Noise in Deep Submicron Digital Design

Kenneth L. Shepard and Vinod Narayanan
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598

Abstract

As technology scales into the deep submicron regime, noise immunity is becoming a metric of comparable importance to area, timing, and power for the analysis and design of VLSI systems. This paper defines noise as it pertains to digital systems and addresses the technology trends which are bringing noise issues to the forefront. The noise sources which are plaguing digital systems are explained. A metric referred to as noise stability is defined, and a static noise analysis methodology based on this metric is introduced to demonstrate how noise can be analyzed systematically. Analysis issues associated with on-chip interconnect are also considered. This paper concludes with a discussion of the device, circuit, layout, and logic design issues associated with noise.

1 Introduction

Noise immunity has always been a metric of interest in analog circuit design where the noise sources of principal concern are those associated with physical effects – shot noise, thermal noise, flicker noise, and burst noise[1]. In contrast, the pervasiveness of digital systems is in great part due to their inherent noise immunity. Digital circuits use a range of analog voltages to define a logic '1' and logic '0', where degradation of these voltages due to noise would cause them to fall out of the valid ranges. High-gain logic gates, the best examples being CMOS inverters, restore these logic values by means of nonlinear voltage transfer characteristics which significantly reduce noise near the high and low voltage rails[2]. These gates are usually referred to as *restoring logic gates.*

The high gain of digital circuits, however, results in new "man-made" noise sources, which can be several orders of magnitude greater than those associated with the physical silicon devices. This first became a problem in the context of mixed digital-analog IC's in which the noisy digital circuits could strongly influence the noise-sensitive analog ones[3]. With the continued scaling of CMOS technology and with performance requirements which are driving designs in the direction of more noise-sensitive dynamic circuits, man-made noise sources are becoming an issue even in purely digital designs. In modern CMOS processes, more levels of interconnect (> 6 levels) are being packed closer together (minimum spacing $< 0.3\ \mu m$), increasing the amount of capacitive coupling between nets. To maintain drive strength in the face of scaled-down power supply voltages, threshold voltages are also scaled lower. Lower threshold voltages result in lower noise margins and increased leakage noise. Noise in deep submicron digital circuits must now be analyzed and designed as a metric of comparable importance to area, timing, and power.

2 Noise in deep submicron digital designs

2.1 What is noise?

We begin this discussion with a few useful definitions. An *evaluation node* is a circuit node that forms the connection between channel connected components in the design. Pass transistors are one exception to this rule, in that it is useful to treat both the source and drain as evaluation nodes. *Static evaluation nodes* are evaluation nodes which always have a conducting path to power or ground. *Static circuits* are circuits in which all evaluation nodes are static. The impedance which holds an evaluation node high or low is referred to as the *node impedance. Dynamic evaluation nodes* are evaluation nodes that during some part of normal system operation are disconnected from power and ground; that is, they have an infinite node impedance and a logic value determined by a charge stored on a capacitor. *Dynamic circuits* are circuits which contain dynamic evaluation nodes.

Noise can be defined as anything that causes the voltage of an evaluation node to deviate from the nominal supply or ground rails when it should otherwise have a stable high or low value (i.e., the node is not switching) as determined by the logic and delay of the circuit.

Noise can be characterized by its peak magnitude relative to the nominal supply and ground rails and its behavior in the time-domain. Noise sources that reduce an evaluation node voltage below the supply level (VDD) are denoted V_H, while noise sources that increase an evaluation node voltage above the ground level (GND) are denoted V_L. Noise may also be bootstrapping if it increases a node voltage above the supply level (V_{H^*}) or below the ground level (V_{L^*}). For the purposes of determining how circuits respond to noise, one can abstract time-domain response into one of two categories – DC noise and pulse noise as shown in Figure 1. These two noise behaviors are discussed in detail in the context of noise sources in Section 2.2. Pulse noise on an otherwise high logic level is, for example, denoted as V_H^{pulse} while DC noise on an otherwise high logic level is denoted as V_H^{DC}.

ICCAD '96
1063-6757/96 $5.00 © 1996 IEEE

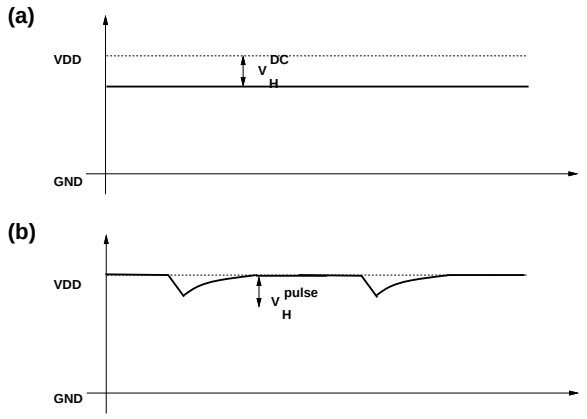

Figure 1: Time domain abstractions for noise: (a) DC noise and (b) pulse noise

2.2 Noise sources in digital systems

The noise sources most relevant to digital design are leakage noise, power supply noise, charge-sharing noise, and crosstalk noise.

2.2.1 Leakage noise

Leakage noise is generally applied to the context of dynamic nodes and comes from two sources. There is leakage noise due to the off current of FETs, which allows charge to drain from or accumulate on the dynamic node. This is largely due to subthreshold current and is directly determined by the threshold voltage and the temperature. Another leakage noise source is minority carrier back-injection into the substrate due to bootstrapping. This is sometimes referred to as substrate noise in the context of mixed analog-digital ICs[4]. One solution for substrate noise is to introduce guard bands, n-type diffusions tied to the supply voltage to collect the minority carrier electrons[3, 5]. Substrate noise becomes less of a problem at scaled power supply voltages because of the need to bootstrap more than $\sim 0.6V$ and, therefore, is not considered in the static noise analysis methodology discussed in Section 3. Leakage noise can also be used in reference to static evaluation nodes, in which the "leakage" is ratioed against a static path holding the node in the opposite direction. Leakage noise is a DC noise source because it changes the steady-state logic high or low voltage value on a time scale which is slowly varying with respect the the system clock, a time scale which we refer to as the *phase time*.

2.2.2 Charge-sharing noise and crosstalk noise

Charge-sharing noise is produced by charge redistribution between a dynamic evaluation node and internal nodes of the circuit. Figure 2(a) shows a circuit in which charge sharing noise is graphed under four conditions in Figure 2(b).[1] In *(i)*, the charge sharing noise is calculated in the absence of a half-latch PFET device for A1, A2, A3, and A4 simultaneously switching from low to high, while B1, B2, B3, and B4 are low. *(ii)* shows the same situation in the presence of a half-latch. In *(iii)*, A1, A2, A3, and A4 are assumed to be logically orthogonal (i. e., at most one of these signals may be '1' at the same time), a Boolean satisfiability constraint that can be verified through binary-decision diagram techniques[7, 8], but no half-latch is used. In *(iv)*, logical orthogonality and a half-latch are assumed.

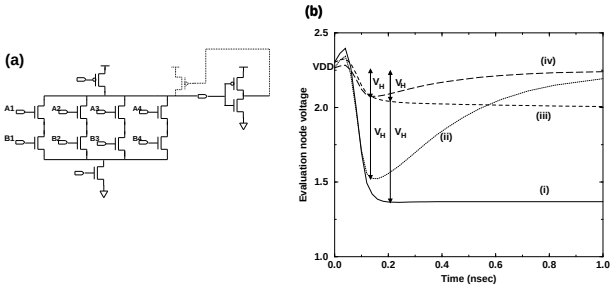

Figure 2: Charge sharing noise: (a) circuit and (b) dynamic node voltage

Crosstalk noise is the voltage induced on a node due to capacitive coupling to a switching node of another net. Figure 3 shows the noise coupling onto three different types of evaluation nodes from a single noise source switching with a fall time of $100ps$ with simple capacitive coupling.

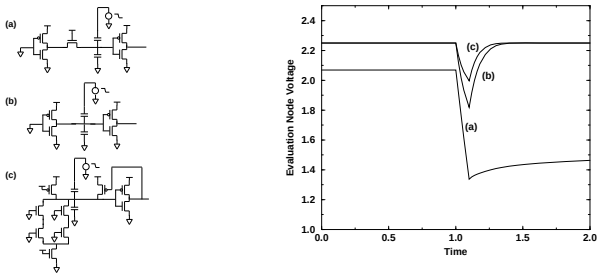

Figure 3: Coupling noise calculated for three circuit topologies (a) Pass transistor (b) Two inverters (c) Dynamic node of domino stage

Both crosstalk and charge-sharing noise are pulse noise sources, in which the leading edge is determined by a switching signal on the chip and the trailing edge is determined by the node impedance charging or discharing the capacitance of the evaluation node. The time constant of this response is referred to as the *restoring time constant*, since it is the time it takes the node to be restored to its static value. For dynamic nodes, as in cases *(iii)* and *(i)* of Figure 2, this

[1] The simulations shown in this paper are performed in a $0.5\mu m$ CMOS process described in detail elsewhere [6].

time constant is infinite (i.e., the node never recovers). For static paths that drive through pass gates, as in Figure 3(a), this time constant can be very long.

2.2.3 Power supply noise

Power supply noise explicitly refers to noise appearing on the supply and ground nets of the chip and coupled onto evaluation nodes through a FET conduction path. Figure 4 shows the actual supply variation as measured on a fourth-level metal sense point near the on-chip cache of a CMOS microprocessor running at a clock period of 7 nsec. Power supply noise contains

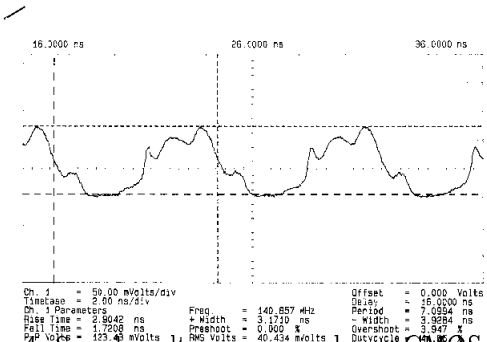

Figure 4: Supply voltage measured on a CMOS microprocessor during peak operation. Peak-to-peak amplitude is 123 mV.

both a DC and sinusoidal content. The DC component of power supply noise is produced by the IR drop through the power and ground nets due to the average current demands over the chip. The DC component of power supply noise can be reduced through a denser or wider interconnect structure for the power and ground network. The sinusoidal component of power supply noise comes from the RLC response of the chip and package to current demands that peak at the beginning of the clock cycle. In addition to sinusoidal variation, a sharply peaked current demand may also produce higher frequency components, which can be largely suppressed by the use or placement of on-chip decoupling capacitors[9]. In Figure 4, there is some higher frequency content around 1 GHz. Since power supply variations vary slowly relative to circuit frequency response, they are generally treated as DC for analysis purposes.

2.3 How do we know when the system will function?

The fundamental goal of a conservative noise analysis methodology is to guarantee that on every evaluation node in the circuit, the correct '1' or '0' value is defined by voltages that fall within one of two valid ranges for all times that a stable high or low logic value should be present as determined by the logic and delay of the circuit. For dynamic evaluation nodes, this goal is essential since no static path exists to restore the correct logic state in the case of noise. It is also essential for many bistable circuits. One example of this would be V_{L^*} coupling noise feeding the pass gate of a latch in which the gate of the pass gate is '0' and the latch stores a '1'. If V_{L^*} exceeds the threshold voltage of the NFET of the pass gate, the latch can be drained and flipped to a '0' without a chance for recovery. Static evaluation nodes in the absence of feedback will, in general, recover from pulse noise if one waits for a restoring time constant to elapse; however, the stability of logic signals due to noise is not something that is practical to manage in predicting performance and cycle time.

One way to ensure functionality for static CMOS logic gates is to define DC noise margins, NM_L and NM_H such that the bistable circuits shown in Figure 5 do not switch and $NM_H + NM_L$ is maximum[10]. This corresponds to biasing each gate at the unity gain point in its DC voltage transfer characteristic. For noise $V_H < NM_H$ and $V_L < NM_L$, gates will always be biased into regions of their voltage transfer characteristics in which noise is attenuated.

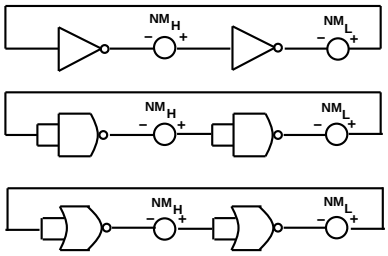

Figure 5: "Latch" circuits which define DC noise margins for CMOS NOR, NAND, and NOT gates.

DC noise margins are much too conservative to apply against the magnitude of pulse noise sources, because they fail to consider the fact that logic gates act as low-pass filters. *Noise tolerance* is a generalized view of noise margins applied to triangular pulsed noise sources, in which different noise margins are calculated for different pulse amplitudes and widths[11].

In order to handle nonrestoring logic gates, such as pass transistors, and provide more precision in handling pulse noise and bootstrap noise sources, one can further generalize the idea of noise tolerance, which applies to an individual logic gate, to the concept of *noise stability* which applies to the entire digital circuit. Consider the case in which noise is present on every evaluation node juxtaposed in time in the worst allowable way. The worst possible magnitude at which noise can exist in the circuit defines a "bias point." At this bias point, the circuit is defined to be noise stable if for any one noise source, a small change in its amplitude of δV_i results in a noise amplitude change on any evaluate node of δV_j such that $|\delta V_j / \delta V_i| < 1$ with $i, j \in \{L, H, L^*, H^*\}$.

2.4 Noise and delay

As an aside, capacitive coupling and power supply variations can also have a direct effect on delay that must also be considered independent of this noise analysis. Leakage noise and power supply noise result in lower or higher supply levels, which reduce

or enhance the current drive of a circuit and consequently increase or decrease the delay. Coupling noise can cause the effective line capacitance to increase or decrease in the presence of simultaneously switching noisy lines, increasing or decreasing the delay.

3 Static noise analysis

In this section, we briefly introduce a static noise analysis methodology, called *Harmony*, which enforces circuit stability. Details will be presented elsewhere[12]. Dynamic simulation is not feasible for checking noise on designs with tens of millions of transistors. Instead, as with timing, static analysis techniques which couple simulations on small numbers of transistors (basically individual gates) with a path trace are used. Worst-case preconditioning assumptions and (in most cases) worst-case temporal relationships drive the simulations. At the same time, global interconnect is handled with more sophistication than local "circuit" interconnect in determining the magnitude of coupling noise. Analysis is based on constructing a noise graph abstraction, which follows very closely the timing graphs used in static timing analysis[13]. As in the case of static timing analysis in which waveforms are abstracted as saturate ramps, the actual dynamic waveforms are simplified to pulse or DC waveshapes. In the case of precharged logic, only noise against the reset state is considered since this is the transition that contains the logical information of the circuit. Power supply noise is also assumed to be characterized by a DC value.

3.1 Constructing a noise graph

The steps to constructing a noise graph from a circuit are outlined below.

3.1.1 Evaluation node identification and direction setting

The evaluation nodes in the circuit are identified. Basic topologies are recognized – restoring logic stages, domino logic stages, simple inverter feedbacks, weak pullups, and pass transistor stages. Evaluation nodes which are precharged in dynamic logic stages are identified as *precharge evaluation nodes*. These nodes potentially require charge-sharing noise calculation as discussed in Section 3.1.3. Transistor directions are set using the simple rules-based approach of Jouppi[14]. This is of principal concern for pass gates. Feedbacks are recognized so that they can be independently preconditioned. Floating capacitors are broken and tied to ground when connected to any nodes other than evaluation nodes (i.e., coupling is only considered to evaluation nodes). This is not a significant source of error since evaluation nodes will be the only nodes with significant interconnection length. Nodes which feed NFET pass transistors are identified as *bootstrap-down evaluation nodes*, since they will be sensitive to V_{L^*} while nodes which feed PFET pass transistors are identified as *bootstrap-up evaluation nodes* since they will be sensitive to V_{H^*}.

3.1.2 Calculate the coupling noise and restoring time constants for each evaluation node in the circuit

For wire lengths less than $\sim 1mm$, we ignore resistance. In addition, all signals coupled to the given evaluation node are assumed to switch simultaneously at a minimum slew time (t_{slew}^{min}). In Section 4, we show how these assumptions are relaxed in the case of long interconnect. Static paths are preconditioned to keep the evaluation nodes quiet against the noise source, including feedback paths. All pass transistors are configured off except that configuration which produces the highest static resistance to supply or ground to produce worst-case coupling noise on the node in question. V_H and V_L due to coupling are calculated at each evaluation node, except precharge evaluation nodes, for which only V_H or V_L is calculated, depending on whether the node is precharge high or low. For bootstrap-down evaluation nodes, V_{L^*} is also calculated while for bootstrap-up evalaution nodes, V_{H^*} is calculated. Each coupling noise waveshape is abstracted as a pulse waveform with a leading edge determined by a maximum slew time (t_{slew}^{max}) and a trailing edge determined by a restoring time constant. Figure 3 shows the typical results of such a simulation.

3.1.3 Charge sharing noise is calculated on all precharge evaluation nodes

Charge-sharing noise is calculated for all *precharge evaluate nodes* identified topologically. In every stack connected to the precharge evaluate node, the top device and bottom device are configured off. All other devices in the leg of the NFET stack are on. The top devices are then switched simultaneously for parallel stacks. In the presence of an evaluate NFET foot device in the domino NFET stack, the foot device is considered on. Logic constraints can be used to reduce pessimism as shown in Figure 2. The charge sharing and coupling noise are summed to determine the total pulse noise on the evaluation node.

3.1.4 Draw the directed segments connecting evalution nodes in the noise graph

Now that the evaluation nodes and pulse noise sources on each of these nodes is defined, the next step is to connect the nodes by directed segments according to the circuit structure to define the noise graph.

There are two basic types of segments in a noise graph – a *restoring* segment and a *propagate* segment. A *restoring* segment connects evaluation nodes in the case that the DC transfer characteristic shows gain. Restoring segments are identified by dashed lines and connect inputs and outputs of restoring logic gates. A *propagate* segment is one which propagates noise from one evaluation node to another without gain. Channel conduction in a pass gate, for example, is represented by a propagate segment. Propagate segments are denoted by solid lines. Both types of segments are labelled by noise type. A power-supply noise segment is a special propagate segment in which the source

node is ground or supply. The power supply noise is propagated as V_L from ground and V_H from supply. For bootstrap-up nodes, V_{H^*} is also propagated from supply and V_{L^*} from ground. Figure 6 shows an example of a noise graph, in this case for a latch driving a domino gate.

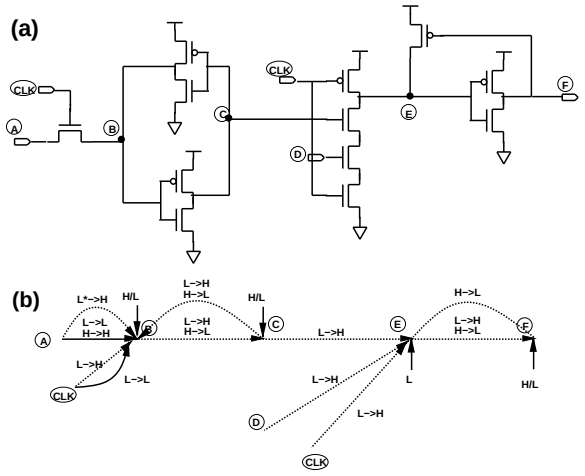

Figure 6: (a) Circuit and (b) noise graph for a latch driving a domino stage

3.2 Propagating noise through a graph search

Once the noise graph is constructed, it is searched in a breadth-first fashion to propagate noise through the network and, in the case of restoring segments, to perform the sensitivity tests required to ensure noise stability.

3.2.1 Propagate segments

On each propagate segment, the peak output voltage waveform is calculated for the DC noise and pulse noise appearing on the input. This peak voltage is propagated as DC noise onto the output evaluation node; therefore, the DC noise appearing on the input is that propagated from previous stages as well as that introduced as power supply noise. This assumption introduces pessimism into the analysis but eliminates the need to propagate time-domain response while taking into account the low-pass filter characteristics of logic gates.

3.2.2 Restoring segments

For restoring logic, there will, in general, be multiple inputs entering the same stage. A set of inputs form a *noise equivalence class* when the worst-case response at the output is produced when those inputs are simultaneously excited by noise. This allows us to perform a worst-case analysis for noise without knowledge of the relative timing of noise events. Two distinct types of simulations are performed for restoring logic segments. The first is a simulation to check stability and the second, performed conditionally based on the results of the first, propagates noise to the output. In both simulations, for a given input and given noise type, V_H or V_L, other conduction paths in parallel with the one in question are preconditioned "on" except for the transistor closest to the evaluate node, which will be noise equivalent to the input under test. Feedback paths are also preconditioned to hold the node at the static level during both simulations.

The first simulation checks the condition of noise stability, which must be done for restoring logic stages since they have gain. Let i denote different inputs of a noise equivalence class. Let $(V_\mu^{DC})_i$ denote the DC voltage on input i, let $(V_\mu^{pulse})_i$ denote the pulse voltage appearing on input i, and let V_ν^{DC} be the output voltage, then for all equivalence classes, for all i in an equivalence class,

$$\left|\frac{\partial V_\nu^{DC}}{\partial (V_\mu^{DC})_i}\right| < 1 \; and \; \left|\frac{\partial V_\nu^{DC}}{\partial (V_\mu^{pulse})_i}\right| < 1 \qquad (1)$$

where $\mu, \nu \in \{L, H, L^*, H^*\}$. This is a sufficient condition to ensure circuit stability. The first of the two conditions localizes the global stability definition of Section 2.3 to an effective stability analysis on a single stage by requiring subunity sensitivity in magnitude against all the noise propagated from previous stages. The derivatives are evaluated at the "bias" point determined by the noise appearing on the inputs of the equivalence class. These sensitivities can be readily calculated in certain timing simulators in the time-domain by the direct or adjoint methods[15, 16]. In the event of a violation, the noise and noise sensitivities for all the inputs in the equivalence class containing the violation are reported. The noise source (input and DC/pulse) of the highest sensitivity is the most likely candidate for repair.

For those equivalence classes which do have stability violations, a second simulation is performed similar to that performed for propagate segments in which the *actual* DC and pulse noise is simultaneously applied to each input in the class to determine the output noise. For those equivalence classes which are in violation, a recovery mechanism exists to provide a worst-case voltage to propagate forward in the path search. In this simulation the same pulse is applied to each input in an equivalence class in the absense of a DC bias to determine the output voltage at the unity gain point. This generally represents the most pessimistic output noise. Figure 7 shows the "pulse transfer characteristic" that would be calculated as part of this simulation for an and-or-invert gate for one particular noise equivalence class with a restoring time constant of $1ns$. The inset shows the configuration used in each case. These look similar in their hysteretic behavior to the AC transfer characteristics of Ref. 17.

3.2.3 Implied rules of the path trace

A DC noise value and a pulse noise value are stored on each evaluation node during path traversal. When

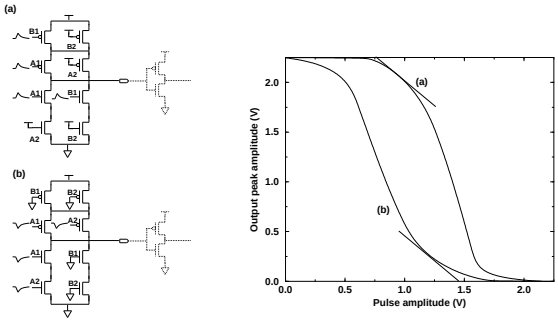

Figure 7: Pulse transfer characteristic and-or-invert static CMOS gate for (a){A1, B1} $V_L \rightarrow V_H$ equivalence class and (b) {A1, A2} $V_H \rightarrow V_L$ equivalence class

multiple restoring segments converge on a node, the propagated noise is the worst of the converged segments. When multiple propagate segments converge on a node, the noise sources are summed. When propagate and restoring segments converge on an evaluation node, the worst of the propagated output noises from the restoring segments is added to the sum of the propagate segment noises.

4 Long interconnect analysis

In calculating coupling as in Section 3.1.2, we made two significant simplifying assumptions regarding interconnect – that we could ignore resistance and that we could assume that all of the coupling sources were simultaneously switching. In calculating the coupling noise in long interconnect, resistance must be considered. In addition, coupling noise calculation on long interconnect is one place where the addition of timing information to break temporal correlation can have a big effect on reducing pessimism in the analysis.

We define the *primary net* as the net on which we wish to calculate the noise at each receiver. The *secondary nets* are those nets coupled to the primary net. There are two possible network simplifications that can be used in long interconnect analysis, both shown in Figure 8. Driver outputs are modeled as resistances and receiver inputs are modeled as capacitances. These linear networks are easily analyzed by moment-matching techniques such as AWE or PVL[18, 19]. Four poles are usually more than adequate to determine the response. In Figure 8(a), only the resistance of the primary net is considered. The resistances on the secondary nets are ignored. In Figure 8(b), the resistances on both primary and secondary nets is considered, which is more accurate since it considers RC delays in the secondary net. The coupling capacitance between the secondary nets and other nets is assumed to be grounded.

In both cases, two types of analysis are possible – one that assumes worst case temporal correlation of noise sources and one that attempts to break temporal correlation with the addition of information from static timing analysis.

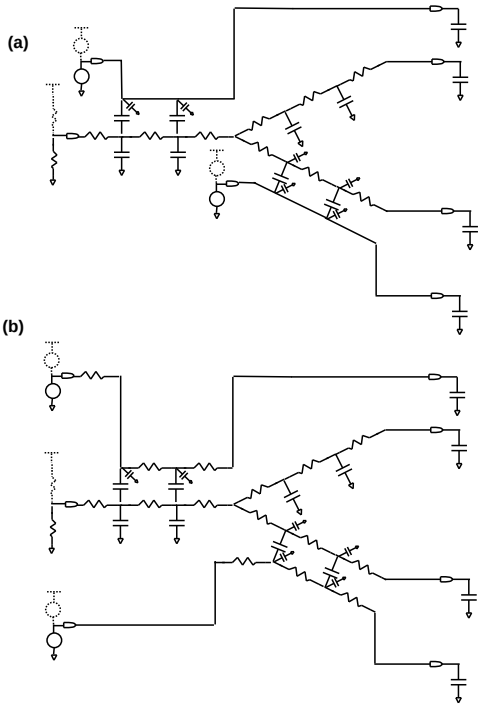

Figure 8: Long interconnect analysis for (a) resistances only on the primary net and (b) for resistances on both the primary and secondary nets

4.1 Worst-case temporal correlation of noise sources

In this case, each noise source is a voltage waveform with a slew time t_{slew}^{fast}. The sources are applied one at a time while the other sources are grounded and the peak magnitudes are added at the receiver. This is equivalent to assuming that the noise voltage arrives at each coupling capacitor at the time required to produce the worst coupling at the quiet receiver.

4.2 Using static timing analysis to break temporal correlation

In this case, we consider the following additional information from static timing analysis – early and late mode arrival times and best case slews (t_{slew}^{BC}) at each secondary net driver. Let the driver of secondary net i begin rising at time t_i. The problem is then to find the times t_i that fall within the valid early and late mode arrival time windows for each secondary net driver that produce the highest peak coupling noise on the given receiver. One can do this by calculating the response for each driver individually and determining the peak position at the quiet receiver. The peak position skews immediately translate back to the arrival time t_i skews. The worst-case combination that satisfies the "timing orthogonality" constraints is selected.

4.3 Inductance

With the increasing use of wide, thick wires and the potential introduction of copper interconnect, inductance is becoming a concern[20]. There are two conditions which must be met for inductance to be significant:

- $\mathcal{R} < \omega\mathcal{L}$, where $\mathcal{R}$ is the resistance per unit length, $\mathcal{L}$ is the inductance per unit length, and ω is the characteristic frequency, roughly determined by the typical slew times (t_r) by $\omega = 2\pi/t_r$.
- $t_r < 2t_f$, where t_f is the time-of-flight given by $t_f = l\sqrt{\mathcal{LC}}$ where $\mathcal{C}$ is the capacitance per unit length and l is the length of the line[21].

Even if these two conditions are satisfied, the far end response will behave like a distributed RC line if $\mathcal{R}l > 2Z_o$, where Z_o is the characteristic impedance of the line, given by $Z_o = \sqrt{\mathcal{L}/\mathcal{C}}$. The complexity in calculating inductance is determining the current return path. Roughly speaking, the current will return through the path of minimum $\mathcal{R} + j\omega\mathcal{L}$. For low ω, the return path will be the path of minimum resistance. In the case of solder-ball flip-chip packaging in which supply and ground are introduced throughout the chip, this will be a return path through the package for most long interconnect runs, a path of very high inductance. As ω increases, the current return will seek a lower inductance (and higher resistance) return on chip. In most cases, it will find a return on a power or ground bus but in the absence of adequate power or ground distribution, current may return through a signal line.

Inductance in the global interconnect has two effects on noise. At the receivers of switching nets, inductive ringing may be a concern. For quiet nets within actively switching environments, inductive coupling tends to worsen noise at the near end while reducing noise at the far end.

To illustrate these points, five parallel lines on $2\mu m$ thick copper interconnect are analyzed as shown in Figure 9(a). The lines are $3.6\mu m$ spaced $1.8\mu m$ apart. Line A is grounded and the current return path assumed for inductance calculations. Line E is actively switched while the other lines are held quiet by active drivers. Figure 9(b) shows the response at the near and far end of line E in the case of RC and RLC modelling. There is ringing at the far end when inductance is considered. Figure 9(c) shows the response at the near and far end of line D. Including inductance in the analysis reduces the noise at the far end while increasing the noise at the near end.

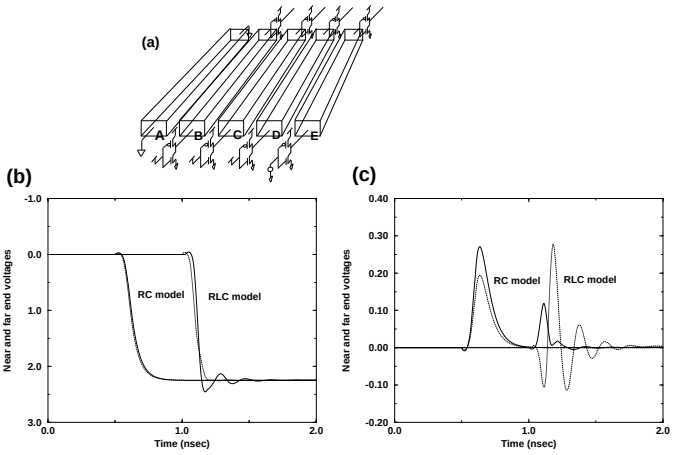

Figure 9: (a) Example long interconnect structure, (b) response on the driver line, and (c) response on the quiet line. Near end is dashed and far end is solid.

5 Design issues in reducing noise

There are many design techniques that can be used to reduce noise. Many of these techniques are enforced as rules early in the design process, while others must be more carefully traded off. Certain circuit topologies can be restricted because of their noise sensitivity. In many cases, NFET-only pass gates are disallowed because of their sensitivity to power supply noise. Pass gates at the ends of long wires are almost always forbidden because of their sensitivity to bootstrapping coupling noise. In general, buffers and inverters can be introduced to clean up noise at the receivers of noisy lines at the expense of additional delay. Charge-sharing noise can be controlled to a considerable extent through the careful use of logic constraints as discussed in Section 3.1.3 and demonstrated in Figure 2 or through the addition of "babysit" devices to statically hold internal nodes high or low as the cost of additional power.

Many device sizing techniques can be used to improve noise margins, almost always at the expense of additional delay. Half-latches can be added or sized up for dynamic nodes. Device lengths can be tuned up to increase thresholds and reduce leakage. Beta ratios can be adjusted to improve one noise margin at the expense of the other.

To reduce coupling noise, the spacing between wires can be increased or signals can be alternately routed with power or ground. These techniques have been applied to constraint-driven routing, primarily in the context of analog or mixed-signal IC's[22]. Sometimes it is sufficient to increase the driver strength to help hold the quiet line quiet, but this is less effective in long wires because of resistive shielding. In some cases when coupling noise is accentuated by simultaneously switching noise sources, buffers or inverters can be added onto noisy nets as delay elements to offset their switching times. In other cases, coupling noise can be reduced by decreasing overall wire length by inserting a repeater. In extreme cases, differential buses or DCVS logic can be used.

6 Conclusions

In this paper, we have defined noise and discussed the noise sources relevant to digital systems. We have also defined a metric, noise stability, for determining design "goodness" and described a static analysis

methodology for verifying a design against this metric. Special modeling issues associated with long interconnect were discussed. Several future issues were noted in the course of the discussion.

7 Acknowledgements

The authors gratefully acknowledge H. Smith and D. Becker for helpful discussions about power supply noise and for providing the data in Figure 4. The authors also acknowledge C. Visweswariah for a careful reading of the paper.

References

[1] P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, John Wiley and Sons, 1984, Chapter 11.

[2] P. Larsson and C. Svensson, "Noise in digital CMOS circuits," *IEEE Journal of Solid-State Circuits*, Vol. 29, No. 6, 1994, pp. 655-661.

[3] L. D. Smith, et al, "A CMOS-based analog standard cell product family," *IEEE Journal of Solid-State Circuits*, Vol. 24, No. 2, 1989, pp. 370-379.

[4] P. K. Su, M. J. Loinaz, S. Masui, and B. A. Wooley, "Experimental results and modeling techniques for substrate noise in mixed-signal integrated circuits," *IEEE Journal of Solid-State Circuits*, Vol. 28, No. 4, 1993, pp. 420-430.

[5] P. E. Gronowski, et al, "A 433 MHz 64b Quad-Issue RISC Microprocessor," *1996 IEEE International Solid-State Circuits Conference*, pp. 222-223.

[6] C. W. Koburger, III, et al, "A half-micron CMOS logic generation," *IBM Journal of Research and Development*, Vol. 39, No. 1/2, 1995, pp. 215-227.

[7] R. E. Bryant, "Graph-Based algorithms for boolean function manipulation," *IEEE Transactions on Computers*, Vol. C-35, No, 8, 1986, pp. 677-691.

[8] A. Kuehlmann, A. Srinivasan, and D. P. LaPotin, "Verity – A formal verification program for custom CMOS circuits," *IEEE Journal of Research and Development*, Vol. 39, No. 1/2, 1995, pp. 149-265.

[9] H. H. Chen, "Minimizing chip-level simultaneous switching noise for high-performance microprocessor design," *Proceedings of IEEE International Symposium on Circuits and Systems*, Vol. 4, 1996, pp. 544-547.

[10] L. A. Glasser and D. W. Dobberpuhl, *The Design and Analysis of VLSI Circuits*, Addison-Wesley, 1985, pp. 205-212.

[11] G. A. Katopis, "Delta-I noise specification for a high-performance computing machine," *Proceedings of the IEEE*, Vol. 73, 1985, pp. 1405-1415.

[12] K. L. Shepard and V. Narayanan, "Harmony: A methodology for deep submicron noise analysis", in preparation.

[13] R. B. Hitchcock, G. L. Smith, and D. D. Cheng, "Timing analysis of computer hardware," *IBM Journal of Research and Development*, Vol. 26, 1982, pp. 100-105.

[14] N. P. Jouppi, "Derivation of signal flow direction in MOS VLSI," *IEEE Transactions on Computer-Aided Design*, Vol. CAD-6, No. 3, 1987, pp. 480-490.

[15] S. W. Director, "The Generalized adjoint network and network sensitivities," *IEEE Transactions on Circuit Theory*, Vol. CT-16, No. 3, 1969, pp. 318-323.

[16] P. Feldman, T. V. Nguyen, S. W. Director, and R. A. Rohrer, "Sensitivity computation in piecewise approximate circuit simulation," *IEEE Transactions on Computer-Aided Design*, Vol. 10, No. 2, 1991, pp. 171-183.

[17] J. M. Zurada, Y. S. Joo, S. V. Bell, "Dynamic noise margins in MOS logic gates," *Proceedings of ISCAS '89*, pp. 1153-1156.

[18] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Transactions on Computer-Aided Design*, Vol. 9, No. 4, 1990, pp. 352-366.

[19] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Pade approximation via the Lanczos process," *IEEE Transactions on Computer-Aided Design*, Vol. 14, No. 5, 1995, pp. 639-649.

[20] A. Deutsch, et al, "Modeling and characterization of long on-chip interconnections for high-performance microprocessors", *IBM Journal of Research and Development*, Vol. 39, No. 5, 1995, pp. 547-567.

[21] H. B. Bakoglu, *Circuits, Interconnects, and Packaging for VLSI*, Addison Wesley, 1990, Chapter 6.

[22] E. Malavasi, E. Charbon, E. Feit, and A. Sangiovanni-Vincentelli, "Automation of IC Layout with Analog Constraints," *IEEE Transactions on Computer-Aided Design*, Vol. 15, No. 8, 1996, pp. 923-942.