNETH L. SHEPARD
Columbia University
CadMOS Design Technology
VINOD NARAYANAN
CadMOS Design Technology

As feature sizes decrease and clock frequencies increase, noise is becoming a greater concern in digital IC design. The authors describe a verification metric, noise stability, which guarantees functionality in the presence of noise, and a CAD technique, static noise analysis, for applying this metric on a chipwide basis.

NOISE HAS TWO DELETERIOUS EFFECTS in digital ICs. When noise acts against a stable logic level on a circuit node, it can transiently destroy logical information carried by the node. If this ultimately causes an incorrect machine state stored in a latch, functional failure will result. Attacking this problem by using dynamic simulation techniques on circuits containing tens of millions of transistors is highly impractical. Instead, we have developed a verification methodology called static noise analysis, which we apply on a chipwide basis using computer-aided design tools.[1] Even when noise does not cause functional failure, it has an impact on timing, affecting both delay and slew. We call this the noise-on-delay effect, and we will show how we apply static timing analysis techniques to its management.[2] First, let's look at the technology trends that have brought noise issues to the forefront.

What's going wrong?

If, as industry analysts predict, the current trend of technology scaling continues, feature sizes will continue to shrink and clock frequencies to increase. Shrinking feature size implies not only shorter gate lengths but also decreasing interconnect pitch and device threshold voltages.

In general, chips consist of more interconnect levels packed closer together. Reduction in the top and bottom areas of a minimum-width wire means that total wire capacitance is decreasing. Resistance, however, is increasing faster, despite efforts not to scale metal thicknesses. Moreover, the fraction of self-capacitance represented by lateral coupling is increasing. Die sizes remain relatively constant as more functionality is integrated on a single chip. Consequently, average wire lengths are also relatively constant, despite the decreasing pitch. These geometry factors have already made RC (resistance-capacitance) delays in the interconnect a significant performance component.

These trends, combined with faster on-chip slew times, also mean that capacitive coupling is becoming a significant source of noise. Furthermore, in many cases, because of this coupling capacitance, one cannot accurately calculate delay without considering the effect of simultaneous switching on coupled nets. Practical efforts to control RC delays through the use of low-resistivity metals (copper), low-dielectric-constant insulators, and wide, thick wiring will require future interconnection analysis to consider inductance and inductive coupling.

Performance demands are also increasing the use of circuit families by which designers deliberately sacrifice noise immunity for performance. While technology scaling re-

0740-7475/98/$10.00 © 1998 IEEE

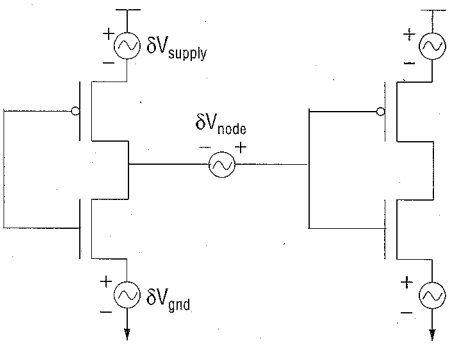


Figure 1. *Modeling man-made noise as evaluation node, ground line, or supply line sources in a circuit.*

sults in lower threshold voltages, the threshold voltage magnitude largely determines noise immunity in these circuits.

Noise sources

Noise is a problem to analog IC designers because it presents a lower bound on the useful amplification of a signal's magnitude. Noise also presents an upper bound on an amplifier's useful gain, ultimately saturating the amplifier if the gain is too high. The types of noise of concern in analog design—thermal, flicker, and shot noise, for example—arise from physical sources.[3]

Physical noise occurs because of the discreteness of electronic charge and the stochastic nature of electronic transport processes. Thermal noise, for example, originates from random motion caused by the thermal energy of carriers. Thermal noise has a flat spectral density. The open-circuit mean-square voltage of a resistor of resistance R due to thermal noise is

$$\langle V_n^2 \rangle = 4kTBR, \quad (1)$$

where k is Boltzmann's constant, T is absolute temperature, and B is the bandwidth of the circuit or measurement device.[3] At 300 kelvins, a 1,000-Ω resistor has a root-mean-square noise of $4n\text{V/Hz}^{1/2}$.

In contrast, digital circuits, by virtue of the large, abrupt voltage swings characteristic of their operation, create deterministic man-made noise several orders of magnitude greater than noise from stochastic physical sources. Problems due to these noise sources were first observed in mixed-signal applications, which plunged highly noise sensitive analog circuits into a noisy digital environment. Although digital circuits create much more noise than analog circuits, digital systems are prevalent because they are inherently immune to noise. Until the introduction of deep-submicron CMOS technology, noise immunity overcame the noisiness of digital circuits. Unfortunately, technology scaling has changed this balance, and noise is now a problem even in purely digital designs.

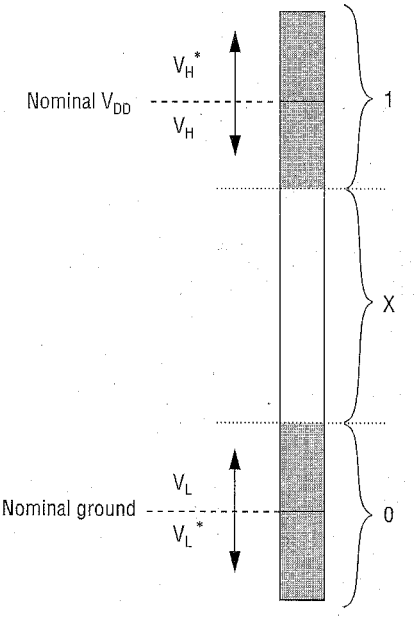


Figure 2. *A range of analog voltages defines the digital 0 and 1.*

As Figure 1 shows, deterministic man-made noise in CMOS digital ICs can be modeled as coming from series-voltage sources on evaluation nodes (δV_{node}), ground lines (δV_{gnd}), and supply lines (δV_{supply}).[4] By "evaluation nodes," we mean nodes in the circuit that carry logical information. These are the inputs and outputs of the circuit's logic gates. Logic gates are generally representations of sets of channel-connected transistors—that is, transistors connected through their sources and drains. Noise modeled with δV_{gnd} or δV_{supply} is called power supply noise. In the context of CMOS digital ICs, therefore, "noise" refers to any deviation from nominal supply or ground voltages at nodes that should otherwise represent stable logic 1 or 0.

In digital circuits, analog voltages on evaluation nodes carry logic information used in computation. Although noise causes these analog voltages to vary, the system still functions as long as the voltages fall into a valid range. If they do not, the circuit's correct functioning cannot be certain. Noise analysis is complex because the voltage ranges that represent valid logic levels depend on the precise time-domain characteristics of noise appearing on evaluation nodes, and on the receiving circuit's sensitivity to this noise. The former is influenced by δV_{node}-modeled noise and power supply noise in the driving circuit. The latter is influenced by power supply noise in the receiving circuit and by the properties of the receiving circuits themselves.

Figure 2 shows valid voltage ranges for a particular eval-

uation node. Voltages in the X region would render the node logically indeterminant. Noise is classified according to the voltage's relationship to the rails:

- V_H noise reduces an evaluation node voltage below the supply level.
- V_H^* noise increases an evaluation node voltage above the supply level.
- V_L noise increases an evaluation node voltage above the ground level.
- V_L^* noise decreases an evaluation node voltage below the ground level.

V_H^* and V_L^* noise, referred to as bootstrap noise, can also cause logical indeterminacy for certain receiving circuits.

The inherent noise immunity of CMOS digital circuits is due to the presence of high-gain restoring logic gates such as the inverter shown in Figure 3, which has a very nonlinear voltage transfer characteristic. As long as noise biases the gate such that noise appearing on the input is attenuated when propagated to the output, the gate acts to restore a valid logic level. If, however, noise biases the gate into a high-gain response, noise on the input will be amplified to the output, probably resulting in functional failure. In subsequent discussion, we use this "bias" condition to define a metric for determining the condition of indeterminacy for each evaluation node in the circuit. We call this metric noise stability.

Man-made noise

The basic cause of all noise in digital circuits is that these circuits use large-signal voltage changes to switch logic levels. Figure 1 showed noise modeled as series-voltage noise appearing directly on evaluation nodes or as power supply noise. We can further classify two types of evaluation node noise: interconnect noise, injected from coupling in the wires of the chip, and circuit noise, injected by or propagated from the circuits themselves.

Interconnect noise. Coupling noise, or cross talk, is primarily due to capacitive coupling between metal lines. Figure 4 shows a highly simplified analysis (neglecting interconnect resistance) of the essential attributes of this noise. In Figure 4a, coupled noise on the victim evaluation node between the two inverters results from switching on the neighboring perpetrator line denoted by the voltage source. In the circuit representation in Figure 4b, C_1 is the capacitance to ground on the victim net, and C_2 is the coupling capacitance to the perpetrator. R_{driver}, the node impedance of the evaluation node, is the effective resistance trying to keep the node quiet.

Typically, the resulting noise has the form of a pulse (Figure 4c). The switching slew on the perpetrator net determines its leading edge, and the restoring time constant $\tau = R_{driver}(C_1 + C_2)$ determines its trailing edge.

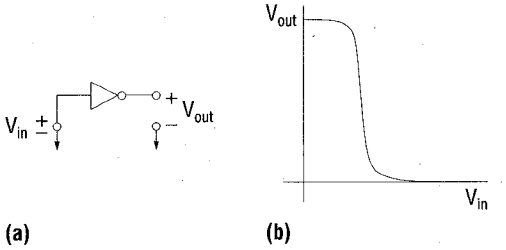


Figure 3. *The inherent noise immunity of digital circuits such as this CMOS inverter* ***(a)*** *results from high-gain restoring logic gates with very nonlinear voltage transfer characteristics as shown in* ***(b)****.*

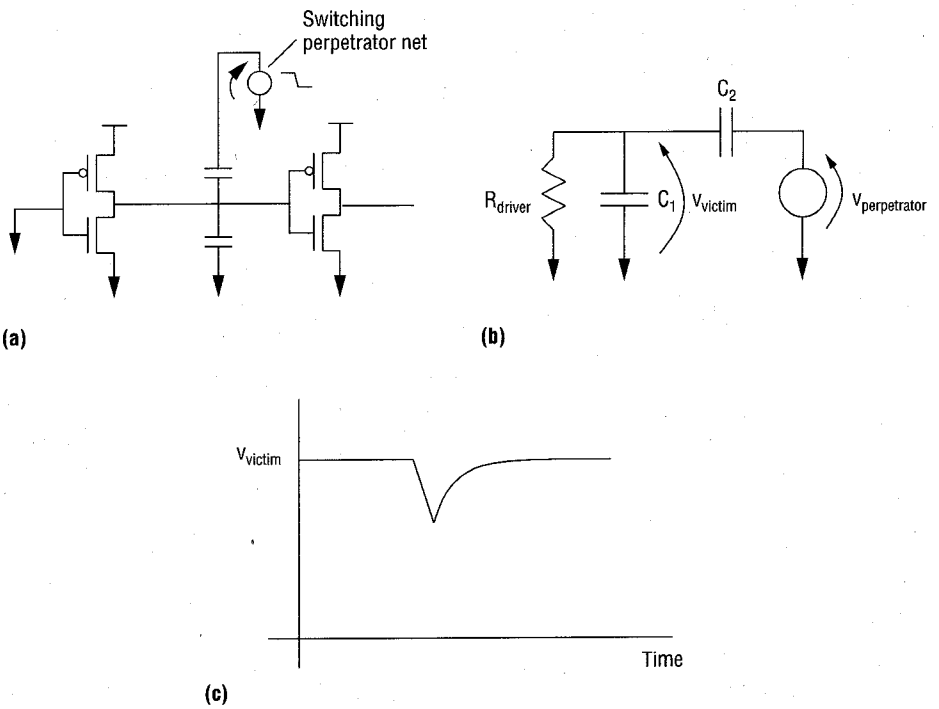


Figure 4. *Capacitive coupling noise: coupling onto an evaluation node between two inverters* ***(a)****; simplified equivalent circuit* ***(b)****; coupling-noise-pulse waveform* ***(c)****.*

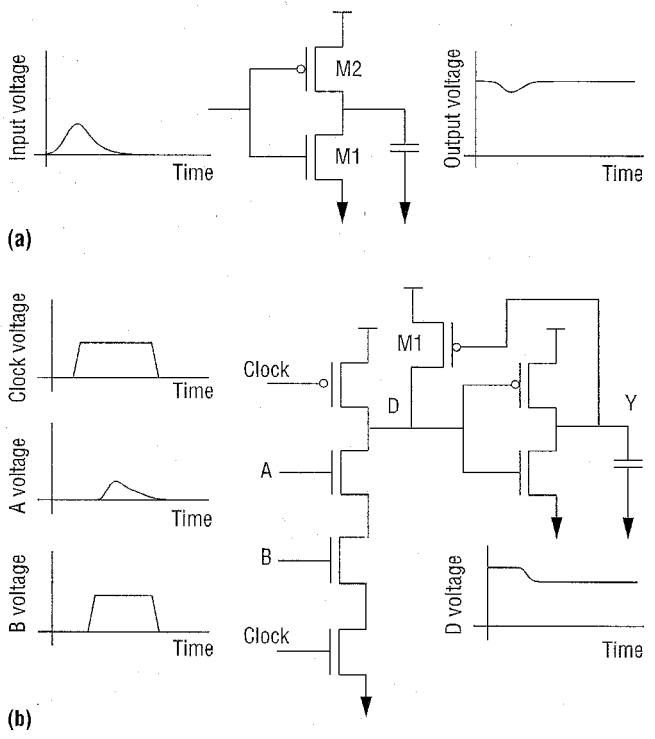


Figure 5. *Circuit noise propagation for a static inverter* ***(a)*** *and for a two-way domino AND* ***(b)****.*

Although capacitive coupling noise is the main source of on-chip interconnect noise, in some cases inductive coupling can be a significant source of coupled noise. In addition, if a driver is oversized relative to the characteristic impedance, an underdamped ringing can result. For inductive effects to be significant,

$$R_{driver}, R_{line} < Z_0 = (\mathcal{L}/C)^{1/2}. \quad (2)$$

R_{driver} is the driver's effective resistance, R_{line} is the line's total resistance, Z_0 is the line's characteristic impedance, and $\mathcal{L}$ and C are the inductance and capacitance per unit length. Designers are working to decrease the R_{line} by using wide-wire routes on thick, upper-level interconnect.[5] The previously described technology changes are also driving this trend. Z_0 is generally increasing due to lower wire capacitances. As a result, the Equation 2 condition is becoming a concern for a rapidly increasing number of nets on a chip.

The complexity of analyzing inductive effects is due to the frequency dependence of the resistances and inductances. This dependence results from a skin effect in each conductor and, more importantly, from a proximity effect that makes the current distribution in various return paths depend on the frequency. At low frequencies, current returns favor low-resistance paths, sometimes very far from the driven wire or through the package. At high frequencies, current returns choose the lowest-inductance path, usually the closest power or ground line, but sometimes a neighboring signal line or collection of signal lines. Extracting and analyzing inductance in the context of timing and noise analysis is an open research problem and will not be considered here.

Circuit noise. Noise can also be propagated onto an evaluation node from the driving gate or injected by other circuit effects in the driving gate. When V_L noise appears on the input of a CMOS inverter, for example, V_H noise propagates to the output. In Figure 5a, *n*-FET (*n*-channel field-effect transistor) M1 turns on, because of the V_L noise appearing at its gate, and tries to bring down the output voltage. This action is fought by *p*-FET M2, which continues to hold the output high. Noise propagates to the output, but, if all goes well, is attenuated relative to the noise appearing on the input (this is the essence of the noise stability metric we present later). Many high-performance circuit styles try to speed up one transition (usually the falling one) at the expense of the other and assign all logical evaluations to the faster edge. We call any circuit that uses this technique a skewed-evaluate circuit. Dynamic circuits are an extreme form of skewed-evaluate circuits, in which the evaluation transitions are unchallenged.

Figure 5b (in the absence of the *p*-FET half-latch device M1) is an example of a dynamic gate (in this case, a two-way AND). When the clock is 0 (the precharge phase), node D is charged to V_{DD} and output node Y carries a logic 0. When the clock goes to 1 (the evaluation phase), and if either A or B is still 0, node D will float with no dc path to ground. Let's consider the case in which B goes high during the evaluation phase but A is still nominally 0. Because D is floating, nothing is fighting to keep node D high. As a result, V_L noise on node A comparable to or greater than the *n*-FET threshold voltage easily propagates to the output. This noise sensitivity is intrinsic to the gate's performance benefit. Node D is also very sensitive to coupled noise because it is floating.

We can bolster this gate's noise immunity by including the *p*-FET half-latch device M1. This device actively fights to keep the dynamic node charged to V_{DD} in the presence of noise. It degrades performance, however, because it fights evaluation of the gate; thus, the noise immunity comes at a cost. The half-latch prevents the gate from being truly dynamic.

Skewed-evaluate circuits are sensitive to subthreshold leakage currents from nominally off devices even in the absence of input noise. Consider again the dynamic circuit of Figure 5b with the gate in the evaluation phase (the clock is 1). If A and B are both 0, these transistors in the *n*-FET stack are nominally off. Subthreshold current proportional to $e^{-qVT/kT}$ still flows, however, acting to pull D below V_{DD}. (V_T is the threshold voltage.) This effect is sometimes called leakage noise.

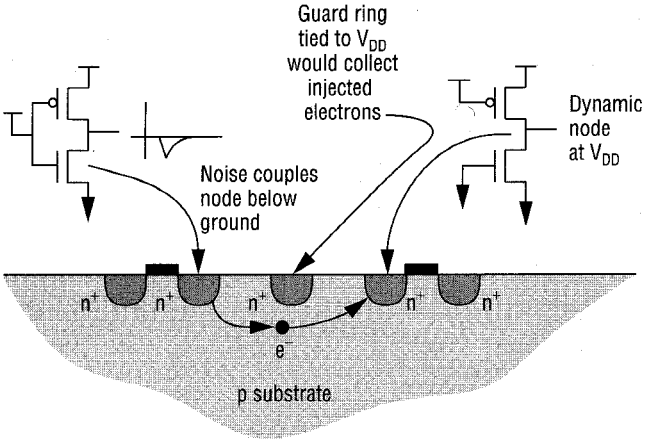


Figure 6. *Minority carriers back-injected into the substrate.*

Stray minority carriers in the substrate are a second possible source of leakage. Figure 6 illustrates one cause of stray minority carriers: injection due to bootstrap noise. In this case, coupling noise on the output of a driver drives the node to a voltage below ground. As a result, electrons are back-injected into the substrate, where they can subsequently be collected by a dynamic node at a positive voltage. To prevent this, designers can add guard rings to capture the stray minority carriers, the usual procedure for off-chip drivers and receivers.[6] The need to bootstrap more than 0.6 V effectively eliminates this mechanism as a leakage source at supply voltages below 2 V.

Another source of stray minority carriers is ionizing radiation, either cosmic rays or alpha emission from packaging metals. Functional failures due to this leakage mechanism are called soft errors.

Figure 7 illustrates another type of circuit noise that applies primarily to skewed-evaluate circuits: charge-sharing noise. The cause of this noise is charge redistribution between a dynamic node and internal nodes of a pull-up or pull-down stack. In the example in Figure 7a, node D is initially precharged to V_{DD}. In Figure 7b, cases 1 and 3 represent a situation in which there is no half-latch device. In cases 2 and 4, the half-latch device is present to restore the node. In cases 1 and 2, B1 to B4 are 0 and A1 to A4 switch to 1. This causes charge sharing between the dynamic node and the internal nodes a, b, c, and d. In cases 3 and 4, only one of A1 to A4 switch, as would be the case if logical constraints dictated that only one of these signals could switch to a logic 1 at a time. In cases 2 and 4, the waveform associated with charge sharing has the same pulse feature as capacitive-coupling noise (Figure 5). Without the half-latch device, node D is dynamic and never recovers from the charge-sharing noise event.

Another circuit style that can be particularly sensitive to

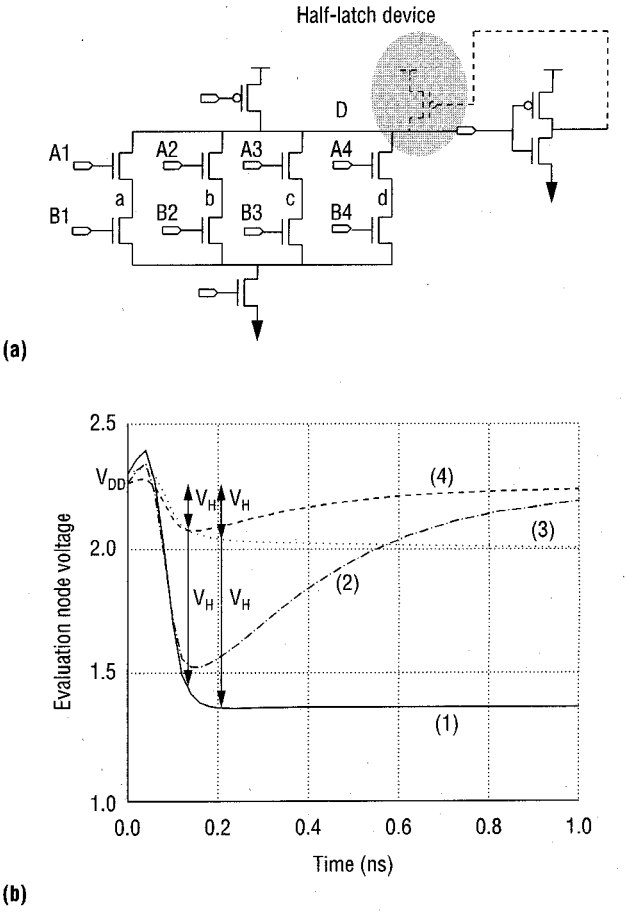


Figure 7. *Charge-sharing noise: typical circuit in which node D is a dynamic or weakly static node susceptible to charge-sharing noise* ***(a)****;* V_H *noise waveforms appearing on node D* ***(b)****.*

noise is pass-gate logic, which is increasingly popular in high-performance designs. Single *n*-FET and *p*-FET pass gates contain a V_T drop, which represents dc evaluation node noise at the pass-gate output. Even when circuits use complementary pass-gate logic to avoid the V_T drop, evaluation nodes in these circuits can have weak static paths to ground, which make the circuits susceptible to coupling noise.

Power supply noise. Power supply noise appears on the on-chip power and ground distribution network. There are two components of power supply noise. The first is variations in the dc power supply and ground levels. To calculate these variations, known as IR drops, we apply a separate analysis of the chip's spatial current demands against a supply- and ground-rail resistance extraction.[7]

A second type of power supply noise is delta-I noise, produced by the simultaneous switching of off-chip drivers and internal circuits, usually synchronized with clock activity.

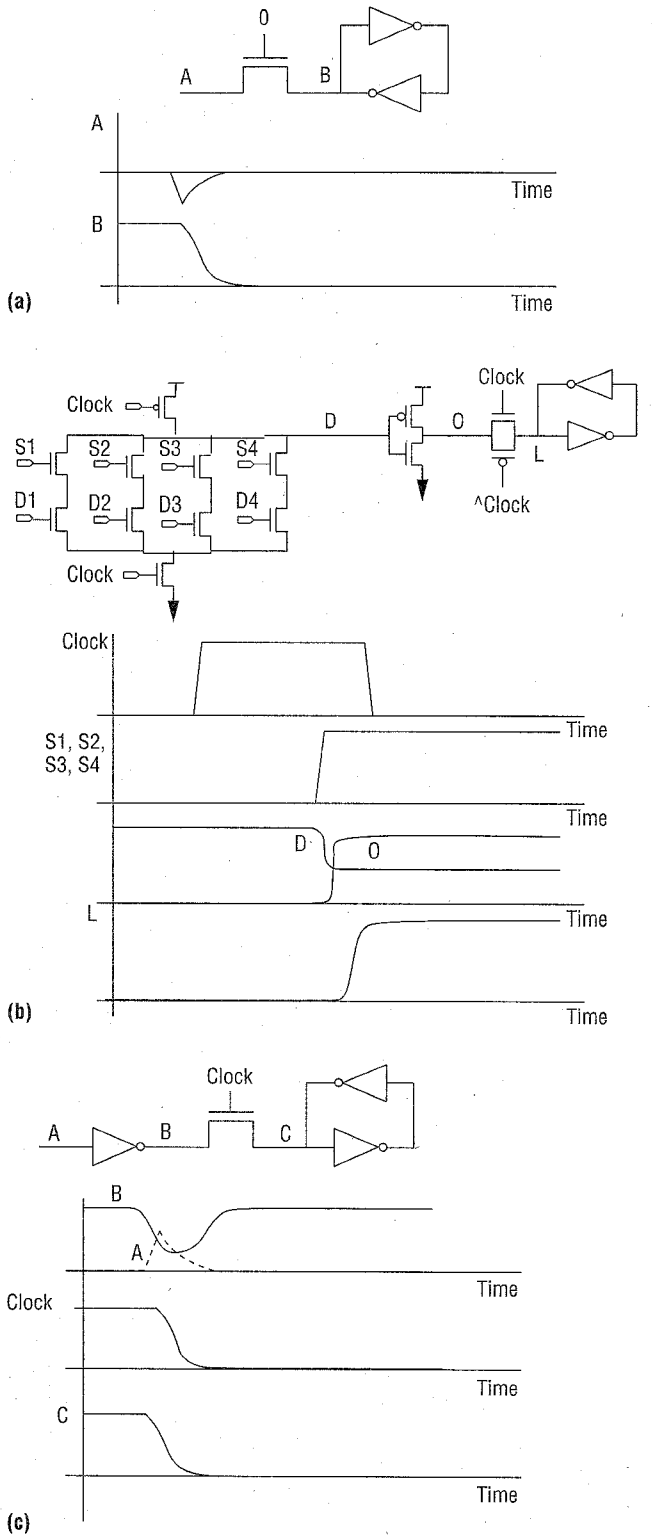


Figure 8. *Examples of functional failures due to noise: V_L^* noise coupled into a pass-gate latch* ***(a)****; charge-sharing noise on a dynamic gate* ***(b)****; coupling noise occurring near a closing clock edge* ***(c)****.*

This sudden demand for current causes variations in the supply and ground rails as the current must be supplied through inductance of the chip power grid and chip-package connection. On-chip decoupling capacitance provides a transient source of charge that can reduce delta-I noise. This capacitance can be due to nonswitching circuits, *n*-well capacitance, or explicit thin-oxide capacitors. Detailed transient analysis of the power grid involves applying models of the circuits' current demands to a detailed RLC (resistance-inductance-capacitance) extraction of the power grid combined with the package.[8] The current models usually take the form of Norton- or Thevenin-equivalent circuits at designated points in the power or ground distribution hierarchy, usually on a designated via layer. In practice, a well-designed on-chip power distribution based on C4 technology[9] is sufficiently rigid that the delta-I variations dominate the dc IR drop.

Stability metrics and static noise analysis

One traditionally analyzes noise in analog circuits by adding noise generators for each possible physical noise source to the complete small-signal equivalent circuit.[3] These noise generators are usually in the form of mean-square voltages or currents, as in the case of thermal noise in Equation 1. By contrast, the highly nonlinear operation of digital circuits and the more deterministic nature of man-made noise sources require an entirely different kind of analysis and verification metric.

Essential stability. To guarantee the chip will function, we must verify that latching structures that hold state do not falsely switch in the presence of noise. Latches can be either static, bistable, positive-feedback configurations of restoring logic gates or dynamic nodes acting as latches, storing state by virtue of the charge on an evaluation node. The act of switching a latch defined by a positive-feedback configuration of restoring logic gates involves making the circuit unstable. Therefore, we call the condition that ensures that latches do not falsely switch the essential stability requirement. In other words, essential stability is the necessary and sufficient condition for the functionality of a digital circuit.

Let's consider three examples of functional failures produced by essential instabilities. In Figure 8a, coupled noise appears at the input of a pass-gate latch. The latch stores a logic 1, and the pass gate is nominally off. A V_L^* noise pulse that exceeds the threshold voltage of the pass gate turns on the pass gate, switching the latch to a logic 0. There is no possibility of recovery because the pass gate is off after the V_L^* pulse is removed. We call this an invariant noise failure because the failure is independent of the relative timing of data and clock.

In the second example, shown in Figure 8b, signals S1, S2, S3, and S4 switch to 1, injecting charge-sharing noise onto dynamic node D (nodes D1, D2, D3, and D4 are 0). This noise propagates to node O, subsequently flipping the latch and producing a failure. This too is an invariant noise failure because once the dynamic node fails, there is no mechanism for recovery. Figure 8c also shows coupling noise causing a false value to be stored in the latch. In this case, the clock edge closes the latch, capturing the erroneous value. This case differs from the others in that static recovery is possible; that is, slowing the clock down can eliminate the noise failure. Therefore, we call this a variant noise failure. The problem with variant failures is that they are not predictable from timing analysis and therefore must be prevented.

Noise stability. We could choose to verify the essential stability condition at each latch. Consider a latch consisting of a bistable feedback configuration of restoring logic gates, as shown in Figure 9. Let x and y be the voltages on nodes A and B. Let f and g be the transfer functions of gates 1 and 2; that is, $y = f(x)$ and $x = g(y)$. The latch will be stable in the presence of series-voltage dc noise sources δV_A and δV_B on evaluation nodes A and B if

$$\left|\frac{\partial f}{\partial x}\frac{\partial g}{\partial y}\right| < 1$$

at the bias point determined by these sources.[4,10] This is the essential stability condition for dc series-voltage noise sources, and it will certainly hold if

$$|\partial f/\partial x| < 1, \text{ and } |\partial g/\partial y| < 1.$$

This is a stronger condition that actually implies that $\delta V_A + \delta V_B$ is maximum. If every restoring logic gate in the circuit meets this second condition, it will certainly never be possible for any positive-feedback configuration to switch. This is the criterion traditionally used to define the worst-case static noise margins.

Static, or dc, noise margins are too conservative to apply against the magnitude of pulse noise such as coupling or charge-sharing noise because they ignore the fact that logic gates also act as low-pass filters. Pulse noise amplitudes can be safely higher than static noise margins would allow, depending on the shape of the pulse. We therefore cast the noise stability condition as follows: Every restoring logic gate, when acted upon by a noise stimulus, must have a dc-noise sensitivity less than one at the maximum (for V_L or V_H^* noise) or minimum (for V_H or V_L^* noise) of the output response. This condition is sufficient to verify functionality.

Let's consider the condition in more detail for the circuit in Figure 10a. In this case, we inject pulse noise into the series-voltage noise source, δV_{node}. In the latch's initial state, node A is low and node B is high. Figure 10b shows the latch's behavior when the peak noise amplitude of δV_{node} is 1.37 V (top graph) and when it is 1.38 V (bottom graph). In the first case, the latch tolerates the noise and does not switch. In the second case, the latch becomes unstable and switches, an essential stability violation.

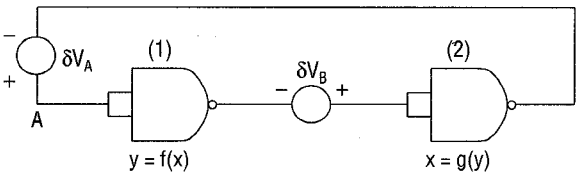


Figure 9. *Latch circuit with dc series-voltage noise sources, δV_A and δV_B.*

Figure 10c shows how this failure would have been detected by the noise stability check on gate 1. The top graph shows the input and output waveforms for a pulse amplitude of 1.1 V applied to the input of gate 1. The bottom graph shows the sensitivity to changes in the dc level. The minimum of the output response is noted as time t_0, at which the magnitude of sensitivity exceeds one. At this noise pulse amplitude, the gate is at the threshold of a noise stability violation. The fact that the latch can actually tolerate an additional 280 mV of pulse noise before switching demonstrates the conservatism of the noise stability approach. Because gate 2 is subunity-biased, gate 1 can tolerate more noise. In practice, this margin is not significant for bistable latch circuits because once a restoring logic gate is biased beyond the unity-sensitivity threshold, the sensitivity magnitude rapidly increases.

The main source of the noise stability test's conservatism is that we apply it at every restoring logic gate, not only at latches. We do this to localize noise failures within a gate or two of the offending noise sources. In practice, noise stability violations, even when they would not result in an essential instability, represent severe design weaknesses that should be corrected. False switching of a dynamic node acting as a latch is defined by the noise instability of the restoring logic gates to which that node fans out.

We must verify noise stability at the most aggressive conditions under which the chip must be functional: fast process, high temperature, and high nominal voltage. Fast process means faster slews, which generate more coupling noise. Fast process corners also mean shorter channel lengths, which usually result in lower V_T. For fast sorts, channel length variations can be a significant source of failure due to noise if the fast process corner is not used for noise analysis. High temperature means that slews are slower, generating less noise. However, high temperature means high-

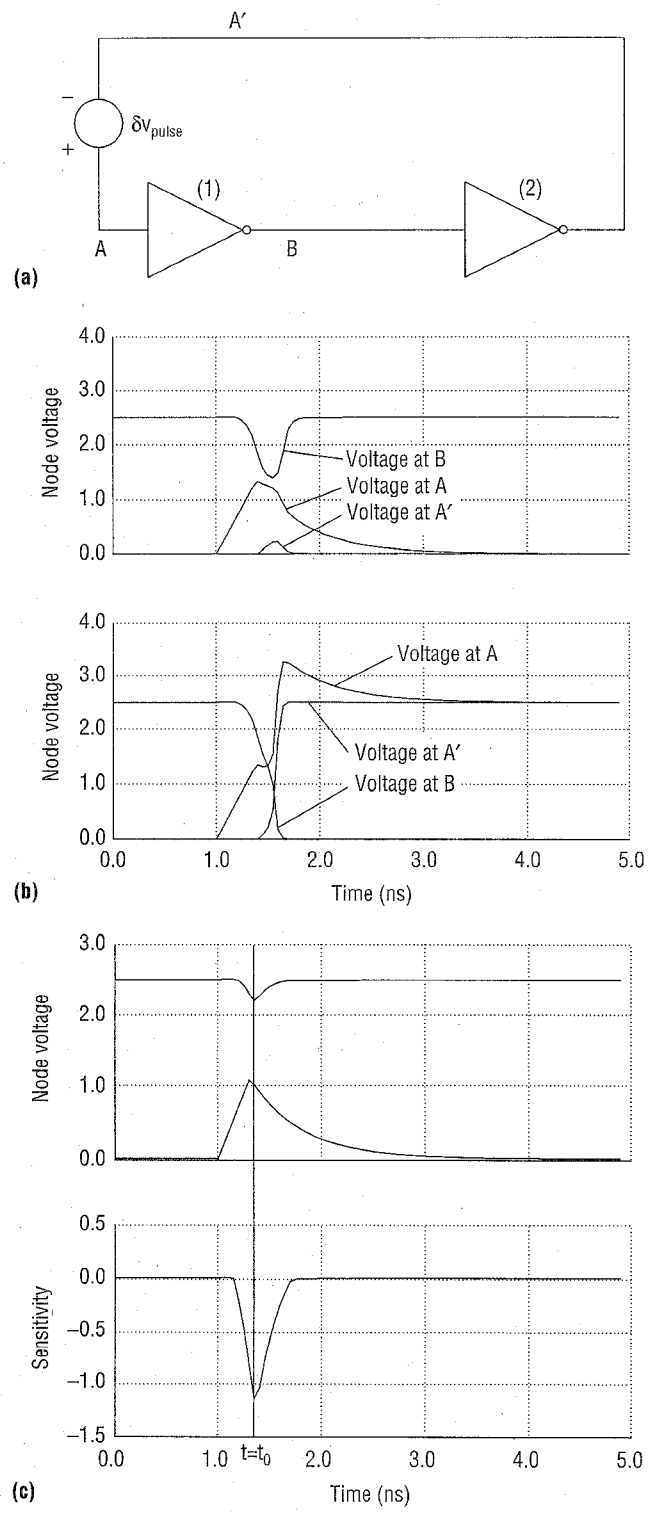


Figure 10. *Functional failure due to pulse noise: the cross-coupled inverter latch* ***(a)*** *does not switch at a pulse amplitude of 1.37 V but is forced unstable at 1.38 V* ***(b)****. Gate 1 is noise-unstable at a pulse amplitude of 1.1 V* ***(c)****.*

er subthreshold currents and more leakage noise, which is usually a much stronger effect. Higher nominal voltages produce faster transitions and higher noise voltage levels relative to V_T.

Static noise analysis. Our static analysis technique combines simulations on small numbers of transistors—groups of channel-connected components (CCCs)—with a path trace to check noise stability practically on a chipwide basis.

The key abstraction in static noise analysis is the noise graph, a directed graph containing all the circuit's evaluation nodes connected by segments that move and transform noise. In many ways, this graph is analogous to the timing graph used in static timing analysis.[7] This similarity, along with the tight interaction of timing and noise analysis, encourages the development of tools combining the techniques. We describe the approach here in general terms; details of an actual hierarchical implementation can be found elsewhere.[1,11]

Several fundamental assumptions underlie static noise analysis techniques:

- Worst-case sensitization conditions drive the CCC simulations used for calculating circuit and interconnect noise. By this, we mean that now the transistor gates are biased before application of the noise stimulus or switching waveform.
- The superposition principle applies approximately in adding (in the time domain) circuit and interconnect noise sources. For noise sources small enough to satisfy the noise stability requirement, active FET channels (those attempting to hold nodes to their static level) are biased in the triode regions of their current-voltage characteristics, justifying this linear assumption. In particular, charge-sharing noise and propagated noise can be calculated on a "single-input-changing" basis and superposed with the coupled noise calculation to find the total noise. The calculation uses the sensitization producing the largest amplitude output noise. Noise sources can be combined only when the sensitization conditions are mutually satisfiable.
- We define worst-case temporal relationships by superposing the peak responses of the circuit and interconnect noise for each allowable noise type (V_L, V_H, V_L^*, V_H^*). (More complex assumptions would be necessary in the case of a ringing response of inductive coupling.)
- Power supply integrity analysis is performed independently, characterized by dc bounds on the local power supply variation. That is, V_{DD}^{max} and V_{DD}^{min} characterize the supply, and Gnd^{max} and Gnd^{min} characterize the ground.

In some instances, it is useful to work with simplified time-domain abstractions for analog noise waveforms (for example, in creating noise abstracts for hierarchical analysis or cell characterization). This approach follows similar approaches used in static timing analysis, in which voltage waveforms are abstracted as saturate ramps. In particular, one can treat the noise on any node as the superposition of a dc value and a pulse. One can further characterize the pulse by a peak value, a leading slew time, and a trailing exponential time constant.

Circuit and noise graph analysis. Figure 11a illustrates the general static noise analysis approach. We use a superposition of simulation results to determine the circuit noise appearing at node Y. In this example, the noise at node Y is the worst-case noise propagated through gate 1, combined with the charge-sharing noise potentially introduced by gate 1. We then superpose this circuit noise with capacitive coupling noise contributed by the switching perpetrator net. We use this sum in the simulation of gate 2 to determine its noise stability and to calculate the circuit noise appearing at its output.

Figure 11b shows the noise graph abstraction for this (very simple) example. There are three types of segments in a noise graph: restoring, propagate, and node injection. Restoring segments cross gates that at some dc bias point have a small-signal gain greater than one. Noise propagates across restoring segments. We must also perform noise stability checks across restoring segments. In the example in Figure 11b, three restoring segments connect the inputs and outputs of gates 1 and 2. Each is labeled with the type of noise propagated by the segment. For example, L → H indicates that the segment propagates V_L noise and transforms it into V_H noise. Correspondingly, propagate segments cross gates that have subunity gain at all dc bias points (there are no propagate segments in the example in Figure 11b). Node injection segments can introduce noise directly onto an evaluation node, superposing it with the propagated noise. Both coupled interconnect noise and charge-sharing noise are modeled as node injection segments.

To begin the analysis of the noise graph, we break the loops in the graph and topologically sort it for traversal. (A similar approach would be used in static timing analysis of the same design.) Our example contains no loops, and the graph is already sorted left to right. In a traversal of the graph, we perform appropriate circuit simulations for each restoring, propagate, and node injection segment. Three types of simulations are necessary, each with its own sensitization assumptions for the driving CCC's transistors:

- Weakest-path sensitization for coupled-noise calculation.
- Sensitization for noise stability and propagated-noise analysis. In Figure 11b, we calculate V_H noise appearing at node Y due to V_L noise appearing at node A. To do this, we hold node B statically at V_{DD} and nodes A and C statically at ground before applying the noise at A. For noise stability checks, we use V_{DD}^{max} and Gnd^{min} for supply and ground connections, and for determining propagated noise, we use V_{DD}^{min} and Gnd^{max}.
- Sensitization for charge-sharing noise calculation. For example, we calculate possible V_L charge-sharing noise injected onto node Y in Figure 11b. By switching node A from V_{DD} to ground, we set nodes A, B, and C to V_{DD} before applying the switching waveform at A.

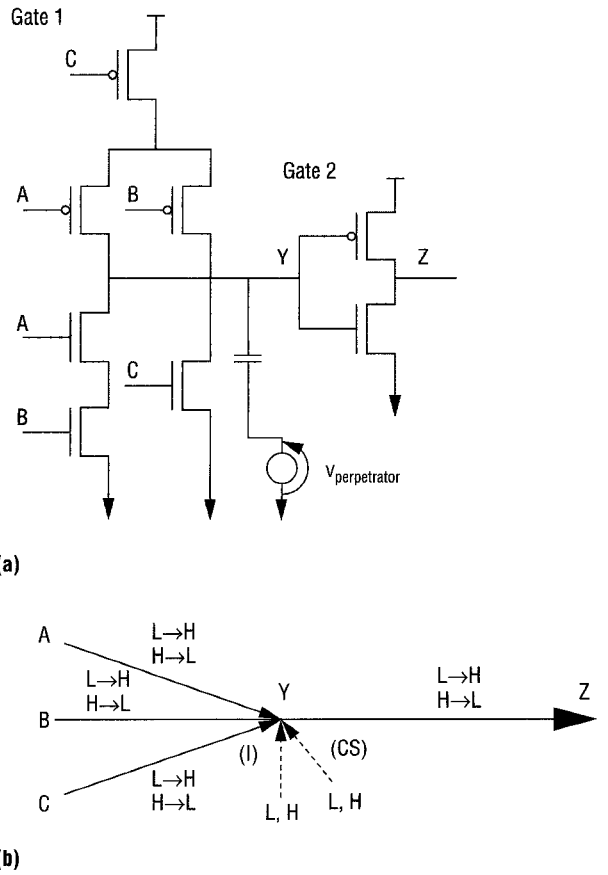


Figure 11. *Establishing the elements of static noise analysis: simple circuit example* ***(a)****; noise graph* ***(b)****. Solid lines denote restoring segments. Dashed lines denote node injection segments for coupled noise (labeled I) and charge-sharing noise (labeled CS).*

In general, to find the noise appearing as an output of a given CCC, we must find the sensitization producing the largest amplitude output noise for each noise type (V_L or V_H). Static noise analysis reports stability violations that oc-

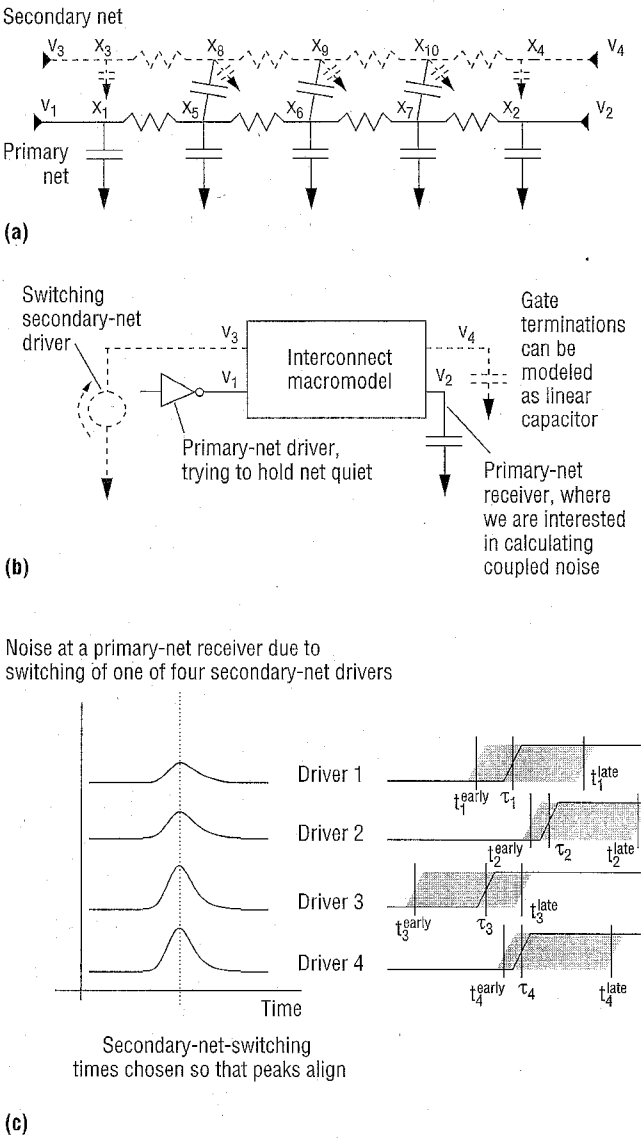


Figure 12. *Interconnect modeling for coupled-noise calculation: unreduced net complex* ***(a)****; interconnect reduced to multiport macromodel* ***(b)****; timing orthogonality analysis* ***(c)****. We superpose capacitive-coupling noise in the time domain if the secondary-driver switching time implied by peak alignment falls into the early- and late-mode timing windows—in this case, for rising transitions. (v: port voltage; x: internal subnode voltage)*

cur in restoring segments due to noise appearing on any input, independently of the sensitization conditions required for the maximum output noise.

Interconnect analysis. To perform the coupled noise calculation, we identify a net complex for each evaluation node in the design. The primary net of the complex is the one for which we are trying to calculate the noise. The complex also includes secondary nets with significant coupling to the primary net.[11] Couplings between the significant secondary nets and nets other than those already in the net complex are grounded. We can use reduced-order modeling techniques to find a multiport admittance macromodel (for example) of this complex for use in the coupled-noise calculation.

We apply an independent voltage source model directly to the secondary-net-driver ports. The primary driver port can terminate with either a linearized model of the driver (that is, a resistor) or the complete gate. In the latter case, we must use techniques such as recursive convolution to perform the simulation. (See Shepard[12] for a review of interconnect analysis techniques and references to works on the subject.) To calculate the worst possible noise at each primary-net receiver, we use superposition and align the peaks of the pulses (c_i) caused by each secondary-net driver as shown in Figure 12. This corresponds to a set of secondary-net-driver switching times τ_i.

The coupled-noise calculation should interact tightly with the static timing analysis of the same design.[11] From the timing model, we can obtain the secondary-net-driver slews; otherwise, we must assume the fastest possible slews. In addition, we can reduce the pessimism of assuming that all secondary-net drivers contribute to coupled noise on primary nets. We do this by disallowing participation of some secondary-net drivers based on a timing orthogonality analysis. The static timing analysis defines timing windows by the earliest and latest possible arrival times (t_i^{early} and t_i^{late}) for each secondary-net driver (see Figure 12c). Only when times τ_i fall within these windows do we allow the noise sources to contribute.

Noise-on-delay and static timing analysis

The static noise analysis methodology we have described focuses on verifying the noise stability condition. But even in a noise-stable circuit, noise, particularly coupling noise, can significantly affect delay. When the primary and secondary nets switch in opposite directions, the coupling capacitors behave as larger capacitors tied to ground, increasing delay and slew. Similarly, when primary and secondary nets switch in the same direction, the capacitances appear smaller, decreasing delay and slew.

Just as we use timing information to reduce pessimism in noise analysis, we can use noise analysis to determine the effect of coupled noise on delay. In static timing analysis, we usually choose a load and net-delay division of interconnect timing. That is, first we independently calculate delay and slew at the output of the driver due to the loading of the interconnect. Then we calculate delay and slew change from the driver to each receiver in the net. In a sense, we treat the interconnect as a distinct segment in the static timing graph.

The noise-on-delay analysis must be an iterative algorithm including the following steps:[11]

1. Perform an initial timing with all secondary nets grounded.
2. Freeze the arrival time windows and slews at each driver.
3. Recalculate all delays, including the coupling effects in the interconnect analysis. (This analysis uses the frozen secondary-net-driver information.)
4. Go to step 2 and repeat until convergence occurs.

Through this process, well-tuned designs (with drivers appropriately sized and RC delays controlled with repeaters or wide-wire routes) will converge. The process may not always assure convergence for a poorly tuned design—for example, a design in the early stages of the design process.

NOISE IS A DESIGN METRIC of comparable importance to timing, area, and power. With technology scaling, it is a problem affecting all types of designs from custom microprocessors to standard-cell ASICs. A noise analysis solution must be capable of analyzing tens of millions of transistors, considering both circuit and interconnect noise, and evaluating the distinct noise tolerances of each node in the circuit.

Successful design methodologies must incorporate a three-level noise strategy. The first line of defense is a set of noise avoidance rules (see box) to guide circuit and interconnect design. These rules should prevent most noise problems without introducing too many area or timing constraints. Next, a detailed static noise analysis of the design should find all possible noise failures. Finally, careful circuit simulation should determine whether the design can tolerate some failures flagged by static noise analysis. D&T

Noise-reducing design techniques

Designers can use a number of techniques and practices to control noise in digital circuits. The most common techniques take the form of circuit design rules. Here are some examples:

- Disallow single *n*-FET or *p*-FET pass gates because of the V_T drop they cause.
- Disallow pass gates at the ends of long wires.
- Disallow long wire runs feeding domino gate inputs.
- Disallow high beta static circuits feeding low beta static circuits, or vice versa.

In addition, design rules can require "baby-sitting" devices added to internal nodes in *n*-FET pull-down stacks of domino gates to mitigate charge-sharing noise, and half-latches added to dynamic nodes. Rules of this sort are sometimes accompanied by a circuit-checking tool. Although design guidelines and rules can go a long way in preventing noise problems, they should never be viewed as a substitute for comprehensive noise analysis.

There are many possibilities for design tuning as part of the trade-off between noise immunity and performance. One possibility is adding noise margin constraints to performance-based optimization. In addition to tuning gate widths, one can tune gate lengths to balance performance against leakage noise. Weaker devices at the top of the *n*-FET stack can also reduce charge-sharing noise in the presence of a half-latch device.

In addition to circuit techniques and optimization, there are also many ways to reduce coupling noise in the interconnect design—for example, increasing spacing between wires or routing signal lines alternately with power or ground. Such techniques can be implemented as constraints in a routing tool.

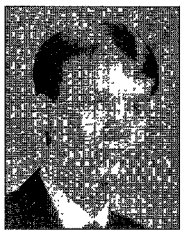
Acknowledgments

We gratefully acknowledge numerous helpful discussions with Charlie Huang and Steven Chan of CadMOS Design Technology and Ron Rose, Abe Elfadel, Chandu Visweswariah, Pete Elmendorf, and Gutuan Zheng of IBM.

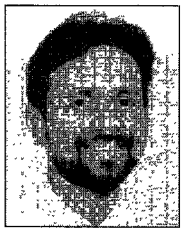
References

1. K.L. Shepard and V. Narayanan, "Noise in Deep Submicron Digital Design," *Proc. IEEE/ACM Int'l Conf. Computer-Aided Design*, IEEE Computer Society Press, Los Alamitos, Calif., 1996, pp. 524-531.
2. R.B. Hitchcock, G.L. Smith, and D.D. Cheng, "Timing Analysis for Computer Hardware," *IBM J. Research and Development*, Vol. 26, No. 1, 1982, pp. 100-105.
3. P.R. Gray and R.G. Meyer, *Analysis and Design of Analog Integrated Circuits*, John Wiley & Sons, New York, 1984.
4. J. Lohstroh, "Static and Dynamic Noise Margins of Logic Circuits," *IEEE J. Solid-State Circuits*, Vol. SC-14, June 1979, pp. 591-598.
5. H.B. Bakoglu, *Circuits, Interconnects, and Packaging for VLSI*, Addison-Wesley, Reading, Mass., 1990.
6. P.E. Gronowski et al., "A 433-MHz 64b Quad-Issue RISC Microprocessor," *IEEE J. Solid-State Circuits*, Vol. 31, No. 11, 1996, pp. 1687-1696.
7. K.L. Shepard et al., "Design Methodology for the G4 S/390 Microprocessors," to be published in *IBM J. Research and Development*, Vol. 41, No. 4/5, 1997, pp. 515-547.
8. H.H. Chen, "Minimizing Chip-Level Simultaneous Switching Noise for High-Performance Microprocessor Design," *Proc. IEEE Int'l Symp. Circuits and Systems*, Vol. 4, IEEE CS Press, 1996, pp. 544-547.

9. L. Miller, "Controlled Collapse Reflow Chip Joining," *IBM J. Research and Development*, Vol. 13, No. 3, 1969, pp. 239-250.
10. J. Lohstroh, E. Seevinck, and J. De Groot, "Worst-Case Static Noise Margin Criteria for Logic Circuits and Their Mathematical Equivalence," *IEEE J. Solid-State Circuits*, Vol. SC-18, Dec. 1983, pp. 803-806.
11. K.L. Shepard et al., "Global Harmony: Coupled Noise Analysis for Full-Chip RC Interconnect Networks," *Proc. IEEE Int'l Conf. Computer-Aided Design*, IEEE CS Press, 1997, pp. 139-146.
12. K.L. Shepard, "Practical Issues of Interconnect Analysis in Deep Submicron Integrated Circuits," *Proc. Int'l Conf. Computer Design*, IEEE CS Press, 1997, pp. 532-541.



Kenneth L. Shepard is an assistant professor of electrical engineering at Columbia University and the chief technology officer of CadMOS Design Technology, San Jose, California. Previously, he was a research staff member and manager in the VLSI Design Department at IBM's T.J. Watson Research Center, where he was responsible for the design methodology for the G4 S/390 microprocessors. He received IBM Research Division awards for contributions to the G4 design, a Hertz Foundation prize for his doctoral thesis, and a National Science Foundation CAREER Early Faculty Development Award. Shepard received his BSE from Princeton University, where he was class valedictorian, and his MS and PhD from Stanford University. He is a member of the IEEE.



Vinod Narayanan is the vice president of research and development at CadMOS Design Technology. Previously, he was a research staff member of IBM's T.J. Watson Research Center, working on various design automation problems. He is interested in many areas of design automation including physical design, floor planning, timing analysis, signal integrity analysis, and design automation for highly integrated systems. Narayanan received his MS and PhD from Syracuse University. He is a senior member of the IEEE.

Address questions or comments to Kenneth L. Shepard, Dept. of Electrical Engineering, Columbia University, 1312 S.W. Mudd Building, New York, NY 10027; shepard@ee.columbia.edu; http://shepard.ee.columbia.edu.