Analytical Modeling and Characterization of Deep-Submicrometer Interconnect

DENNIS SYLVESTER, MEMBER, IEEE AND CHENMING HU, FELLOW, IEEE

Invited Paper

This work addresses two fundamental concepts regarding deep-submicrometer interconnect. First, characterization of on-chip interconnect is considered with particular attention to ultrasmall capacitance measurement and in-situ noise evaluation techniques. An approach to measuring femto-Farad level wiring capacitances is presented that is based on the concept of supplying and removing charge with active devices. The method, called the charge-based capacitance measurement (CBCM) technique, has the advantages of being compact, having high-resolution, and being very simple. We also present a novel time-domain measurement scheme for on-chip crosstalk noise that is based on the use of cascaded high-speed differential pairs to compare a user-defined reference voltage to the unknown noise peak value. The noise measurement technique complements a delay measurement to directly evaluate the impact of capacitive coupling on delay for various victim and aggressor driver sizes as well as arbitrary waveform timing and phase alignments.

The second area of emphasis in this work is analytical interconnect modeling. Several important effects are modeled, including a rigorous crosstalk noise model that also includes a timing-level model. Results from this noise model show it to provide accuracy within 10% of SPICE for a wide range of input parameters. The noise model can also be calibrated and verified with comparison to the noise measurement scheme described in this work. A fast Monte Carlo approach to modeling the circuit impact of back-end process variation is presented, providing a better depiction of real 3-sigma performance spreads compared to the traditional skew-corner approach. Finally, a comprehensive system-level performance model called Berkeley Advanced Chip Performance Calculator (BACPAC) is developed that accounts for a number of relevant deep-submicrometer system design issues. BACPAC has been implemented online and is useful in exploring the capabilities of future very large scale integration systems as well as determining trends and tradeoffs inherent in the design process.

Keywords—Capacitance measurement, crosstalk, inductance, interconnections, noise measurement, ultralarge-scale integration.

I. INTRODUCTION

A. CMOS Scaling Overview

Advances in complementary metal–oxide–semiconductor (CMOS) technology over the past 25 years have led to an explosion in the performance of integrated circuits (ICs). The concept of CMOS scaling refers to the miniaturization of MOS transistors in a systematic manner such that the new smaller devices are faster, more power-efficient, and reliable [1]. There are several branches of scaling including ideal, constant-voltage, and quasi-ideal. In general, MOSFET channel lengths (L) and oxide thicknesses (T_{ox}) are reduced along with supply voltages (V_{dd}). This results in a shorter channel transit distance and smaller voltage swing yielding faster devices. In addition, threshold voltages (V_{th}) are reduced to maintain sufficient current drive with the dropping supply voltage. The most fundamental limitations to scaling devices are the exponential increase in leakage current with reduced L and difficulty in fabricating ultrathin gate oxides. While these roadblocks are significant, the advances in CMOS device performance over the past two decades are expected to continue for at least another ten years. A scaling factor S defines the change in a certain physical parameter (e.g., gate oxide thickness) from one technology generation to the next. For instance, changing the channel length from 0.35 to 0.25 μm gives S the value 0.72 for L. A typical scale factor is 0.7 for a number of important parameters.

The delay of a transistor τ can be modeled to the first order using the expression

$$\tau = \frac{C_L V_{\text{swing}}}{I}$$  (1)

where C_L is the load capacitance, V_{swing} is the voltage swing of interest (e.g., 50% of the supply voltage), and I is the drive current of the device. By examining how these three parameters scale from one process to the next, we can estimate how device delay will be affected as well. Table 1 presents a generalized look at ideal scaling for deep-submicrometer
(DSM, feature sizes <0.25 μm) processes. It has been documented that, beginning at about the 0.35-μm generation, further increases in drive current (normalized to device width) are difficult to achieve because of velocity saturation, mobility degradation, and parasitic source-drain resistance [2]. However, even with constant drive current we see that transistor delay decreases by $S$ each generation, yielding faster devices with each technology shrink. Since device area is also reduced quadratically, we expect a constant power density with ideal scaling. In the quest for additional functionality, chip area is rising slowly with technology advancement. Thus, it is anticipated that the total device power consumption will increase.

### B. Interconnect Scaling

In direct contrast to the performance advantages inherent in MOS transistor scaling is the phenomenon of interconnect "reverse" scaling. Reverse scaling refers to the concept that smaller interconnections actually yield larger delays due to the rapidly shrinking cross-sectional area of the wire that is used to conduct current. In this paper, we will use the terminology of front-end and back-end processes. Front-end processes are defined as steps in the fabrication process that create MOS transistors. These steps include deposition and etching of metals as well as interlevel dielectric (ILD) deposition. A diagram and photo of the back-end of a modern IC is shown in Fig. 1.

**Local and Global Wires:** Interconnect scaling can be broken down into two distinct components; local and global scaling. The distinction between the two can be made by first defining a local wire as a connection within a functional unit that spans only a small number of gate dimensions (known as a gate pitch). These local wires tend to be on the length scale of 10–500 μm in current technologies (~0.18 μm). Global wires serve to connect separate functional units and can have significantly larger wirelengths. One way to view this is that the length scale of local wires is set by the size of an individual gate which is very small. On the other hand, global wirelengths are set by both the size of a functional unit and the size of the entire chip since they span at least one functional unit and could attain a chip-side in length for the limiting case. These definitions serve to highlight the primary difference between the two major types of on-chip wiring.

Interconnect scaling approaches for local and global wires are summarized in Table 2. Two common scenarios are shown for both local and global wires; the ideal scaling rules are valid for both types of wires. In the case of local interconnections, a quasi-ideal approach to scaling can be taken that scales vertical and horizontal back-end parameters differently. Also, global wiring may employ a constant-dimension type of "scaling" rather than the ideal case.

**Table 1**

<table>
<thead>
<tr>
<th>Scaled Parameter</th>
<th>Ideal Scaling Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W$, $L$, $T_{ox}$, $X_{j}$</td>
<td>$S$</td>
</tr>
<tr>
<td>Substrate Doping, $N_{sub}$</td>
<td>$1/S$</td>
</tr>
<tr>
<td>Voltages: $V_{th}$, $V_{dd}$</td>
<td>$S$</td>
</tr>
<tr>
<td>$I_{dss}$ (for fixed W/L)</td>
<td>$S$</td>
</tr>
<tr>
<td>$C_{gate}$</td>
<td>$S$</td>
</tr>
<tr>
<td>$R_{in}$ ($\approx V_{dd}/I_{dss}$)</td>
<td>1</td>
</tr>
<tr>
<td>$\tau$ ($\approx CV/I$)</td>
<td>$S$</td>
</tr>
<tr>
<td>Power ($\approx CV^{2}I$)</td>
<td>$S^{2}$</td>
</tr>
<tr>
<td>Power $\cdot \tau$</td>
<td>$S^{2}$</td>
</tr>
<tr>
<td>Area / Device</td>
<td>$S^{2}$</td>
</tr>
<tr>
<td>Power / Area</td>
<td>1</td>
</tr>
</tbody>
</table>

The ideal scaling rules for interconnect basically serve to provide sufficient packing density for highly integrated designs. Since gates are being rapidly scaled down in size, more and more wires are needed for communication. Therefore, both linewidth and spacing are decreased by $S$ in each generation. The wire and ILD thicknesses are also reduced by $S$ for ease of process integration and roughly constant capacitances (per unit length). Local wires see a reduction in line length by the scale factor $S$. This is due to the reduction in gate pitch by $S^{2}$ (an $S$ reduction in each dimension that translates directly to the shorter wirelength). The wire-length reduction in local interconnects results in a constant RC delay. It should be noted here that the results of these interconnect scaling scenarios are predicated on the use of fixed materials, aluminum (Al) and silicon dioxide (SiO$_2$), for example. Changes in materials (e.g., copper wiring and low-$k$ dielectrics) may alter the results somewhat but the general trends remain. From Table 2, we also see an increase in current density by $1/S$ since the cross-sectional area of the wire is reduced quadratically while the current drive is only reduced linearly with $S$.

In quasi-ideal scaling, the vertical dimensions are scaled more slowly than the horizontal dimensions, resulting in a tall and narrow wire geometry after time. For example, with $S = 0.7$, the line and ILD thicknesses are scaled by only 0.837. Starting with a square wire, after two generations the scaled wire is 43% taller than it is wide. The performance advantages of this scaling scenario include the preservation of packing density since the horizontal dimensions (width and space) are still fully scaled. In addition, the quadratic
increase in resistance per unit length is dampened by the slightly larger line thickness. This leads to a better RC delay scale factor of $\sqrt{S}$ that more closely tracks the increase in transistor switching speed shown above.

There are two major problems with the quasi-ideal scaling approach. First, the increase in line thickness results in a higher aspect ratio (defined as the ratio of line thickness to linewidth) that yields more coupling capacitance ($C_c$) to neighboring wires. The issue of coupling capacitance is discussed in more detail in the next section but in general the rise in $C_c$ leads to enhanced coupled noise effects that degrade both signal integrity and delay predictability. The second problem is that manufacturing high-aspect ratio lines is difficult since a deep and narrow trench must be completely filled with metal to eliminate possible opens or resistance fluctuations. For this reason, lines with aspect ratios of greater than 2.5 can be hard to reliably mass-produce.

The ideal scaling scenario for global wires has one fundamental difference from that of local interconnects. The length scale for global wires is set by the chip-side length (as well

<table>
<thead>
<tr>
<th></th>
<th>Local Wiring</th>
<th>Global Wiring</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ideal Scaling</td>
<td>Quasi-ideal Scaling</td>
</tr>
<tr>
<td>Linewidth &amp; Spacing</td>
<td>$S$</td>
<td>$S$</td>
</tr>
<tr>
<td>Wire Thickness</td>
<td>$S$</td>
<td>$\sqrt{S}$</td>
</tr>
<tr>
<td>ILD Thickness</td>
<td>$S$</td>
<td>$\sqrt{S}$</td>
</tr>
<tr>
<td>Wirelength</td>
<td>$S$</td>
<td>$S$</td>
</tr>
<tr>
<td>Resistance (per unit length)</td>
<td>$1/S^2$</td>
<td>$1/S^{3/2}$</td>
</tr>
<tr>
<td>Capacitance (per unit length)</td>
<td>1</td>
<td>- 1 (possible small increase)</td>
</tr>
<tr>
<td>RC delay</td>
<td>$1/S$</td>
<td>$\sqrt{S}$</td>
</tr>
<tr>
<td>Current density</td>
<td>$1/S$</td>
<td>$1/\sqrt{S}$</td>
</tr>
</tbody>
</table>
as functional unit size), which is not shrinking, as are gate dimensions. As a result, the global RC delay scales as $1/S^3$ rather than remaining constant. To arrive at this figure, global wirelength is assumed to scale up as $\sqrt{S}$, or by about 20% per generation. Using a typical scale factor of 0.7, this translates to an unacceptable 192% rise in RC delay from one technology to the next. The different length scaling of global and local wires is the main reason behind the evolving paradigm shift in chip design from device-centric to interconnect-centric.

In order to combat these rising RC delays, a constant dimension “scaling” approach to global wires has been suggested elsewhere [3] and is also advocated in [4]. Since the primary back-end parameters are fixed and not reduced, it is not actually an approach to scaling. The main concept is that by maintaining wide and thick wires at the upper metallization levels, a low RC product can be kept for global wires. As seen in Table 2, the global RC product only rises by $1/S$ in this scenario and combined with the introduction of new interconnect materials can be further limited. A positive side effect of constant dimension scaling is the suppression of noise effects since line spacing is not reduced. The foremost drawback to this approach is the potential lack of routing resources since packing density is being sacrificed at the expense of performance.

C. Motivation

Given the previous discussion of device and interconnect scaling, we can see a paradigm shift occurring in modern integrated circuit design. Whereas in the past the majority of design effort has been expended in transistor optimization and careful device placement, new emphasis is now being placed on the routing of interconnect between these devices. The “reverse scaling” phenomenon associated with on-chip wiring indicates that smaller interconnect dimensions can yield slower signal transmission. In addition to the rising importance of interconnect delay, other important effects such as coupling capacitance and its associated crosstalk noise and delay implications, power dissipation, increased process variation, and inductance also become more prominent due to scaling.

This paper will explore interconnect delay and noise models and characterization techniques, and discuss future trends in these areas. To begin, we motivate interconnect-related research by examining a number of key design metrics and how they are influenced by interconnect scaling. These metrics include delay, crosstalk noise, power, and process variability and they are all under increasing scrutiny by designers in the deep-submicrometer era.

Delay/Timing Closure: Increasing delay in the wires leads to poor prelayout estimates of delay. Since specifics of interconnect routes (e.g., line length, width) are not known prior to layout, statistical models based on prior design experience are used. These models are called wireload models and may result in large over- or underestimates in wirelength. This leads to errors in timing estimation and yields iterations in the design process. This problem is known as timing closure because the iterations may not always converge quickly; see [5] for a more detailed description of this issue.

Noise: In the quasi-ideal scaling approach to local interconnect, wires become taller and narrower. Combined with the shrinking linewidths and spacings in modern processes, designers are left with tall wires that are very close together. The result of this scaling scenario is a marked increase in coupling capacitance in the DSM regime. Fig. 2 shows the rise in $C_{c}/C_{\text{total}}$ for scaled processes where $C_{\text{total}}$ is the total wire capacitance (consisting of capacitance to upper and lower ground planes as well as $C_{c}$). Currently, for minimum pitch wires, $C_{c}$ accounts for roughly 70% of the total wire capacitance. This capacitive coupling to neighboring wires means that voltage changes on one wire can adversely affect the voltage level of another wire. The capacitance values in Fig. 2 were calculated using a two-dimensional (2-D) field solver for minimum pitch lower-level metals with upper and lower ground planes present.

Noise in a digital system can be defined as anything that causes a node to deviate from $V_{\text{dd}}$ or ground when it should otherwise have a stable high or low value [5]. Coupling capacitance is a source of noise in that it causes such deviations to occur. Noise sources in turn lead to signal integrity problems, and we now introduce two such signal integrity problems caused by $C_{c}$.

First, crosstalk noise results when a quiet victim line is acted upon by one or more neighboring aggressor lines. The coupling capacitance between the victim and aggressor lines is partially charged by the driving gates of the aggressors, which yields an unwanted voltage spike on the victim line. If this voltage spike is large enough it can cause logic faults (especially in dynamic circuits or pass-transistor logic). In the case of bootstrapped noise where the victim voltage level goes above $V_{\text{dd}}$ or below 0 V, there can be reliability concerns due to enhanced device stress (hot carriers) and possible forward-biased drain-substrate p–n junctions. The magnitude of the voltage spike, $V_{x}$, is a complex function of driver strength, $C_{\text{c}}$, fan-out capacitance, and wiring resistance. To the first order, however, $V_{x}$ can be viewed as proportional...
to the ratio of $C_c$ to $C_{\text{total}}$. As we have already discussed, this ratio is rising in modern processes due to tight pitches and high-aspect ratio lines. Therefore, we anticipate a rise in crosstalk noise in scaled technologies.

The second form of noise caused by $C_c$ is dynamic delay. While fundamentally related to the crosstalk noise problem, dynamic delay specifically refers to situations where the victim line is switching rather than static. Due to the fact that a victim line has substantial capacitance to neighboring wires rather than ground, the delay of the victim system (driver + wire) becomes a function of the neighboring wire switching activity. When the nearby aggressor lines are static, $C_c$ acts roughly as a ground capacitance. However, if the victim and aggressor switch simultaneously, the capacitance between these lines sees a different voltage swing $\Delta V$ than in the static case. For instance, if two capacitively coupled nodes have the same voltage waveforms, then the capacitance between these nodes sees a net voltage swing of $0 V$—the capacitance is not charged/discharged at all. If the same two nodes have complementary voltage responses, then the total voltage swing across the capacitor is $2V$, where $V$ is the voltage swing of each waveform. This is equivalent to switching the positive and negative electrodes of a capacitor—the effective voltage swing is $2V$. The result of this argument is that when aggressors and victims switch in opposite directions, a larger amount of charge needs to be supplied from the drivers to switch $C_c$ than in the static case. This yields a higher delay for the same circuit configuration, the only difference being the neighboring signal activity. Dynamic delay is problematic since it becomes difficult to perform static timing analysis. Neighboring wires must be examined to determine the likelihood of simultaneous switching, which greatly increases the complexity of the timing analysis problem.

**Power:** The insertion of inverters or noninverting buffers along a long global wire to reduce delay is called repeater insertion. With a proper insertion approach, the dependency of delay on wirelength can be reduced from quadratic to linear. In modern designs, global wires are usually broken up into segments approximately 3–6 mm in length using large repeaters (30–80 times minimum size). Repeater insertion is beneficial not only for delay values but it also maintains good slew rates and increases noise immunity for global nets. A serious drawback of significant use of repeaters is that such large gates add to power consumption and also take up chip area. In application-specific integrated circuits (ASICS), package limitations exist—cost considerations make plastic packaging a necessity. Plastic packages pose strict restrictions on power dissipation, meaning that widespread repeater use in ASICS may not be entirely feasible for packaging reasons. Another implication of repeater use that has recently been brought up is the proliferation of vias required to periodically connect top-layer routing levels to silicon [7]. This may have a strong impact on the routing of a chip since vias effectively block underlying signal routing tracks, reducing the available wire area.

**Process Variation:** The ability to pattern millions of submicrometer-sized features on a die the size of a thumbnail is a remarkable achievement. However, even with the tremendous lithography capabilities of modern IC manufacturers, it is impossible to produce completely identical devices over an entire 200- or 300-mm wafer, or even a much smaller die. Differences between supposedly identical features in the same process are known as process variation. As lithography tools are pushed to their limitations, the likelihood of process variation increases. With enhanced process variation, the actual performance of a fabricated circuit becomes more unpredictable. In order to maintain decent yields, IC manufacturers need to consider process variation at the design phase so that even worst case fabrication conditions will result in a working part. The general approach to modeling process variation is to design using worst case (slow), nominal (typical), and best-case (fast) SPICE transistor models [8]. These model files are generated from extensive measurements of device test structures.

An inherent difficulty in designing using worst case conditions is accurately defining “worst case.” Typically a worst case device model file may be generated by setting a number of relevant parameters (e.g., oxide thickness, channel length) to their $3\sigma$ values. This may prove too pessimistic, however, since many device parameters are correlated such that this generation methodology yields device models with unrealistically poor performance. The danger in overestimating process variation is that potential performance is being sacrificed for no reason other than poor models. Underestimating the amount of process variability, however, reduces the parametric yield, which drives up operating costs.

One way to more accurately assign worst case conditions is to use Monte Carlo simulations. Monte Carlo approaches begin with a set of predefined probability distribution functions (PDFs), typically normal distributions that are randomly sampled in each simulation or trial. Many simulations are performed and the results are taken as an average over a large number of observations/trials. In this manner, Monte Carlo simulations closely approximate a real-world system by maintaining the original PDFs. In general, Monte Carlo techniques are accurate but time consuming.

The majority of effort in modeling process variation is expended in determining realistic worst case transistor SPICE model files. With the rise in interconnect delay in scaled processes, a larger amount of work needs to be put into determining the extent of back-end process variation. This work will describe one Monte Carlo based approach to linking interconnect process variation to circuit performance.

**D. Interconnect Characterization and Modeling**

From the above discussion, we can see that interconnect analysis is important in realizing a successful circuit design project. Models of interconnect performance range in complexity from simple first-order linear models to complex moment-matching techniques. In all cases, such models need to be confirmed experimentally as well as...
through simulation. Actual silicon measurement has several key advantages over simulation; exact geometric information required for direct TCAD tool calibration is often unavailable. Also, process variation is inherently taken into account in an experimental setting. Indeed, examining the magnitude and sources of process variation is perhaps the most important application of measurement data. Another important application of interconnect characterization is to validate new models and check assumptions that are made in such models.

The models described in this work are analytical in nature. The availability of closed-form solutions is important for designers since these sorts of models shed insight on key design tradeoffs and parameter dependencies. In addition, with the rise in system complexity (multimillion to 100-million gate designs) model efficiency is vital and analytical models can be used in inner optimization loops of the design process due to their great speed. In fact, some of the best applications of these models are as screening tools that are applied to very large designs. After screening, a much smaller subset of items (they could be nets or gates) can be more rigorously analyzed with higher complexity models.

II. CHARACTERIZATION

In this section, we present a few novel approaches to measuring interconnect parasitics and performance. This section is not comprehensive and there exist a number of established measurement techniques that are not covered, including time-domain reflectometry and frequency-domain 

A. Charge-Based Capacitance Measurement (CBCM)

In order to give circuit designers an accurate assessment of speed and noise issues, parasitic capacitances due to interconnect must be well described. Currently, this is most often done using extensive computer simulations. A new measurement-based technique, charge-based capacitance measurement (CBCM) [9], has been developed to characterize interconnect capacitances. This simple, compact, and sensitive test structure can be used to measure any interconnect capacitance structure.

There is a pressing need for a simple and elegant measurement technique for on-chip interconnect capacitances. Conventional LCR meter based methods [10] are insufficient to measure small capacitances. They require large area parallel-plate capacitors, which eliminates their use in test chips and/or scribe lines. Prior approaches to compact characterization methods have not been shown to be effective in measuring very small capacitances (i.e., 10 fF and below) and such works fail to identify and evaluate major sources of error in the methodologies [11], [12]. S-parameter based measurements are very accurate for long transmission line structures and can be used to measure frequency-dependent RLC line parameters. Again, however, this technique does not lend itself to scribe line usage or the measurement of small capacitances.

Test Structure Description: The CBCM test structure is shown in its entirety in Fig. 3. It consists of an NMOS and PMOS transistor connected as in a typical CMOS inverter; however, the gate inputs to each device are different. To achieve the highest level of resolution, a second test structure should be placed next to the basic structure. This second structure is identical to the first in every way, except that it does not include the capacitance structure that is to be characterized. In the case of Fig. 3, the right inverter has an interconnect load consisting of a metal 1 line and a metal 2 to metal 1 crossover capacitance. The left inverter does not contain the crossover component.

The input signals denoted \( V_1 \) and \( V_2 \) in Fig. 3 are shown in Fig. 4. These waveforms are nonoverlapping so that only one of the two transistors is conducting at any given time and can be generated either on-chip or off-chip. In an actual CMOS inverter, there is a component of short-circuit current that
flows when both devices are on simultaneously. This component is typically about 10% of the total current during a switching transient. By using separate input signals, CBCM eliminates this source of error and ensures that all current being drawn from the supply voltage is being used to charge the load capacitance.

The operation of CBCM is described as follows. When the PMOS transistor turns on, the NMOS transistor is off and the capacitances in the circuit (both interconnect and device) are discharged completely. Once the PMOS is turned on, it begins to charge these capacitances to \( V_{dd} \). Given enough time, the PMOS will have completely charged all capacitances and its current will return to zero. By measuring the amount of current drawn by the PMOS device from the supply voltage through an ammeter placed at \( V_{dd} \), the total capacitance of the circuit can be determined. The second test structure does not include the target capacitance; hence, its overall capacitance is lower. Since the only difference between the two circuits is the target capacitance, the difference in measured current is directly proportional to this target capacitance. Similarly, current could be measured through the source of the NMOS devices (assuming they are separated) to achieve the same results. It is worthwhile to point out here that the shape of the PMOS device current waveform is not significant in this measurement technique. Only the total current delivered is of interest and its value can be determined with any dc ammeter.

The qualitative explanation supplied above can be summarized in just a few simple equations to provide the quantitative basis of CBCM. The crux of CBCM is the equation of a linear capacitor

\[
Q = CV. \tag{2}
\]

For a known voltage, we need only to be able to measure charge \( Q \) to determine any capacitance of interest. By focusing on the average current drawn from a power supply over a given time interval (the period of the signals \( V_1 \) and \( V_2 \)), the amount of charge is directly measured. Viewing charge as current multiplied by time yields another expression

\[
I = CV/t \tag{3}
\]

where \( t \) is the time interval mentioned previously that can be controlled and monitored using the on-chip or off-chip signal generator in Fig. 3. Now, applying these concepts directly to CBCM, we see that the measured currents \( I \) and \( I' \) in Fig. 3 can be subtracted and the new term, \( I_{\text{net}} \), can be inserted into (3) to give

\[
I_{\text{net}} = I - I' \tag{4}
\]

\[
I_{\text{net}} = CV_{\text{dd}}f/t. \tag{5}
\]

In (5), \( C \) represents the target capacitance to be measured. Rewriting (5) in terms of frequency, the resulting equation provides a clean and simple way to describe CBCM analytically

\[
I_{\text{net}} = CV_{\text{dd}}f \tag{6}
\]

where \( f \) is defined in Fig. 4. Now we can see with the help of Fig. 3 that by controlling the supply voltage and frequency, we can directly obtain the capacitance under test by subtracting the average currents from both inverters.

In our results, the input signals \( V_1 \) and \( V_2 \) are generated on-chip using a voltage-controlled oscillator (VCO), enabling the frequency of operation to be varied by applying different dc biases to the VCO. The first structure characterized was that of a single metal 2 to metal 1 overlap with an area of only 2.25 \( \mu \text{m}^2 \). This geometry was chosen to demonstrate the high resolution of CBCM, as well as to refine a robust extraction methodology.

The extraction algorithm used with CBCM ensures an accurate result by measuring the same capacitance repeatedly while varying one of the two independent variables from (6). For instance, at a given supply voltage of 5 V, the operation frequency can be varied and a measurement can be made at each frequency. By plotting the net current from (4) versus frequency, a straight line is obtained whose slope is equal to the product of the target capacitance and the supply voltage. This method can similarly be applied by varying the voltage at a fixed frequency. The results of these measurements for a single metal 2 to metal 1 overlap (area of 2.25 \( \mu \text{m}^2 \)) are seen in Fig. 5. The best fit lines closely match the data points, indicating very little error and a high degree of repeatability in the measurements.

The average capacitance value found from the set of measurements is 0.4431 fF with a standard deviation that is only 1.2% of the mean. RMS error is less than 0.5% in this case as well. The source of this variation is simply measurement equipment and rounding error. However, a closer look at all components of error in CBCM sheds insight on the effective design of test patterns.

There are two clear sources of potential error in CBCM. First, when measuring very small capacitances, the resolution of the equipment being used could result in erroneous measurements. Second, mismatch between the original test structure and the second structure placed nearby to act as a reference will result in measurement error. Concentrating on the former, we find that with modern high-resolution amme-
ters, equipment error can be safely neglected when operating frequencies are in the megahertz range. We have found that the dominant source of error in CBCM can be attributed to mismatch between inverters.

Ideally, with two inverters placed near each other, there would be no process variation and the devices and loads could be assumed to be identical. However, variation does exist and it is the most significant component of error in CBCM. Variation can occur in any device parameter, including threshold voltage, gate oxide thickness, and other important parameters. We limit this discussion to transistor width variations that result in parasitic device capacitance discrepancies. Due to transistor width variations, the device capacitances themselves will be slightly mismatched. Thus, the capacitances that each inverter charges will be slightly different and as a result, \( J_{\text{neq}} \) from (3) will be partially in error.

To estimate the resolution limit as a result of device mismatch, we need to estimate the amount of device mismatch and its impact on (6). Since transistor capacitances, including \( C_{\text{overlap}} \) and \( C_{\text{junction}} \), are directly proportional to width, they are also directly proportional to width variations. For a pessimistic gate width variation of 4% between close-proximity devices, the resolution limit is set at 0.04 + \( C_{\text{device}} \). From this back of the envelope calculation, it is seen that the accuracy of CBCM is both process-dependent and layout-dependent. First, a better and more controllable process will result in a lowering of the 0.04 factor, possibly to 0.02 or lower. Second, by using smaller or narrower devices in CBCM structures, the resolution limitation due to mismatch can be minimized through the \( C_{\text{device}} \) term. A typical CBCM inverter may have \( W/L \) ratios of 10 and 5 for the PMOS and NMOS devices respectively.3 These figures translate into a junction capacitance term (the dominant component of device capacitance) of approximately 2 fF. Thus, for a well-controlled process the resolution limit of CBCM is under 0.1 fF.

Additional Results: By varying the width of an isolated metal line over substrate with a constant length, a linear capacitance versus linewidth plot results, from that area and fringing components of the capacitance can be found. Fig. 6 shows metal 2 capacitance to substrate as a function of drawn width for both measurement and simulation (RAPHAEL is used [13]). The intercepts of the two lines are essentially identical, while the slopes are different. The slope in this figure corresponds to the area component of the capacitance to substrate. CBCM yields 19.6 aF/\( \mu \)m\(^2\) for \( C_{\text{area}} \), while simulation gives 15.5 aF/\( \mu \)m\(^2\). Data on 32 fabrication lots for this process is provided by the manufacturer, giving an average \( C_{\text{area}} \) of 20.4 aF/\( \mu \)m\(^2\), with values ranging from 11 to 27. The y-intercept of Fig. 6 corresponds to a linewidth of zero. The extrapolated capacitance at this point is equal to the total fringing capacitance of the line. RAPHAEL results match CBCM \( C_{\text{fringe}} \) values very well.

Discrepancies between RAPHAEL and CBCM may result in \( C_{\text{area}} \) due to substrate effects that are not taken into account in the simulator. Also, the interconnect structures used in these measurements were fairly long (\( L = 135 \mu \)m) compared to their width. A long thin metal line will have a much larger fringing component of capacitance than area component. This fact makes the measurements particularly sensitive to \( C_{\text{area}} \). In the future, structures of this type should be designed with roughly similar areas and perimeters to minimize potential error [10].

In this particular test chip, our interwire structures were designed to measure the additional capacitance brought on by the presence of a neighboring wire.4 Fig. 7 presents measurement and simulation data for four different spacings.

3 Minimum width devices exhibit a larger degree of process variation than slightly wider devices; hence, a safety factor of 2 to 3 over \( W_{\text{min}} \) should ideally be used when designing CBCM test structures.

4 Note this added capacitance does not correspond to the coupling capacitance as defined elsewhere in this paper.
An assumption made in many analytical interconnect models to provide simplicity is that an array of lines behaves the same as a continuous plate when dealing with interlayer capacitances [15]. We tested this assumption by placing metal 1 lines increasingly closer together underneath a metal 2 plate. We then measured the capacitance on the metal 1 lines increasingly closer together underneath a metal 2 plate. Each overlap was 1.5 \( \mu \text{m} \). We found a saturating effect where capacitance was only as high as a continuous plate when dealing with interlayer capacitances. Metal 2 length is held constant at 135 \( \mu \text{m} \), while spacing between metal 1 lines varied between 1.5, 3, and 4.5 \( \mu \text{m} \). We tested this assumption by placing a small set of CBCM structures, a simple analytical fit could be made for \( C_{\text{interv}} \). Implementing this expression in a layout extraction program, very accurate capacitance values for long parallel lines could be calculated. It should be recalled that the \( 1/d^2 \) relationship was found for additional capacitance and not for wire-to-wire capacitance, which is typically modeled with a \( 1/d \) relationship. CBCM can also be used to accurately measure coupling capacitance, as defined elsewhere in this work. One approach to doing this is described in [14], which uses a three-step superposition method and a multibranch circuit structure to find the actual coupling capacitances.

An assumption made in many analytical interconnect models to provide simplicity is that an array of lines behaves the same as a continuous plate when dealing with interlayer capacitances [15]. We tested this assumption by placing metal 1 lines increasingly closer together underneath a metal 2 plate. We then measured the capacitance on the metal 2 plate. Each overlap was 1.5 \( \mu \text{m} \times 2 \ \mu \text{m} \), and spacing between metal 1 lines varied between 1.5, 3, and 4.5 \( \mu \text{m} \). We found a saturating effect where capacitance was only increased by a few percent when decreasing spacing from 3 to 1.5 \( \mu \text{m} \). Fig. 8 shows our data compared to RAPHAEL simulations. Simulations show a similar saturating effect, although it takes place more gradually, or equivalently, at smaller spacings. In this analysis, an ILD thickness corresponding to dense metal 1 was used. This results in the slight undershoot by RAPHAEL at 18 and 24 lines. By varying the ILD thickness within given process specifications (typically 20% or more of variation), a range of capacitances can be determined and can be seen from the error bars in Fig. 8 to result in better agreement with CBCM. CBCM implicitly takes any ILD variation into account since it is based on measurement data. As a result of using CBCM, we estimate that a metal density of 33% or greater (spacing \( = 2 \times \text{width} \)) can be approximated as a plate with negligible loss of accuracy.

B. Interconnect Delay and Noise Characterization

We now turn to the prediction of interconnect performance. In contrast to the previous section on capacitance measurement, we propose direct measurement of interconnect performance using in-situ techniques that will allow newly developed delay and noise models to be verified and validated.

Errors incurred by interconnect models must be carefully monitored and controlled since the impact of interconnect on system performance is rising. Due to high on-chip integration levels, very efficient models are required that tend to make a number of simplifying assumptions. These assumptions need to be investigated by comparison with on-chip measurement results. The introduction of new interconnect materials such as copper and low-k dielectrics heighten the need for interconnect performance verification; their true impact cannot be determined without characterization. Also, the increasing number of interconnect-related effects (such as dynamic delay and inductance) further emphasize the necessity of new interconnect validation techniques.

Indirect in-situ techniques remove the need for external probing, which fundamentally changes the system that is being measured. Previous work in the area of interconnect performance characterization has failed to reproduce the actual on-chip system that occurs in an actual product. For example, [16] is an extensive report on measuring characteristics of long on-chip interconnections, but no active drivers are used. Instead, time domain reflectometry (TDR) is used to stimulate each transmission line resulting in a very different situation than a MOSFET driving a global wire. Likewise, \( S \)-parameter based frequency-domain measurements are limited in that they mainly characterize line parameters (RLC) for long transmission lines and do not give insight to the interaction between devices and interconnections (i.e., a wire on a real chip is driven by a gate, not a network analyzer) [17]. In general, frequency-domain approaches are ideal for measurement of high-frequency phenomena such as skin effect but unsuitable for recreating the actual environment seen in real designs.

In [18], some delay and noise measurement results are presented without a clear explanation of the measurement methodology. In general, the measurement technique applied in [18] is a brute-force approach where the end of the victim line connects to a pad that is directly probed for its voltage level. By including the additional pad and probe capacitance in simulations, a comparison can be made. In reality, the large pad capacitance would not be present so the technique interferes with the realistic case.

The delay measurement methodology now presented is based on [19]. That work detailed an accurate in-situ measurement technique, which is based on the use of analog circuits to extract circuit performance metrics. Using compara-
tors, the authors of [19] measured the delay and noise waveforms of a number of circuit configurations in a 0.25-μm CMOS technology. Their work has been extended in the following ways [20].

- We propose a novel and more accurate method of determining peak crosstalk noise.
- We investigate the impact of dynamic delay on modern processes.
- Our test chip implementation allows for variable aggressor/victim signal arrival times.
- The aggressor driver strength can be changed to examine the impact on noise and delay characteristics.

Two test chips were fabricated to explore the capabilities of the new measurement methodologies. The designs emphasize the importance of interconnect coupling effects including both crosstalk noise and dynamic delay. This section focuses on the measurement techniques developed and presents simulation and preliminary measurement results to demonstrate their advantages over existing methods.

**Delay Characterization:** Fig. 9 illustrates the block diagram of an interconnect measurement site. Each site contains a number of important circuit elements:

- three wide range comparators (WRC) [21];
- three variable impedance noninverting drivers (two for aggressors, one for victim);
- one accurate peak detector (APD) circuit for peak noise measurement;
- three interconnections driven by variable impedance drivers and terminated by MOS gate capacitances;
- two dummy interconnections to mimic a bus structure, tied to ground through active devices on the near-end and terminated by MOS gate capacitances.

Tri-state buffers are used to determine whether the noise or delay measurement mode is selected. Two tri-state buffers are present in each site and they are connected to the output of the APD and WRC at the end of the victim line.

Four sites are combined into a group. A group has a single set of inputs and outputs and multiplexers and demultiplexers are used to determine that of the four sites is active at any time. Two test chips were fabricated: one in a 0.35-μm four-level metal CMOS process and the other in a 0.25-μm five-level metal CMOS process. These chips have very similar designs and specifications, with approximate dimensions of 2.2 x 7.2 mm. The primary test equipment includes a digital oscilloscope, pulse generator, and a multichannel voltage source; a socketed DUT board is used to interface the packaged chip with the testing equipment.

Compared to [19], we concentrate on realistic wirelengths and configurations. For instance, a 10-mm minimum pitch metal 1 line is not an interesting test case since it would never be used in an actual design. We limited our line lengths to 6 mm since lines longer than this will be buffered in modern designs and also focused on using global layers for longer lines. Each test site emulates a bus structure with parallel signal lines, as shown in Fig. 10, and loading lines above and below that act as ground planes.

In the delay mode, the delay of the victim line can be calculated by measuring the delay from out0 to out2 in Fig. 9. Alternatively, the delay of the RC transmission line alone can be found by comparing out1 and out2. By changing \( V_{\text{ref}} \), we can sample the victim waveform at different points during its transition. Figs. 10 and 11 show the procedure to find the line delay and the complete waveform, respectively. In Fig. 11, only three data points are shown for clarity; any amount can be used for greater resolution. The delay of the comparator must be determined and removed from the total measured delay. For this purpose, a calibration structure is included that
consists of just a comparator and the subsequent buffering system (the same as in an actual site). The reference voltage and input slope can be adjusted to determine their impact on the comparator delay.

A full explanation of the measurement scheme is available in [19]. In our implementation, we focused on the impact of noise on delay. The aggressor input is separated from the victim input and we allow the victim and aggressors to have independently varying driver strengths. The variable impedance drivers have seven possible on-resistances that are specified by a three-bit control sequence. By allowing the aggressors to have variable impedances (rather than a fixed 60 Ω as in [19]), we can investigate the role of aggressor drive strength on noise characteristics (as will be discussed later in Section III-B). We can also examine the timing alignments (between victim and aggressors) that result in worst case noise since we can input the victim and aggressor signals independently. The width of such timing windows is important in that circuit designers often try to design their circuits so there are no possible timing hazards due to noise. Doing so requires that no neighboring wires switch within a certain timing window. If this window is a large portion of the system cycle time, the task becomes very difficult. Fig. 12 defines aggressor and victim switching phases.

Noise Characterization: The noise characterization technique we employed in our test chips is different from that of [19]. In that work, the authors use the same comparator setup as for the delay measurement described above. By changing the reference voltage, a noise waveform can be reconstructed similarly to the delay waveform. The largest \( V_{\text{ref}} \) value for which the comparator output switches is the peak noise value. In [19], it is stated that this measurement technique yields 20%–30% underestimation of the peak noise for sharp noise peaks due to comparator bandwidth limitations. Through simulations, we verified this result and found that the noise peaks do not need to be very sharp for a significant error to result. Furthermore, more advanced processes yield sharper noise peaks so the conventional noise measurement technique of [19] will not necessarily improve with the higher bandwidth provided in future technologies. The measurement results of [19] concentrate on slow, broad noise waveforms with widths of several nanoseconds that are fairly unrealistic.

To alleviate these concerns, we developed a novel accurate peak detector (APD) circuit, composed of a cascaded low-gain high-speed differential amplifier and pseudodynamic latch. The APD is shown in Fig. 13(a) and has an input voltage \( (t_{\text{in}}) \) and a user-defined reference voltage \( (V_{\text{ref}}) \). The APD compares the input signal to \( V_{\text{ref}} \) and converts the result into an easily measured single-transition output signal. By concentrating on peak noise, which is of the most interest to designers concerned with signal integrity, the APD gives significantly better accuracy for \( V_{\text{peak}} \) than the more general approach of the WRC. From simulation, we have found that the APD captures a noise pulse when the \( V_{\text{ref}} \) crossing time is about 100 ps. This corresponds to an error in peak voltage measurement of about 10% with a 1-ns noise pulse base width. For the same conditions, the WRC gives about 35% error and requires a crossing time of nearly 400 ps to do so. These estimates are pessimistic since actual
noise waveforms are flatter at the peak than the triangular waveform we used.

The pseudodynamic latch and level-shifter are used to propagate a single-transition signal to the output pad. The timing diagram for the noise measurement procedure is shown in Fig. 13(b). When the input signal drops below $V_{\text{ref}}$ (we concentrate on the case where the victim is held nominally at $V_	ext{dd}$ and aggressors switch low), the differential pair steers the current such that the input side of the pair sees a higher voltage than the reference side. In the quiescent state the input voltage is $V_{\text{dd}}$, which is higher than $V_{\text{ref}}$. Since the noise pulse is a transient phenomenon, the input signal will only be smaller than $V_{\text{ref}}$ for a short amount of time after that the signals return to their quiescent state.

In order to maintain the flipped state of the output, we latch in any change of state occurring at the output of the differential amplifier. The level shifter acts to buffer up the output of the differential amplifier to rail-to-rail voltages. The pseudodynamic latch has a logic high tied to its input. After buffering, $V_{\text{out}}^+$ and $V_{\text{out}}^-$ from the level shifter are taken as the clock and clock_bar inputs of the latch. In their quiescent state, the output is isolated from the input of the latch. This output value is reset to zero to begin a measurement. If the reference voltage is crossed, the clock and clock_bar inputs of the latch switch to make the latch transparent, passing the $V_{\text{dd}}$ at its input to the output. Thus, when taking a noise measurement, we are looking for a low-to-high transition at the output pad to signify that $V_{\text{ref}}$ was crossed. When the reference voltage is not exceeded [the dashed line in Fig. 13(b)], the output signal is static.

This noise measurement technique can be extended to measure inductive coupling effects by activating dummy lines as shown in Fig. 9. A significant difference was not observed for the particular processes and driver sizes used in these designs according to simulation. However, this is an area for future investigation.

**Results and Discussion:** In Fig. 14, we have reconstructed several waveforms (out2 from Fig. 9) and plotted them along with SPICE simulations of the same structure. In this case, we are considering a 6-mm metal 4 line with quiet aggressor lines. The simulated 10%–90% rise times are within 10% of the measured cases (simulation uses a 10-lump interconnect model to represent the distributed line). The rise time of 1.25 ns for the smallest victim drive strength (Victim = 1, $R_{\text{ca}} = 703$ Ω) is 3.5 times longer than the case with the strongest victim driver (Victim = 7, $R_{\text{ca}} = 96$ Ω). For the strongest driver case, the dominant component of the rise time is the wire RC delay (distributed 10%–90% delay = 0.9RC), which accounts for approximately 65% of the total rise time.

Measured results on the impact of timing difference between victim and aggressor are shown in Fig. 15. This diagram is called a delay change curve (DCC). Changing the aggressor arrival time $T_a$ relative to the victim arrival time $T_v$ alters the victim line delay due to changes in the effective coupling capacitance between the lines. $T_\text{dly}$ in the figure roughly corresponds to the base width of the noise waveform. The peak delay change made by out-of-phase inputs for the different victim sizes shown are about 90 and 220 ps. These numbers can be used as the required timing margin of the receiver. We also see that the delay change rapidly approaches zero if there is a timing difference between inputs. This implies that if layout tools can consider the switching timing of the neighboring parallel lines and avoid routing simultaneously switched lines next to each other, the required timing margin becomes much smaller. Since the worst case timing variation may be too pessimistic, the use of DCCs would be indispensable for these tools.

Fig. 16 looks at the timing window width and magnitude of dynamic delay for bus structures where the victim and aggressor have the same driver strength. This scenario is very common in buses that may run the entire chip-side length (periodically buffered for optimal delay). As the drivers get stronger, the timing window is smaller since the victim can hold its interconnect more tightly to ground.

Fig. 17 presents noise measurement data from the APD technique. In this instance, the interconnect configuration is a minimum pitch 6-mm metal 4 line. The aggressor is fixed at its maximum strength ($R_{\text{ca}} = 96$ Ω) for worst case noise. Results are presented for all seven possible victim strengths. As expected, a stronger victim driver creates a less resistive
On-chip crosstalk is a major concern in ULSI circuits due to scaling linewidths, increasing aspect ratios, and larger die sizes [2]. Also, due to reduced noise margins and larger ground bounce, noise issues become even more important. However, with several million nets in a modern design, detailed simulation of crosstalk noise on each net is highly inefficient. A rapid and accurate crosstalk estimation technique is needed to quickly screen nets that violate noise margins.

The most basic crosstalk model is the charge-sharing model, presented in many circuit textbooks and review papers [23], [24]. This simple model considers only the ratio of coupling capacitance to total lumped capacitance. This model has been found to introduce extremely large errors (e.g., >500%) in another work [25]. Most high-level electronic-design-automation (EDA) tools that account for crosstalk noise also only consider the capacitance of the signal lines [26]. This simplification can produce significant errors that may lead to unpredicted circuit failures under certain conditions. A more accurate model needs to account for the various resistive components of the circuit. In [27], a closed-form model based on RC transmission line analysis is presented, but driver modeling is not discussed and the analysis is limited to step inputs. Two other models [25], [28] approximate the driver with a resistor and a voltage source, but signal line resistance is neglected. In deep-submicrometer designs, line resistance is appreciable and cannot be ignored.

In this section, we present a general closed-form crosstalk model that takes into account driver strength and line resistance. This model is simple, accurate, and provides an excellent basis for a crosstalk screening tool. We demonstrate its accuracy in deep-submicrometer logic gates, concentrating on inverters and NAND gates, by comparing it with distributed SPICE simulations. The model can be used in conjunction with timing macromodels or other timing-level tools, as driver rise time is an important parameter in crosstalk calculation.

Model Derivation: Crosstalk is a complex form of electromagnetic coupling between two or more conducting lines. In order to obtain a tractable analytical expression for peak crosstalk noise, several assumptions will be made. First, the aggressor gate is modeled as a ramped voltage source with rise/fall time, $T_r$. This rise time can be obtained by the use of a gate-level timing simulator or through the use of analytical expressions such as those presented in [29]. A timing macro-model is presented later for this purpose. A second assumption is that all interconnect and load capacitances (including fan-out gates and drain junction capacitances) are modeled as a lumped capacitance to ground, excluding the coupling capacitance. Finally, the victim line driver is modeled as an effective resistance. This resistance is equal to the inverse of victim network. Further increases in victim driver size will not yield substantial drops in peak noise due to this high ratio.

III. ANALYTICAL MODELING

A. Crosstalk Modeling

From Fig. 17 we also see a saturating trend where larger victim drivers do not give substantial noise reductions past a certain point. This trend is a result of the line resistance dominating the total resistance to ground (from the end of the victim line). At a victim strength of 7 in the figure, the line resistance accounts for 86% of the total resistance in the path to the supply rail for the end of the victim line, which reduces noise. Simulation results and results from the crosstalk model presented in the next section are compared to the experimental data with good agreement found (within 10%).

The measurement results of Fig. 17 are a significant improvement over previously published results; they demonstrate that crosstalk modeling is necessary. Even in the case of a strong victim, when the noise peak is sharpest, matching between simulation, measurement, and model is very good. The figure shows that the noise model of [22] consistently underestimates the noise peak as found by measurement and simulation. This is due in part to the fact that both the actual test chip and simulation netlist consider inductance (both mutual and series), whereas the model includes RC effects only. By neglecting mutual inductance in the simulations, the noise peak is decreased slightly so that the maximum model error versus simulation is reduced to <6% (from 7.2%).

From Fig. 17 we also see a saturating trend where larger victim drivers do not give substantial noise reductions past a certain point. This trend is a result of the line resistance dominating the total resistance to ground (from the end of the victim line). At a victim strength of 7 in the figure, the line resistance accounts for 86% of the total resistance in the victim network. Further increases in victim driver size will not yield substantial drops in peak noise due to this high ratio.
the slope of a device’s $I_d-V_d$ curve at the origin and needs to be found just once for a given technology as it scales linearly with device width. Since the victim device is operating only in the linear region, this is a valid assumption.

The resulting equivalent circuit is shown in Fig. 18. In the figure $R_a$, describes the aggressor line resistance while $R_v$ is equal to the sum of victim driver resistance and line resistance. The aggressor gate is modeled as a voltage source with ramp rate $V_{DDL}/T_r$, and crosstalk is denoted as $V_x$. $C_a$ and $C_v$ are the sum of all ground capacitances for the aggressor and victim, respectively. These totals include drain junction capacitance, fan-out capacitance, and interconnect capacitance to upper and lower ground planes.

From circuit analysis principles, the victim line voltage $V_x$ is found to be

$$V_x = \frac{R_a C_v V_{DDL}}{\tau_0 T_r} \left( \tau_0 + \tau_1 e^{-t/\tau_1} - \tau_2 e^{-t/\tau_2} \right)$$

$$0 \leq t \leq T_r$$

$$V_x = \frac{R_a C_v V_{DDL}}{\tau_0 T_r} \left[ \frac{1}{\tau_1} \left( e^{-(t/\tau_1)} - e^{-\left(\left(t-T_r\right)/\tau_1\right)} \right) - \frac{1}{\tau_2} \left( e^{-(t/\tau_2)} - e^{-\left(\left(t-T_r\right)/\tau_2\right)} \right) \right]$$

$$T_r \leq t.$$  

In these equations, $T_r$ is the rise/fall time at the output of the aggressor driver (equivalent to the input of the interconnect network) and $\tau_0$, $\tau_1$, and $\tau_2$ represent different time constants of the equivalent circuit. The time constants are defined as

$$\tau_0 = \left[ \frac{R_a C_v + R_v (C_v + C_a)}{2} - 4 R_a R_v \left( C_v C_a + C_a C_v + C_v C_a \right) \right]^{1/2}$$

$$\tau_1 = \left[ \frac{2 R_a R_v (C_v + C_a + C_v C_a)}{R_a (C_a + C_v) + R_v (C_v + C_a + \tau_0)} \right]$$

$$\tau_2 = \left[ \frac{2 R_a R_v (C_v + C_a + C_v C_a)}{R_a (C_a + C_v) + R_v (C_v + C_a - \tau_0)} \right].$$

The peak value of crosstalk noise, $V_{\text{max}}$, can be found by differentiating (8) as this peak occurs at $t > T_r$

$$V_{\text{max}} = \frac{R_a C_v V_{DDL}}{\tau_0 T_r} \left( \tau_1 Y_1 \left( \frac{Y_1}{Y_2} \right)^{\tau_1/\left(\tau_1-\tau_2\right)} - \tau_2 Y_2 \left( \frac{Y_1}{Y_2} \right)^{\tau_2/\left(\tau_1-\tau_2\right)} \right)$$

where

$$Y_1 = e^{-T_r/\tau_1} - 1$$

For slow rise times ($T_r \gg \tau_2$), (12) can be seen to approach a limiting value of $R_a C_v V_{DDL}/T_r$. The model presented in [27] is a special case of (12) when $R_a = R_v$, $C_a = C_v$, and the gate output is a step. In this case, (12) reduces to

$$V_{\text{max}} = \frac{V_{DDL}}{2} \frac{C_v}{C_a + C_v}.$$  

Our new model allows for different values of $C_a$ and $C_v$, which is almost always the case in actual designs. Also, there are many cases where an aggressor line runs parallel to a victim for less than the full length of either line. The general nature of this model can handle such cases. Thus, the model can be seen to have wider applicability than those mentioned above.

**Lumped versus Distributed:** Deep-submicrometer interconnect is difficult to model by a single-lump RC model. However, to derive a simple analytical model, a lumped topology is often assumed. For our model to accurately represent distributed interconnect the capacitances are scaled to account for the distributed properties of an RC line. Fig. 19 shows the representative cross-section of the three-line system utilized throughout this work. An upper ground plane is not considered here. Based on the Elmore delay model, the ground capacitances $C_a$ and $C_v$ are scaled by a factor of 0.5 [for an $N$-step ladder, the factor $(N+1)/2N$ should be used] to approximate a fully distributed scenario. The capacitances are reduced since the distributed capacitors only see the upstream resistance, rather than the total line resistance [30]. Coupling capacitance is scaled by a different, larger factor $\alpha$ that is technology-independent and given by

$$\alpha = (1 - \chi) e^{-T_r/\tau_1} + \chi$$

$$\chi = 0.5 \times \left( 1 + \left( \frac{R_{\text{lev}}}{R_{\text{lev}} + R_{\text{int}}} \right) \right).$$

The parameter $\chi$ is an empirical factor accounting for the ratio of victim driver ($R_{\text{lev}}$) and line ($R_{\text{int}}$) resistances and is equal to 1 for shorter lines (device-dominated) and 0.5 for long lines (interconnect-dominated). As can be seen from the above equations, $\alpha$ is equivalent to $\chi$ for a sufficiently slow rise time, but is equal to 1 for a step input.
Worst case crosstalk occurs in the three-line case shown in Fig. 19. Due to the above model’s generality, it can be extended to accurately describe the case involving two aggressor lines. As the two aggressor lines can be considered to be in parallel, the ground capacitances $C_{G1}$ and $C_{G2}$ and aggressor line resistances $R_{a1}$ and $R_{a2}$ can be substituted with $(C_{G1}+C_{G2})/(R_{a1}+R_{a2})$. Likewise, $C_L$ will consist of two components as well, $C_{L1}$ and $C_{L2}$. Capacitance values such as $C_{G1}$, $C_{L1}$, and $C_{L2}$ should be re-extracted for the three-line case, as they will be somewhat different from the two-line scenario.

As mentioned earlier, this model can be used with electronic computer-aided design (ECAD) tools that provide gate output rise times to determine peak crosstalk. Fig. 20 demonstrates the accuracy of the model, including the capacitance scaling scheme, when the aggressor gate rise time is known. The maximum error is less than 4% and highlights the accuracy of the lumped-to-distributed scaling approach. In this and all subsequent cases, a 50-lump RC network is used in SPICE to simulate crosstalk using 0.25-μm device and interconnect parameters unless otherwise specified.

Transistor-Level Model: We now introduce a timing macromodel for use with (7)–(17) above. The rise times calculated in this section are defined by the 10%–90% convention. While there are a number of CMOS delay models in the literature [27], [29], [31], there are few simple expressions available to evaluate the rise time at the output of an interconnect-loaded gate. Since it is this value that is equivalent to the aggressor ramp rate shown in Fig. 18, an accurate model needs to be developed for use with our crosstalk model if timing-level ECAD tools are not available.

The rise time of a gate driving an interconnect can be broken into three distinct parts. First is the intrinsic rise time of the device with no load and a step input at the gate. The value of this delay is proportional to the effective resistance of the device multiplied by the capacitance at the output node

$$t_{r1} = \frac{0.5V_{dd}C_{out}}{I_{bat}}, \quad (18)$$

where device resistance is calculated as $0.5V_{dd}/I_{bat}$. The output capacitance consists primarily of junction and overlap device capacitances. This term is small, in the range of 5–10 ps for inverters, and is independent of device width as the geometry dependencies of $I_{bat}$ and $C_{out}$ cancel. For older technologies or large gates (e.g., four-input NAND), this term can become sizable but in most cases is overshadowed by the delay due to capacitive loading.

The second component of rise time is the input waveform dependency. Since the input to any gate is not an ideal step, there is less drive current supplied initially due to the fact that $V_{gs} < V_{dd}$. Simulations show this input waveform dependency to be linear, both in propagation delay [29], [31] and rise time. Specifically, Fig. 21 shows that the relationship between input and output transition time with no load is linear over a wide range of rise times and technologies. Thus, the following expression describes this dependency well:

$$t_{r2} = k_2t_{in}, \quad (19)$$

Here $k_2$ is a constant for a given technology and varies between 0.1 and 0.2, signifying that 10%–20% of the input transition time is reflected in the output delay or rise time. This term can be very significant, especially for slow input signals and small load capacitances. In the case of a 0.25-μm inverter analyzed in this work, $t_{r2}$ constitutes 31% of the total rise time at a line length of 1 mm when the input transition time is 200 ps. Inaccurate modeling of this term, or the assumption of a step input, will result in large errors in rise time estimation. Previous attempts [29], [31] have been made to relate $k_2$ to the ratio of $V_t$ to $V_{dd}$. From Fig. 21, we see no clear pattern for $k_2$ (the slope of each line) with regard to technology scaling. As $V_t/V_{dd}$ is increasing as supply voltages scale downward, we would expect to see a consistent trend in $k_2$ versus technology. A technology-specific empirical formulation will be more accurate although an estimation of $k_2$ equal to 0.15 seems to yield good results for most deep-submicrometer technologies.
Finally, capacitive loading comprises the largest portion of $T_r$. A modified analytical model based on [32] results in the expressions

$$t_{\tau 3} = 1.2\lambda C_L \left( \frac{V_I - 0.1V_{dd}}{I_{d\text{est}}} + \frac{(V_{dd} - V_I)}{2I_{d\text{est}}} \right) \cdot \ln \left( \frac{19V_{dd} - 20V_I}{V_{dd}} \right)$$

$$\lambda = 1 - \left( \frac{R_{a,\text{int}}}{R_{a,\text{int}} + \sqrt{3}R_{a,\text{dev}}} \right)^4$$

In these expressions, $\lambda$ accounts for the shielding of capacitance by line resistance, $C_L$ is the interconnect capacitive load ($C_o + C_v$ in Fig. 19), and the factor of 1.2 accounts for short-circuit current. $R_{a,\text{int}}$ and $R_{a,\text{dev}}$ refer to the aggressor line resistance and the aggressor device resistance, respectively. As the aggressor device moves throughout the linear, saturation, and cutoff modes of operation, the average device resistance is estimated as $(V_{dd}/I_{d\text{est}})$. Short-circuit current is overestimated for large line lengths but does not result in large errors in $V_{\text{max}}$. Equation (21) is an empirical determination of resistive shielding and can be further optimized for individual technologies.

By summing $t_{\tau 1}$, $t_{\tau 2}$, and $t_{\tau 3}$ a simple expression for $T_r$ is obtained. Results are shown in Fig. 22 for two different gate topologies and loading conditions. Overestimation at long line lengths is due to the fact that short-circuit current is nearly negligible when $C_L$ is very large. However, the sensitivity of $V_{\text{max}}$ to rise time is very high for short lines but drops dramatically for longer line lengths. For example, at a line length of 300 $\mu$m, a 15% increase in $T_r$ results in a 13% drop in $V_{\text{max}}$ for a 0.25-$\mu$m inverter. At a line length of 4 mm, a 15% change in $T_r$ yields less than a 0.5% change in $V_{\text{max}}$. For this reason, overestimation of $T_r$ for large loads is tolerable to obtain higher accuracy at shorter line lengths.

Fig. 23 demonstrates the fit of the crosstalk model compared to SPICE simulations for a three-input NAND inverter when using the above timing model to obtain $T_r$. Error is less than 10% for all line lengths up to 1 cm and in most cases is less than 5%. To demonstrate the wide applicability of this model, the victim driver size is varied by an order of magnitude in a three-input NAND gate ($W = 10–100$ $\mu$m) and the resulting crosstalk is plotted in Fig. 24. Finally, aggressor driver sizes are varied to obtain a large range of rise times. The timing model is used to calculate $T_r$ in each case and the crosstalk values are still in very good agreement with SPICE. These results are also shown in Fig. 24. The maximum error for $V_{\text{max}}$ in these cases is 12%, verifying that both the crosstalk model and the timing model are valid over a wide range of parameters.

B. Dynamic Delay

We now turn our attention to the effects of noise on timing that were first introduced in Section I-B. It can be seen that due to the increase in coupling capacitance as a percentage of overall net capacitance, the potential exists for variability in the delay of a circuit depending on the switching activity of neighboring nets that have significant coupling to the original circuit. This is a relatively new modeling problem that makes static timing analysis more difficult since the delay
Worst case dynamic delay occurs when two neighboring wires switch in the opposite direction as the victim simultaneously. The circuit is now a dynamic phenomenon that depends on the activity of nearby signals. There are two methods of modeling the additional delay brought on by noise, which are discussed next.

Fig. 25 demonstrates the problem; the middle line is called the victim and switches in the opposite direction of the two neighboring nets (out of phase). The neighboring nets are aggressors and are considered to switch simultaneously with the victim net for worst case delay. In reality, it has been shown in [33] that this situation does not correspond exactly to the worst case delay value. Instead, the aggressor input arrival times can be staggered such that they cause a non-monotonic response on the victim net during switching. The resultant delay is typically about 10% worse than the simultaneous switching scenario. By assuming all nets in the design run parallel to other nets for their entire length and that neighboring signals switch opposite simultaneously with the signal of interest, typical noise analysis is already very pessimistic. Finding the true worst case input arrival times adds unnecessary complexity to the first-order analysis below.

**Modeling Approaches:** The first method of modeling the additional delay brought on by noise is explained by examining the network in the context of Miller’s theorem [34]. Miller’s theorem states that when there are two nodes connected with an admittance $Y$, the network can be equivalently represented by isolating the two nodes from each other and placing new admittances between each node and the ground node. The Miller effect is demonstrated graphically in Fig. 26. The values of the new admittances, $Y_1$ and $Y_2$, are equal to $Y \times (1 - K)$ and $Y \times (1 - 1/K)$ where $K$ is the voltage gain from node 1 to node 2. If we assume that the nets switch in opposite directions at the same slew rate, the voltage gain is simply equal to $-1$ since no amplification occurs. Thus, the new values for admittances $Y_1$ and $Y_2$ are both $2 \times Y$. In the figure, we substitute the coupling capacitance $C_c$ for admittance $Y$. Therefore, we replace this floating capacitance by a capacitance to ground equal to $2 \times C_c$ for each line. This factor of 2 is referred to as a switch factor (SF). Examining the victim net of Fig. 26, we have added a total of $2 \times C_c \times 2$; one for each side where there is an aggressor net.

In the complementary situation, both nets switch in the same direction (in-phase). In this case, the gain between the nodes is equal to 1, which sets $Y_1$ and $Y_2 = 0$. By inserting the new coupling capacitance term into a delay expression, we see that the impact of noise on delay can be quite significant depending on the ratio of $C_c$ to the total load capacitance. This method of incorporating noise effects into delay is the traditional method and can be accurate for lines that do not have a significant amount of resistance and that experience similar transition times. Long lines with significant resistance (relative to the driver resistance) or substantially different rise times may need more accurate modeling approaches such as that described next.

A second method is based on the recent observation that the neighboring wires can be seen as an added load for the victim gate. As such, we should be able to calculate the additional charge required to charge these new loads [35]. By examining the voltage spike experienced on the victim line when it does not switch and the aggressors do switch (the crosstalk noise scenario), we can find an upper bound on the amount of charge needed to counteract the influence of the aggressors. Fig. 27 shows the crosstalk noise spike $V_\text{noise}$ that occurs when the victim line is quiet and both aggressors are switched simultaneously. The value of $V_\text{noise}$ can be calculated using the crosstalk noise model of Section III-A. Now, by superimposing this voltage waveform on the quiet voltage response of the victim, we can find an upper bound on the “noisy” victim delay. The 50% delay of the composite signal is equal to the delay of the original quiet delay to the voltage point $V_{\text{bd}}/2 - V_{\text{zr}}$. This point is shown in Fig. 27 by the vertical black line intersecting both the solid (original) and dashed (composite) waveforms. Delay can simply
be computed by using a modified version of Sakurai’s delay model [27]. The exact expression is

$$T_v = 0.1R_{ag}C_{w} + \ln\left(\frac{1}{1 - v}\right)$$

$$\cdot \left(R_{dev}(C_j + C_{in}) + R_{dev}C_{w} + R_{w}C_{m} + 0.4R_{ag}C_{w}\right).$$

(22)

In this equation, $v$ is the voltage point of interest normalized to the supply voltage. For instance, if $V_x$ is 10% of $V_{dd}$, then $v$ becomes equal to $0.5 + 0.1 = 0.6$. In the case of bootstrapped noise where the victim driver is being helped by aggressor switching (in phase), $v$ becomes smaller than 0.5, which yields a smaller calculated delay. When there is no consideration of noise, $v$ becomes 0.5 and (22) becomes equivalent to Sakurai’s original delay expression. The reason that this approach results in an upper bound is that the crosstalk noise voltage is calculated with the aggressors switching in the absence of noise. The impact of the victim line switching will serve to slow down the transition times of the aggressors, leading to a somewhat smaller value of $V_x$. However, since we are usually looking for conservative estimates of noise, this approach is sufficient in that respect. The above discussion holds for a linear system—in actuality, due to MOSFET resistance, the problem is nonlinear. Thus, we are using an approximation to determine the impact of noise on delay.

**Model Evaluation:** To evaluate the accuracy of the above two modeling approaches to dynamic delay, we performed a group of SPICE runs to quantify the quiescent, in-phase, and out-of-phase delays for a number of circuit configurations. The technology used in the simulations was a commercial 0.25-μm CMOS process using BSIM3v3 model files. The circuit configurations consisted of two-input NAND gates driving a distributed RC interconnect with a fan-out of a single NAND gate (the size of the fan-out varied for local and global cases). We first simulated metal 2 routing at lengths of 250, 500, 750, and 1000 μm at minimum pitch as well as a case using minimum width but twice the minimum spacing (minimum width and spacing in this technology were 0.4 μm). Interconnect resistance and capacitance parameters were extracted using a 2-D field solver. The victim device widths were fixed at either 5 or 10 μm, while the aggressor device widths were set to 5, 10, and 20 μm. Results are shown in Fig. 28 for both the switch factor-based and crosstalk-based modeling approaches. Sakurai’s delay expression was used for both models and the crosstalk noise model of Section III-A was used to calculate $V_x$. All delay times refer to the output pull-down case.

A key point in Fig. 28 is that the SF-based model does not consider the important impact of aggressor driver strength on dynamic delay. SPICE results and the crosstalk-based model show that larger aggressors significantly increase the magnitude of dynamic delay. The crosstalk-based model tracks the SPICE results well with a typical error in this figure of 5% to 8%. The figure also shows that the crosstalk-based model is not always an upper bound on dynamic delay. This could be due to a few possibilities including underestimation of $V_x$ by

the crosstalk noise model or error in the delay model itself. In most cases though, the crosstalk-based model does provide an upper bound on dynamic delay with accurate results.

Compiled results from all simulations show that, in general, the SF-based model shows fair accuracy for very limited cases while the crosstalk-based model tracks both the aggressor strength dependency as well as the line length dependency very well. For various interconnect configurations, typical errors are in the range of 5%–15% for crosstalk-based and 10%–50% for SF-based models.

The primary problem with the SF-based approach is that it does not account for aggressor drive strength in its formulation. However, it can be extended to do so. In the explanation of Miller’s theorem above, the gain $K$ was mentioned. In the simple discussion, the gain was set to be either $+1$ or $-1$ depending on signal activity. In reality, a better determination of $K$ should examine the signal slopes of the victim and aggressor and take the ratio of these slopes to be the gain (with a negative sign implying opposite switching directions). For example, if the aggressor driver switches twice as fast as the victim, it is expected to supply more noise. This could be modeled in the Miller approach with a larger gain that effectively increases the switch factor. Recent work has begun to look in this direction and we feel this is an area that may find applications in areas such as static timing analysis and noise-aware routing. Results have shown that $-1$ and 3 are the actual lower and upper bounds on the switch factor, rather than 0 and 2 as previously assumed [36].

**C. Monte Carlo Process Variation Modeling**

Modern chemical-mechanical polishing (CMP) techniques, used to planarize interlevel dielectrics (ILD), often result in a significant process spread of ±20% about the nominal ILD thickness [37]. For large linewidths, this will result in a roughly proportional variation in interconnect capacitance, translating directly to delay variation. For

\[\text{Note that the signal slew rates themselves are functions of the effective coupling capacitance, making this an iterative approach.}\]
minimum pitch wiring, ILD variation may result in significant crosstalk fluctuation as the shielding capability of the available ground planes is a function of the oxide thickness. Thus, the variation of interconnect can have a large effect on overall circuit performance and this variation should be taken into account throughout the design process.

While many approaches have been presented to model the statistical variation of MOS devices [38]–[40], relatively little work has been done in the area of statistical interconnect modeling [41], [42]. Presently, the only technique well developed in this area is the worst case skew-corner method. In this basic approach, all process and design parameters (e.g., linewidth, metal thickness, ILD thickness) are set to their worst case values and the resulting resistance and capacitances are calculated. These values then correspond to a worst case delay for the given circuit. Given that these worst case process parameters are typically 3-σ cases, one can determine the joint probability (JP) for this skew-corner case using (23)

$$JP = \prod_{i=1}^{N} p(i)$$

where $p(i)$ is the probability density of the $i$th parameter and $N$ is the total number of parameters. In this case, we assume independence of the input parameters, which is a reasonable assumption that is supported by experimental data. For five parameters, each at their 3-σ value (one-sided to represent worst-case only), the $JP$ is $3.71 \times 10^{-15}$. Clearly the skew-corner method is overly pessimistic in selecting a 3-σ process corner. A more realistic approach is needed to prevent designers from overdesigning products to meet inflated specifications.

**General Methodology:** We now present a fast and accurate methodology to assess the impact of interconnect variation on circuit performance metrics such as delay, crosstalk noise, and rise time. Fig. 29 shows a flow-chart describing the key points of the approach. The user provides a set of required input parameters that include the nominal values of linewidth, ILD thicknesses, etc., as well as their process spreads. It is assumed that all input parameters are normally distributed with a known variance, $\sigma^2$. This assumption can be justified by observing actual on-chip measurements of parameters such as linewidth and ILD thickness (see Fig. 30).

A fixed pitch is assumed in this work, with line spacing equal to the variable linewidth subtracted from this pitch (i.e., correlation between spacing and linewidth is -1). Given the nominal parameters, a single SPICE run is used for the determination of device characteristics and model parameters (for the analytical models described next). This step insures accuracy of any fitting parameters in the delay models.

At this point, a large number of randomized input parameter sets are generated using a Monte Carlo approach. These sets replace the single worst case input vector that is used in the skew-corner method. With a 90% confidence level, 2000 sets yield a mean with $<$1% error and a standard deviation with less than 3% error. Given 2000 sets of input parameters, the program calculates 2000 resistances, capacitances, delays, rise times, and crosstalk values. The flexibility of the approach allows users to choose either a simulation-based or analytical approach in determining circuit performance. For enhanced speed analytical models should be employed for the calculations. Inherent speed limitations on simulation-based approaches may result in a much smaller number of data sets being used and a correspondingly higher error estimate [42]. We briefly report on the errors incurred in using analytical models instead of simulators in the following sections.

This stochastic approach to interconnect modeling results in entire distributions of performance criteria as opposed to a single worst case value. This fact allows designers to obtain any arbitrary performance figure, such as 95% points or 2-σ values, rather than just a single 3-σ point. In addition, the 3-σ values found using this new methodology are much more tightly bound (i.e., show less variation), giving the designer.
more flexibility in the wire planning and floorplanning design phases.

The interconnect geometry of interest in this section consists of three parallel lines embedded in a larger array with both the immediate upper and lower ground planes present (see Fig. 31). Analytical models are taken from [22], [27], [43] and are used to calculate capacitances, delays, and crosstalk values. While the crosstalk model was discussed in detail above, a few minor modifications to the capacitance and delay models are now pointed out.

**Capacitance Model:** There are relatively few empirical capacitance expressions available in the literature for the exact geometry of Fig. 31. In [43], the authors present a straightforward model for this case that does not allow for different upper and lower ILD thicknesses. As these variables are separated in the statistical approach, we introduce a slight modification to Chern’s model to allow for this in (27). The capacitance formulas used are as follows:

\[
\begin{align*}
C_{\text{left}}(w) &= \frac{W}{H_1} \left( 1 + 0.685e^{-T/1.345S} - 0.9686e^{-S/1.21H_1} \right) \\
C_{\text{right}}(w) &= \frac{W}{H_2} \left( 1 + 0.685e^{-T/1.345S} - 0.9686e^{-S/1.21H_2} \right) \\
C_{\text{line}}(w) &= \frac{T}{S} \left( 1 - 1.8097e^{-H/0.31S} - 1.302e^{-H/0.482S} - 0.1292e^{-T/1.421S} \right) \\
\end{align*}
\]

(24)–(27)

In addition, the resistance can be calculated as a simple function of the line cross-sectional area and the line resistivity (\(\rho\), assumed to be constant although this is not required)

\[
R = \frac{\rho}{WT}. \tag{28}
\]

**Delay Model:** To approximate worst case delay values, we use a switch factor of 2 in calculating the total wire capacitance. The delay formulas used are taken from [27] with a modification made to determine the effective device resistance

\[
\begin{align*}
T_d &= R_D(C_{\text{out}} + C_{\text{in}}) + (R_D C_{W}) \\
T_f &= R_D(C_{\text{out}} + C_{\text{in}}) + (R_D C_{W}) \\
R_D &= K \frac{V_{DD}}{T_{ESAT}} \
\end{align*}
\]

(29)–(31)

The fitting parameter \(K\) is found to be different for the 50% delay and rise times (10% to 90%).\(^6\) The single SPICE run incorporated in this approach finds the values of \(K\) for both \(T_d\) and \(T_f\), ensuring an excellent fit of the delay models to the simulation results. Since the fitting parameter is calculated for specific circuit environments in our methodology, the corresponding error will be small.

A 2-D field solver and SPICE are used to verify the accuracy of the preceding analytical models. In these cases, minimum-pitch wiring in a 0.35-\(\mu m\) process is simulated for a line length of 5 mm routed in metal 2. The correlation coefficients of the analytical models are between 0.972 and 0.999 while the slopes of the scatter plots range from 0.98 to 1.14.

**Results and Discussion:** To investigate the improvement in 3-\(\sigma\) performance estimation using the new method, a test case is examined with a 5 mm metal 2 line driven by a fixed-size driver with varying interconnect pitches. Fig. 32 demonstrates the calculated process spreads for delay using three approaches: skew-corner, analytical, and simulation-based stochastic methodologies. From Fig. 32

---

\(^6\)Note that this phenomenon is explained in Section III-D by using a simple analytical model for device resistance.
Monte Carlo based methods predict about 50% less performance variation for the test cases. Furthermore, analytical and simulation-based methods are highly correlated in their results. The larger expected performance spread in the skew-corner model implies that designers will need to leave some performance “on the table” in order to meet 3-σ specifications.

More results are shown in Table 3, where each of the three performance metrics is examined at four different pitches. Again, a fixed driver size is used \( W_{n/p} = 20/40 \) µm for a metal 2 line of length 5 mm. Improvement over the skew-corner method is calculated as \( (1 - \frac{\sigma_{realistic}}{\sigma_{skew-corner}}) \). Substantial improvements, in the range of 33%–63%, are seen when applying the new stochastic methodology. In addition, delay and rise time values found using the analytical models are within a few percent (<4%) of the full-simulation cases for both mean values and 3-σ points (error values for means are not shown here). Crosstalk results include smaller pitches as these are of more interest for noise. Errors in crosstalk estimation are slightly higher (5%–7%) but still show strong correlation at various pitches and a significant ∼35%–45% improvement over the worst case methodology. These results are the first attempt at investigating the impact of process variation on noise.

The runtimes for the analytical approach is over 3 orders of magnitude less than the full simulation case. Approximately 50% of the runtime in the analytical methodology is due to the single SPICE run, implying that larger data sets (e.g., 10 000) will not incur much additional time penalties while further reducing error levels. More importantly, the model-based approach allows for varying pitches, metal layers, materials, geometries, driver sizes, etc., to be investigated while remaining feasible in terms of runtime.

**Application to Clock Network:** As an important application of the stochastic back-end variation model, we investigate the impact of interconnect variation on clock skew in a buffered \( H \)-tree clock network [44]. Clock skew is vital to

---

**Table 3**

<table>
<thead>
<tr>
<th>Pitch (µm)</th>
<th>% Error</th>
<th>SIMS</th>
<th>Analytical</th>
<th>% Error</th>
<th>SIMS</th>
<th>Analytical</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6</td>
<td>54.9</td>
<td>48.2</td>
<td>0.1</td>
<td>58</td>
<td>57.9</td>
</tr>
<tr>
<td>3</td>
<td>1.9</td>
<td>61.5</td>
<td>56.8</td>
<td>3.7</td>
<td>50.3</td>
<td>59.6</td>
</tr>
<tr>
<td>5</td>
<td>1.8</td>
<td>62.8</td>
<td>56.2</td>
<td>3.4</td>
<td>55.1</td>
<td>56.9</td>
</tr>
<tr>
<td>10</td>
<td>0.1</td>
<td>57.8</td>
<td>52.0</td>
<td>1.0</td>
<td>52.4</td>
<td>52.4</td>
</tr>
</tbody>
</table>

**Crosstalk**

<table>
<thead>
<tr>
<th>Pitch (µm)</th>
<th>% Error</th>
<th>SIMS</th>
<th>Analytical</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.2</td>
<td>39.7</td>
<td>34.8</td>
</tr>
<tr>
<td>1.4</td>
<td>7.5</td>
<td>42.8</td>
<td>35.4</td>
</tr>
<tr>
<td>1.8</td>
<td>7.0</td>
<td>46.1</td>
<td>35.6</td>
</tr>
<tr>
<td>2.2</td>
<td>7.3</td>
<td>45.3</td>
<td>32.8</td>
</tr>
</tbody>
</table>
Fig. 33. Buffered $H$-tree with device sizes, wirelengths, and linewidths shown. Left and right sides are symmetric, routing uses metal 4, and each branch has length of 10 mm.

Fig. 34. Distribution of clock skew found using analytical models. Typical clock skew in modern microprocessors is 50–150 ps.

system performance and must be precisely modeled in order to eliminate timing problems. Fig. 33 shows a schematic of a simplified clock tree with device and wire dimensions. Ignoring process variation, the skew of this network is zero as all branches and drivers are equivalently sized. In most microprocessor designs, process variation is accounted for by inclusion of device variation only (e.g., deviations in effective channel length) [45]. Due to the large wirelengths seen in clock trees, the impact of back-end process variation may not be negligible.

Applying the analytical stochastic approach to the clock tree in Fig. 33, 3-$\sigma$ skew due to interconnect variation is found to be 20 ps. Using the skew-corner method, we obtain a worst case clock skew of 55 ps. Results are shown in Fig. 34.

This analysis takes only back-end process variation into account, meaning that these values can be taken as the interconnect contribution to clock skew. Variation in device current drive will increase the absolute value of the skew. Furthermore, we do not consider variation in the load capacitance for different branches of the $H$-tree. Additional branches in the clock tree increase the discrepancy between the stochastic and skew-corner approaches. In this analysis we consider variation in rise time as well as 50% delay since intermediate buffers exhibit different delays due to the input slope dependence of CMOS gates. At the clock frequencies seen in today’s high-performance microprocessors, modeling error of even 20 ps of clock skew can be significant.

D. BAC PAC

System-level performance models can be defined as first-order models that attempt to capture the majority of relevant system design issues in order to provide useful predictions or early feedback to designers. An early example of a system-level performance model is SUSPENS (Stanford University System Performance Simulator), which was developed in the late 1980s [46]. Since the introduction of SUSPENS, on-chip interconnect has become a primary performance bottleneck. This issue was addressed somewhat in an updated cycle-time model by Sai-Halasz, which accounts for multilevel metallization [3].

The need for new system-level performance models is illustrated by the rise of new DSM effects that are limiting performance in modern designs. For instance, power dissipation has become a major issue in the design of high-speed microprocessors as well as application-specific integrated circuits (ASICs). Previous models have only given token consideration to power issues. To be more accurate, a hierarchical approach to power estimation should be employed such as that in [47]. Also, the impact of noise is a relatively new phenomenon in digital designs. Noise can be a reliability problem as well as a timing issue. No previous system-level models have sufficiently addressed noise concerns. Finally, important topics such as wirability, yield, and new packaging technologies (e.g., flip-chip) have a significant impact on chip performance.

Berkeley Advanced Chip Performance Calculator (BAC PAC) has been developed to address the above concerns. The model is available online at http://www-device.eecs.berkeley.edu/~dennis/BAC PAC. An overview of BAC PAC is shown in Fig. 35. This section highlights some of the new models incorporated into BAC PAC and presents preliminary results that point to important design considerations in future VLSI systems.

Delay Analysis: The drive for higher performance in ASICs and microprocessors is often quantified by the clock speed. In order to predict operating frequency in BAC PAC, a model of a critical path is specified by the user. Key device and interconnect parameters are used in conjunction with analytical wirelength estimations to optimize device sizes in terms of performance. Based on the user-defined critical path, local and global delays are computed. Clock frequency is determined by the sum of these delays as well as process
variation, clock skew, and latch delays. This section explains this process in more detail.

The first input to BACPAC is the technology parameters for the process of interest, in particular the feature size $f_x$. This value gives the program an idea for later default parameters such as metal linewidths, dielectric constants, etc. At this point, the user inputs expected values for back-end parameters such as contacted pitch, line thickness, ILD thickness, and material properties (resistivity, dielectric constant). BACPAC then proceeds to calculate the resistance and capacitance of each metal layer. The analytical capacitance formulas are from [43], based on the assumption of minimum pitch routing with upper and lower ground planes.

Device characteristics calculated by BACPAC include the device capacitances, both junction and input, as well as an effective device resistance value to be used in delay equations. While capacitance calculations are straightforward (see [32]), we present a novel means of determining device resistance. Driver resistance is computed using a new technique that relates a device’s current and voltage relationships directly to its effective resistance. It is a physical derivation, as opposed to empirical efforts in the past [48]. We begin by looking at the driver-interconnect network in two fashions, as depicted in Fig. 36. First, we can view the network as a lumped RC system where the 50% delay is defined as 0.69RC. We can also see the system as a current source charging the load capacitance. In this instance the 50% delay point is set by $C_{\text{load}} \times (V_{dd}/2)/I_{\text{supply}}$. If we assume that the driver supplies $I_{\text{load}}$ throughout the voltage swing of interest, we can equate these two expressions to find the effective device resistance (note that we are neglecting wire resistance, which is a good assumption in local routing with high-impedance drivers)

$$R_{\text{dev}} = \frac{C_L V_{dd}}{0.69(2\alpha) C_L I_{\text{load}}} = \frac{0.725 V_{dd}}{\alpha I_{\text{load}}},$$

(32)

In this expression, the parameter $\alpha$ represents the fact that $I_{\text{load}}$ will not be supplied throughout the voltage swing of $V_{dd}/2$. Due to velocity saturation effects, however, a current close to $I_{\text{load}}$ will be conducting throughout most of the time period, so we expect $\alpha$ to have a value near 1. From simulations and published $I_d$-$V_d$ curves, we have determined $\alpha$ to be approximately 0.9. Solving (32) with $\alpha = 0.9$, we find that device resistance for 50% delay is equal to 0.805$V_{dd}/I_{\text{load}}$. Accuracy of this expression was confirmed using HSPICE. In addition, the method is easily extendable to calculate effective device resistance for 10%–90% rise time and other arbitrary delay values.

Wirelength Modeling: Wirelength models are typically based on Rent’s Rule [49]. BACPAC uses Donath’s model [50], which serves as a good upper bound on expected wirelengths for a given Rent’s exponent. Wirelengths are usually calculated in terms of gate pitches. Gate pitch is traditionally determined by taking the square root of the chip area divided by the number of gates. According to [51] and [52], the size of a cell is set by the contacted metal pitch of the lower-level metals in a process. Investigating a two-input NAND, we see the size of this cell for standard drive strengths is 4 MP across by 16 MP high, where MP is the contacted metal pitch of bottom-level metals [52]. Other gates are either wider or narrower depending on the number of inputs and drive strength. Thus, in a typical high-performance cell library, the average gate has an area of 64 MP$^2$.

Gate pitch is then obtained by scaling the ideal value of $(64 \text{ MP}^2)^{1/2}$ using the silicon efficiency ($S_e$). Silicon efficiency is defined as the percentage of silicon that contains gates (typically ~50%). We obtain

$$P_g = \frac{8 \text{ MP}}{\sqrt{S_e}}.$$  

(33)
If \( s_c \) is one, then the gate pitch becomes simply equal to 8 MP. When \( s_c \) is 0.5, the gate pitch is equal to 11.3 MP. From the above, average point-to-point wirelength is determined. Multiple fan-out nets are extrapolated using a modified linear empirical expression from [3].

**Device Sizing:** In order to limit the impact of interconnect performance, driving gates should be sized properly. Given the average wirelengths for a range of fan-outs (typically 1 through 4), BACPAC calculates the size of a gate that no longer yields a predetermined reduction in delay for a unit-size width increase. This method is user-defined by a sizing criterion, \( C \), which is typically in the range of 2%-5%. When an increase in gate width in a single \( W/L \) ratio does not yield a reduction in delay of \( C \), BACPAC stops increasing the size and fixes the gate width as optimal. Although most designs are interconnect-limited as opposed to device-limited [53], arbitrary increases in device sizing will lead to area penalties. These area penalties also translate to increased average wirelengths. Unless extremely small \( C \) values are used (e.g., 0.5%), this should not be a problem. Gate size optimization is discussed in more detail in [4].

**Critical Path:** Users are free to create their own critical paths. They may define any distribution of fan-outs within the range of 1 to 4. The critical path logic depth is also user-defined. This is important as it can vary widely from company to company and design to design. Logic depths in modern microprocessors, for example, range from 10 to 25. Overall, however, we expect logic depth to decrease slowly from generation to generation in the interest of getting higher performance out of DSM designs. A buffered global wire is also included in the critical path. BACPAC determines the buffer sizes based on a new area-optimal repeater expression.

The procedure to determine the global wirelength applies Donath’s model to the global level. Large functional blocks of gates (e.g., 50 000 gate modules [4]) are modeled as individual gates. The size of these modules is found using the gate pitch from (33) and then used to calculate the global gate pitch. Donath statistics are applied in a straightforward manner to calculate an average global wirelength. It should be noted that it is an average global wire and not pathological wires that are being used in BACPAC. Other system-level models typically use corner-to-corner lines, which are increasingly rare due to efficient floorplanning tools.

At this point, total cycle time is found using average wirelengths (local and global), device and repeater sizes, and a delay expression formulated by Sakurai [27]. We add a term to this expression to account for input rise-time dependency based on [29]. This term can account for 30% or more of the total stage delay. Latch set-up times and propagation delays are also included in the overall clock cycle and these times are based on ASIC libraries.

The impact of noise on timing is also considered. Noise modeling in BACPAC assumes that all stages of the critical path have worst case noise features. Namely, all gates drive lines that have two neighboring parallel minimum-pitch wires simultaneously switching in the opposite direction as the victim line. These approximations provide pessimistic noise results, which will serve as a lower bound on the speed of the design.

**Power Analysis:** We now outline a hierarchical approach to power modeling based on [47]. This approach isolates the different components of power consumption on a chip and then attempts to model them individually. We also examine leakage and short-circuit power.

- **Standard Cell Logic and Local Wiring:** We assume a hierarchical layout style like that discussed in [4], where the designer uses blocks of 50 000 to 100 000 gates to limit the impact of interconnect while maintaining reasonable flexibility and design times. Looking carefully at one of these blocks and using the previous discussion of gate pitch, we calculate the physical size of each block. To determine the interconnect capacitance, we determine the wiring requirements within the module and use the resultant total length with an average capacitance per unit length. The wiring requirements of a block are found by extrapolating the critical path model to the entire module.

  Total switching device capacitance consists of oxide, overlap, and junction capacitances. In addition, we consider the impact of internal nodes of a multi-input logic gate. From the delay analysis section, BACPAC has determined the optimal device sizing in local routing. Calculations are made for both with and without noise scenarios, so that comparisons can be made in terms of power versus delay tradeoffs. All logic gates are assumed to be the optimal size—this may result in an overestimate of sizing in some noncritical cases but will also underestimate some device sizing when longer wires are present in certain paths. Overall, we expect that this assumption will be slightly pessimistic, leading to somewhat higher power numbers.

- **Global Interconnect:** Once a typical global interconnect is found, we seek to determine the number of global wires in a design. Based on Rent’s Rule (and defining a global interconnect as a wire that runs between or among blocks), we calculate the number of global nets in a design. The total number of repeaters needed is a function of the ratio between average global wirelength and critical length and is used to determine the total buffer capacitance.

- **Clock Distribution:** BACPAC clock distribution models are based on the buffered \( H \)-tree, which is the most common clock distribution network in use today. The cluster size is set by the maximum amount of skew allowable in the design. The total distance that must be traversed in this instance is \( L \), where \( L \) is the side-length of a cluster. In order to find the maximum \( L \) that meets the \( T_{\text{skew}} \) requirement, we use Sakurai’s delay expression [27] to find the delay between a gate just at the output of the lowest level clock buffer and a gate in the corner of the cluster (see Fig. 37). In this case, we expect the dominant source of delay to be the charging of latch capacitances along the wire resistance. Since these latches will not be lumped at the end of the line, but distributed along the length \( L \) of the wire, we model this RC component as distributed,
which differs from Sakurai’s initial formulation. In addition, a 10% process variation is assumed for both device and interconnect characteristics, which serves to reduce $L$. The analytical formulation gives results within 10% of simulations.

Once a maximum cluster size is determined, the required number of drivers and total wirelength can be estimated based on the regularity of the $H$-tree structure, leading to a total clock load capacitance. Since the activity factor for clock networks is 1 (there is a power-consuming transition every clock cycle), the power consumption from this component can be large.

- **I/Os:** The power modeling of I/O drivers is relatively straightforward. It consists of first determining the number of signal pads for a design based on Rent’s Rule. Once this value is determined, the pad is optimally driven by a cascaded chain of inverters where each stage is larger by a constant factor. We make an important enhancement to the previous approach by developing a more accurate model for the total driver chain capacitance, which accounts for the intrinsic output capacitance of the buffers. HSPICE simulations have confirmed the validity of this new model.

- **Memory (On-Chip Caches):** Memory power is estimated by examining the capacitances being charged in read and write operations. Detailed expressions similar to those in [47] are developed that take into account the lower signal swing typically seen for bit lines in on-chip memories.

- **Leakage:** In order to track the increasing importance of leakage current in future designs, we have implemented leakage power models into BACPAC. The calculations are based on an expression for static current in an MOS device [54]. The total chip-level static power consumption is found by determining the total device width in a design. Leakage current from both NMOS and PMOS devices are of interest here, so the total device width is calculated. Ignoring memory, we estimate device width by including repeaters, logic gates, and clock drivers.

- **Short-Circuit:** Based on [55], BACPAC calculates short-circuit power consumption by determining the length of time that the short-circuit current flows ($t_{\text{rise}}$) as well as the peak value of the current ($I_{\text{peak}}$). Combining delay expressions from [27] with the device resistance method explained earlier, $t_{\text{rise}}$ is found for each fan-out configuration. $I_{\text{peak}}$ is found using the alpha-power law model for $I_{\text{ds}}$ to estimate drain current at $V_{GS} = V_{GD}/2$ [29]. Short-circuit power dissipation per gate is found, which is then extrapolated to the entire design by focusing on logic gates only.

Wirability Analysis: BACPAC determines the wirability of a system by calculating the available routing resources. In this discussion, we look only at the logic portion of a design as the wiring requirements for memory are generally much lower than that for logic [3]. Ideal routing capacity is reduced by several factors that have been ignored or roughly approximated in previous system-level performance models, including power distribution, via blockage, and clock routing.

The power distribution network uses significant routing resources, especially at the top layers. BACPAC first determines power grid dimensions by fixing the peak IR voltage drop at 4% of $V_{DD}$. Analytical expressions are derived to describe the IR drop of an arbitrary layer as a function of metal linewidth. When the reliability constraints are met, the percentage of routing resources used is calculated. BACPAC allows for both wirebonding and C4 flip-chip packaging in constructing a power grid. Results indicate that wirebonding is a nonscalable technology due to rising die sizes, dropping noise margins, and increasing current levels.

It has been estimated empirically that for metal layers with equivalent pitches, an upper layer blocks 12%–15% of an underlying layer due to its need to connect to the substrate using vias [3]. However, when larger metal pitches are used on higher levels, the amount of blockage can be reduced if we are using relatively fixed-size vias. The relationship between via blockage and metal pitch can be shown to be linear. With a multilevel interconnect system, vias connecting the top layer to the substrate necessarily block all underlying levels. Thus, metal 1 is blocked by all subsequent layers resulting in a sizable loss in metal 1 routing capacity. Fortunately upper layers usually have significantly larger pitches than bottom levels, reducing the via penalties associated with multilevel interconnect. BACPAC calculates exact via blockages for each layer according to the user-specified metal pitches.

Clock distribution also serves to reduce available routing resources. Due to the regularity of $H$-tree structures, the total wiring required for such a network can be found fairly accurately. Given the number of clusters in the $H$-tree, the total wirelength is given by a simple analytical expression in terms

\begin{equation}
\text{wirelength} \approx L_n \times \text{factor}
\end{equation}
of the chip-side length. Additional “within-cluster” routing is approximated using a heuristic that allocates wires from the central driver to the perimeter of the cluster in all directions.

Routing tools cannot fully utilize all the available routing resources for a given design. This is mainly due to the algorithms used within the routing tools. BACPAC models this effect by first calculating the available routing resources after clock routing, power/ground routing, and via blockages. At this point, the routing area is multiplied by a routing efficiency factor to give the estimated available routing resources. This routing efficiency factor is set at 0.5 in this work. This value is based on discussions with CAD engineers from industry.

Results and Discussion: BACPAC has been used to investigate future VLSI systems, with an emphasis on microprocessors. Default parameters are used in many instances; defaults use conservative values for parameters such as drain current, threshold voltage, dielectric constant, etc. Looking at Fig. 38, one can see that BACPAC default predictions are very close to those from the 1999 ITRS for global clock frequency [2]. Logic depth is 15 throughout technologies in the default case. When the logic depth is reduced (by 1 per generation), ITRS expectations can be exceeded with significant improvements seen at sub-0.1 μm technologies. The impact of noise is seen to be about an 8%–10% drop in critical path speed. The main effect of noise on system design is the need for enhanced current drive to offset the higher overall capacitances. The larger devices lead to more power dissipation in addition to a delay penalty.

Fig. 39 looks at performance in terms of megahertz per watts and finds that increases in transistor current and module design size (from 50,000 to 100,000 gates/module) do not appreciably alter performance. However, the use of silicon-on-insulator (SOI) technology does yield significantly higher performance due to its lower device capacitances and improved leakage characteristics.

Global wires are examined and the use of scaled global wires (reducing pitch each generation) is found to have detrimental effects on both speed and power (due to excessive buffering). As explained in [56], the use of “fat” [3] or unscaled global wires will be necessary to achieve performance goals without heavy power and delay penalties.

Total power consumption is seen to rise quickly in Fig. 40, exceeding ITRS predictions beyond 0.1 μm. Power calculations are based on the assumption that 70% of the devices in a design are allotted for memory. With billion transistor designs at 0.05 μm, a larger portion of the device count may be memory-related. This would drop BACPAC power estimations but also result in less powerful and capable de-
signs. In addition, leakage is seen to become significant with scaling $V_{th}$ values, even at a conservative $V_{th}$ of 150 mV at the 0.05 $\mu$m technology node. For instance, at 0.05 $\mu$m, a generic microprocessor could see $\sim$6% of its total power go to static consumption. Clock distribution uses a buffered $H$-tree, which generally lead to lower power consumption than grid-based designs. This explains the relatively small amount of power consumed in clocking in Fig. 41.

A wirability analysis of a generic 0.05 $\mu$m microprocessor has found an optimal wiring architecture to consist of nine metal levels, six of which are dedicated to global wiring where global wiring is defined as routing among 50,000 gate modules. While local wiring dimensions will remain to be set by lithography limits, global wires will need to be carefully optimized to allow for pathological routes while supplying sufficient routing resources for the entire design.

IV. TRENDS AND DISCUSSION

As discussed in Section III-A, crosstalk noise can be expected to worsen with technology scaling due to a number of inevitable trends including tighter pitches and larger aspect ratios. However, with static CMOS as the dominant logic family, crosstalk as a reliability phenomenon is not as serious as with dynamic, pass-transistor, or other logic families with less noise immunity. For this reason, noise in the form of dynamic delay is likely to be a more serious concern in digital design. New approaches to including dynamic delay in static timing analysis have recently appeared [57], [58]; they are based on the concept of pruning unaffected nets by examining the possible switching windows for each signal compared to its neighbors (timing orthogonality). These methodologies are good first-cut approaches but in the future, dynamic delay will need to be considered in the design phase. On-the-fly noise analysis will require efficient analytical models that can accurately handle multiple aggressors and location-dependent coupling. These models will be used to drive routing and should also be coupled to driver sizing, as this work has shown the driver characteristics strongly impact noise. Switch factor models seek to replace the coupling capacitance with an effective coupling capacitance (see Section III-B). Closed-form expressions that describe the coupling multiplier as a function of driver and line parameters will help in both the routing and post-route analysis stages. Finally, functional orthogonality is required to further prune unaffected nets; by functional orthogonality we refer to neighboring nets that cannot switch simultaneously due to their logic functionality.

On-chip inductance has been ignored for the most part in comparison to printed circuit board (PCB) and multichip module (MCM) interconnections. Because of smaller dimensions on a chip, inductance values have been negligible until now. At today’s high frequencies and, just as importantly, even faster rise times, inductive effects have become relevant. To illustrate, impedance is defined as

$$Z = R + j\omega L$$

where $\omega$ is the angular frequency ($\omega = 2\pi f$). Resistance dominates total impedance at low frequencies, especially in narrow wires. But for wide copper wires and high frequencies, the resistance can be fairly low such that $R$ is overshadowed by $j\omega L$. At this point, inductive effects come into play.

It is important to realize that the clock frequency is not the best indicator of relevant frequencies. When dealing with digital signals in general, the maximum frequency of interest is related to the edge rate or rise time of the pulse. Through a Fourier transform, a signal with a rise time of $t_r$ has frequency components up to $(1/t_r)$. This can be conservatively defined as the signal bandwidth. A better definition for signal bandwidth is $(0.35/t_r)$, which corresponds to cutoff frequencies at the $\sim$3 dB points [59]. For instance, a 1-GHz signal with 100 ps rise/fall times has a conservative bandwidth of 10 GHz and a 3-dB bandwidth of 3.5 GHz. In reality, frequency components at the conservative bandwidth are normally not important. For a realistic depiction, the 3-dB points should be used. As a result of this relationship between signal edge rate and inductance, clock signals are the most susceptible nets to inductive effects since they have stringent requirements on edge rate to maintain logic functionality. Clock nets also tend to be wide to limit resistance, further emphasizing the importance of inductance.

Consequences of on-chip inductance include ringing and overshoot effects, added or reduced delay (compared to the RC case), reflections of signals due to impedance mismatch, and inductive coupling between lines. Ringing effects can be problematic for clocks since these glitches can be observed as transitions and lead to faulty logic switches. Coupling due to inductance is very significant in bus structures and is a nonlocal problem since mutual inductance does not decay quickly with distance as does coupling capacitance.

The main difficulty in modeling on-chip inductance is the determination of the current return path. Inductance arises from time-varying currents along a path that leads to the generation of a magnetic field. The path is generally considered to be a closed loop (in a solenoid for example) and flux is proportional to the area of the loop. However, in on-chip applications, there is no clear return path for a wire, just a source and a sink. While ground or $V_{th}$ will serve as the current return path, it is hard to determine exactly which power or ground line “closes the loop.” In addition, nearly signal lines may act as AC grounds and provide partial current return as well. Therefore inductance extraction is a much more difficult procedure than resistance or capacitance extraction.

While full-chip RC extractors have recently become practical due to fast algorithms and powerful workstations, a good approach to inductance extraction has not been found. To alleviate this problem, tools need to make first order approximations to derive fairly accurate inductance values for global nets. It is the global nets that will see inductive effects
first and by using pessimistic approximations, designers can guarantee sufficient signal integrity. Once inductance values are found for key nets, steps can be taken to limit the effect of this inductance on circuit performance. One example would be shield wire insertion algorithms that provide stable and nearby current return paths for inductance-sensitive nets (clocks, buses, etc.). Furthermore, the presence of long-range mutual inductive coupling will be a top modeling priority in the future. No current approaches are available for extraction of mutual inductance.

In addition, characterization techniques for inductance need to be developed to calibrate future extractors, similar to CBCM for capacitance. While a simple relationship between charge and capacitance forms the foundation of CBCM, there are no analogs in the case of inductance. Nonetheless, on-chip inductance characterization is necessary if we are to move from RC to RLC modeling. Compact RLC models, similar to the RC models presented in [27], are needed to help guide designers and for use in fast screening tools. For instance, pessimistic inductance extraction results could be inserted into a simple RLC model to determine whether inductive effects are significant. Additional complexity models would not be required if a certain threshold (e.g., 5% delay change from RC to RLC) is not crossed under worst case assumptions.

It is unclear how well the insertion of shield wires works in limiting inductive coupling noise. If this technique does not provide the necessary noise immunity, other approaches will need to be used. An example is differential low-swing global interconnect in buses. This approach has the advantage of increased speed (due to the smaller voltage swing) and good noise immunity (common-mode rejection). In particular, early results demonstrate significantly better noise characteristics with respect to inductive coupling when compared to standard shielding wire approaches [60]. The most significant drawback to this approach may be the presence of substantial off-state currents.

Systematic process variation occurs due to established limitations of processing equipment, such as stepper-induced imaging nonuniformity due to lens aberrations or ILD thickness sensitivity to pattern density in chemical-mechanical polishing. These limitations can be characterized, modeled, and incorporated into the design process. For example, one can build empirical models to define the ILD thickness as a function of pattern density [60]. Using these models, routers can be aware of constraints on pattern density (currently done by foundries using metal fill) and design accordingly. Full-chip RC extraction would be more accurate as well since calibrated dielectric thicknesses are used rather than nominal values. Another example is in the front-end where the center of the die has smaller polysilicon critical dimensions (CD); a wafer map can be generated a priori based on test chips with the results used to modify the effective channel length or device speed in place and route timing models [62].

The current state of the art in delay modeling includes the use of effective capacitance models in standard cell library characterization [63]. Effective capacitance, or $C_{eff}$, accounts for resistive shielding of downstream capacitance, which is more of an issue in DSM since line resistances are growing. With resistive shielding, the output of the driving gate responds more quickly since it effectively “sees” less capacitance. The $C_{eff}$ concept replaces the total capacitance used in the library characterization delay tables with an effective capacitance value that is a function of the driving cell and the interconnect network characteristics. The value of $C_{eff}$ is typically found using an iterative process, with an initial estimate of $C_{eff} = C_{local}$, where $C_{local}$ is the total lumped capacitance at the output of the gate. While this approach has worked well, increasing system complexity and resistive shielding effects imply that noniterative $C_{eff}$ models would be well suited to delay calculation in inner optimization loops while also speeding the library characterization process [64].

Modern delay models range in complexity from the simple Elmore delay model [65] to detailed moment matching techniques [66]. Typically, the amount of model complexity increases as the design process progresses; simpler models are acceptable during synthesis while post-route analysis requires the utmost in accuracy. However, with increasing resistive shielding, faster rise times, and inductive effects coming into play, models must be able to handle such effects. The Elmore delay is the workhorse of the design automation community; the Elmore delay is the first moment of the impulse response. The Elmore delay is fast since it can be expressed by a simple closed-form expression. In addition, it is useful because it provides an upper bound on 50% delay; there are many instances when we would rather overestimate delay than underestimate it. The Elmore delay model allows us to guarantee this. Finally, the Elmore approximation gives high fidelity with the actual solution (under a more accurate delay model, such as full SPICE simulation). Elmore works best when the rise time is slow. Due to its high fidelity, it can also be used accurately to determine the relative delays between fan-out nodes in a balanced (or nearly balanced) interconnect tree. For very fast transition times and fan-out nodes near the source, Elmore delay can have significant errors (e.g., 50%).

Additional moments (beyond the first which is simply the Elmore delay) are easy to calculate but going from moments to delays, as is done in moment matching techniques such as asymptotic waveform evaluation (AWE), requires nonlinear iterations that are slow. The more moments that are used, the slower the solution. Thus, moment matching in general is too complex for inner loops of the design process—it is more often relegated to later stages of design where the accuracy versus runtime tradeoff is more heavily skewed toward accuracy.

Thus, a void has formed where the Elmore model is too inaccurate for modern deep-submicrometer technologies but moment matching techniques are too computationally intensive to handle the rising system complexity. Simple models, such as two-pole RLC models, are needed that can provide the needed accuracy in near-Elmore delay calculation times. There has been renewed interest in this area recently [67], [68].
V. CONCLUSION

Characterization of interconnect is vital since interconnect strongly impacts circuit performance. Given that it is the interaction of devices and wires that matters, all efforts should be made to create realistic test structures that truly reflect the scenarios found in actual IC products. To this end, in this paper we demonstrated an active approach to measuring ultrasmall interconnect capacitance, which has become the new industry standard in parasitics characterization. In addition, a novel means of measuring interconnect noise related phenomenon (both dynamic delay and crosstalk noise) was described. This technique is useful for analytical model verification and to recognize and validate key interconnect trends in deep-submicrometer CMOS. It also contributes information on the timing windows that must be avoided to eliminate noise-related delay variation.

We also motivated the need for fast and accurate analytical models; such models can be used for attacking the rising system complexity issue by providing a screening tool mechanism for full-chip analysis. Furthermore, simple closed-form expressions are required for design purposes; intuitive models that provide guidance to designers also help drive synthesis and placement tools toward optimal chip implementations. In this work, an analytical crosstalk noise model is presented that accounts for issues such as line resistance, driver strength, and nonzero input transition times. All model parameters are typically available in an EDA environment, making the model ideal for rapid crosstalk estimation and signal integrity verification. This model is the first silicon-confirmed crosstalk noise model.

We proposed a new stochastic approach to account for on-chip interconnect variation, which is based on the use of actual process distributions and incorporates analytical models to quickly generate realistic \(3\sigma\) performance at the prelayout design phase. The runtime improvement allows for more extensive interconnect optimization to be performed early in the design process and the improved accuracy results in well-defined and realistic bounds on delay and noise.

BAC PAC was introduced to account for new DSM effects within the framework of a system-level performance model. It can be used to investigate trends with respect to various input parameters, whether they are device, interconnect, or system-level properties. In addition, the models in BAC PAC have been subsequently used in creating the Gigascale Silicon Research Center Technology Extrapolation System (GTX) \[69\], which adds significant flexibility to existing system performance models.

ACKNOWLEDGMENT

The authors wish to thank J. Chen, T. Sato, Y. Cao, and S. Nakagawa for their significant contributions to this work.

\[9\] For further information on other aspects of BAC PAC, readers are referred to: http://www-device.eecs.berkeley.edu/~dennis/bacpac/bacpac_models.html.

\[10\] GTX is available for download at http://vlscad.cs.ucla.edu/GSRC/GTX.

REFERENCES

Sylvester and Hu: Analytical Modeling and Characterization of Deep-Submicrometer Interconnect


Dennis Sylvester (Member, IEEE) received the B.S. degree in electrical engineering summa cum laude from the University of Michigan, Ann Arbor, in 1995, and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1997 and 1999, respectively.

He is an Assistant Professor in the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor. Previously, he was a Senior R&D Engineer in the Advanced Technology Group of Synopsys. He also worked at Hewlett-Packard Laboratories, Palo Alto, CA, from 1996 to 1998. During his graduate studies, he held a Semiconductor Research Corporation (SRC) Graduate Fellowship. He has published numerous papers in his field of research, which includes interconnect characterization and modeling, on-chip crosstalk, noise-aware timing analysis, and back-end statistical variation.

Dr. Sylvester was awarded the 2000 David J. Sakrison Memorial Prize for the best dissertation in the Berkeley EECS department. He also received the 2000 Beatrice Winner Award at ISSCC, two outstanding research presentation awards from the SRC, and a best student paper award at the 1997 International Semiconductor Device Research Symposium. He has given tutorials and invited presentations at several conferences and workshops and serves on the technical program committee for the International Conference on Computer-Aided Design (ICCAD) as well as Technical Program Chair for the 2001 International Workshop on System-Level Interconnect Prediction.
Chenming Hu (Fellow, IEEE) received the B.S. degree from the National Taiwan University and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1968, 1970, and 1973, respectively. He is Chancellor’s Professor of Electrical Engineering and Computer Science at the University of California, Berkeley. He was an Assistant Professor at Massachusetts Institute of Technology for three years. He was the Board Chairman of the East San Francisco Bay Chinese School and is a frequent advisor to industry and educational institutions. His present research areas include microelectronic devices, thin dielectrics, circuit reliability simulation, and nonvolatile memories. He has authored or co-authored four books and more than 500 research papers and supervised 60 doctoral students.

Dr. Hu is a member of the U.S. National Academy of Engineering, an Adjunct Professor of Peking University, and Honorary Professor of the Chinese Academy of Science. In 1991, he received the Excellence in Design Award from Design News and the inaugural Semiconductor Research Corporation Technical Excellence Award for leading the research of IC reliability simulator, BERT. He received the SRC Outstanding Inventor Award in 1993 and 1994. He leads the development of the MOSFET model BSIM3v3 that has been chosen as the first industry standard model for IC simulation by the Electronics Industry Association Compact Model Council and was given an R&D 100 Award in 1996 as one of the 100 most technologically significant new products of the year. The IEEE awarded him the 1997 Jack A. Morton Award for his contributions to the physics and modeling of MOSFET reliability. Also in 1997, he received UC-Berkeley’s highest honor for teaching—the Distinguished Teaching Award. In 1998, he received the Monie A. Ferst Award from Sigma Xi for encouragement of research through education.